HAL
open science

Infinite Computations in Algorithmic Randomness and Reverse Mathematics

Paul-Elliot Anglès d'Auriac

To cite this version:

Paul-Elliot Anglès d'Auriac. Infinite Computations in Algorithmic Randomness and Reverse Mathematics. Logic in Computer Science [cs.LO]. Université Paris-Est, 2019. English. NNT : 2019PESC0061 . tel-02936757v2

HAL Id: tel-02936757

https://theses.hal.science/tel-02936757v2

Submitted on 15 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Paris-Est
École Doctorale MSTIC

Thèse de doctorat
Logique Mathématique

Paul-Elliot Anglès d'Auriac

Infinite Computations in Algorithmic Randomness and Reverse Mathematics

Thèse dirigée par : Pierre Valarcher et Benoît Monin

Jury:

Noam Greenberg	Victoria University of Wellington	Rapporteur
Merlin Carl	Europa-Universität Flensburg	Rapporteur
Pierre Valarcher	UPEC	Directeur de thèse
Benoît Monin	UPEC	Directeur de thèse
Serge Grigorieff	Université Paris Diderot	Président du jury
Julien Cervelle	UPEC	Examinateur
Ludovic Patey	Université Lyon 1	Examinateur
Brigitte Vallée	Université de Caen	Examinateur

Remerciements

Tout d'abord, cette thèse n'aurait pu voir le jour sans mes deux directeurs, que je remercie pour cela. Benoît Monin, mon directeur et encadrant de thèse, a eu le courage de me prendre en thèse à peine nommé maître de conférence. J'ai pu profiter de son intérêt pour le calcul en temps infini, qu'il m'a transmis, jusqu'à ses qualités humaines. J'ai aussi eu la chance d'avoir Pierre Valarcher comme directeur de thèse officiel, il a veillé à ce que tout se passe bien du début à la fin. I bow to my two reviewers, Merlin Carl and Noam Greenberg, for taking the time to carefully read this long document. It is a tedious and long task, thank you very much for your work. Je remercie aussi le reste du jury, Serge Grigorieff, Julien Cervelle, Brigitte Vallée et Ludovic Patey..

Depuis qu'Alexander Shen et Laurent Bienvenu m'ont initié à la calculabilité par un excellent cours, Laurent n'a pas cessé d'être l'homme de la situation. Je le remercie pour tout : son enseignement de la science, son enseignement de la recherche, et pour m'avoir pointé de nombreuses d'opportunités scientifiques qui ont beaucoup influencées cette thèse. Arrivé en deuxième année, j'ai eu l'immense privilège de régulièrement côtoyer Ludovic Patey. Sa réputation scientifique n'est plus à faire, mais sa disponibilité, son partage de connaissance, sont également impressionnant. Merci encore.

Merci à toutes les organisations qui m'ont permis de continuer à me former pendant la thèse. Chi Tat Chong and the IMS and their speakers (in particular Ted Slaman) for organizing such interesting summer school each year. Thanks to JSPS and its Summer Program, which allowed me to work with Takayuki Kihara and Keita Yokoyama, whom I also warmfully thank.

Merci à tous mes collègues du LACL, toujours aidant pour des questions d'ordre scientifique comme pratique. Et merci à mes étudiants pour m'avoir donné le goût d'enseigner.

Merci à mes professeurs, et institutions, qui m'ont guidé vers cette voie, la logique mathématique. Merci au Master LMFI, à son organisation parfaite, ses cours spécialisés et ses professeurs investis, n'hésitez pas une seconde à y aller ; merci aussi à l'ENS Lyon pour m'avoir ouvert à l'informatique fondamentale dans sa généralité.

Contents

Remerciements **iii**

Introduction en français **1**

Introduction in English **13**

I Preparation **25**

I.1 Basics of Mathematical Logic . . . 25
I.1.1 Formulas . . . 25
I.1.2 Theories and Models . . . 26
I.1.3 Gödel's code . . . 27
I.2 Computability Theory . . . 28
I.2.1 Basics and Turing Machines . . . 28
I.2.2 Oracles and reducibilities . . . 30
I.2.3 Computable enumeration . . . 32
I.3 Gödel's theorems . . . 32
I.3.1 Gödel's completeness theorem . . . 32
I.3.2 Gödel's incompleteness theorems . . . 33
I.4 Topology and Computability . . . 34
I.4.1 Computability in topological spaces . . . 34
I.4.2 Topological properties of Cantor and Baire spaces . . . 36
I.4.3 Effectively open and closed sets . . . 37
I.5 Hierarchies of complexity . . . 37
I.5.1 The Arithmetical hierarchy . . . 38
I.5.2 The finite Borel hierarchy . . . 40
I.6 Ordinals . . . 41
I.6.1 Basic properties . . . 41
I.6.2 Computable ordinals . . . 42
I.6.3 Constructive ordinals . . . 43
I.7 Transfinite hierarchies . . . 46
I.7.1 The Borel Hierarchy . . . 46
I.7.2 The effective Borel Hierarchy . . . 47
I.7.3 Hyperarithmetic sets . . . 48
I.7.4 Complexity above the Borel sets . . . 48
I.8 Reverse Mathematics . . . 50
I.8.1 Second Order Arithmetic . . . 50
I.8.2 Encoding objects in Second Order Arithmetic . . . 52
I.8.3 The Big Five . . . 52

II Infinite time computations **57**
II.1 Higher Computability . . . 59
II.1.1 Lightface analytic and co-analytic sets . . . 59
II.1.2 Properties of higher recursion . . . 60
II.1.3 Higher computable sets . . . 62
II.1.4 Higher recursion with sets of reals . . . 63
II.2 α-recursion . . . 64
II.2.1 Godel's constructible universe . . . 65
II.2.2 The definition of α-recursion . . . 67
II.2.3 Σ_1-induction for limit and admissible ordinals . . . 68
II.2.4 Some α-recursive predicates . . . 70
II.2.5 Stability . . . 71
II.2.6 Projectibility . . . 73
II.2.7 An α-simple subset of $\mathbb{N}$ . . . 74
II.3 Infinite Time Turing Machines . . . 75
II.3.1 The Infinite-Time Turing Machines model . . . 75
II.3.2 Computational power of ITTMs . . . 77
II.3.3 The three notions of writability . . . 79
II.3.4 The λ-ζ-Σ theorem . . . 82

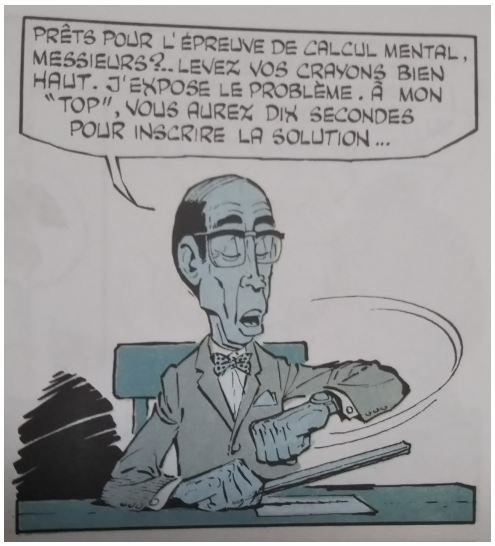
III The Reverse Math of Hindman's Theorem **91**
III.1 Hindman's Theorem . . . 92
III.1.1 Statement . . . 92
III.2 Multiple proofs . . . 94
III.2.1 The most effective proof . . . 94
III.2.2 The less effective proof . . . 99
III.2.3 The ultrafilter proof . . . 101
III.3 The Reverse Mathematics of HT . . . 104
III.3.1 Upper bound . . . 105
III.3.2 Lower bounds . . . 107
III.3.3 Right-match, Left-match and Full-match . . . 110
III.4 Ordinal Analysis of Hindman's proof . . . 116
III.4.1 Largeness notions . . . 117
III.4.2 The strategy . . . 119
III.4.3 Proof of Theorem III.71 . . . 121
III.4.4 Ordinal analysis of n-FinHT . . . 122
III.4.5 Conclusion on the ordinal analysis . . . 127

IV AC and ATR in Weihrauch lattice **129**
IV.1 Reducibilities on multivalued function . . . 130
IV.1.1 Medvedev and Muchnik reducibilities . . . 130
IV.1.2 Weihrauch Reduction . . . 131
IV.1.3 Algebraic structure of the Weihrauch lattice . . . 133
IV.1.4 Arithmetical Weihrauch reduction . . . 134
IV.2 Choices in the Weihrauch lattice . . . 135
IV.2.1 Single Choices . . . 135
IV.2.2 Restrictions on single choice . . . 137
IV.2.3 Axioms of Choices . . . 139
IV.3 Restricted Analytical Axioms of Choice . . . 141

IV.3.1 Axiom of finite Choice . . . 142
IV.3.2 Axiom of all-or-finite Choice . . . 144
IV.3.3 Axiom of cofinite Choice . . . 147
IV.3.4 Axiom of finite-or-cofinite Choice . . . 149
IV.3.5 Conclusion on the restrictions . . . 152
IV.4 More on ATR_0 . . . 152
IV.4.1 Comparability of Well-Ordering . . . 153
IV.4.2 Σ^1_1 separation . . . 154
IV.4.3 Perfect Set Theorem . . . 155
IV.4.4 The Method of Pseudo-Hierarchies . . . 156
IV.5 ATR in the Weihrauch lattice . . . 158
IV.5.1 The Weihrauch degrees of ATR . . . 158
IV.5.2 Comparing the different versions of ATR . . . 162
IV.5.3 Relation between choices and ATR. . . . 163
IV.5.4 Weihrauch degrees of theorems at the ATR_0 level . . . 164
IV.6 Separating Σ^1_1 axioms choice and dependent choice . . . 166
IV.6.1 Supportive orders and pseudo well orders . . . 166
IV.6.2 Complexity of Supp and PWO . . . 171
IV.6.3 ATR_2 versus Σ^1_1-AC_ω and Σ^1_1-DC_ω . . . 173

V Randomness and genericity within ITTMs 175
V.1 Notions of typicality . . . 177
V.1.1 Measure Theory . . . 177
V.1.2 Algorithmic Randomness . . . 179
V.1.3 Baire categoricity . . . 179
V.1.4 Genericity . . . 180
V.2 Forcing in the constructibles . . . 182
V.2.1 Borel codes . . . 182
V.2.2 The naming system . . . 183
V.2.3 The canonical Borel sets . . . 185
V.3 Randomness within ITTMs . . . 187
V.3.1 Main definitions . . . 188
V.3.2 ITTM-randomness . . . 190
V.3.3 Martin-Löf randomness in the constructibles . . . 195
V.4 Genericity within ITTMs . . . 201
V.4.1 Genericity over the constructibles . . . 202
V.4.2 Main definitions . . . 204
V.4.3 ITTM-genericity and ITTM-decidable genericity . . . 204

Introduction



Gotlib, RàB T2. Le matou matheux

Nous introduisons le sujet de cette thèse en partant du plus général vers le plus précis, non pas en une seule étape mais à l'aide d'agrandissements successifs du niveau de précision, d'une manière de plus en plus détaillée atteignant ultimement un résumé précis de la thèse et de ses contributions.

Un travail mathématique

Ce document présente un travail de recherche en mathématiques. Le but des mathématiques est d'étudier certains concepts abstraits, afin de mieux comprendre leurs propriétés et par conséquent de mieux appréhender les concepts eux-mêmes. Nous donnons sans tarder un exemple : nous sommes tous familiers avec le concept de moyenne, et d'évènement aléatoire. Ainsi, on dira facilement : « si je lance une pièce en l'air il y a une chance sur deux qu'elle tombe sur pile ». Ou « si je vais au casino, j'ai plus de chances de repartir plus pauvre, que plus riche ». Ces phrases sous-entendent l'existence d'une notion d'aléatoire, difficile à définir dans le monde réel qui nous semble suivre la causalité et donc être étranger au concept d'aléatoire. Mais les mathématiques nous le permettent, ils peuvent faire exister cette notion de manière précise dans un monde abstrait. Et une fois le concept proprement défini, nous avons les outils pour l'étudier, le disséquer et dévoiler ses propriétés les plus inattendues. Ainsi, la théorie des probabilités nous apprend qu'il existe des évènements qui n'ont pas de moyennes ! Imaginez une boîte contenant des boules numérotées, et avec un bouton qui à chaque pression extrait une boule de la boite, tirée aléatoirement parmi toutes les boules selon une certaine distribution. Le tirage est donc indépendant des précédents. Pour certaines distributions, l'étude mathématique nous dit que le concept de moyenne, associé à ce tirage, n'existe pas. On pourrait être tenté de calculer la moyenne des nombres qui sortent, après de nombreux tirages, mais celle-ci ne se stabiliserait pas : très rarement, un nombre colossalement grand serait tiré de la boîte, augmentant la valeur moyenne calculée sans qu'il n'y ait de borne à cette augmentation.

Les mathématiques nous ont donc appris quelque chose à propos du concept de probabilité, que l'on n'aurait pas forcément deviné lorsque celui-ci était juste une idée, une intuition sur le fonctionnement du monde. Elles nous ont appris bien d'autres choses sur les probabilités, dont de nombreuses se vérifient et permettent de comprendre mieux le monde bien réel et concret.

La logique mathématique

Commençons enfin notre premier raffinement vers ce qui nous intéresse dans cette thèse : la logique mathématique. La logique mathématique est un domaine particulier des mathématiques, elle correspond à l'étude d'un concept particulier, le concept même des mathématiques. Seulement rappelons nous, la première étape dans l'étude d'un concept abstrait est de le définir formellement, de le faire exister concrètement dans un monde abstrait.

Ce domaine des mathématiques fait donc exister de manière concrète les concepts de théorème, de preuve, de théorie dans le monde des mathématiques, en leur donnant une définition formelle, des contraintes qu'ils doivent satisfaire. Ces contraintes, ces définitions sont calquées parfaitement sur notre idée de ce que sont les mathématiques, nous sommes donc confiants en cette représentation des concepts abstraits dont nous sommes partis.

Nous nous trouvons ainsi en présence de deux types de théorèmes : ceux auxquels on vient de donner une existence formelle en les définissant dans les mathématiques, et ceux dont on parle lorsque l'on fait des mathématiques, mais qui n'ont pas d'existence précisément, formellement définie. Ces derniers sont dits faisant partie de la *méta-théorie*, les mathématiques que l'on utilise dans la vraie vie, pour étudier les mathématiques que l'on vient de définir.

Bien sûr, si les mathématiques de la méta-théorie correspondent réellement aux mathématiques, on peut définir à l'intérieur de ces dernières à nouveau les notions de preuves, théorèmes, et formules. En d'autres termes, si les mathématiques peuvent étudier les mathématiques, alors les mathématiques peuvent étudier les mathématiques étudiant les mathématiques. Il est bien sûr temps de s'arrêter là, en résumant l'essentiel : en logique, il y a deux types de mathématiques, celles que l'on étudie, et celles que l'on utilise pour étudier. Les unes sont une modélisation des autres, mais les deux sont bien distinctes.

Ce besoin d'étudier les mathématiques provient initialement d'un besoin de fondations solides et puissantes aux mathématiques : solides dans le sens qu'elles ne permettent pas de prouver une chose et son contraire, et puissantes dans celui de pouvoir prouver suffisamment de théorèmes. Cet objectif a guidé la majeure partie des contributions des premiers logiciens, et de nombreux résultats surprenants, inattendus ont été trouvés. Ainsi, malgré un plan d'attaque, proposé par Hilbert, qui connut tout d'abord un certain succès, Gödel démontra que de telles fondations n'étaient pas possibles. Mais tant pis : la logique mathématique était née.

Cette thèse ne s'intéresse pas aux fondations des mathématiques. En effet, bien qu'ayant été la motivation initiale, la logique ne s'arrête pas à celle-ci. C'est une théorie qui permet l'étude de nombreux nouveaux concepts, ou d'anciens concepts vus sous un nouvel angle, et qui possède de nombreuses applications et connexions à d'autres domaines des mathématiques. Cette thèse s'inscrit dans le contexte de la théorie de la calculabilité, et c'est donc le moment d'opérer un nouveau zoom en direction de celle-ci.

La théorie de la calculabilité

La théorie de la calculabilité, ou plus simplement la calculabilité, est l'étude du concept de calcul. Un calcul est l'application successive de règles simples, partant d'un état et se terminant dès qu'une règle l'indique. Par exemple, l'addition telle que l'on apprend à la poser correspond à un calcul : c'est une succession d'additions mécaniques de deux chiffres, que l'on applique chiffres par chiffres, en ajoutant au besoin des retenues, et qui se termine lorsque l'on a considéré tous les chiffres, par la somme des entiers initiaux.

Ce concept de règles mécaniques, aussi appelé "algorithme" du nom du mathématicien Perse Al-Khwârizmî, n'a à ce point de l'introduction qu'un sens vague. Passer du concept à une réalité dans le monde mathématique, satisfaisante et convaincante, ne fut pas chose aisée ! En effet, il n'est pas évident de trouver un ensemble de règles suffisamment puissantes pour englober tout ce

qui serait dit calculable par l'intuition, sans permettre de règles qui offrent trop de possibilités. Et pour chaque candidat à une telle définition, il faut encore se convaincre que c'est la bonne ! Ainsi, plusieurs définitions de la notion de calcul furent proposées, certaines n'étant pas convaincantes car elles n'englobaient pas assez de calcul possible, d'autres car elles étaient trop alambiquées et semblaient loin de l'intuition de calcul. Finalement, c'est Turing qui mit tout le monde d'accord, en définissant ses "machines de Turing", sorte de modèle d'un ordinateur primitif.

Ainsi donc, la définition formelle de calculable finit par être consensuelle. Grâce à celle-ci, nous sommes ainsi capables de définir précisément ce qu'est un ensemble calculable : c'est un ensemble d'entiers A tel qu'étant donné un entier n, le problème de savoir si $n \in A$ peut être résolu à l'aide d'un simple calcul, au sens de Turing. Nous sommes aussi en mesure de prouver qu'il existe des ensembles non calculables. Mais plus que cela, la définition de Turing nous permet de définir un calcul *relatif* à un ensemble d'entiers : en quelque sorte, si B est un ensemble d'entiers, la calculabilité relativement à B correspond à l'étude des calculs qui sont possibles si B est accessible au calcul. Cela permet ensuite de s'abstraire des ensembles d'entiers, et de ne parler plus que de *contenu calculatoire*. Deux ensembles ont ainsi le même contenu calculatoire si les notions de calculabilités relativement à l'un ou à l'autre sont les mêmes. L'étude des contenus calculatoires, nommés degrés Turing, fut au centre des préoccupations, et une structure très riche et complexe fut découverte.

Donnons maintenant quelques exemples de contenus calculatoires. Bien que l'on n'ait pas encore défini formellement la notion de calcul, nous appelons à l'intuition, en soulignant que cette notion de calcul correspond exactement à ce qui peut être calculé par un ordinateur qui ne serait pas limité par sa mémoire. Prenons l'ensemble des entiers pairs, P. La donnée de P n'aide en rien à calculer de nouveau : on peut se passer de cette donnée car on peut facilement la retrouver, en regardant le dernier chiffre d'un entier, qui sera pair si et seulement si ce dernier chiffre est 0, 2, 4, 6 ou 8. P n'a donc aucun contenu calculatoire particulier. Cependant, considérons maintenant l'ensemble des calculs qui finissent par s'arrêter. Il semble difficile de prédire si un calcul va s'arrêter, et on ne peut pas le tester en l'exécutant, car s'il ne se termine pas, nous ne le saurons jamais. Et cette donnée peut être utile pour générer de nouveaux calculs, qui pourront éviter de s'engager dans des boucles infinies. Cet ensemble est connu sous le nom du *problème de l'arrêt*. Donnons un dernier exemple. La résolution d'équations est un des problèmes majeurs des mathématiques. Est-ce que l'on pourrait ramener leur résolution à une suite mécanique de règles, un calcul ? En particulier dans le cas d'un système d'équations simples, ne contenant que des entiers, des sommes et des multiplications, un théorème de Matiyasevitch nous dit que cela n'est pas possible, et même que le contenu calculatoire nécessaire pour résoudre ces équations est exactement celui du problème de l'arrêt, ni plus, ni moins.

À ce point là il n'est pas encore clair en quoi la théorie de la calculabilité fait partie de la logique mathématique. Ceci tient en grande partie au fait qu'il existe de nombreux liens entre calculabilité, et définissabilité. Ainsi, la calculabilité d'un ensemble peut se ramener au fait que l'ensemble est définissable par une formule ayant certaines propriétés. Définissons une formule Δ^0_0 comme une formule de l'arithmétique dont tous les quantificateurs (les assertions de la forme "pour tout n", et "il existe m") sont bornés. Définissons les formules Σ^0_1 comme celles qui ne peuvent comporter que des quantifications "il existe" qui soient non bornées, et Π^0_1 celles dont les seules quantifications non bornées sont celles de la forme "pour tout". Alors, un ensemble est calculable si et seulement si il est définissable à la fois par une formule Σ^0_1 et par une formule Π^0_1. Un ensemble qui est définissable par une formule Σ^0_1 sera lui un ensemble *calculatoirement énumérable*, ou ensemble c.e., c'est à dire un ensemble tel qu'un calcul puisse énumérer exactement ses éléments. Le problème de l'arrêt est c.e. car il est définissable par une formule Σ^0_1, un calcul c s'arrêtant si et seulement si "il existe une étape t tel que le calcul c s'arrête à l'étape t".

Les liens entre définissabilité et calculabilité ne sont pas si étonnants, étant donné que ce sont

deux notions de complexité pour des ensembles : l'une mesure la difficulté à le définir, l'autre la difficulté à le calculer. La mesure de complexité d'ensembles est par ailleurs un thème récurrent en logique mathématique, central aussi en théorie descriptive des ensembles qui jouera aussi un rôle dans cette thèse.

Fort de cette analogie entre calculabilité et définissabilité, le mouvement suivant est d'appliquer les nombreuses méthodes, constructions et outils puissants développés lors de l'étude des degrés Turing, à d'autres domaines de la logique mathématique dont la définissabilité est un élément clé, et il est temps de zoomer à nouveau.

Calculabilité et définissabilité

Nous arrivons au plus gros niveau de granularité de cette introduction. Au moins trois domaines profitent de la calculabilité pour jeter un jour nouveau sur la définissabilité, et utiliser ses nombreuses méthodes de construction. À l'instar des précédents agrandissements, nous allons présenter ces trois domaines en effectuant un travelling : la théorie descriptive des ensembles, puis les mathématiques à rebours et enfin la théorie algorithmique de l'aléatoire. Nous terminerons ce travelling par une présentation du caractère commun à notre étude de ces trois domaines, le point de départ de cette thèse : la notion de calcul infini, elle aussi très liée à la définissabilité, et donc applicable aux domaines susnommés.

La théorie descriptive des ensembles

La théorie descriptive des ensembles est l'étude de la complexité des sous-ensembles de $\mathbb{R}$, d'un point de vue topologique. On prendra généralement le point de vue d'associer un élément de $\mathbb{R}$ à son écriture en base 2 : un réel est donc assimilé à une suite infinie de 0 et de 1, et l'on nomme l'ensemble de ces suites infinies l'espace de Cantor, noté 2^ω.

Les sous-ensembles de l'espace de Cantor les plus simples, au sens de la théorie descriptive des ensembles, sont les ensembles de la forme $\{x \in 2^\omega : x \text{ commence par } \sigma\}$ pour σ une suite finie, c'est à dire l'ensemble des réels dont l'écriture commence par une suite de chiffres finie et fixée. Une *description* d'un tel ensemble est simple : il suffit de donner la chaîne σ. À noter que dans $\mathbb{R}$, cela correspond à des ensembles ouverts. Et pour cause : les ensembles ouverts, ainsi que les ensemble fermés, constituent le premier niveau de complexité de la théorie descriptive, après les ouverts-fermés.

Le niveau de complexité descriptive juste au dessus des ouverts et des fermés est constitué des intersections dénombrables d'ouverts, et des unions dénombrables de fermés, traditionnellement nommés F_σ et G_δ. Les niveaux suivants de complexité de description topologique continuent ainsi, en prenant des intersections dénombrables, ou des unions dénombrables, d'ensembles de complexité moindre.

Les motivations à classer les sous-ensembles des réels par complexité proviennent entre autres de la théorie des ensembles. Le but est de montrer que les ensembles "pas trop compliqués" se comportent bien. Ainsi, même si des ensembles de grande complexité existent nécessairement, la plupart des mathématiques usuelles manipulent des ensembles de complexité raisonnable. Ainsi, par exemple les ensembles dont la complexité se trouve dans la précédente hiérarchie, dite hiérarchie Borélienne, sont tous déterminés, ils ne contredisent pas l'hypothèse du continu, deux propriétés de bon comportement.

À première vue, il ne semble pas y avoir de liens entre la complexité topologique que l'on vient d'esquisser et la calculabilité, et pourtant ils existent ! Pour les faire apparaître, il faut rajouter de l'effectivité dans les unions et les intersections. La complexité "effectivement topologique" que l'on définit ainsi possède de bonnes propriétés, comme celle d'avoir un nombre dénombrable

d'éléments pour une complexité donnée. Une fois qu'un résultat est prouvé sur la complexité effective, on peut le réhausser au niveau de la complexité topologique en relativisant le résultat : cette technique marche car la complexité topologique est l'union des complexités effectivement topologiques relativisées à un réel. L'étude de la complexité topologique effective est appelée *théorie descriptive effective.*

Les mathématiques à rebours

Dans cette thèse, le second domaine qui profite de la calculabilité est le domaine des mathématiques à rebours. Commençons par parler à nouveau des origines de la logique : le besoin de fondations aux mathématiques. Ces fondations furent bâties sur ce qu'on appelle la théorie des ensembles, une théorie où tous les objets sont des ensembles, y compris les entiers. Cela implique donc un travail de codage des objets définis usuellement en mathématique dans cette théorie. Avec les axiomes de Zermelo et Fraenkel, et éventuellement un axiome supplémentaire, toutes les mathématiques usuelles sont prouvables dans cette théorie. Cependant, elle implique l'existence d'objets gigantesques n'apparaissant jamais naturellement, et les axiomes ne correspondent pas aux étapes d'un raisonnement fait lors de la preuve d'un théorème.

Les mathématiques à rebours s'attachent à étudier non pas les fondations de toutes les mathématiques, mais elles se restreignent aux raisonnements effectués en arithmétique du second ordre, c'est à dire les raisonnements qui n'impliquent que les entiers et les ensembles d'entiers. Cela inclut la grande majorité des mathématiques développées jusqu'à maintenant. Par exemple, la théorie des nombres et la théorie des graphes entrent dans ce cadre, car les objets qu'ils étudient sont, au codage près, seulement des entiers. La théorie de Ramsey, qui elle étudie les structures émergeant dans les ensembles infinis d'entiers, est largement étudiée du point de vue des mathématiques à rebours. Beaucoup d'autres domaines, bien que se concentrant sur des objets qui ne sont ni des entiers ni des ensembles d'entiers, peuvent tout de même être étudiés dans ce cadre : c'est le cas de l'analyse dans sa grande majorité, car par exemple une fonction continue peut être codée en un ensemble d'entiers.

Le plan des mathématiques à rebours pour étudier les raisonnements de l'arithmétique du second ordre, était le suivant : étant donné un résultat important d'un domaine des maths, isoler les axiomes utilisés dans la preuve, c'est à dire les étapes qui ne nécessitent pas d'explications, qui sont vraies car elles sont évidentes ; puis une fois ces axiomes exhibés, tenter de prouver les axiomes eux-mêmes à partir du résultat. Ainsi, si le théorème et ses axiomes sont équivalents, on peut dire que ces axiomes sont les axiomes *minimaux* permettant de prouver le théorème. Le terme de mathématiques à rebours vient de la technique de preuve qui part du théorème pour arriver aux axiomes, à l'inverse de l'ordre usuel. Cependant, le problème clé d'une question de mathématiques à rebours peut aussi être de trouver une preuve utilisant moins d'axiomes.

La plus grande découverte de cette étude est que l'écrasante majorité des théorèmes développés au cours des siècles, dans de très nombreux domaines, sont équivalents à un parmi cinq systèmes d'axiomes correspondant chacun à un type de raisonnement, voire même à un programme philosophique de fondations des mathématiques [1]. Ces cinq systèmes d'axiomes, réunis dans le *Club des Cinq*, correspondent aux principes suivants :

1. Le premier, RCA_0, est le système d'axiomes nécessaire au développement des mathématiques constructives. Il correspond aux arguments les plus simples, ceux qui nous permettent d'effectivement construire l'objet que nous définissons. Par exemple, si l'on prouve

1. À savoir : le constructivisme de Bishop, les mathématiques finitaires de Hilbert, la prédicativité de Weyl et Fefermann, le reductionisme prédicatif de Friedman et Simpsons, puis pour finir l'imprédicativité de Feferman *et al.* [60]

l'existence d'un ensemble d'entiers à l'aide de ce système d'axiomes, alors l'ensemble sera calculable.

2. Le second, WKL_0, est le système d'axiomes correspondant à l'usage de la compacité dans un argument, un moyen de ramener une preuve sur le fini à une preuve sur l'infini.
3. Le troisième, ACA_0, est le système d'axiomes permettant de construire des objets qui ne soient pas totalement constructifs, mais dont la construction ne dépend que d'énoncés arithmétiques, c'est à dire de formules ne parlant que d'entiers.
4. Le quatrième, ATR_0, est le système d'axiomes permettant les constructions par induction *transfinie*, c'est à dire étape par étape, mais dont le nombre total d'étapes afin d'atteindre l'objet final dépasse le fini.
5. Le cinquième et dernier, Π^1_1-CA_0, est un système d'axiomes permettant de construire des objets de complexité presque déraisonnable. Cependant, des théorèmes naturels nécessitent tout de même l'existence d'un tel système d'axiomes, le rendant plus que légitime.

Ainsi qu'on peut le voir, la théorie de la calculabilité pointe déjà son nez à la porte des mathématiques à rebours. En effet, les axiomes correspondent à des concepts de calculabilité, et nombre de constructions de ce domaine permettent des séparations entre théorèmes et axiomes, c'est à dire permettent de montrer qu'un théorème ne peut pas impliquer un système d'axiomes.

L'implication de la communauté de calculabilité dans les mathématiques à rebours a permis aussi de montrer qu'on peut définir un "contenu calculatoire" à un théorème, c'est à dire ses implications en termes de calculabilité. Par exemple, considérons le théorème de Bolzano-Weierstrass, qui stipule qu'une suite infinie de réels bornés admet une sous-suite convergente. Étant donné une telle suite calculable, la sous-suite convergente ne pourra pas nécessairement être calculable, et le contenu calculatoire commun à toutes les sous-suites convergentes est donc en quelque sorte "impliqué" par le théorème. Cependant, le cadre donné par les mathématiques à rebours n'était pas adapté à l'étude précise du contenu calculatoire des théorèmes.

Le cadre adapté, l'outil qu'il fallait pour étudier le contenu calculatoire des théorèmes, existait en réalité déjà, et était utilisé en analyse : c'est la réduction de Weihrauch. Celle-ci permet en particulier de comparer les théorèmes de la forme "pour tout quelque chose, il existe autre chose tel qu'une troisième chose soit vraie". L'idée est de considérer ce type de théorèmes comme un problème, avec des instances dont il faut trouver une solution. Sous cet angle, le théorème dit que toute instance admet une solution, et le contenu calculatoire d'un tel théorème correspond à la difficulté de trouver la solution étant donnée l'instance. On peut ensuite utiliser la réduction de Weihrauch pour comparer la difficulté de résolution de deux problèmes : un problème P_1 est plus facile à résoudre qu'un problème P_2, si étant donné une instance de P_1, je peux en trouver une solution si l'on me donne une boîte qui me permet de résoudre l'instance que je souhaite de P_2, sachant que je ne peux utiliser la boite qu'une seule fois.

La théorie algorithmique de l'aléatoire

Le dernier domaine important de cette thèse, toujours en relation directe avec la calculabilité, est la théorie de l'aléatoire algorithmique. Nous avons tous une idée de ce à quoi ressemblerait une suite de "pile" et de "face", obtenue en lançant 1000 fois de suite une pièce. Pourtant, toutes les suites de 1000 "pile" ou "face" ont la même probabilité de se produire. Pourquoi donc, la suite "pile, pile, pile, pile, pile, ..." nous paraît-elle moins aléatoire que la suite "pile, face, face, pile, face, pile, pile..." ? Il y a une dissonance entre notre intuition d'une suite aléatoire et notre intuition des probabilités. Tandis que la théorie des probabilités formalise ce dernier concept, la théorie algorithmique de l'aléatoire se concentre sur notre concept de suite aléatoire, réconciliant ce concept avec celui de probabilité.

Comment définir formellement ce qu'est une suite aléatoire ? En accord avec l'exemple du paragraphe précédent, nous commençons par discuter le cas des suites finies (mais présumées longues). En plus de la suite constante à "pile", bien d'autres suites ne doivent pas non plus être considérées comme provenant d'un processus aléatoire, comme par exemple la suite constituée de "pile, face" répétés. La suite des décimales de π prise à partir de la 100ème décimale peut sembler aléatoire à première vue, mais plus du tout dès que l'on se rend compte de ce fait. Toutes ces suites très longues ont été décrites à l'aide de quelques mots, c'est à dire d'une suite très courte. Mais il me serait impossible de décrire une suite aléatoire en si peu de mots, l'intuition nous dit qu'il serait peu probable qu'elle soit justement obtenue, parmi toutes les suites possibles !

C'est comme cela que l'aléatoire est défini pour les suites finies : plus précisément, il est *mesuré* en fonction de la plus petite description qui permet de retrouver la chaîne, c'est la complexité de Kolmogorov. Cette complexité ne peut dépasser la longueur de la chaîne, car en donnant la chaîne elle-même, on peut trivialement la "retrouver". Et toute suite de longueur n aura au moins la complexité de l'écriture de n. Plus la complexité d'une suite de longueur n est proche de n, plus elle peut être considérée comme aléatoire, et plus elle est proche de la complexité de l'écriture de n, moins elle semble aléatoire.

Si l'on appelle suite infinie aléatoire une suite obtenue par la répétition infinie d'un lancer de pièce, alors à chaque étape la suite finie de lancers devra avoir une mesure de son caractère aléatoire proche du maximum, c'est à dire que sa complexité de Kolmogorov devra être proche de n. Cette définition, appelée l'aléatoire de Martin-Löf, est une des plus importantes, car elle peut être caractérisée par trois approches possibles , toutes raisonnables :

1. *L'incompressibilité* : l'approche par complexité de préfixes, celle que nous avons utilisée à l'instant pour décrire un ensemble aléatoire. Si un ensemble est obtenu par un lancer infini de pièces, à chaque étape il devra sembler aléatoire, et réciproquement s'il semble aléatoire à chaque étape, il devra sembler aléatoire dans son ensemble.

2. *L'imprévisibilité* : une suite semble aléatoire s'il est impossible de deviner le prochain élément de la suite à partir des précédents, avec plus qu'un succès sur deux. Plus précisément, si l'on doit parier sur ce que sera l'élément suivant, une somme que l'on choisit à chaque étape, on ne doit pas pouvoir devenir aussi riche que l'on veut.

3. *La banalité* : une suite aléatoire ne doit avoir aucune propriété exceptionnelle. Une propriété exceptionnelle est une propriété telle que l'on ne s'attende pas à ce qu'une suite aléatoire l'ait, comme par exemple le fait d'avoir toujours deux "piles" à la suite. Et si la théorie des probabilités ne nous dit pas pourquoi une suite semble plus aléatoire qu'une autre malgré leurs équiprobabilités, elle nous dit bien quelles sont les propriétés qu'il est très improbable d'avoir : ce sont les propriétés de mesure 0. Parmi ces propriétés de mesure 0, on ne choisit que celles qui ont une description pas trop complexe, car toute suite est dans un ensemble de mesure 0, comme $\{x\}$.

L'aléatoire de Martin-Löf possède donc des définitions équivalentes, chacune suivant l'un de ces trois paradigmes. Mais il n'est pas pour autant une définition parfaite, dans le sens où certaines suites qui nous semblent ne pas être aléatoires, sont tout de même Martin-Löf aléatoires. Par exemple, elles ne devraient pas pouvoir avoir un contenu calculatoire arbitrairement grand, et pourtant un théorème de Kučera et Gács montre que tout ensemble, quel que soit sa difficulté, est calculé par un Martin-Löf aléatoire. Cette notion n'est donc en ce sens "pas assez forte" pour représenter notre intuition.

Mais il existe de nombreuses autres notions plus ou moins fortes, une hiérarchie de niveaux d'aléatoirité, tous définis à l'aide d'un des trois paradigmes donnés précédement. La théorie algorithmique de l'aléatoire est l'étude de toutes ces notions, leurs forces, et leurs faiblesses.

Le calcul en temps infini

Il nous reste à aborder le dernier point important pour comprendre de quoi il s'agit dans cette thèse. C'est le spectre par lequel nous allons investiguer les domaines que nous avons précédemment évoqués, le *calcul à temps infini*. En effet, étant donné l'apport que la calculabilité a à offrir aux mathématiques, il est justifié de s'attendre à ce que des variations de cette notion soient aussi utiles.

Mais quelle est l'intuition d'un calcul à temps infini ? Difficile de parler même d'intuition, car n'étant pas quelque chose que l'on rencontre dans la vie de tous les jours, l'infini n'est pas le concept le plus transparent que l'on puisse envisager. Et pourtant il est très présent en mathématiques, il est même *essentiel* à certains résultats, qui ne le mentionnent pourtant même pas dans l'énoncé. Afin de familiariser le néophyte, passons en revue quelques types d'infinis.

Le premier type d'infini est un "pseudo" infini, un infini en puissance, dans le sens qu'il n'est pas atteint. C'est l'infini qui correspond aux résultats du type :

$$\lim_{n\to\infty} \frac{n^2}{2n+1} = \infty.$$

Cette expression se lit "tendre vers l'infini", et non "être égal à l'infini", car cela n'aurait pas de sens pour n d'être *égal* à l'infini. Tendre vers l'infini signifie s'en rapprocher sans jamais s'arrêter, et sans borne finie à son rapprochement. Ce n'est donc pas cela qui nous permettra de définir un calcul infini, car un calcul de ce type d'infini n'attend jamais réellement une étape infinie, il ne pourra jamais profiter de son infinité d'étapes précédentes.

Le premier à avoir défini un infini actuel est Cantor, ce qui lui valut une grande résistance de la part de certains de ses collègues. Pourtant, il existe un exemple simple d'infini, en tant que nombre d'éléments d'un ensemble, que l'on peut donc voir comme un infini actuel : l'ensemble des entiers naturels. Cette notion d'infini en tant que "nombre d'éléments" d'un ensemble est appelé la *cardinalité* : un ensemble à 5 éléments a une cardinalité de 5, tandis que $\mathbb{N}$ a une cardinalité infinie. Deux ensembles A et B auront même cardinalité si l'on peut mettre en relation un à un les éléments de A et de B, tel que tout élément de A est en relation avec exactement un élément de B, et réciproquement. Ainsi, bien que l'ensemble des entiers $\mathbb{N}$ contienne strictement l'ensemble des entiers pairs $2\mathbb{N}$, ces deux ensembles ont la même cardinalité : on peut mettre en relation l'élément n de $\mathbb{N}$ à l'élément $2n$ de $2\mathbb{N}$. Cantor montra qu'il existe de nombreuses cardinalités infinies, par exemple la cardinalité de $\mathbb{R}$ est strictement plus grande que celle de $\mathbb{N}$.

Profitons de ce bref passage sur les cardinaux pour parler d'une illustration assez imagée des étonnantes propriétés de l'infini, en tant que cardinalité. Hilbert, fameux mathématicien allemand, décida un jour d'ouvrir un hôtel qu'il baptisa sobrement : l'hôtel de Hilbert. Mais Hilbert ne fait pas les choses à moitié, son hôtel possède une infinité de chambres ! Elles sont numérotées c_1, c_2, c_3... et ainsi de suite. C'est le succès absolu ! Et c'est aussi très pratique, comme le montre l'anecdote suivante : un soir de grande pluie, un voyageur isolé arrive trempé à l'hôtel, qui est plein cette nuit là. Le réceptionniste n'y peut rien, mais sur l'insistance du voyageur, appelle Hilbert qui aura peut-être une solution, mathématicien qu'il est. Et c'est bien sûr le cas : Hilbert demande simplement à l'occupant de la chambre c_n de se déplacer dans la chambre c_{n+1}, qui vient donc d'être libérée par la même opération. Et comme il n'y a pas de chambre c_0, la chambre c_1 est donc libre !

Un peu plus tard, cette même nuit là, une troupe de scouts arrive à l'hôtel. Hilbert, qui accueille tout le monde et même les scouts, demande au Grand Scout combien de personnes compte la troupe, prêt à refaire la même opération jusqu'à ce qu'ils aient tous une chambre de libre. Le Grand Scout compte à voix haute : « Voyons voir, il y a s_1, s_2, s_3... nous sommes une infinité ! ». Comment faire, se dit le réceptionniste, l'hôtel est éjà plein ! Mais déjà, Hilbert a

une solution : il demande à l'occupant de la chambre c_n d'aller à la chambre c_{2n}, libérant ainsi toutes les chambres impaires, et envoie le scout s_i à la chambre c_{2i+1}. Nous laissons au lecteur le soin de résoudre le problème survenu cette même nuit, où une infinité de cars b_1, b_2, b_3... chacun rempli d'une infinité de touristes t_1^i, t_2^i, t_3^i... où i est le numéro du car, débarquèrent à l'hôtel. Ce problème laissa un affreux mal de crâne au réceptionniste.

Cette digression n'ayant que trop duré, parlons maintenant de l'infini qui nous permettra de définir un calcul infini : l'infini *ordinal*, ou *bien-fondé*. En effet, le concept de cardinalité d'un ensemble ne semble pas adapté pour définir un calcul, qui est une succession d'étapes caractérisée par l'application de règles simples, et se terminant sous des conditions simples. La décomposition en étapes suggère que l'infini que l'on définit est sous la forme d'un ensemble ordonné : chaque élément de l'ordre correspond à une étape du calcul, et l'ordre est infini. Cela implique une variété bien plus grande d'infinis que dans le cas de la cardinalité, qui nous permettait de réordonner les éléments, comme lorsque Hilbert déplace les occupants des chambres !

Des infinis ordonnés, seul une part nous intéresse, celle qui nous permet de définir chaque étape de calcul à partir des précédentes. Ces types particuliers d'ordre sont appelés les ordres *bien fondés*, et chaque ordre a un représentant appelé *ordinal* ne contenant que l'essence de l'ordre. Pour comprendre le calcul en temps infini, il faut bien comprendre cette notion d'ordinal. Nous appellerons dans cette introduction les éléments des ordinaux des *étapes*. Les ordinaux eux-mêmes sont bien ordonnés, les premiers étant les ordinaux finis, correspondant à un nombre fini d'étapes, les unes après les autres. Après tous ces ordinaux finis, arrive le premier ordinal infini, usuellement noté ω. Cet ordinal correspond au même ordre que celui sur les entiers, c'est à dire une infinité d'étapes, les unes après les autres. C'est l'ordinal correspondant au fait de "tendre" vers l'infini, sans jamais l'atteindre. Après ω vient $\omega + 1$, l'ordinal constitué de toutes les étapes de ω auquel est rajouté une étape après toutes les autres : c'est le premier ordinal qui correspond pour un calcul à l'atteinte effective d'une étape infinie, au dessus de toutes les autres. Mais un calcul infini ne s'arrête pas là, il peut avoir besoin encore de nombreuses étapes pour mettre à profit l'infinité d'étapes précédentes. Après $\omega + 1$ vient donc l'ordinal $\omega + 2$, puis $\omega + 3$, $\omega + 4$ et pour tout n, $\omega + n$. Et plus grand que tous les ordinaux précédents, nous avons l'ordinal $\omega + \omega = \omega 2$, constitué de ω étapes après lesquelles viennent ω autres étapes. On peut continuer comme ça et définir $\omega 2 + n$, puis $\omega 3$, $\omega 4$, $\omega 5$... et enfin ω^2. Si le vertige ne nous a pas encore gagné, il est clair que ω^2 n'est pas une borne aux infinis possibles, car il suffit de rajouter une étape supplémentaire à la fin, l'ordinal $\omega^2 + 1$. Et ainsi de suite, on définit ω^n, puis ω^ω, puis $\omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}}$, et nous arrêterons ici l'explication qui pourrait continuer *infiniment longtemps*.

Nous savons maintenant à quoi ressemblent les étapes d'un calcul infini. Il nous reste à définir quelles sont les règles que l'on peut utiliser, et en particulier aux étapes limites, c'est à dire les étapes qui n'ont pas d'étape immédiatement précédente contrairement aux calculs avec un nombre fini d'étapes ! Dans cette thèse, nous allons utiliser trois types de calcul en temps infini : la calculabilité d'ordre supérieur, l'α-récursivité et les machines de Turing à temps infini. Ces trois types de calcul à temps infini, s'ils ont beaucoup de points communs et paraissent surtout différents par la durée des plus long calculs, sont définis de manières très différentes. Les deux premiers viennent de la relation entre définissabilité et calculabilité que nous avons déjà présentée dans cette introduction. La calculabilité d'ordre supérieur est une variation de la définissabilité par des formules Δ_1^0, Σ_1^0 et Π_1^0, en des formules Δ_1^1, Σ_1^1 et Π_1^1, c'est à dire que les quantifications ne sont plus limitées aux entiers, mais aux réels. La calculabilité qui en découle peut être vue comme infinie, de temps borné par un ordinal particuler, ω_1^{CK} le premier ordinal non calculable. L'α-récursivité étend la quantification des formules à une bien plus grande classe d'ensembles, les constructibles de Gödel. Non seulement elle permet de borner la longueur des temps de calculs à un ordinal arbitraire α, mais elle permet aussi de calculer des ensembles d'ordinaux au lieu des ensembles d'entiers habituels. Enfin, les machines de Turing à temps infini prennent un tout autre

chemin, et ne se basent pas sur la relation entre définissabilité et calculabilité. Au contraire, elles consistent en une modification de la machinerie des machines de Turing pour leurs permettrent de s'exécuter le long des étapes infinies constituées par les ordinaux.

Avant d'attaquer l'aride résumé de la thèse et des principales contributions, il est temps de faire une pause rafraîchissante, et de reparler de l'hôtel de Hilbert. Celui-ci était construit sur une ligne de métro, qui partait du centre-ville, passait par les arrêts a_1, a_2, a_3... et enfin a_ω, où était l'hôtel. Cette ligne avait été construite par Cantor [2] pour montrer tous les avantages de l'infini actuel à ses détracteurs. La ligne était sobrement nommée le métro de Cantor. Chaque rame partait du centre-ville en direction de l'hôtel, et à chaque station, exactement une personne descendait s'il n'est pas vide, et invariablement 10 personnes montaient. Cependant, chaque jour un nombre différent de personnes arrivaient à l'hôtel, certaines fois finis, certaines fois infinis, et de temps en temps le métro arrivait même vide ! Le réceptionniste se demandait comment le même nombre de montées/descentes pouvait induire des résultats différents, et il se décida finalement à poser la question à Cantor, qui séjournait à l'hôtel en ce moment.

La réponse de Cantor fut la suivante : « Bien qu'à chaque arrêt le même nombre de personnes montent et descendent chaque jour, cela ne veut pas dire que ce sont chaque fois les même personnes qui font les mêmes actions. Les passagers arrivant à l'hôtel sont ceux qui sont rentrés dans le métro et ne sont pas sortis avant d'arriver à l'hôtel. Si chaque voyageur sort dès que parmi les autres personnes de la rame, il est celui à avoir voyagé le plus longtemps, alors personne ne restera indéfiniment dans le métro, car à tout moment le nombre de personnes dans le métro est fini, et quand une personne rentre elle aura donc un nombre fini d'arrêts à attendre avant de sortir. Si au contraire, à chaque arrêt sur les 10 personnes qui rentrent, 9 décident de rester indéfiniment et une seule décide de sortir au prochain arrêt, alors la rame arrivera pleine. Et dans la saison touristique, où une infinite de personnes montent et une seule descend à chaque station, la rame peut-elle aussi arriver vide d'après toi ? ».

Mais Cantor n'avait pas prévu de rame de retour, ce qui était bien embêtant, même si les astuces de Hilbert permettaient de toujours trouver de la place dans l'hôtel. Il était bien difficile d'en construire une qui passerait exactement par les mêmes arrêts qu'à l'aller, car l'annonce sonore « Prochain arrêt, a_γ » ne marchait pas au départ de l'hôtel. Ainsi, il décida de faire passer le métro par un autre chemin, et de construire les arrêts $a_{\omega+1}$, $a_{\omega+2}$, ... jusqu'àu centre-ville, l'arrêt a_{ω_1} où ω_1 est le premier ordinal non dénombrable. Ainsi, la ligne dessert de nombreux arrêts supplémentaires. Mais à la grande surprise de ce pauvre réceptionniste, décidément bien naïf, personne n'arrivait jamais au centre-ville, la rame était toujours vide quelque soient qui monte et qui descend à chaque arrêt [3]. La démonstration de ce fait est laissée en exercice au lecteur.

Résumé et contributions de la thèse

Cette thèse présente à la fois des résultats de mathématiques à rebours et d'aléatoire algorithmique, dont le dénominateur commun est l'importance de l'utilisation du calcul en temps infini. Elle est découpée en cinq chapitres, dont les deux premiers contiennent tous les préliminaires nécessaires, et les trois suivants sont le travail de thèse à proprement parler.

Le chapitre I se veut une large présentation de toutes les notions qui serviront dans cette thèse. Trop brève pour être une vraie introduction au sujet, elle rappelle les résultats, et souvent

2. L'histoire crédite plutôt Hilbert, mais j'ai décidé de le remplacer par Cantor, car Hilbert a déjà un hôtel, et Cantor n'eut pas une vie facile, entre sa dépression et les attaques de ses pairs sur son travail novateur et remarquable.

3. Qui suivaient toujours la rêgle suivante : à chaque arrêt 1à personnes montent toujours, et exactement une personne descend *sauf* si la rame est vide.

leur preuve, qui joueront un rôle suffisamment important par la suite. Ainsi, ce chapitre sert de rappel plus que d'introduction.

Le chapitre II se concentre sur l'introduction des différents types de calcul en temps infini que nous considérons ici : la calculabilité d'ordre supérieur dont les ensembles calculables sont les Δ^1_1, l'α-récursivité, et les machines de Turing à temps infini. Cette partie se veut aussi comme un préliminaire, mais elle est bien plus spécialisée que le chapitre précédent. Bien que soient présents quelques résultats originaux, la principale contribution de cette partie est de présenter de manière rapprochée, et dans une certaine mesure comparée, les différents types de calcul en temps infini existant.

Le chapitre III est l'étude du théorème de Hindman dans le cadre des mathématiques à rebours. C'est le seul chapitre où le calcul en temps infini ne joue pas de vrai rôle, mais c'est un chapitre important. Le théorème de Hindman est tiré de la théorie de Ramsey, il stipule qu'étant donné un coloriage des entiers en un nombre fini de couleurs, il existe un ensemble infini H dont toutes les sommes finies d'éléments distincts de H ont une même couleur. L'appartenance de ce théorème au Club des Cinq n'est pas connue, le seul résultat est qu'il doit être entre ACA_0 et ACA_0^+. D'un point de vue plus calculatoire, il existe une instance calculable dont toute solution calcule le problème de l'arrêt, tandis que toute instance a une solution calculable en la $\omega+2$-ième itération du problème de l'arrêt. L'étude dans ce chapitre se place à la fois du point de vue de la calculabilité, en étudiant un des passages clés de la preuve consistant en la construction d'objets combinatoires intermédiaires nommés *full-match*, et du point de vue de la théorie de la preuve en effectuant une analyse ordinale du théorème.

La contribution majeure de ce chapitre se trouve sans doute dans le théorème III.41 et les questions III.54 et III.46. Mettre une question comme contribution peut sembler étonnant, mais la raison est que cette question était censée être résolue, avant que le présent travail ne découvre une erreur dans la preuve, ainsi qu'une preuve que le résultat plus fort de la preuve erronée ne pouvait être vrai, contenue dans le théorème III.41. Cette découverte ouvre de nouvelles directions de recherche pour résoudre la question principale sur le théorème de Hindman. La seconde contribution est l'analyse ordinale de ce théorème, dont l'aboutissement est le théorème III.91.

Le chapitre IV est l'étude d'un niveau particulier du zoo des degrés de Weihrauch, situé autour du principe d'induction arithmétique transfinie. Ce principe, bien que très important en mathématiques à rebours, est encore peu étudié du point de vue des degrés de Weihrauch. En particulier les différentes manières de représenter ce principe dans cette hiérarchie ne sont pas encore toutes explorées. ATR est relié au principe de choix analytique, que nous étudions aussi dans ce chapitre, en particulier ses restrictions.

La contribution majeure de ce chapitre est clairement le théorème IV.100 séparant les principes d'axiomes de choix et de choix dépendant, les choix s'effectuant dans des sous-ensembles analytiques de $\mathbb{N}$. Cela résout une question posée par [13] et [47], à l'aide d'une version de ATR introduite par Goh [51]. Ce chapitre introduit aussi de nouveaux degrés reliés à ATR aux définitions IV.57, IV.64, IV.68, IV.70, et IV.71. Pour finir, une étude extensive des restrictions du choix analytique est effectuée, aboutissant aux Theorèmes, propositions et corollaires de la section IV.3, dont un résumé est présent dans la conclusion IV.3.5.

Finalement, le chapitre V est un développement du domaine de l'aléatoire algorithmique relativement aux machines de Turing à temps infini. Ce domaine a été initié par Carl et Schlicht [19] peu après le début de cette thèse, et ce chapitre est dans la continuation de leur travail. En particulier les versions ITTMs des notions standards d'aléatoire algorithmique sont définies, ainsi que leur équivalent d'α-récursivité. Ce chapitre se veut être le développement d'un cadre précis pour l'étude de l'aléatoire des ITTMs, en particulier il suppose le minimum d'admissibilité dans ses résultats afin de pouvoir les appliquer à l'ordinal Σ des ITTMs, connu pour ne pas être admissible.

Les contributions principales de ce chapitre sont de trois natures. La première est le développement d'un cadre précis pour l'étude de l'aléatoire en temps infini. La seconde est à nouveau une question, la Question V.43 sur la séparation de deux classes, une question dont la réponse paraissait si évidente qu'elle n'avait pas été investiguée dans le papier fondateur [19] introduisant les définitions. Enfin, la troisième consiste en une progression en direction de la résolution de la précédente question, comme le théorème V.37, et ainsi qu'une résolution de son équivalent catégorique à l'aide du Théorème V.59. Nous répondons aussi à plusieurs questions de [19] dans la section V.3, dont le théoreme V.41 en est un remarquable représentant.

Introduction

We introduce the subject of this thesis, going from the most general to the most specific, not in one big step but using many successive zooming, getting more and more detailed, eventually reaching a precise summary of the thesis and its contributions.

A mathematical work

This document presents a research work in mathematics. The goal of mathematics is to study abstract concepts, in order to better comprehend their properties and therefore the concept themselves. We start with an example: everyone, or at least most people, are familiar with the concept of "mean", and random events. So, one would easily say: "if I toss a coin, there is one chance out of two that I get head". Or "if I go to the casino, there are more odds that I leave less rich, than richer". These sentences suggest the existence of a notion of randomness, hard to define in the real world, which seems to us to follow a causality rule, therefore somewhat "unrelated" to the concept of randomness. But the mathematics allow us to define this notion, to make it actually exist in a precise way but in an abstract world. And once the concept is defined, we have the tools to study it, and unveil its most unexpected properties. For instance, probability theory teaches us that there are random events that do not have a mean! Think of a box, each time we press a button on it, a ball with a number written on it is outputted from the box, picked at random along all possible balls following a certain distribution. The random drawing is done independently of the previous ones. For some distribution, the mathematical study tells us that the concept of mean does not exist. One could be tempted to compute the mean of the outputted numbers, after a huge amount of drawings, but this computed mean would not stabilize: very rarely, a colossally big number would be drawn, increasing the computed mean, eventually reaching any bound fixed in advance on the mean.

Mathematics taught us something about the concept of probability, that would have been hard to guess when they were only an idea, an intuition on the way the world works. They taught us many more things on probability, many of them being verified and allowing us to understand better the world we live in.

Mathematical Logic

Start our first enlargement toward what interests us most in this thesis: mathematical logic. Mathematical logic is a particular domain of mathematics which corresponds to the study of a particular concept, the concept of mathematics itself. Recall that the first step in the study of an abstract concept is to define it formally, to make it exists concretely in an abstract world.

This field of mathematics makes therefore exist formally defined versions of the concepts of theorem, proof and theory in the world of math. These definitions are modeled exactly from

idea of what are mathematics, we are confident that what they model is actually what we had in mind, even though there is no way to prove it.

We are in presence of two types of theorems: those we just gave a formal meaning, and those we informally speak of when we do math. The latter ones are said being part of the *meta-theory*, the mathematics that we used in real life to study the *theory*, the math that we just defined.

Of course, if the mathematics of the meta-theory do correspond to the usual mathematics, we can define inside the latter the notions of proof, theorem and formula again. In other words, if the mathematics can study the mathematics, then the mathematics can study the mathematics studying the mathematics. It is time to stop and summary the essential: in logic, there are two types of mathematics, the one we study and the one we use to study. The former is a model of the latter, and both are distinguished.

This need of studying mathematics come initially from a need of solid and powerful foundation to mathematics: solid in the sense that they do not allow to prove one thing and its opposite, and powerful in the sense of being able to prove sufficiently many theorems. This goal guided the contributions of most early logicians, and many unexpected results were found. Despite a precise plan, made by Hilbert, which first had some success, Gödel showed that such perfect foundations do not exist. But good things were done: mathematical logic was born!

This thesis does not focus on foundations of mathematics. Indeed, even if it has been the initial motivation, mathematical logic is not bound to it. It allows the study of numerous new concepts, or older concepts from a new angle, and has many applications and connections with other fields of mathematics. This thesis stands in the context of computability theory, and it is time to focus our frame on it.

Computability theory

Computability theory is the study of the concept of computation. A computation can be seen as a successive application of simple rules, from an initial state and halting whenever a rule says so. For instance, addition as we learn to do it by hand corresponds to a computation: it is a successive application of addition of two digits, applied to the digits of two numbers in some way involving carries, ending when all digits have been considered with the sum of the two original numbers.

This concept of mechanical rule, also called "algorithm" from the name of the Persian mathematician Al-Khwârizmî, is at this point of the introduction only vaguely defined. Going from the concept to a reality in the mathematical world, satisfying and convincing, was not an easy thing! Indeed, it is not obvious to find a set of rules sufficiently powerful to include all that would be intuitively computable, without adding rules allowing too many possibilities. And for each candidate to such a definition, it is yet to convince peers that it is the one! Several definitions of the notion of computation have been proposed, some of them been not convincing by lack of computational power, others because they were overly complicated and unnatural, seamed to far from the initial intuition. Eventually, Turing made everyone agree with his eponymous "Turing machines", a kind of model for a primitive computer.

So, a formal definition for computation made it to a consensual state. Thanks to it, we are now able to define precisely what is a computable set: it is a set of integers A such that given an integer n, the problem of knowing if $n \in A$ can be solved using a simple computation, in the sense of Turing. We were able to prove that there exists non computable sets. More than that, Turing's machine allows to define a computation *relative* to a set of integers: in some ways, if B is a set of integers, the computability relative to B consists of the study of the computations that are possible if B is accessible to the computation. It allows us to abstract ourselves from the set of integers, and to speak only about the *computational content*. Two sets have the same

computational content if the notions of computability relative to one or the other are the same. The study of computational contents, named Turing degrees, has been in the center of concerns, and a very rich and complex structure was unveiled.

Let us now give a few examples of computational contents. Even if we have not formally defined the notion of computation, we appeal to the intuition, emphasizing that this notion corresponds exactly to what can be computed by a computer which would not be limited by its memory. Consider the set of even numbers, P. The set P does not help to compute anything new: we can do without it, as its information can easily been retrieved by a computation, checking if the last digit is a 0, 2,4,6, or 8. P has no particular computational content. However, consider now the set of computations that reach a halting state. It seems difficult to predict if a computation will stop, and there is no way to test it by executing the computation, as if it never stops, we will never know it. But having access to this set might be useful in some computation, to avoid ending in infinite loops. This set is known as the *halting problem.* Let us give one last example. The resolution of equations is one major problem of mathematics. Could we reduce their resolution to a mechanical step by step computation? In particular, in the case of a system of simple equations containing only integers, sums and multiplications, a theorem of Matiyasevitch tells us that it is not possible, and even more it tells us that the computational content needed to resolve these equations is exactly the one of the halting problem, not more, not less.

At this point, it is not even clear why computability theory is part of mathematical logic. This is due to the fact that there exists many links between computability, and definability. Thus, the computability of a set can be reduced to the fact that the set can be defined by a formula having certain properties. Define a Δ^0_1 formula as an arithmetical formula all of whose quantifiers (the parts of the form "there exists" and "for every") are bounded. Define Σ^0_1 formula as those that are allowed unbounded "there exists", and Π^0_1 those that are allowed unbounded "for every". Then, a set is computable if and only if it is definable both by using a Σ^0_1 formula, and by using a Π^0_1 formula. A set definable only by a Σ^0_1 formula is a *computably enumerable* set, or c.e. set, that is a set such that there exists a computation that enumerates exactly its elements. The halting problem is c.e. as it is definable using a Σ^0_1 formula: a computation halts if and only if "there exists a step t such that the computation halts after t steps of execution".

The links between definability and computability could be expected, as both are a notion of complexity for sets: one measure the difficulty to define it, while the other the difficulty to compute it. The measure of complexity of sets is a recurring theme in mathematical logic, central also in descriptive set theory which has also a role in this thesis.

With this analogy in mind between computability and definability, the next move is to apply the different methods, constructions and powerful tools developed during the study of Turing degrees, to other domains of mathematical logic where definability is a key element. Now, let's take a closer look on these domains.

Computability and definability

We are now at the most precise level of granularity of this introduction. At least three domains use computability to shed new lights on definability, and profit of its many methods of construction. We will present these three domains as a tracking shot: first, descriptive set theory, then reverse math and algorithmic randomness. We will end this tracking shot by a presentation of the common part to our study of these three domains, the initial point of this thesis: the notion of infinite time computations, itself also tied closely to definability, and therefore applicable to the above-mentioned domains.

Descriptive set theory

Descriptive set theory is the study of the complexity of subsets of $\mathbb{R}$, from a topological point of view. We will generally decide to associate an element of $\mathbb{R}$ to its base 2 decomposition: as such a real is assimilated to an infinite sequence of 0s and 1s, and we call the set of these infinite sequences the Cantor space, noted 2^ω.

The most simple subsets of the Cantor space, from the descriptive set theoretical meaning, are the sets in the form $\{x \in 2^\omega : x \text{ extends the string } \sigma\}$, where σ is a fixed finite string; or the set of reals whose base 2 development starts with a finite and fixed in advance sequence of bits. A *description* for such a set is simple: it suffices to give the sequence σ. Note that in $\mathbb{R}$, it corresponds to open intervals. And for good reasons: the open sets, as well as the closed sets, make the first level of complexity of descriptive set theory, just after the clopen sets.

The complexity level just after the open and closed sets is made of countable intersections of open sets, and countable unions of closed sets, traditionally named F_σ and G_δ. The next levels of complexity go on like this, by taking countable intersections, or countable unions, of sets of lesser complexity.

The motivations to classify subsets of the reals using their complexity comes, among others, from set theory. The goal is to show that the "not too complicated" sets behave well enough. Thus, even if sets of high complexity necessarily exist, most of the usual mathematics only manipulate sets of reasonable complexity. For instance, for sets in the hierarchy defined above, called the Borel hierarchy, are all determined, they do not contradict the continuum hypothesis, two property of well behaviour.

It does not seem at first sight to be any link between the topological complexity of the Borel hierarchy and computability, and yet they exist! To unveil them, one has to add some effectivity in the unions and intersections. The "effectively topological" complexity that we obtain has many good properties, such as having only countably many members of a given complexity. Once a result is proven one the effectively topological complexity, it is almost always possible to raise it to topological complexity by a technique called *relativization*: this technique uses the fact that the topological complexity is nothing more than the union of effectively topological complexities relativized to a real. The study of the effective topological complexity is called effective descriptive set theory.

Reverse mathematics

The second field of this thesis that benefits from computability is called reverse mathematics. Let us talk again of the origin of logic: the need for strong and solid foundations of mathematics. These foundation were built on what we call set theory, a theory where all the objects are sets, including numbers. It therefore implies an effort of coding the usual objects of math into sets. With the axioms of Zermelo and Fraenkel, and sometimes one additional axiom, all of the usual mathematics are provable in this theory. However, it implies the existence of gigantic objects, never showing up naturally by themselves, and the axioms of this theory do not correspond to the a reasoning step made in the proof of a theorem.

The reverse mathematics study not foundations of all mathematics, but restrict themselves to the reasoning made in second order arithmetic, that is mathematics using only integers and sets of integers. It includes a vast majority of mathematics developed until now. For instance, number theory and graph theory fit in this framework, as the objects they study are only integers, up to coding. Ramsey theory, studying structures emerging from infinite sets of integers, is widely studied from the point of view of reverse mathematics. Many other fields, even if they concentrate on integers that are neither integers, nor sets of integers, can still be studied in this framework:

it is the case of analysis in its vast majority, as for instance a computable function can be coded in a set of integers.

The plan of reverse mathematics to study second order arithmetic reasoning, was the following: given an important result of a given domain fitting in the framework, to isolate the axioms used in the natural proof, that is the steps of the proof that did not demand any explanation, which are true as they are "obvious"; then once these axioms have been isolated, to try to prove the axioms themselves assuming the result. Thus, if the theorem and its axioms are equivalent, we can say that they are the *minimal* axioms needed to prove the theorem. The name "reverse mathematics" comes from the technique of proof starting from the theorem and trying to prove the axioms, reversed from the usual way. However, the key problem of a reverse mathematical question can also be to find a proof using less axioms.

The biggest discovery in this study was that the vast majority of theorems developed during centuries, in many domains, are equivalent to one out of five axiomatic systems, each corresponding to a type of reasoning, or even to a philosophical program of foundation of mathematics [4]. These five axiomatic systems, known as the *Big Five*, correspond to the following principles:

1. The first one, RCA_0, is the axiomatic system needed to develop constructive mathematics. It corresponds to the most simple arguments, the ones which allow to actually build the object that we define. For instance, if we prove the existence of a set in this system, then the set is computable.
2. The second one, WKL_0, is the axiomatic system corresponding to the use of compactness in an argument, a way to reduce a proof on the finite to a proof on the infinite.
3. The third one, ACA_0, is the axiomatic system allowing to build objects that are not totally constructive, but whose construction only depends on arithmetical statements, the statement that only mention integers.
4. The forth one, ATR_0, is the axiomatic system allowing constructions by *transfinite* induction, that is step by step, but with the total number of steps needed exceeding the finite.
5. The fifth and last one, $\Pi^1_1\text{-}\mathrm{CA}_0$, is the axiomatic system allowing to construct object of a very high complexity. However, some theorems still need the existence of such an axiomatic system, making those constructions legit.

As we can see it, computability theory is not far in the previous principles. Indeed, the axiomatic systems correspond to computability concepts, and many constructions of this domain allow separation of theorems and axioms, that is allow to show that one theorem cannot imply an axiomatic system.

The involvement of the computability theory community in reverse mathematics allowed the discovery of the fact that we can define a "computable content" to a theorem, corresponding to its implications in terms of computability. For instance, consider the Bolzano-Weierstrass theorem, stipulating that an infinite sequence of bounded reals must admit a converging subsequence. Given a computable sequence, the converging subsequence may not be computable, and the computable content common to all of the converging subsequences is in some ways "implied" by the theorem. However, the framework given by reverse mathematics is not ideal for the precise study of computable content of theorems.

The good framework, the tool that was needed to study the computational content of theorems, actually already existed in analysis: it is the Weihrauch reduction. This reduction allows in particular to compare the theorems of the form "for every something, there exists an other

4. They are : the constructivism of Bishop, the finitistic reductionism of Hilbert, the predicativism of Weyl and Fefermann, the predicative reductionism of Friedman and Simpsons, and finally the impredicativism of Feferman *et al.* [60]

thing, such that some property is true". The idea is to consider these types of theorems as problems, with instances of which one has to find a solution. Under this angle, the theorem says that all instances have a solution, and the computational content of such a theorem corresponds to the difficulty of finding a solution given the instance. We can then use Weihrauch reduction to compare the difficulty of the resolution of two problems: a problem P_1 is easier than a problem P_2, if given an instance of P_1, I can find a solution if I am given a box solving the instance I want of P_2, but that I can use only once.

Algorithmic randomness

The last important domain of this thesis, still in direct relation with computability, is algorithmic randomness. We all have an idea of what would look like a sequence of "head" and "tail" obtained by tossing 1000 times a coin. However, all sequences of 1000 tossing of a coin have the same odd to happen. Then, why does the sequence "head, head, head, head..." seems less random than the sequence "head, tail, tail, head, tail, head, head,..."? There is a dissonance between our intuition of a random sequence and of our intuition of probability. When probability theory studies the latter, algorithmic randomness concentrate on the concept of random sequences, conciliating it with probability.

How to formally define what is a random sequence? In agreement with the example of the previous chapter, we begin by discussing the case of finite sequences (presumed long). In addition to the sequence constantly "head", many other sequences should not be considered coming from a random process, such as for instance the sequence made of a succession of "head, tail" repeated. The sequence of digits of π taken from the 100-th decimal can seem random at first glance, but not at all when we realize this fact. All these very long sequences have been defined using only simple words, that is using a short sequence of letters. But, it would be impossible for me to give an example of a seemingly random with so few words, intuition telling us that the odds of obtaining such a string out of all possibles strings is very low!

This is how we define randomness for finite strings: more precisely, it is *measured* by the length of the smallest description allowing to retrieve the string, the Kolmogorov complexity. This complexity cannot exceed the length of the sequence, as giving the entire sequence is a description of it. And all sequences of length n will have as complexity at least the complexity of n. The more the complexity of a string is close to n, the more it can be considered random; and the more it is close to the complexity of n, the less it seems random.

If we call "infinite random sequence" a sequence obtained after the infinite repetition of tossing a coin, then at each step the finite sequence of tossing will have a measure of its randomness close to the maximum, that is its Kolmogorov complexity will be close to n. This definition, called Martin-Löf randomness, is one of the most important, as it can be characterized by three possibles approach, all of them reasonable:

1. *Incompressibility*: the approach by prefix complexity, is the one that we just used to define a random sequence. If a sequence is obtained by an infinite repetition of tossing a coin, at each step it will have to seem random, and conversely if it seems random at each steps, it will seem random as a whole.
2. *Imprevisibility*: a sequence seems random if it is impossible to guess which will be the next element, given its predecessor, with more than one odd out of two. More precisely, if at each step we can bet some amount that we chose on the next element of the sequence, we mustn't be able to be rich.
3. *Banality*: a random sequence mustn't have any exceptional property. An exceptional property is a property that we do not expect a random sequence to have, such as the fact of always having two "head" in a row. And if probability theory does not tells us why some

sequence look more random than others, it does tell us what are the properties that it is unlikely to have: the properties of measure 0. Out of these properties of measure 0, we only chose the not too complex ones, as every sequence is in a measure 0 set, such as $\{x\}$.

Martin-Löf randomness is a robust notion with different equivalent definitions, each following one of these paradigm. But that does not mean it is a perfect definition for our intuition of randomness, in the sense that there exists sequences that should not be considered random, but still are Martin-Löf random. For instance, they should not have an arbitrary powerful computational content, and yet a theorem of Kučera and Gács shows that every set, no matter its computational content, is computable by some Martin-Löf random. This notion is therefore "not strong enough" to represent our intuition perfectly.

There exists many other randomness notions stronger and weaker, a full hierarchy of randomness, all defined using one of the three paradigm given above. Algorithmic randomness as a field is the study of all these notions, their strength, and their weakness.

Infinite time computations

It remains to tackle the last important point in order to understand what it is about in this thesis. It is the spectrum through which we will investigate the domain previously mentioned, the *infinite time computation*. Indeed, given the benefits computability theory bring to other fields, it is justified to expect the same kind of benefits from variations of computability.

But what would be the intuition of infinite computations? It can be hard to speak about intuition, as not being something we meet a lot in every day life, infinite is not the most obvious concept. And yet, it is very present in mathematics, it can even be *essential* to some results, not even mentioning it in their statement. In order to familiarize a neophyte, begin by reviewing several infinities.

The first type of infinity is a "pseudo" infinity, a potential infinity, in the sense that it is never reached. It is the infinity corresponding to the statements of the type:

$$\lim_{n\to\infty} \frac{n^2}{2n+1} = \infty.$$

This expression can be read "as n *goes* to infinity", and not "*is* infinity", as it would not even make sense of n to equal infinity. Going to infinity means to get closer and closer to it, never stopping, with no bound on the convergence. This type of infinite would not help us to define an infinite computation, as such an infinite computation for this type of infinite would never actually reach an infinite step, it would never be able to benefit from the infinitely many steps just done.

The first to define an actual infinite is Cantor, which earned him a huge resistance from some of his colleagues. However, there exists a simple type of infinite, as the size of a set, that we can see as an actual infinite: the set of natural numbers is infinite. This notion as the "number of elements" of a set is called the *cardinality*: a set with five elements has a cardinality of 5, while $\mathbb{N}$ has an infinite cardinality. Two sets A and B will have the same cardinality if we can put the elements of A and B in a one to one relation, such that every element of A is in relation with exactly one element of B, and reciprocally. Thus, even if the set $\mathbb{N}$ strictly contains the set of even numbers $2\mathbb{N}$, these two sets have the same cardinality: we can put in relation the element n of $\mathbb{N}$ with the element $2n$ of $2\mathbb{N}$. Cantor showed that there exists many infinite cardinalities, for instance the cardinality of $\mathbb{R}$ is strictly bigger than the cardinality of $\mathbb{N}$.

Let us use this part on cardinality to speak about an imaged illustration of the amazing properties of the infinite cardinality. Hilbert, a famous German mathematician, decides one day to open a Hotel that he simply named: Hilbert's Hotel. But Hilbert is not a man to do things in

small size, and his hotel has infinitely many rooms! They are numbered r_1, r_2, r_3... and so on. And its an absolute success! And it is also very useful, as the following anecdote suggests: One day of heavy rain, a lonely traveler makes it to the hotel, wet to the bones. But the hotel is full this night, and the receptionist does not help! On the insistence of the traveler, he calls Hilbert, hoping that as a mathematician, he might have a solution. And he does! Hilbert simply asks the occupant of the each room r_n to move to the room r_{n+1}, which has been freed by the same process. And as there is no room r_0, the room r_1 is now empty!

An hour later, a group of scouts reaches the hotel. Hilbert, who welcome everyone and even the scouts, is ready to redo the same operation several time to accommodate the group. He asks the Great Scout how many people are there in the group. The Great Scout starts to count: "Let's see, there is scout c_1, c_2, c_3... we are an infinity!". How will we do this, thinks the receptionist, the hotel is already full! But already Hilbert has a solution: he asks to the occupant of each room r_n to move to the room r_{2n}, freeing in this way all odds rooms, and send the scout c_i to the room r_{2i+1}. We leave to the reader the care to solve the problem that happened this exact same night, where an infinity of buses b_1, b_2, b_3,... with the bus b_i being full of tourists t_1^i, t_2^i, t_3^i... queued at the hotel. This problem left a horrible headache to the receptionist.

This digression being already too long, let us now speak about the type of infinite that will allow us to define infinite computations: the *ordinal*, or *well-founded*, infinite. Indeed, the concept of cardinality does not seem to fit the need to define an infinite computation, which corresponds to a sequence of steps characterized by the application of a simple rule, halting on simple conditions. The decomposition in steps suggests that the infinite we need has the form of an ordered set: each element of the order will correspond to one step of computation, and the order will be infinite. This implies a much wider variety of infinity that in the case of cardinality, which allowed us to reorder the elements, just as when Hilbert moves the occupant of the rooms!

Out of the ordered infinities, only some are of interest for us, those that allows to define each step of computation from the previous steps. These particular types of linear orders are called *well-founded* orders, and each of them has a representative called *ordinal* containing only the essence of the order. To understand infinite time computations, one has to understand well this notion. We will call in this introduction "steps" the elements of an ordinal. The ordinals themselves are (well) ordered, the first one of them are the finite ordinals, corresponding to a finite number of steps, one after the other. After all these finite ordinals, comes the first infinite ordinal, usually written ω. This ordinal corresponds to the same order as the one of the integers, that is an infinity of steps, one after the other, like an infinite ladder. It's the ordinal corresponding to the potential infinite of "going to infinity", never reaching it. After ω there is $\omega + 1$, the ordinal consisting of all the steps of ω and a supplementary step after all the others: its the first ordinal where a step actually reached infinity. But an infinite computation does not stop here, it may need several other steps to benefit the infinitely many previous steps already done. After $\omega + 1$ comes the ordinal $\omega + 2$, followed by $\omega + 3$, $\omega + 4$, and for all n, $\omega + n$. Above all of them lies $\omega + \omega = \omega 2$, consisting of ω steps, followed by ω other steps. We can continue and define the ordinals $\omega 2 + n$, $\omega 3$, $\omega 4$, $\omega\omega = \omega^2$. Again, there is no reason for ω^2 to be a bound, as it suffices to add a step after it to have a bigger ordinal, $\omega^2 + 1$. And so forth and so on, we define, ω^n, ω^ω, and $\omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}}$. We stop here our examples of how far the infinite ordinals can go, before it becomes *infinitely long*.

We now know what kind of steps will be in an infinite time computation. It remains to define what are the simple enough rules that we can use, and in particular at the limit steps, that is at the steps which do not have one immediately preceding step, unlike the steps of a finite computation! In this thesis, we will use three types of infinite time computations: higher computability, α-recursion and infinite time Turing machines. These three types, if they have many common points and seem different only by the length of the allowed computations, are defined in very

different ways. The first two come from the relation between definability and computability that we already mentioned. Higher computation is a variation of definability by Δ^0_1, Σ^0_1, Π^0_1 formulas, in Δ^1_1, Σ^1_1, Π^1_1 formulas, that is the unbounded quantifications are no longer limited to the integers, but range over the reals. The computability which surprisingly results from these definitions can be seen as computation with infinite time but still bounded by an ordinal, ω_1^{CK} the first non computable ordinal. The α-recursion follows the same principle, but extends the quantifications of the formulas to a much larger class of sets, the Gödel's constructibles. Not only it defines infinite time computation with time bounded by an arbitrary large α, but it also allows to compute sets of ordinals instead of the sets of integers usually computed. Finally, infinite Time Turing machines take a different path, and do not bases itself on the relation between definability and computability. Rather, it modifies the hardware of a regular Turing machine to allow them to be executed along ordinal stages.

Before starting the arid summary of the thesis and its principal contributions, we take the time to make a refreshing break, and to speak again about Hilbert's Hotel. This hotel was located near a subway station of a line starting from the center of the city, and going through the stops s_1, s_2, s_3... and s_ω, the nearest stop to the hotel. This line was built by Cantor [5] to show his detractors all the advantages of an actual infinite. The line was simply called Cantor's subway. Each train left the center in the direction of the hotel, and at each station, invariably 10 persons get on the train, and exactly one person get off the train. However, each day a different number of people get to hotel, sometimes a finite number, sometimes an infinite number, and sometimes even an empty train! This puzzled the receptionist, who wondered how the same number of ins and outs in the subway could lead to different outcomes in the station s_ω. He decided to ask Cantor, who was in the hotel at this moment.

Cantor's answer was the following: "Even if at each stop the same number of people get in and out of the train each day, this does not mean that it is the same people who do the same actions each days. The passengers arriving to the hotel are those who entered the train at an earlier step, and never left before the hotel. If each passenger decide to leave once he is the first, out of the people with him in the train, to have entered it, then no one will stay until the hotel: indeed when someone enters the train there are finitely many people there, so once they all have leaved he will also leave. If, contrary to this, of the 10 passengers entering at each stops, one of them decides to leave at the next stop while the 9 others decide to stay until the hotel, then the train will arrive full. What about in the touristic season, where infinitely many people get in and only one gets out at each station, do you think the metro can still reach the hotel empty?".

However, Cantor did not plan a way to come back, which was annoying even if Hilbert's always found some tricks to host the new people. It was quite difficult to build one that would go through the same stops in reverse order, as no one knew what should the announcement for the next stop say when leaving the hotel: "next stop: a_n" was wrong for all n. Therefore, Cantor decided to make the way back using another path, with stops $a_{\omega+1}$, $a_{\omega+2}$, $a_{\omega+3}$... until the city center, a_{ω_1} where ω_1 is the first uncountable ordinal. This way, several new people get to have a subway station near their house. But at the great astonishment of this poor receptionist, decidedly naive, no one never reached the city center, no matter what the passengers did [6] ! The proof of this fact is left to the reader as an exercise.

5. The history credits Hilbert, but I decided to replace him with Cantor, as Hilbert already owns a hotel, and Cantor did not had an easy life, from his depression to the attacks of his peers on his remarkable work.

6. They still followed the rule that we precise in this case: at each steps, 10 people get in the train, and exactly one gets out *except* if the train is empty.

Summary and contributions of the thesis

This thesis presents both results in the field of reverse mathematics and algorithmic randomness, with the common point being the importance of infinite time computations. It is split in five chapters, the first two of them contain all the necessary preliminaries, and the three last ones contain the actual work of this thesis.

Chapter I presents itself as a large presentation of all the notions that will be helpful in this thesis. Too short for being a true introduction to a neophyte, it still has the merits of recalling many results, and often proofs, that will play an important role in the later parts.

Chapter II focuses on the introduction of the several types of infinite time computations that we will use: higher computability whose recursive sets are the Δ^1_1 sets, α-recursion, and infinite time Turing machines. This part can also be viewed as a preliminary, but much more specialized than the previous one. Even if some original results are included, the main contribution of this chapter is to present in a common document, and in a somewhat compared manner, the main notions of infinite time computations.

Chapter III is the study of Hindman's theorem in the framework of reverse mathematics. It is the only chapter on which infinite time computations do not play an important role, but it is nonetheless an important chapter. Hindman's theorem comes from Ramsey's theory, it states that given a finite coloring of the integers, there exists an infinite set H, such that all finite sums of distinct elements of H share the same color. The belonging of this theorem in the Big Five is unknown, the only result is that its strength has to be between ACA_0 et ACA_0^+. In a more effective point of view, there exists a computable coloring such that every solution H for Hindman's theorem would compute the halting problem, while we know that every computable instance has a solution computable in the $\omega + 2$-th iteration of the jump. The study in this chapter is both from the point of view of computability, by studying one of the key part of the proof consisting in the construction of an intermediate combinatorial object named *full-match*, and from the point of view of proof theory by making an ordinal analysis of the proof.

The main contribution of this chapter certainly is Theorem III.41 and Questions III.54 and III.46. Emphasizing a question as a main contribution may sound strange, but the reason is that the question was supposed to be solved, before this work exhibit an error in the proof, as well as the proof that the stronger result in which lied the error was in fact false, as attested by Theorem III.41. This discovery opened new research directions toward solving the main question on Hindman's theorem. The second contribution is the ordinal analysis of the proof of Hindman's theorem, of which TheoremIII.91 is the conclusion.

Chapter IV is the study of a particular level in the zoo of Weihrauch degrees, located near the principle of Arithmetical Transfinite Reduction. This principle, despite its importance in reverse mathematics, is not well known from the point of view of Weihrauch degrees. In particular, the different ways to embed this principle in a Weihrauch degree are not yet fully explored. ATR is linked with the analytic version of the axioms of choice, that we also study in this chapter.

The main contribution of this chapter is with no doubt the Theorem IV.100 separating the axioms of choice and dependent choice, each choice being made from an analytic subsets of $\mathbb{N}$. This solve a question asked in [13] and [47], using a version of ATR introduced by Goh [51]. This chapter also introduces new degrees related to ATR with Definitions IV.57, IV.64, IV.68, IV.70, and IV.71. Finally, an extensive study of the restrictions of analytic axioms of choice is made, leading to the theorems, propositions and corollary of section IV.3, which are summed up in the conclusion section IV.3.5.

Finally, chapter V is a development of the domain of algorithmic randomness for infinite time Turing machines. This domain was initiated by Carl and Schlicht [19] soon before the beginning of this thesis, and this chapter is a continuation of their work. In particular, the ITTMs versions

of standard randomness notions are defined, as well as their equivalent in α-recursion. This chapter aims at being the development of a precise framework for the study of randomness in ITTMs, in particular the fewest requirement are made in the statement of its results, in order to be able to apply them to the ordinal Σ which is known to not be admissible.

The main contributions of this chapter are of three different natures. The first one is the development of the precise framework for randomness in infinite time. The second is Question V.43 on the separation of two classes, a question whose answer seemed so obvious that it had not been investigated in the founding paper [19] introducing all the definitions. Finally, the third one consists of a progression toward a resolution of the precedent question, as in Theorem V.37, as well as a resolution of its categorical counterpart in Theorem V.59. We also answer several open questions of [19] in section V.3, among which Theorem V.41 is an important one.

Chapter I

Preparation

I.1 Basics of Mathematical Logic

Gotlib, RàB T2. Le matou matheux

Every mathematical field is centered around the study of a particular concept. For instance, number theory studies the concept of numbers, with regards to addition and multiplication, with basic objects being the integers. From the many fields of mathematics, we can give a few examples: analysis roughly consists of the study of functions of real numbers; while algebra studies algebraic structures such as groups, rings and fields... An introduction to any of these field starts with defining the most basic objects, the object that are to be studied by the field, from function to group or probability measure. Then, if Mathematical Logic is centered around the notion of logic, what are the objects that should be defined in the first place?

Logic refers to reasoning, and therefore proofs, theorems, theories and formulas. This means that these notions will be our basic objects, and there will be theorems about theorems, definitions about definitions, proofs about proofs... There seems to be some kind of circular definition: in order to define a formula, one already has to use formulas. In fact, we will only define a "model" of our intuition of formulas, just as in probability theory, when we model our intuition of probability. So there will be the "formulas", the "proofs" and the "theorems" that we define and study as mathematical objects. To distinguish them from the usual meaning of formulas, proof and theorems, we will call the latter members of the *meta-theory*. The meta-theory is the framework in which we do mathematics, whereas the theory will be some object that we study.

Therefore, we need to give a formal meaning to these syntactic notions that we always use to write mathematics.

I.1.1 Formulas

The first thing to define is the language, in which the formula will be defined.

Definition I.1

A *language* $\mathscr{L}$ is a couple $((f_i)_{i\in I}, (P_j)_{j\in J})$ where each f_i is a symbol for a function with a certain arity, while each P_j is a symbol for a predicate, for arbitrary set of indices I and J.
The 0-ary functions of $\mathscr{L}$ are called constants. Terms are formed syntactically from variables, constants and functions of positive arity.

For instance, the *language of arithmetic* consists of the constant 0, the 1-ary successor function S, the 2-ary functions $+$ and $\times$ and the 2-ary predicate $\leq$. The *language of set theory* consists just of the binary predicate $\in$. Another example is the language of group theory, with a binary function for the group operation, and optionally a constant for the neutral element.

Given a language, we are able to build formulas.

Definition I.2

- A *formula without quantifier* in a language $\mathscr{L}$ is a formula built using the predicate "$=$", predicates of $\mathscr{L}$ and logical connectors such as $\wedge$, $\vee$, $\neg$, $\implies$, and the constant $\perp$ (meaning "false").
- A *first order formula* in a language $\mathscr{L}$ is a formula where we can additionally use first order quantifiers.
- A *second order formula* in a language $\mathscr{L}$ additionally allows second order quantifiers and the predicate $a \in X$ if a is a term of $\mathscr{L}$ and X is a second order variable.
- A *sentence* is a formula with no unbounded variable.

I.1.2 Theories and Models

For the moment, we only have defined formulas in a syntactic way. That is, we defined everything without specifying any "meaning". Of course, the meaning, or semantic, of everything such as the logical connectors and quantifiers is clear to any mathematician.

In the following, we give two meanings to these connectors. One is how they behave in proofs, the second is how they are interpreted in models. As we use their classical logic interpretation, which is the one of usual mathematics, we omit the details in the definitions.

Syntax

Definition I.3

Let $\mathscr{L}$ be a language, T be a set of formulas of $\mathscr{L}$ and ϕ a single formula. We say that T *proves* ϕ, in symbol $T \vdash \phi$ if there exists a proof of ϕ using as axioms only the formulas in T and the axiom of equality.

For example the logical connector $\wedge$ is defined as a symbol, but we give it some meaning in the previous definition by specifying how it works in proofs: $A \wedge B$ proves A and proves B, and if T proves A and proves B, then T proves $A \wedge B$.

A theory is then just a collection of sentences: the axioms of a theory.

Definition I.4

A *theory* on a language $\mathscr{L}$ is a set of sentences from the language. A theory T is *consistent* if there is no proof of $\perp$, otherwise it is *inconsistent*. A theory is *complete* if for every formula ϕ of $\mathscr{L}$, either $T \vdash \phi$ or $T \vdash \neg\phi$.

Usually, the axioms define the behaviour of the predicates and function symbols of a language. For instance, the axioms of group theory define the behaviour of the operator symbol. Two different theories on the same language can give different meaning to the same symbol.

Models

We now take an orthogonal approach: instead of defining the meaning by axioms (logical or in theories), we give them an interpretation in a model.

Definition I.5

Let $\mathscr{L} = ((f_i)_{i\in I}, (P_j)_{j\in J})$ be a language. An *$\mathscr{L}$-structure* $\mathcal{M}$ is a tuple $(M, (g_i)_{i\in I}, (Q_j)_{j\in J})$, where M is any set, $g_i : M^{n_i} \to M$ where n_i is the arity of f_i and $Q_j \in \mathcal{P}(M^{m_j})$ where m_i is the arity of P_j. We call g_i the interpretation of f_i, and Q_i the interpretation of P_i.
A *second order $\mathscr{L}$-structure* is an $\mathscr{L}$-structure $\mathcal{M}$ with an additional set $S \subseteq \mathcal{P}(M)$.

Similarly to when we define proofs, the meaning we give to the symbols lies on the following definition. Once again, we do not give the details on the interpretations of usual connectors and quantifiers.

Definition I.6

Let $\mathcal{M}$ be an $\mathscr{L}$-structure, and ϕ be an $\mathscr{L}$-sentence. We say that ϕ *is true in* $\mathcal{M}$ and write $\mathcal{M} \models \phi$ when ϕ is true in M when interpreting functions and predicates by their $\mathcal{M}$-interpretations. We sometimes also allow to add elements of M as constants in the language $\mathscr{L}$, and if a formula uses such constants it is said to be *with parameters*.
If $\mathcal{M}$ is a second order $\mathscr{L}$-structure and ϕ is a second order $\mathscr{L}$-formula, then when ϕ is interpreted in $\mathcal{M}$, the second order quantifiers range over the set $S \subseteq \mathcal{P}(M)$ of $\mathcal{M}$.

Definition I.7

We say that an $\mathscr{L}$-structure $\mathcal{M}$ is a *model of* T, and write $\mathcal{M} \models T$, if for every sentence $\phi \in T$, we have $\mathcal{M} \models \phi$.

For instance, in the language of groups, one can define the theory T containing the axioms of a group. Then, a model of this theory is any group. Closer to logic, $\mathbb{N}$ is a model of Peano Arithmetic, a theory containing some rules about addition and multiplication, as well as induction. There exists other models of Peano Arithmetic, that we call "non-standard models".

I.1.3 Gödel's code

We previously defined every syntactic notions, such as proofs and formulas, using the definition of a language. However, all of these are really just a sequence of symbols over a finite alphabet, containing for instance variables, logical connectors, but also parenthesis. Some se-

quences are well-formed, some others might not be. There is a way to combine those sequences of symbols by assigning a fixed number to each of them.

Gödel made this encoding of finite sequences in a clever way, so that it is possible to create integer codes for sequences, and then recover every element of the sequence, in a very effective way. Using this, he was able to show that already in Peano arithmetic, the statement "is provable in PA" taking a code for a formula as an input, could be defined.

In this thesis, we will call Gödel code an integer, coding a finite sequence (such as a formula), such that it is easy from the code to recover the initial object.

I.2 Computability Theory

I.2.1 Basics and Turing Machines

What would be a formal definition of the computable functions, for our intuitive notion of "computation in the real world"? This question, in our modern world full of computers, seems easier now than when it was first asked. It took many efforts from great scientists to give a convincing answer, and there has been several attempts. Some have failed to model all the computable functions, such as the now called "primitive recursive" functions. Some other were not convincing enough, such as Herbrand-Gödel's general recursive functions [1]. Soon after, Turing gave the definition of his now famous Turing Machines, a model of computation much closer to what it should represent (a computation in the real world), leading to the Church-Turing Thesis:

▶ **Thesis I.8** (Church-Turing Thesis)**.** The Turing machine model succeeds in modelling what we mean by computations in the real world.

Of course, it is not possible to prove or disprove this thesis. One can only give it credit, or discredit, by comparing specific example of what a Turing machine can and cannot compute to what we expect to be computable and uncomputable. In this document, we take it as granted and do not distinguish the Turing machine model and the natural notion of computation.

We first give a formal definition, and then an informal one. Usually, the informal definition is much easier to understand, and the actual coding of the concept into a formal definition do not add any value. The reason we care to still give a formal definition of a Turing Machine here, as a 5-uplet, is that the important Definition II.40 of Infinite Time Turing Machine in Chapter V will be based on Turing Machines. Of course, we want this central definition to be formal in order to have a strong basis to the theory.

Definition I.9

- A *(one-tape) Turing machine* can be formally defined as a tuple $M = \langle \Sigma, Q, q_0, q_h, \delta \rangle$ where
  - Σ is the alphabet;
  - Q is a finite, non-empty set of "states";
  - $q_0, q_h \in Q$ are the "initial state" and "halting state";
  - $\delta : (Q \setminus \{q_h\}) \times \Sigma \to Q \times \Sigma \times \{-1, 1\}$ is a partial function called the "transition function".
- A *configuration* is a triple (t, q, h), where t is an infinite word in the alphabet Σ and corresponds to the values on the tape; q is a state corresponding to the current

1. The general confusion between recursive and computable comes from this terminology.

state of the machine; and h is an integer corresponding to the current position of the head on the tape.

— A *computation* is a sequence of configurations $(c_i)_{i<n}$ following the rules of δ. A computation starts with the head at position 0 and the initial state q_0. If $c_i = (t_i, q_i, h_i)$, then $\delta(q_i, t_i(q_i)) = (q_{i+1}, a, b)$ where $h_{i+1} = h_i + b$ and $t_{i+1}(n) = t_i(n)$ for $n \neq h_i$ and $t_{i+1}(h_i) = a$.

— A computation is *halting* if it reaches the halting state q_h.

— Let cod be a coding of integers into finite strings of 0 and 1. If M is a Turing machine, we write $M(n) \downarrow$ if there exists a halting computation starting with $\mathrm{cod}(n)$ written on the tape. We write $M(n) = m$ if it halts, with $\mathrm{cod}(m)$ written on the tape.

— A function $f : \mathbb{N} \to \mathbb{N}$ is *computable* if there exists a Turing Machine such that for every n, $M(n) \downarrow = f(n)$.

Informally, a Turing Machine is finite set of instructions, with an unbounded memory. The finite set of instructions corresponds to the code of a program. Each instruction can only read one bit of the memory, so any finite computation depends on only a finite part of the tape.

We note that this model is very robust, as long as it keeps infinite memory, with few constraint on its access. For instance, one can consider Turing machines with several tapes, any finite alphabet, they are all equivalent in terms of computability. Tapes can be also bi-infinite. Using a coding of n-tuples of integers into a single number, one can define computable functions with multiple inputs.

Since then, many models of computation equivalent to Turing Machines have been defined. Some of them are programming language that are used for actual programming, such as assembly, C++. Some others are more abstract, such as recursive functions and various lambda-calculus. One important thing shared between all these models is that the execution of some computation is not too complex, in other word, is itself computable. This can be formalized by the fact that there exists an interpreter, a program that executes a computation given the code of a program.

To code a Turing machine, we will use the Gödel encoding. Then, a Turing machine that can execute any other one given its code, will be called a universal Turing machine.

Definition I.10

Given a Gödel code $e \in \mathbb{N}$ for a Turing machine, we will write φ_e for the corresponding Turing machine. Therefore, $(\varphi_e)_{e\in\mathbb{N}}$ is an enumeration of all Turing machines. We write $\varphi_{e_0} \simeq \varphi_{e_1}$ if they are equal as partial functions, that is they are defined on the same inputs, and if they are defined they have the same value.

Theorem I.11 (Existence of universal Turing machines)

There exists a Turing machine U such for every Turing machine φ_e and input n, $U(\langle e, n\rangle) \simeq \varphi_e(n)$.

This is clear assuming Church-Turing Thesis. Not only it is computable to execute φ_e given a code e, but it is also computable to manipulate those codes. For instance, given two codes e_0 and e_1 it is possible to give the code e_2 of the composition of φ_{e_0} and φ_{e_1}. One of the historically most famous theorem on manipulating codes is the S-m-n Theorem, stating that we can hard-code an input into a code. In other words, there exists a computable S such that if e is a code

for a Turing Machine of two inputs, then for all m and n, $\varphi_{S(e,m)}(n) \simeq \varphi_e(m,n)$.

Again, those code manipulations are natural for anyone used to programming, assuming Church-Turing thesis. There is one theorem that is more surprising and deserves attention: Kleene's recursion theorem.

Theorem I.12 (Kleene's recursion theorem)

Let f be a computable function. Then, there exists some $e \in \mathbb{N}$ such that $\varphi_e \simeq \varphi_{f(e)}$.

Proof. Let s be a computable function such that $\varphi_{s(e,n)}(m) \simeq \varphi_e(n,m)$. Let $e \in \mathbb{N}$ be a code such that

$$\varphi_e(n,m) \simeq \varphi_{f(s(n,n))}(m).$$

Then, we claim that $s(e,e)$ is a fixed-point in the sense of the theorem: we have that

$$\varphi_{s(e,e)}(m) \simeq \varphi_e(e,m) \simeq \varphi_{f(s(e,e))}(m).$$

■

The consequence of this theorem is that, when defining a Turing machine, one can always suppose that it has access to its own code, regardless on how circular this definition may seem. For instance, here is how we define a Turing machine that outputs its own code. Let f be the computable function, which to any n associates a code for a Turing machine outputting n on any input. Let e be a fixed-point of this. Then, $\varphi_e \simeq \varphi_{f(e)}$ is a Turing machine which outputs e on any input. These machines are called "quines".

I.2.2 Oracles and reducibilities

One of the nice feature of Turing machines is that it is really easy to relativize the notion of computation.

Definition I.13

We call *Turing machine with oracle*, or *Turing functional* a Turing machine with an additional read-only tape, called the oracle tape. A *computation with oracle* X is any usual computation that additionally starts with X written on the oracle tape. If e is the code for a Turing machine with oracle, we write $\Phi_e^X = Y$ if Φ_e^X is the characteristic function of Y. We suppose $\Phi_e^\emptyset = \varphi_e$.

The following definition is maybe the most important of computability, allowing to define degrees of unsolvability, corresponding to the "difficulty of computing an uncomputable program":

Definition I.14 (Turing and many-one reducibilities)

We say that $A \subseteq \mathbb{N}$ is *Turing reducible to* $B \subseteq \mathbb{N}$ if there exists an oracle Turing Machine Φ_e such that $\Phi_e^B = A$. In this case, we write $A \leq_T B$.
We say that A is many-one reducible to B if there exists a computable function f such that for every n, we have $n \in A$ iff $f(n) \in B$. In this case, we write $A \leq_m B$.

Many one reducibility is a special case of Turing reducibility where the Turing machine can only check one cell of the oracle tape and output its value directly.

▶ **Proposition I.15.** *The relations $\leq_T$ and $\leq_m$ are pre-orders.*

Definition I.16 (Turing degrees)

The relation $\equiv_T$ defined by $A \equiv_T B$ if $A \leq_T B$ and $B \leq_T A$ is an equivalence relation. We call *Turing degree* a degree in this equivalence relation.

A Turing degree corresponds to a computational power: all of its members can compute exactly the same sets. The study of the structure of Turing degrees has kept computability theorist busy for a long time, and many things are still unknown. The first easy thing is to show that the degrees are not all 0, the degree of computable sets, an easy statement that we will see in the next section. Another important question that lead to the development of an essential technique, is the existence of two incomparable degrees, solving Post's problem of whether there exists non-recursive incomplete c.e. sets, named after Emil Post.

There is an easy counting argument to show that there exists a non-computable set: there are only countably many Turing Machines, while there are uncountably many subsets of the integers. Therefore, most of the sets must be non-computable. However, this does not give an example of a non-computable set. The first and most prominent such example is the halting set.

Definition I.17 (Halting problem, Jump)

The *halting problem* is, given a code for a Turing Machine, the problem of deciding if the Turing machine will stop. The corresponding set is:

$$\emptyset' = \{e : \varphi_e(e) \downarrow\}.$$

More generally, one can relativize this definition to any $X \subseteq \mathbb{N}$:

$$X' = \{e : \Phi_e^X(e) \downarrow\}.$$

We also call X' the *jump of* X.

It is clear that $X' \equiv_T \{\langle e, n\rangle : \varphi_e(n) \downarrow\}$. We now prove that this indeed defines a non-computable set, by showing that $X' >_T X$.

▶ **Proposition I.18.** *For every X, $X <_T X'$.*

Proof. It is clear that $X \leq_T X'$, as one can check if a fixed n is in X by testing if the Turing Machine which halts only if its oracle has a 1 in the n-th position, do halt or not.

Now, suppose that $X \geq_T X'$ via Ψ. Let f be the function which, given a code e for a Turing Machine, outputs the code of the following procedure: given an oracle Y, halts if and only if $\Psi^Y(e) = 0$. By the recursion theorem I.12, let e be a fixed-point of f. Then, ϕ_e^X halts if and only if $\Psi^X(e) = 0$ if and only if $X'(e) = 0$, a contradiction. ■

▶ **Proposition I.19.** *If $A \leq_T B$, then $A' \leq_T B'$.*

Proof. Let f be a computable function such that $f^B = A$. In order to decide if $\Phi_e^A(e)$ halts, let e_0 be the Turing functional such that $\Phi_{e_0}^X = \Phi_e^{f^X}(e)$. Then, $\Phi_{e_0}^B(e_0) = \Phi_e^A(e)$, so $e \in A'$ if and only if $e_0 \in B'$. ■

I.2.3 Computable enumeration

We give an important notion of computability theory: the notion of computable-enumeration.

Definition I.20

We say that a set $A \subseteq \mathbb{N}$ is *computably enumerable*, or *c.e.* if it is the range of a computable function. If e is any integer, we write W_e for the range of the e-th Turing machine φ_e.

In other words, there exists a computable process that, to any integer n, associates one element of A. All elements of A are accessed this way. We often see a c.e. set as containing the elements enumerated by a Machine that never halt, but from time to time output a number: the machine that outputs $f(0)$, $f(1)$, $f(2)$, and so on.

Note that historically these sets have been called recursively enumerable, due to the previous definition of recursive functions. As "computable" is more accurate than "recursive", there has been some efforts to change every occurrence of r.e. to c.e. Despite those efforts, now both formulation can be found in the literature. In this thesis, we try to avoid the word "recursive" when it means computable, however we restrained from modifying the whole terminology of α-recursion theory.

As there is an enumeration of partial computable functions φ_e, there is also an enumeration of computably enumerable sets.

Definition I.21

We call W_e the range of φ_e. Then, $(W_e)_{e\in\mathbb{N}}$ is an enumeration of the computably enumerable sets.

I.3 Gödel's theorems

I.3.1 Gödel's completeness theorem

We gave two different meaning of the syntactic notion of formulas. The first one stays syntactic, the notion of provability, while the second is more focused on semantic: the ability to be interpreted in a model.

There is one link that we expect: if $\mathcal{M}$ is a model of T and T proves ϕ, then ϕ is true in $\mathcal{M}$. This can be proved by induction on the length of the proof, but the result is natural as we gave the same meaning of logical connectors in the proof definition, and in the model definition.

However, the precise link between syntax and semantic, or provability and modelling, is not obvious. Gödel's answered this in his famous completeness theorem.

Theorem I.22 (Gödel's completeness theorem)

Let $\mathscr{L}$ be a language and T be a theory. Then, T is consistent if and only if there exists a model of T.

Gödel's completeness theorem is very important. We will give two examples of how to use it: one to show a very easy incompleteness fact, and one to build a model of a particular theory.

Take the language of groups (a constant for the neutral element, a binary operator) together with the axioms of groups in this language, grouped in the theory T_G. Then there exists a model which is abelian, and another one which is not abelian. Therefore, by Gödel's completeness

theorem, neither $\forall a, b, ab = ba$ nor $\neg(\forall a, b, ab = ba)$ can be proved in T_G, and T_G cannot be complete.

Now, we continue with a more profound example. Let T_A be a theory in the language of arithmetic that attempts to define the integers. Now, add a constant symbol x, and consider $T = T_A \cup \{x > \underline{n} : n \in \omega\}$, where $\underline{n} = S(S(\cdots S(0)\cdots))$ iterated n times. Suppose T is not consistent, there must exist a proof of $\perp$ which therefore uses finitely many axioms from T. But this is impossible, as $\mathbb{N}$ is a model of every finite sub-theory of T. Therefore T is consistent, and there exists a model of T_A which is not $\mathbb{N}$ as it contains an element greater than any integer.

The previous reasoning is known as the compactness theorem, which is a corollary of the completeness theorem but can also be proved without the use of the syntactic proof, using ultrafilters.

Theorem I.23 (Compactness Theorem)

Let T be a theory. If every finite subset of T has a model, then T has a model.

Proof. Towards a contradiction, suppose T is not consistent. Then T proves $\perp$, and as a proof uses finitely many axioms, a finite subset of T is not consistent. But then, by Gödel's completeness theorem, a finite subset of T has no model which is a contradiction with the hypothesis.

Therefore, T is consistent, and by Gödel's completeness theorem T has a model. ■

I.3.2 Gödel's incompleteness theorems

When mathematical logic started to receive more focus, a need to give the whole mathematical field a strong basis and formal definitions was developed. For instance, the notion of infinitesimal was used a lot without a precise definition. Hilbert had a plan to settles strong foundations for all mathematics.

Hilbert's plan was in several step. The first one was to find an axiomatic system is which all of mathematics could be proved. The second step was to prove the coherence of this system within a simple system that he called "finitistic" and that was supposed to be based only on reasoning of finite things.

Hilbert's plan was quite popular and started making progress. For instance, the coherence of simple systems (less than PA) was proved. The coherence of PA was proved using simple means except for one single step. However, removing this step seemed very difficult. And at this point Gödel's theorem condemned the whole program.

The first incompleteness theorem

The first Gödel theorem tackles the first step of Hilbert's plan. Basically, a theory where the axioms are recursively enumerable (that is, when given a formula, if it is an axiom it will be recognized as an axiom), there exists a formula that is unprovable.

Theorem I.24 (Gödel's first incompleteness theorem)

Let T be a computably enumerable theory containing PA. Then, either T is inconsistent, or T is incomplete.

As an example, we give a proof of a related result with a much shorter proof. Let T be a recursively enumerable theory containing the language of arithmetic, and such that $\mathbb{N} \models T$.

Then either T is incomplete or T is inconsistent.

We admit that the following collection of formulas $\{\varphi_{\underline{n}}(\underline{n}) \downarrow : n \in \mathbb{N}\}$ is a collection of arithmetical formula of T.

As T is complete, for every $n \in \mathbb{N}$, either there exists a proof of "$\varphi_n(n) \downarrow$", or there exists a proof of "$\neg\varphi_n(n) \downarrow$". As T is consistent, only one of the previous two incompatible sentences is provable, and because $\mathbb{N} \models T$, only the true one must be computable. Therefore, we have a way to compute the halting set: look for a proof that $\varphi_n(n)$ halts or a proof that $\varphi_n(n)$ does not halt, and output the corresponding answer.

Compared to this result, Gödel had to deal with several additional things. First, he had to encode into Peano arithmetic the notion of formula, sentence and proof. This step corresponds to admitting that $\varphi_n(n) \downarrow$ is an arithmetical formula. Then, he also had to use a fixed point argument.

The second incompleteness theorem

The first incompleteness theorem leaves no hope for the first step of Hilbert's program. However, the unprovable formula of Gödel's first incompleteness theorem does not have any interesting meaning, and the question of its veracity would never be asked in normal mathematics. So, there might subsist some hope to have a good axiomatic system, where all of "natural" mathematics could be proved (at least all the previous theorems), on which the second step of Hilbert's plan could be done.

If this were true, we would get an axiomatic system very secure: there would exist an elementary proof of its consistency. If most mathematics could be done inside this system, then most mathematics would be secure and it would be very satisfying. Gödel's second incompleteness theorem takes care of destroying this hope.

Theorem I.25 (Gödel's second incompleteness theorem)

Let T be a recursively enumerable theory containing PA. Then, the arithmetical formula consisting of the consistence of T is unprovable in T.

One must have a stronger system to prove the coherence of a given system. But then, the coherence of the stronger system is questioned, and there is no point in proving the coherence of some theory assuming the coherence of a stronger theory!

So Hilbert's program should be dead. However, proof theorists continued to pursue this program in some weakened versions and have made major discoveries. For instance, they grouped the "non finitistic" reasoning in a single concentrated step. And even if we know we will never prove its consistency, mathematics feel secure after so many years of usage, and no one doubt of its consistency.

I.4 Topology and Computability

I.4.1 Computability in topological spaces

In a previous section, we introduced computable functions relative to some oracle. We showed that this allows to define relative computability for subsets of the integers, by using the characteristic functions of sets. In this section, we show that this can be extended to many topological spaces, given a convenient presentation of them. In this thesis we will only be interested in

Cantor and Baire space, so we give only the presentation of these spaces. We also highlight the interaction between topology and computability.

Definition I.26

We call *string of length* k a function in $k \to \mathcal{A}$, where $\mathcal{A}$ is the alphabet. An infinite string is a function in $\mathbb{N} \to \mathcal{A}$. The length of a string σ is written $\mathrm{lh}(\sigma)$ or $|\sigma|$. If σ, τ are two strings, we say that τ *extends* σ (written $\sigma \prec \tau$) if τ extends σ as partial functions. We write $\sigma^\frown\tau$ for the concatenation of two strings, and $\sigma \restriction n$ for the initial prefix of σ of size n.
The set of strings in alphabet $\mathcal{A}$ is written $\mathcal{A}^{<\omega}$. The empty string is written ϵ.

Definition I.27 (Cantor and Baire space)

We call *Cantor space* the set 2^ω of infinite sequences of bits, equipped with the product of the discrete topology on 2. This topology is generated by the open sets $[\sigma] = \{x \in 2^\omega : \sigma \prec x\}$ for all $\sigma \in 2^{<\omega}$. We call *reals* the elements of the Cantor space, in relation with infinite binary expansion of reals between 0 and 1.
We call *Baire space* the set ω^ω of infinite sequences of integers, equipped with the product of the discrete topology on ω. This topology is generated by the open sets $[\sigma] = \{x \in \omega^\omega : \sigma \prec x\}$ for all $\sigma \in \omega^{<\omega}$.

It remains to show how to define computations on Cantor and Baire space. Using Turing Machines, it seems quite easy for Cantor space, as the tapes of Turing Machine already consist of infinite sequences of 0 and 1. We say that a real x is computable if there exists a Turing Machine that never halts but instead write the infinite sequence x on its tape. Relative computability is defined again using oracle.

For Baire space, it is only slightly less straightforward, as we need to code sequences of integers into sequences of bits. This can be done by many computable [2] methods, such as $(x_n) \mapsto 0^{x_0}10^{x_1}10^{x_2}1\cdots$. We give the general framework for topological spaces with a fixed countable base.

Definition I.28

Let (X, τ) be a topological space. Then, $(U_i)_{i\in\mathbb{N}}$ is a *recursive presentation* if it is a basis for τ and there exists a computable function f such that $U_i \cap U_j = U_{f(i,j)}$. A *recursively presented topological space* is a topological space with a recursive presentation.
If $(U_i)_{i\in\mathbb{N}}$ is any sub-base for τ, one can easily define a recursive presentation of τ using the bijection between integers and finite sequences of integers. Therefore, we sometimes consider sub-bases as recursive presentation.

The canonical recursive presentation of the Cantor space is $([\sigma])_{\sigma\in 2^{<\omega}}$, while the canonical one for the Baire space is $([\sigma])_{\sigma\in\omega^{<\omega}}$.

Definition I.29

Let (X_0, τ_0) and (X_1, τ_1) be two topological spaces with $(U^i_n)_{n\in\mathbb{N}}$ being a recursive presentation of τ_i. Then, we say that $f : X_0 \to X_1$ is *computable* if it is continuous and if the set $\left\{(m, n) : U^0_m \subseteq f^{-1}(U^1_n)\right\}$ is computably enumerable.

2. In the sense of Definition I.29

We say that an element $x \in X_0$ is *computable* if there exists a computable function from the trivial topology of the set with a unique element to X_0 such that x is in the image.

In informal words, in order to compute an object x using an oracle y, it must be able to compute it at any precision, that is find all neighbourhood containing x. In its computation, the machine will be given approximations of y (understand, neighborhoods of y) at arbitrary precision.

It becomes even clearer when the topological spaces are also metric spaces. For instance, one can see $\mathbb{R}$ as a recursively presented topological space generated by $U_{\langle i,j \rangle} = \mathcal{B}(q_i, \epsilon_j)$ where q_i and ϵ_j are rationals (coded by the integers i and j) and $\mathcal{B}(q_i, \epsilon_j)$ is the open ball centered on q_i of radius ϵ_j. A function $f : \mathbb{R} \to \mathbb{R}$ is computable if it can find a rational ϵ-close to $f(x)$, given rationals arbitrary close to x.

Remark that every continuous function is computable relatively to some $x \in 2^\omega$.

I.4.2 Topological properties of Cantor and Baire spaces

We first give a few topological facts about Cantor and Baire space

Theorem I.30

The Cantor space is perfect [a], compact, 0-dimensional [b] and metrizable.

a. Closed without isolated point
b. It has a basis of clopen sets

The most important property to remember here is the fact that Cantor space is compact. This is sometimes known as a weak version of Kőnig's lemma, which states that each infinite tree with finite branching has an infinite branch.

We also specify a metric for Cantor and Baire spaces. Let $x \neq y \in \omega^\omega$. We define $d(x, y)$ to be 2^{-k} where k is the smallest integer such that $x(k) \neq y(k)$.

Theorem I.31

The Baire space is perfect, 0-dimensional, and metrizable.

In fact, there is a characterization of those two spaces by topological properties. In the theorem we only included the properties that will be interesting to us.

We will often use the following characterization of closed sets by trees.

Definition I.32 (Tree and infinite paths)

We say that a set $T \subseteq \omega^{<\omega}$ is a *tree* if it is closed under prefix. We write $[T] = \{x : \forall n, x \restriction n \in T\}$ the set of infinite branches of T.

Theorem I.33

Then, $A \subseteq \omega^\omega$ is closed if and only if there exists a tree $T \subseteq \omega^{<\omega}$ such that $A = [T]$. Similarly, $A \subseteq 2^\omega$ is closed if and only if there exists a tree $T \subseteq 2^{<\omega}$ such that $A = [T]$.

Proof. Let T be a tree. Then, $[T]$ is the complement of $\bigcup_{\sigma \notin T}[\sigma]$. Conversely, if A is a closed set, then $A = [T]$ where $T = \{\sigma \in \omega^{<\omega} : \exists x \in A, \sigma \prec x\}$. ■

I.4.3 Effectively open and closed sets

We define the notion of effective open and closed set.

Definition I.34

A set $A \subseteq \omega^\omega$ is *effectively open* if there exists a computably enumerable set $W \subseteq \omega^{<\omega}$ such that $A = \bigcup_{\sigma \in W}[\sigma]$. An *effectively closed set* is the complement of an effectively open set.

Theorem I.35

A set is effectively closed if and only if there exist a co-c.e. tree $T \subseteq \omega^{<\omega}$ such that $A = [T]$, if and only if there is a recursive tree $T \subseteq \omega^{<\omega}$ such that $A = [T]$.

Proof. Let $W \subseteq \omega^{<\omega}$ be such that the complement of A equals $\bigcup_{\sigma \in W}[\sigma]$. Then, define T to be $\omega^{<\omega} \setminus \{\tau : \exists \sigma \prec \tau, \sigma \in W\}$, a co-c.e. set. By construction, T is a tree. Now if $x \in [T]$ then x avoid being in any $[\sigma]$ for $\sigma \in W$, so $x \in A$.

Conversely, if $T \subseteq \omega^{<\omega}$ is co-c.e. let $W = \omega^{<\omega} \setminus T$, a c.e. set. We have that $x \in T$ if and only if $x \notin [\sigma]$ for any $\sigma \in W$, so the complement of $[T]$ is $\bigcup_{\sigma \in W}[\sigma]$.

Now suppose that T is a co-c.e. tree. Define $\widehat{T} = \{\sigma \in \omega^{<\omega} :$ all prefixes of σ are in T at stage $\mathrm{lh}(\sigma)\}$. Then, $[T] = [\widehat{T}]$ and the result follows. ■

We also give the link between all these notions and their computable counterparts.

Theorem I.36

The following holds:

1. A set is open if and only if it is effectively open in some set X.
2. A set is closed if and only if it is effectively closed in some set X.
3. A function is continuous if and only if it is computable in some set X.

This theorem establish a clear link between topology and computability. It actually is very useful in some areas of topology. For instance, the field of effective descriptive set theory is the application of computability tools to descriptive set theory. Descriptive set theory studies the complexity of subsets of reals, from a topological point of view. We see in the next sections the effective and non-effective hierarchies on these sets.

I.5 Hierarchies of complexity

In this section, we study the different hierarchies of complexity using several means: logic, computability and topology.

I.5.1 The Arithmetical hierarchy

Subsets of $\mathbb{N}$

We define the arithmetical hierarchy using definability. We will define a notion of complexity in arithmetical formulas, and then classify subsets of the integers using this notion.

Definition I.37 (Σ^0_n, Π^0_n and Δ^0_n formulas)

Let $\mathscr{L}$ be the language of arithmetic. We say that an $\mathscr{L}$-formula is Δ^0_0, Σ^0_0 or Π^0_0 if all its quantifiers are bounded, that is of the form "$\exists n < t$" or "$\forall n < t$" where t is an $\mathscr{L}$-term. Then, we say that:

— A formula ϕ is Σ^0_{n+1} if $\phi = \exists i \in \mathbb{N}, \psi(i)$ where ψ is Π^0_n.
— A formula ϕ is Π^0_{n+1} if $\phi = \forall i \in \mathbb{N}, \psi(i)$ where ψ is Σ^0_n.
— A formula ϕ is Δ^0_{n+1} if it equivalent to both a Σ^0_{n+1} and a Π^0_{n+1} formula.

When relativized with a set $X \subseteq \mathbb{N}$, we write $\Sigma^{0,X}_n$, $\Pi^{0,X}_n$ and $\Delta^{0,X}_n$.

Note that the notions of being Σ^0_n and Π^0_n is a syntactic notion, while being Δ^0_n is not. Therefore, it depends on the model on which we interpret the formula, which may or may not be $\mathbb{N}$.

Definition I.38 (Σ^0_n, Π^0_n, Δ^0_n sets)

We say that a set $A \subseteq \mathbb{N}$ is Σ^0_n (respectively Π^0_n, Δ^0_n) if it is Σ^0_n-definable (respectively Π^0_n, Δ^0_n), that is there exists a Σ^0_n (respectively Π^0_n, Δ^0_n) formula $\phi(n)$ with

$$i \in A \quad \iff \quad \mathbb{N} \models \phi(i).$$

The hierarchy of the Σ^0_n and Π^0_n sets form what we call the *arithmetical hierarchy*. A set in the arithmetical hierarchy is said to be *arithmetic*.

The arithmetical hierarchy is strict, and has strong links with computability.

Recall the definition of the halting problem. We claim that $\emptyset'$ has a Σ^0_1 definition. More than that, we have the following:

▶ **Proposition I.39.** — *The predicate "$\varphi_e(n)$ halts in t step" is Δ^0_1.*
— *The predicate "$\varphi_e(n)$ halts" is Σ^0_1.*

Proof. For the first item, we claim that $\varphi_e(n)$ halts in t steps if and only if there exists a trace of the computation of $\varphi_e(n)$ of length t that halts, if and only if every trace of computation of $\varphi_e(n)$ of length t does halt.

The second item is clear, as $\varphi_e(n)$ halts if and only if there exists t such that $\varphi_e(n)$ halts in t steps. ■

In fact, $\emptyset'$ is universal in the Σ^0_1 sets.

▶ **Proposition I.40.** *If A is Σ^0_1, then there exists a computable function f such that $i \in A$ if and only if $f(i) \in \emptyset'$.*

We say that $\emptyset'$ is a many-one Σ^0_1-complete set. This suggests a link between computability and the arithmetical hierarchy. We extend this to iterations of the jump.

Definition I.41

If X is any set, we define by induction the sets $X^{(0)} = X$ and $X^{(n+1)} = (X^{(n)})'$. The set $X^{(n)}$ is called the *n-th jump* of X.

▶ **Proposition I.42.** *Let n be any integer. Then, A is Σ^0_n if and only if there exists a computable function f such that $i \in A \iff f(i) \in \emptyset^{(n)}$.*

Proof. We show that $\emptyset^{(n)}$ is Σ^0_n. The empty set is clearly Σ^0_0. Now suppose $\emptyset^{(n)}$ is Σ^0_n, via formulas ϕ. We have that $e \in \emptyset^{(n+1)}$ if and only if $\exists \sigma \in 2^{<\omega}, \exists t \in \mathbb{N}$ such that $\varphi^\sigma_e(e)[t]\downarrow$ and $\sigma \prec \emptyset^{(n)}$. Now $\sigma \prec \emptyset^{(n)}$ is a Δ^0_{n+1} statement: $\forall n < \mathrm{lh}(\sigma)$, $\sigma(n) = 1 \iff n \in \emptyset^{(n)}$. So $e \in \emptyset^{(n+1)}$ is Σ^0_{n+1} definable.

Then, it is clear that if there exists function reducing A to $\emptyset^{(n)}$, then A is Σ^0_n. Now we prove the converse by induction, so suppose A is Σ^0_{n+1}-definable. Then, $n \in A$ if and only if $\exists m, \langle n, m\rangle \in B$ where B is Σ^0_n-definable, and therefore computable in $\emptyset^{(n)}$. Let $f(n)$ be the code of the Turing machine which on oracle $\emptyset^{(n)}$ searches for such an m, and stops if it finds one. Then, $n \in A$ if and only if $f(n) \in \emptyset^{(n+1)}$. ■

▶ **Corollary I.42.1.** *The arithmetical hierarchy is strict.*

Theorem I.43

Let $A \subseteq \mathbb{N}$ be any set. Then, $A \in \Delta^0_{n+1}$ if and only if $A \leq_T \emptyset^{(n)}$.

Proof.

Suppose that A is computable in $\emptyset^{(n)}$ via the Turing machine φ. Then, $n \in A$ (resp. $n \notin A$) if and only if there exists a time t and a string $\sigma \prec \emptyset^{(n)}$ such that $\varphi^\sigma(n) \downarrow= 1$ (resp. $\varphi^\sigma(n) \downarrow= 0$). Therefore, A is Δ^0_{n+1} definable.

Now, using Proposition I.42, every Σ^0_n set is $\emptyset^{(n)}$ computable, and so are their complement the Π^0_n sets.

Now, suppose A is Δ^0_{n+1}. Then, $n \in A$ if and only if $\exists m, \phi(n, m)$ if and only if $\forall m, \psi(n, m)$, with ϕ and ψ being respectively Π^0_n and Σ^0_n, so their validity is computable in $\emptyset^{(n)}$. Now, the Turing machines with oracle $\emptyset^{(n)}$ with input n can search for the smallest m such that either $\phi(n, m)$ or $\neg\psi(n, m)$ and answer accordingly, making A computable in $\emptyset^{(n)}$. ■

We also have the following important link between definability and computability. This theorem is the characterization of computability we will use to generalize it in other settings.

Theorem I.44

Let $A \subseteq \mathbb{N}$ be any set. Then:

1. A is computably enumerable if and only if A is Σ^0_1.
2. A is co-computably enumerable if and only if A is Π^0_1.
3. A is computable if and only if A is Δ^0_1.

Proof. We prove the first item. Remark that Σ^0_0-definable sets are computable, as bounded quantifiers can be checked in finite time. If A is Σ^0_1 via $\phi \equiv \exists t, \psi(n, t)$, then at time T enumerates the $n \leq T$ such that $\exists t \leq T$ such that $\psi(n, t)$. Clearly, this enumerates all A

and nothing else.

If A is c.e. then $n \in A$ if and only if $\exists m, t$ such that $\varphi_e(n)$ halts in t step and $\varphi_e(m) = n$, where φ_e is the Turing machine whose range is A.

Now, the second item follows from the first as the complement of a c.e. set is a co-c.e. set and the complement of a Σ^0_1 is a Π^0_1 set. The third item is a consequence of Theorem I.43. ■

Subsets of the Baire and Cantor spaces

In this section, we show how the arithmetical sets for 2^ω and ω^ω spaces are linked to topology.

Definition I.45 (Σ^0_n, Π^0_n, Δ^0_n sets)

Let $\mathscr{X}$ be the Baire or the Cantor space. We say that a set $A \subseteq \mathscr{X}$ is Σ^0_n (respectively Π^0_n, Δ^0_n) if it is Σ^0_n-definable (respectively Π^0_n, Δ^0_n), that is there exists a Σ^0_n (respectively Π^0_n, Δ^0_n) formula ϕ with

$$\forall x \in \mathscr{X}, x \in A \quad \iff \quad \mathbb{N} \models \phi(x).$$

In this definition, the arithmetical formulas have a second order free variable. We first show that the Σ^0_1 subsets of the Baire and Cantor spaces correspond to the effectively open sets.

▶ **Proposition I.46.** *A set $A \subseteq \mathscr{X}$ where $\mathscr{X}$ is the Baire or the Cantor space, is Σ^0_1 if and only if it is effectively open. It is Π^0_1 if and only if it is effectively closed.*

Proof. Suppose that $\exists n, \phi(X, n)$ is a Σ^0_1 formula defining A. As ϕ has only bounded quantifiers, given some n it needs only a finite amount of X to decide if it is true or false. Then, $x \in A$ if and only if there exists n and $\sigma \prec x$ such that already we have $\phi(\sigma, n)$. Then we have $A = \bigcup\{[\sigma] : \exists n, \phi(\sigma, n)\}$.

Suppose that A is effectively open, of code e. Then, $x \in A$ if and only if $\exists \sigma \in W_e$ such that $\sigma \prec x$. ■

With the same proof, one can show the corresponding fact for the n-th level. A union (or intersection) is effective if the codes for the element of the union (intersection) are given by a computable enumeration. In our case, a code for a Σ^0_n set is a Gödel code for a Σ^0_n formula defining it, while a code for a Π^0_n is a code for a Π^0_n formula defining it.

▶ **Proposition I.47.** *A set is Σ^0_{n+1} if it is an effective unions of Π^0_n sets. It is Π^0_{n+1} if it is an effective intersection of Σ^0_n sets.*

We have seen that Σ^0_1 sets are open sets, so Π^0_2 sets are intersection of open sets, which is called G_δ in topology. Like in Proposition I.46, the G_δ sets are exactly the relativized Π^0_2 sets. The topological version of the arithmetical sets is called the finite Borel hierarchy.

I.5.2 The finite Borel hierarchy

The Borel hierarchy is a measure of the topological complexity of sets. Compared to the previous definitions, it does not measure in any way the effective content of the set. Therefore, all subsets of $\mathbb{N}$ would be of the lowest topological complexity: they are clopen sets.

Definition I.48 (Finite Borel Hierarchy)

Let $A \subseteq \mathscr{X}$, where $\mathscr{X}$ is Baire or Cantor space. Then:
- A is $\mathbf{\Delta^0_0}$, $\mathbf{\Sigma^0_0}$ or $\mathbf{\Pi^0_0}$ if it is clopen.
- A is $\mathbf{\Sigma^0_{n+1}}$ if it is a countable union of $\mathbf{\Pi^0_n}$ sets.
- A is $\mathbf{\Pi^0_{n+1}}$ if it is a countable intersection of $\mathbf{\Sigma^0_n}$ sets.
- A is $\mathbf{\Delta^0_n}$ if it both $\mathbf{\Pi^0_n}$ and $\mathbf{\Sigma^0_n}$.

A set is in the finite Borel hierarchy if it is Δ^0_n for some $n \in \mathbb{N}$.

The finite Borel hierarchy is closed under finite unions, finite intersections and complement. However, it is not closed under countable unions, as countable unions of A_n where each A_n is $\mathbf{\Sigma^0_n}$ might not be finite Borel. In order to continue the closure under countable union, one need to continue the construction after the finite level: a transfinite construction. The tool to do this is the ordinals.

I.6 Ordinals

I.6.1 Basic properties

Ordinals are fundamental to mathematical logic. They are the backbone of set theory. They appear in model theory, computability theory, topology... It is almost surprising to see how they appear in proof theory. This thesis is of course not spared: ordinals will play an important role in most chapters.

In ordinary language, an ordinal number is a position in an enumeration: the *first*, the *second*, the *third* and so on. Mathematical language extends this to infinity and beyond. For instance, suppose you are trying to count the real numbers. Then, you pick one, label it "first", pick another one, label it "second", and continue this way for all integer. You get a list of reals indexed by $n \in \mathbb{N}$, but as we know that $\mathbb{R}$ does not have the same cardinality as $\mathbb{N}$, there must exist some real that we have not enumerated. So in order to give every real an ordinal number, one need to continue enumerating them, and the first ordinal number after all the previous one would be ω, followed by $\omega+1$, $\omega+2$,... But again, after infinitely many steps we would only have enumerated countably many reals. So the next one is labelled $\omega + \omega = \omega \times 2$, then $\omega \times 2 + 1$... It continues up to $\omega \times n$, ω^2, ω^ω...

Ordinal numbers is the notion required for any kind of induction: definitions and proofs. Integers allow to do induction up to any finite level. Infinite ordinals continue the induction after infinitely many steps, and in this sense it is called transfinite induction.

Definition I.49

A *well founded* relation R on some set A is a relation such that for every $B \subseteq A$, there exists an element $b_0 \in B$ such that $\forall b \in B$, $\neg b R b_0$.
A well-order is a (strict) well founded linear order.

We are usually not interested in the exact support of a well-order, but rather by its structure. Therefore, we usually consider the equivalence class by bi-embedding, and specify representative of these equivalence classes: the ordinals.

Definition I.50

An ordinal is a transitive set, well-ordered by the "membership" relation.

Ordinals are interesting as representative of well-orders. For instance, each ordinal is the set of smaller ordinals, in the sense of embeddability. The smallest well-order is the empty one, we write it 0 and it is an ordinal. Then, $1 = \{0\}$ is also an ordinal, as well as $2 = \{0; 1\}$. The ordinals form a class, and this class is well-ordered as any set has an element, and the set of elements below is well-ordered. We call ω_1 the first uncountable ordinal.

▶ **Proposition I.51.** *Every well-ordered set is in bijection with some ordinal, called its* order-type.

Each ordinal is either the empty ordinal, a successor ordinal if it is of the form $\{\alpha\}$ where α is an ordinal (and in this case we write it $\alpha+1$) or a limit ordinal, that is the limit of an increasing sequence of ordinals for the order topology. We now give an example of definition by induction using ordinals.

▶ **Proposition I.52.** *For every ordinal α, there exists a unique function $+_\alpha : \alpha \times \alpha \to \alpha$ such that for all $\beta, \gamma < \alpha$ we have:*
- $\beta +_\alpha 0 = 0$,
- $\beta +_\alpha \gamma = (\beta +_\alpha \delta) + 1$ *if* $\gamma = \delta + 1$,
- $\beta +_\alpha \gamma = \lim_{\delta<\gamma}(\beta +_\alpha \delta)$ *if* γ *is limit.*

Proof. Let α be an ordinal. We prove by induction on γ that there exists such a $+_{\alpha,\gamma}$ defined up on $\alpha \times \gamma + 1$. If $\gamma = 0 = \emptyset$, the result is clear by choosing $\beta +_{\alpha,0} 0 = \beta$.

If $\gamma = \delta + 1$ and the result is true for δ, it suffices to extend $+_{\alpha,\delta}$ to $\alpha \times \gamma$ by defining $\beta +_{\alpha,\gamma} \gamma = (\beta +_{\alpha,\delta} \delta) + 1$.

If $\gamma = \lim_{\delta<\gamma} \delta$ and the result is true for every $\delta < \gamma$, it suffices to extend $+_{\alpha,\delta}$ to $\alpha \times \gamma$ by defining $\beta +_{\alpha,\gamma} \gamma = \lim_{\delta<\gamma} \beta +_{\alpha,\delta} \delta$. ■

Of course, if $\alpha < \beta$, then $+_\beta$ restricted to α is equal to $+_\alpha$, so we will drop the subscript notation. One can also define ordinal product and exponentiation this way.

As we will mainly be in the setting of second order arithmetic, we need sets to only contain integers. Therefore, instead of speaking of ordinals, we will speak of codes for ordinals, the codes being elements of Cantor space.

Definition I.53 (Encoding of ordinals)

An *encoding for an ordinal* α is any well-order x on a subset of $\mathbb{N}$ with order type α, modulo the bijection $\mathbb{N}^2 \to \mathbb{N}$. More precisely, it is a single real $x \in 2^\omega$ such that the relation $n \lhd m \Leftrightarrow x(\langle n, m\rangle) = 1$ is a well order of order-type α.
We write $\mathbb{WO} \subseteq 2^\omega$ for the set of codes of ordinals below ω_1. We write $\mathbb{WO}_{<\alpha} \subseteq 2^\omega$, $\mathbb{WO}_{\leq\alpha} \subseteq 2^\omega$ and $\mathbb{WO}_{=\alpha} \subseteq 2^\omega$ for the set of codes for ordinal that are respectively strictly below, below or equal, and equal to α.
In order to be consistent with the notation for recursive ordinals, we define $\mathbf{WO} = \{(e, X) : \Phi_e(X) \in \mathbb{WO}\}$.

I.6.2 Computable ordinals

Trivially, only countable ordinals can be encoded. We will be interested in the special case when ordinals have recursive encoding. In this case we can code them into single integers.

Definition I.54 (Recursive code and ordinals)

A *recursive code* for a ordinal α is an integer e such that the relation has domain $\{n \in \mathbb{N} : \varphi_e(n,n)\downarrow\}$ and such that $n \leq m$ iff $\varphi_e(n,m) = 1$ is a well order of order-type α. We call $\leq_e$ this well-order.
We say that an ordinal α is *recursive* if it has a recursive code, and we write $\mathsf{WO} \subseteq \mathbb{N}$ for the set of recursive codes for ordinals. If α is a recursive ordinal, we write $\mathsf{WO}_{<\alpha} \subseteq \mathbb{N}$, $\mathsf{WO}_{\leq\alpha} \subseteq \mathbb{N}$ and $\mathsf{WO}_{=\alpha} \subseteq \mathbb{N}$ for the set of codes for ordinal that are respectively strictly below, below or equal, and equal to α.
We define analogously the relativized notion of *recursive in X codes and ordinals*. We define $\mathsf{WO}^X \subseteq \mathbb{N}$ to be the set $\{e : \varphi_e^X \text{ is a well-order}\}$.
We write $\mathbf{WO}^X = \mathsf{WO}^X \times \{X\}$ and $\mathbb{WO}^X = \{\Phi_e(X) : e \in \mathsf{WO}^X\}$.

It is clear that the recursive ordinal form an initial segment of the ordinals. Indeed, if α is recursive via φ_e, and $\beta < \alpha$, then there exists some n such that e_0 is a code for β, where φ_{e_0} is the restriction of φ_e on the set of m with $m \leq_e n$.

The relativization of WO is straightforward, however we need to also define a version where the oracle is added in the set to have a pendant of $\mathbf{\Sigma^0_n} = \bigcup_{X \in 2^\omega} \Sigma^0_n(X)$. It is clear that WO^X and $\mathbf{WO}^X$ are one-to-one equivalents (relative to X), and we have $\mathbf{WO} = \bigcup_{x \in 2^\omega} \mathbf{WO}^X$.

As there are only countably many computable ordinals, there must exist non-computable ordinals. The smallest such one exists as ω_1 is a well-ordered.

Definition I.55 (ω_1^{CK})

We call *Church-Kleene omega 1* and write ω_1^{CK} the smallest uncomputable ordinal. We write ω_1^X for the smallest ordinal that is non-computable in X.

As the computable ordinals form an initial segment, every ordinal below ω_1^{CK} is computable. It is worth noting that we have $\omega_1 = \bigcup_{X \in 2^\omega} \omega_1^X$.

We emphasized that well-orders are the good framework to conduct construction by induction. So we would expect that computable ordinals would allow some kind of construction by induction where the result is computable. Recall the definition of the ordinal sum in Proposition I.52. Even in this simple definition, we had to react in different ways depending on the form of the ordinal: whether it is 0, a successor ordinal or a limit ordinal.

Suppose we wanted to define a computable version of the ordinal sum, that worked on ordinal codes. Some things would lack in a recursive code for ordering: Indeed, it is non-computable, given a code for a computable ordinal, to decide if it codes for 0, a successor ordinal or a limit ordinal. Similarly it is uncomputable to find a code for the predecessor of a successor ordinal. A way to overcome this difficulty is to consider codes that are more precise: they specify this kind of information, for the ordinal and its elements. We call these constructive codes, and show that recursive ordinals have a constructive code.

I.6.3 Constructive ordinals

The first thing to see in order to build constructive codes is the relation between well-orders, and trees. Indeed, trees are partial orders, and they also can code ordinals.

Definition I.56

Let T be a tree. Then, the relation $\sigma \leq_T \tau$ meaning "σ extends τ in T" is a partial order. It is well-founded if and only if $[T] = \emptyset$. In this case, we say that T is *well-founded*.
If T is a well-founded tree, we define $|\cdot|_o : T \to \mathrm{Ord}$ by induction on T such that $|\sigma|_o = \sup\{|\tau|_o + 1 : \tau \text{ is a direct child of } \sigma\}$. We define $|T|_o$ to be $|\epsilon|_o$.
A *tree-code* for an ordinal α is a code for a computably enumerable well-founded tree T with $|T|_o = \alpha$.

In order to prove that recursive ordinals and ordinals with tree-codes are the same, we need the useful notion of Kleene-Brouwer ordering.

Definition I.57 (Kleene-Brouwer ordering)

We define the *Kleene-Brouwer ordering* on $\omega^{<\omega}$, denoted $\leq_{\mathrm{KB}}$ to be the following order: if $\sigma, \tau \in \omega^{<\omega}$, then:

1. If $\tau \prec \sigma$, then $\sigma <_{\mathrm{KB}} \tau$,
2. If t is the smallest such that $\sigma(t) \neq \tau(t)$, then $\sigma(t) < \tau(t)$ implies $\sigma <_{\mathrm{KB}} \tau$.

Let $T \subseteq \omega^{<\omega}$ be a tree. We define $\leq_{\mathrm{KB}(T)}$ to be $\leq_{\mathrm{KB}}$ restricted to the elements of T.

▶ **Proposition I.58.** *The recursive ordinals and the tree-coded ordinals are the same.*

Proof sketch Let α be a recursive ordinal with code e. Then, the tree of strictly decreasing sequences for $\leq_e$ is a tree-code of α.
Let T be a tree. Then, $\leq_{\mathrm{KB}(T)}$ is a code for a recursive ordinal α such that $|T|_o \leq \alpha$. We conclude using the fact that the recursive ordinals are an initial segment. ■

Tree-codes for ordinals do not add the additional information needed to conduct recursive transfinite induction. However, we are now able to add labels to the nodes of the tree.

Definition I.59 (Constructive code for recursive ordinal)

We say that e is a *constructive code* if e is the code of a computable well-founded tree such that:

1. Each node has either 0, 1 or infinitely many direct children, and is labelled accordingly 0, 1 or ∞.
2. If σ has infinitely many direct children $(\tau_n)_{n\in\mathbb{N}}$ such that $\tau_n = \sigma^\frown a_n$ with a_n strictly increasing, then $|\tau_n|_o < |\tau_{n+1}|_o$.

We say that an ordinal is *constructive* if it has a constructive code. We call $\mathcal{O}$ the set of constructive codes. As usual, we define $\mathcal{O}_{\leq\alpha}$, $\mathcal{O}_{<\alpha}$ and $\mathcal{O}_{=\alpha}$ for restrictions of $\mathcal{O}$, and $\mathcal{O}^X$, $\mathcal{O}^X_{\leq\alpha}$, $\mathcal{O}^X_{<\alpha}$ and $\mathcal{O}^X_{=\alpha}$ for their relativization to X.

This time, constructive codes contain the information in the label of the root. If it is labelled 0, then it codes for the ordinal 0. If it is labelled 1, then it codes for a successor ordinal, and a constructive code for the predecessor is the unique child of the root. If it is labelled ∞, then it is a limit ordinal such that the children form an infinite increasing sequence of constructive codes, converging to the limit ordinal. Therefore we say that *a is the successor of b* if the root of a is labelled 1 and b is its sub-tree whose root is the unique child of the root of a; we say that *a is the limit of* $(b_n)_{n\in\mathbb{N}}$ if the root of a is labelled ∞ and b_n are its sub trees, whose roots are the

children of the root of a in increasing order.

It remains to show that recursive ordinals and constructive ordinals (those with constructive codes) are the same. It is clear that a constructive ordinal is recursive by Proposition I.58. We will now show that every recursive ordinal is constructive, in two steps: first, we prove that every recursive ordinal is bounded by a constructive ordinal, and then we show that constructive ordinals are closed downward.

Theorem I.60

For every recursive ordinal α, there exists a constructive code c for an ordinal β with $\alpha \leq \beta$.

Proof. By Proposition I.58, let T_0 be a computably enumerable tree with $|T_0|_o = \alpha$. We will modify T so that we are able to label it, and so that it is computable, without decreasing its value for $|\cdot|_o$.

First, we modify T into T_0 by adding infinitely many leaves to every node of T. Therefore, we can safely label nodes of T with "∞", and label nodes that we added with "0".

Then, we modify the computably enumerable tree T_0 into a computable tree T_1. The tree T_1 uses the fact that given a code of a Turing machine e and an integer N, one can uniformly find another code e_0 with $\varphi_e \simeq \varphi_{e_0}$. Therefore, we define T_1 to be the enumeration of T_0 such that if $\sigma^\frown n$ is enumerated before $\sigma^\frown m$, then $m < n$. This makes T_1 computable, but with the same ordinal value.

Finally, we ensure that at every step the ordinal value of an increasing sequence of children has increasing associated ordinals. We do this by modifying T_1 by f where f is the following: if $\sigma \in T_1$ is a leaf or a singleton, then $f(\sigma) = \sigma$, otherwise let a_n be the increasing sequence of direct children of σ. Then the new children of σ is an increasing sequence of constructive codes b_n such that b_n codes for the sum of $|f(a_i)|_o + 1$ for $i \leq n$. Then, $T = f(T_1)$ proves the theorem. ■

▶ **Corollary I.60.1.** WO *is many-one reducible to* $\mathcal{O}$. *In its relativized version, for every* $X \in 2^\omega$, WO^X *is many-one reducible to* $\mathcal{O}^X$.

We now show that constructive ordinals are closed downwards.

Theorem I.61

The constructive ordinals are closed downward. Moreover, there exists a function f such that if a is a constructive code for α, then $W_{f(a)}$ is an enumeration of constructive codes such that any $\beta < \alpha$ has a constructive code in $W_{f(a)}$.

Proof sketch We prove by induction that if e is a constructive code for an ordinal α and T is the associated tree, then for any $\beta < \alpha$ there exists some $\sigma \in T$ such that $|T \upharpoonright \sigma|_o = \beta$, where $T \upharpoonright \sigma = \{\tau : \sigma^\frown\tau \in T\}$.

If $\alpha = 0$, the result is trivial as there is no $\beta < 0$. If α is successor, then the root of T has a unique child σ and $|T \upharpoonright \sigma|_o = \beta$, and for every $\gamma < \beta$ we also have the result by induction hypothesis, so we have the result for α.

If α is limit, then the root of T has infinitely many children σ_n, such that $|T \upharpoonright \sigma_n|_o = \gamma_n$ and γ_n converging to α. If $\beta < \alpha$, let n be such that $\beta < \gamma_n$. By the induction hypothesis, there exists a $\tau \in T \upharpoonright \sigma_n$ such that $|T \upharpoonright \sigma_n \upharpoonright \tau|_o = \beta$. Therefore, $|T \upharpoonright \sigma_n{}^\frown\tau|_o = \beta$.

Therefore, $f(a)$ is the code that enumerates the constructive codes $T \upharpoonright \sigma$ for any σ in T. ■

▶ **Corollary I.61.1.** *Constructive ordinals and recursive ordinals are the same. More than that,* WO *is many-one equivalent to* $\mathcal{O}$*. In its relativized version, for every* $X \in 2^\omega$*,* WO^X *is many-one equivalent to* $\mathcal{O}^X$*.*

We now use the previous result to define an iteration of the Turing jump along the constructive ordinals.

Definition I.62

Let α be a recursive ordinal, and a be a constructive code for it. We inductively define $\emptyset^{(a)}$ to be:

1. The empty set if a codes for the ordinal 0,
2. The set $\emptyset^{(b)}$ if a is the successor of b,
3. The set $\bigoplus_{n\in\mathbb{N}} \emptyset^{(b_n)}$ if a is the limit of $(b_n)_{n\in\mathbb{N}}$.

The jump is an operation essentially meant for Turing degrees. We expect the transfinite iteration of the jump along α, the set $\emptyset^{(a)}$, to be independent from the coding a of α. This is the case by a result of Spector.

Theorem I.63 (Spector)

If a and b are constructive codes for respectively α and β with $\alpha \leq \beta$, then $\emptyset^{(a)} \leq_T \emptyset^{(b)}$ uniformly in a and b.

▶ **Corollary I.63.1** (Spector's uniqueness theorem)**.** *If* α *is an ordinal with constructive codes* a *and* b*, then* $\emptyset^{(a)} \equiv_T \emptyset^{(b)}$*, in a uniform way.*

Thus, we are allowed to ignore the constructive coding of an ordinal when we are only interested in Turing degree properties. If α is a constructive ordinal, we write $\emptyset^{(\alpha)}$ for the Turing degree of $\emptyset^{(a)}$ for any constructive code a of α.

Note that there is another way to define iterations of the jump, using recursive codes instead of constructive ones. We will see this construction in Definition I.89.

I.7 Transfinite hierarchies

Now that we have defined ordinals, we are able to extend the arithmetical and Borel hierarchies beyond the finite case. As it is simpler in the sense that it does not require any effectivity, we start by extending the finite Borel hierarchy to any ordinal below ω_1, the smallest uncountable ordinal. In the effective case we will need to code the elements of the hierarchy to be able to make the union and intersection effective. Finally, we extend the arithmetical hierarchy of subsets of integers.

I.7.1 The Borel Hierarchy

Let us define the hierarchy.

Definition I.64 (The Borel hierarchy)

Let $A \subseteq \omega^\omega$. Let α be an ordinal, we say that:

1. A is $\mathbf{\Sigma^0_1}$ if it is open.
2. A is $\mathbf{\Sigma^0_\alpha}$ if it is the union of countably many sets, each of them being $\mathbf{\Pi^0_{\beta_n}}$ for some $\beta_n < \alpha$.
3. A is $\mathbf{\Pi^0_\alpha}$ if it is the complement of $\mathbf{\Sigma^0_\alpha}$ set.
4. A is $\mathbf{\Delta^0_\alpha}$ if it is both $\mathbf{\Sigma^0_\alpha}$ and $\mathbf{\Pi^0_\alpha}$.

First, note that no set is added at step $\mathbf{\Sigma^0_{\omega_1}}$ or $\mathbf{\Pi^0_{\omega_1}}$. Indeed, every countable union (or intersection) of $\mathbf{\Pi^0_{\alpha_n}}$ set where every α_n is countable is already $\mathbf{\Sigma^0_\beta}$ for some countable β, as there is always a countable upper bound on the sequence $(\alpha_n)_{n\in\omega}$. However, the hierarchy is strict before ω_1.

The obtained hierarchy can be seen as a layering on the construction of a set closed by countable union and intersection, and containing the open and closed sets. We start by adding every countable union and intersection. Then, this will add some set, so we also have to add countable unions and intersections of these sets. We continue like this up to ω, but even at this point, there will be a countable union consisting of A_n being a $\mathbf{\Pi^0_n}$ set. So this union is not already added and we have to continue the completion.

The Borel hierarchy is used primarily in Descriptive set theory, as a topological measure of complexity, the simplest set being the clopen sets. It has the following properties:

— For every α, the classes $\mathbf{\Sigma^0_\alpha}$ and $\mathbf{\Pi^0_\alpha}$ are closed under finite unions and intersections.
— For every α, the class $\mathbf{\Sigma^0_\alpha}$ is closed under countable union and the class $\mathbf{\Pi^0_\alpha}$ is closed under countable intersection.
— For every $\alpha < \beta < \omega_1$, we have $\mathbf{\Pi^0_\alpha} \cup \mathbf{\Sigma^0_\alpha} \subseteq \mathbf{\Delta^0_\beta}$.

I.7.2 The effective Borel Hierarchy

We now effectivize Definition I.64. The problem is that for being able to give an effective union, each member of the union must be coded by an integer. Therefore we define codes along with the hierarchy.

Definition I.65 (The effective Borel hierarchy)

Let α be an ordinal.

— A Σ^0_1 code is an integer of the form $c = \langle 0, e\rangle$. The set $\bigcup_{\sigma\in W_e}[\sigma]$ has code c.
— A Π^0_α code is an integer of the form $c = \langle 1, c_0\rangle$, where c_0 is a Σ^0_α code. The complementary of A has code c, where A has code c_0.
— A Σ^0_α code is an integer of the form $c = \langle 2, e\rangle$, if each $c_0 \in W_e$ is a $\Pi^0_{\beta_{c_0}}$ code, $\sup\{\beta_{c_0} + 1 : c_0 \in W_e\} = \alpha$. Then the set $\bigcup_{c_0\in W_e} A_{c_0}$ has code c where each set A_{c_0} has code c_0.

A set $A \subseteq \omega^\omega$ is said to be Σ^0_α (resp. Π^0_α) if it has a Σ^0_α (resp. Π^0_α) code. It is said to be Δ^0_α if it is both Σ^0_α and Π^0_α. The notions relativized to X are written $\Sigma^0_\alpha(X)$, $\Pi^0_\alpha(X)$ and $\Delta^0_\alpha(X)$.

It is clear that every Σ^0_α set is a $\mathbf{\Sigma^0_\alpha}$ set. Conversely, for every $\alpha < \omega_1^{\mathrm{CK}}$ there exists a $\mathbf{\Sigma^0_1}$ which is not Σ^0_α. We saw that the Borel hierarchy stops adding new sets at step ω_1. In the case of the effective Borel hierarchy, there is no Σ^0_α code for any $\alpha \geq \omega_1^{\mathrm{CK}}$: indeed, from a Σ^0_β

code, one can enumerate a well-founded tree of Baire space, linearize it using Kleene-Brouwer ordering, and the resulting ordinal is at least β. So, β cannot be above ω_1^{CK}.

The effective Borel hierarchy is a combination of topological and effective complexity, for subsets of Baire and Cantor space. It still has the following properties:

— For every α, the classes Σ^0_α and Π^0_α are closed under finite unions and intersections.
— For every α, the classes Σ^0_α is closed under effective union and Π^0_α is closed under effective intersection.
— For every $\alpha < \beta < \omega_1$, we have $\Pi^0_\alpha \cup \Sigma^0_\alpha \subseteq \Delta^0_\beta$.

We now show how the Borel and effective Borel hierarchies are intertwined.

Theorem I.66 (Lusin, [52])

For every X and $\alpha < \omega_1^X$, there exists a set that is Σ^X_α but not $\mathbf{\Pi^0_\alpha}$.

▶ **Corollary I.66.1.** *The Borel hierarchy does not collapse before level ω_1. The effective Borel hierarchy does not collapse before level ω_1^{CK}.*

I.7.3 Hyperarithmetic sets

We now extend the arithmetical sets up to ω_1^{CK}. This hierarchy being on subsets of $\mathbb{N}$, it is only a measure of the effective complexity, but not the topological complexity as all subsets of $\mathbb{N}$ are open. It really is the counterpart of the effective Borel hierarchy.

Definition I.67 (Hyperarithmetical hierarchy)

Let α be an ordinal.

— A Σ^0_1 code is an integer of the form $c = \langle 0, e\rangle$. The set $\bigcup_{\sigma\in W_e}[\sigma]$ has code c.
— A Π^0_α code is an integer of the form $c = \langle 1, c_0\rangle$, where c_0 is a Σ^0_α code. The complementary of A has code c, where A has code c_0.
— A Σ^0_α code is an integer of the form $c = \langle 2, e\rangle$, if each $c_0 \in W_e$ is a $\Pi^0_{\beta_{c_0}}$ code, $\sup\{\beta_{c_0} + 1 : c_0 \in W_e\} = \alpha$. The set $\bigcup_{c_0\in W_e} A_{c_0}$ has code c where each set A_{c_0} has code c_0.

A set is said to be Σ^0_α (resp. Π^0_α) if it has a Σ^0_α (resp. Π^0_α) code. It is said to be Δ^0_α if it is both Σ^0_α and Π^0_α. We say that A is *hyperarithmetical* if it is Σ^0_α for some $\alpha < \omega_1^{\mathrm{CK}}$.

There is a very nice equivalence between the hyperarithmetical sets and the iterations of the jump.

Theorem I.68

A set is $\Delta^0_{\alpha+1}$ if and only if it is computable by the α-th jump. It is Σ^0_α if and only if it is many-one reducible to the α-th jump.

▶ **Corollary I.68.1.** *A set $A \subseteq \mathbb{N}$ is hyperarithmetic if and only if there exists a computable ordinal α such that $A \leq_T \emptyset^{(\alpha)}$.*

I.7.4 Complexity above the Borel sets

The task of ordering the sets of reals, although we already gave several hierarchies, is far from being complete. Indeed, the sets that are at some level of a hierarchy, also known as the

Borel sets, is "negligible" compared to the total number of sets. Precisely, the cardinality of $\mathcal{P}(\omega^\omega)$ is $2^{2^{\aleph_0}}$, while the cardinality of the Borel sets is only $2^{\aleph_0}$, as there are countably many hyperarithmetic sets and every Borel set is hyperarithmetic in some oracle.

We saw that the first ω level of the hierarchy correspond to definability with quantifiers over the integers. In order to extend the Borel and effective Borel hierarchies, we will this time allow quantifiers over sets of integers. We will see that this greatly increases the definability power and that already at the first level, it includes the whole (effective) Borel hierarchy. Each level in this projective (or analytical in the effective case) hierarchy has a huge complexity gap between it and its successor, in particular many nice properties (such as measurability) tend to become undecidable. Therefore, instead of considering all levels up to the collapsing of the hierarchy as in the Borel case, we will restrict ourselves to only one quantifier on ω^ω.

We show how to define complexity for sets that have higher complexity than the Borel hierarchy.

Definition I.69

A Σ^1_1 formula is a formula of the form $\exists X \in \omega^\omega, \Phi(X)$ where ϕ is arithmetic. A Π^1_1 formula is a formula of the form $\forall X \in \omega^\omega, \Phi(X)$ where ϕ is arithmetic. A Δ^1_1 formula is a formula that is equivalent both to a Π^1_1 and a Σ^1_1 formula.

Definition I.70

A set $A \subseteq \mathscr{X}$ where $\mathscr{X}$ is 2^ω, ω^ω or $\mathbb{N}$ is Σ^1_1 (respectively Π^1_1, Δ^1_1) if there is a Σ^1_1 (respectively Π^1_1, Δ^1_1) formula φ such that

$$\forall n \in \mathscr{X}, n \in A \iff \varphi(n).$$

We now give the boldface, non effective version of this. The boldface do not directly use the notion of formula, although we could just use Definition I.70 with formulas with a real parameter. Instead, we give the more "historic" definition.

Definition I.71

A set $A \subseteq \mathscr{X}$ where $\mathscr{X}$ is 2^ω or ω^ω is $\mathbf{\Sigma^1_1}$ if it is the projection of a Borel subset of $\mathscr{X} \times \omega^\omega$.A set is $\mathbf{\Pi^1_1}$ if it is the complement of a $\mathbf{\Sigma^1_1}$ set, and it is $\mathbf{\Delta^1_1}$ if it is both $\mathbf{\Sigma^1_1}$ and $\mathbf{\Pi^1_1}$.

In a paper from 1905, Henry Lebesgue himself made a mistake about the measurability of $\mathbf{\Sigma^1_1}$ sets. He stated that projections of Borel sets of the plane into the real line are Borel sets. This mistake was found ten years later by Suslin, who then started the study that eventually led to descriptive set theory.

Note that Definition I.71 is really the non-effective version of Definition I.70. To see this, remark that if A is $\mathbf{\Sigma^1_1}$, then there exists a Borel set B such that $x \in A$ if and only if $\exists g \in \omega^\omega, (x, g) \in B$. We recognize the existential quantifier over the Baire space, however the effective counterpart of Borel sets is hyperarithmetic, and not arithmetic as in Definition I.70. The fact that $\mathbf{\Sigma^1_1} = \bigcup_{X \in 2^\omega} \Sigma^1_1(X)$ needs the following result:

▶ **Proposition I.72.** *A set $A \subseteq \mathscr{X}$ is $\mathbf{\Sigma^1_1}$ if and only if it is the projection of a closed subset of $\mathscr{X} \times \omega^\omega$, if and only if it is the range of a continuous function $F : \omega^\omega \to \mathscr{X}$.*

A Σ^1_1-formula is always equivalent to $\exists X \in \omega^\omega, \forall x, \Phi(X, x)$ and $\exists X \in 2^\omega, \forall x \exists y, \Psi(X, x, y)$ for some Δ^0_0 formulas Φ and Ψ.

▶ **Corollary I.72.1.** *The Σ^1_1 sets A are exactly the projections of the effectively closed sets of the Baire space. By Theorem I.35, A is Σ^1_1 if and only if $A = p([T])$ where $T \subseteq \mathscr{X} \times \omega^\omega$.*

As a Corollary, we are able to define simple codes for Π^1_1 and Σ^1_1 sets:

Definition I.73

A Σ^1_1 code is an integer $n = \langle 0, e\rangle$ where e is a code for an effectively closed set A of $\mathscr{X} \times \omega^\omega$ and the associated set is $p(A)$ (where p is the projection along the first coordinate). A Π^1_1 code is an integer $n = \langle 1, e\rangle$ where e is a Σ^1_1 code, and the corresponding set is the complement of A if A is a Σ^1_1 set of code e.
A $\mathbf{\Sigma^1_1}$ code is a real $\langle 0, x\rangle$ where x is a code for a closed set A of $\mathscr{X} \times \omega^\omega$ and the associated set is $p(A)$. A Π^1_1 code is a real $n = \langle 1, x\rangle$ where x is a Σ^1_1 code, and the corresponding set is the complement of A if A is a Σ^1_1 set of code x.

We will further study those sets in Chapter II, as they can be seen as infinite time computations along ω_1^{CK}.

I.8 Reverse Mathematics

Reverse mathematics is a field that seek to find the necessary and sufficient axioms needed to prove a theorem. In some sense, this looks similar to set theory, but it is quite different. First, set theory is a foundation of all mathematics in a very basic but unifying way. This requires a lot of unnatural coding of mathematical objects into sets, the only kind of object in set theory. The axioms of ZF are consensual, they acts as a base theory, the question about which other axioms are needed, and what they imply falls on less convincing sentences such as the Axiom of Choice, the Constructibility of the Universe, the Continuum Hypothesis... These axioms are at the very least not obvious.

In reverse mathematics, we are interested in Second Order Arithmetic. Therefore, there are different types of objects: integers, and set of integers, also called "reals". This allows the language to be more natural than set theory, and to do formal proofs in a more similar way to what we do in real life. All axioms that we consider in reverse mathematics are true with no doubt, the question is more whether they are necessary to prove some result.

We need to define which is our base theory, the minimal axioms that we need to do the simplest form of mathematics. They consists of two type of axioms: the first order part, the one that define the integers, and the second order part, the one that defines the sets of integers.

I.8.1 Second Order Arithmetic

Definition I.74

The language of second order arithmetic $\mathscr{L}_{\mathcal{A}}$ is a two-sorted language, with first sort intended for integers and the second for set of integers. It contains the signature $\{0; S; +; \times; <; =; \in\}$, where 0 is a constant, S is a unary function for first-sort terms, $+$ and $\times$ are binary function for first sort terms, $<$ and $=$ are predicates for first-sort terms and $\in$ is a binary predicate relating first and second sort terms.
It contains quantification for both sort, we will write quantification over sets of integers using upper case letters and those for integers using lower-case letters.

First Order Part

The first step is to define the first order part of our base theory, the axioms that are specially directed toward defining the first sort of the language $\mathscr{L}_{\mathcal{A}}$. We will use the simple axioms from PA:

Definition I.75

We write P_0^- for the collection of the universal closure of the following axioms:

1. $m+1 \neq 0$
2. $m+1 = n+1 \implies m = n$
3. $0 = m \vee \exists n : n+1 = m$
4. $m + 0 = m$
5. $m + (n+1) = (m+n) + 1$
6. $m \times 0 = 0$
7. $m \times (n+1) = (m \times n) + m$
8. $m = 0 \vee 0 < m$
9. $m \not< 0$
10. $m < n+1 \Leftrightarrow (m < n \vee m = n)$

These axioms define the behaviour of $+$, $\times$, $<$. However, the integers still have too much margin and a model can behave very differently from regular integers. We need to add one fundamental property: the induction.

In the case of second order arithmetic, we add the "second order induction axiom" instead of a scheme of axioms.

Definition I.76

We write P_0 for the collections of axioms of P_0^- together with the second order induction axiom

$$\forall X, 0 \in X \wedge (\forall n, n \in X \implies n+1 \in X) \implies \forall n, n \in X$$

Therefore, the second order part has an influence on the first order part, as it defines how much induction will take place.

Second order Part

The second order part of arithmetic consists of the true statement on the sets of integer. One of the most important specifies that definable sets exist.

Definition I.77

If $\mathcal{C}$ is a class of formula, we call $\mathcal{C}$-comprehension the collection of axioms of the form:

$$\exists X : \forall n, n \in X \Leftrightarrow \phi(n)$$

where ϕ is a formula in $\mathcal{C}$.

Definition I.78

By *full second order arithmetic*, we mean P_0 together with comprehension for the class of all formulas of second order arithmetic.

When devising subsystems of full second order arithmetic, we will mainly restrict comprehension to a subsystem, and sometimes add axioms that do not corresponds to a specific type of comprehension.

I.8.2 Encoding objects in Second Order Arithmetic

In order to speak about theorems of mathematics in Second Order Arithmetic, we need to be able to speak about sufficiently many objects. The language allows to deal with integers and sets of integers. This allows to speak about elements of $\mathbb{R}$, using for instance the binary expansion. However, some fields such as analysis need to deal with functions on reals. Should we switch to Third Order Arithmetic?

In fact, Second Order Arithmetic is sufficient for most of mathematics. We are able to code higher order objects such as functions or subsets of reals into a single real. Usually, every notion that has an effective counterpart can be coded into a real. So for instance, every Borel set can be encoded, as $\mathbf{\Delta^1_1} = \bigcup_{X \in 2^\omega} \Delta^1_1(X)$. One can use this to code a Borel set by a couple consisting of an oracle together with an effective Borel code, which is a integer. Usually, there is also a direct way to code an object, such as for ordinals. From this comes the difference between the definitions of $\mathbb{WO}$ and **WO**.

Of course, most of the set of reals are not encodable in Second Order Arithmetic, as the cardinality of codes is less than the cardinality of $\mathcal{P}(\mathbb{R})$. But references to those sets almost never appear in natural mathematics.

I.8.3 The Big Five

When looking for the necessary axioms of theorems from many different fields of mathematics, a surprising scheme emerged: most of them were equivalent to one out of five linearly ordered systems of axioms. These systems correspond to various kind of arguments used in proofs, but they also correspond to the computational complexity of the objects involved in the proofs. A very good book on the subject is by Simpson, and can be found in [60].

Recursive Comprehension Axiom scheme

The first of them will be the base theory in most of our coming comparisons of theorems and principles. It is the simplest system that allows to conduct computations in mathematics.

Definition I.79 (RCA_0)

RCA_0 is the system of axioms consisting of P_0 together with Δ^0_1-comprehension and Σ^0_1-induction scheme. Note that Δ^0_1 is not a syntactic notion, therefore we need to clarify the axioms: they consists of the following axiom for every ϕ and ψ Σ^0_1 formulas:

$$[\forall n, \phi(n) \Leftrightarrow \neg\phi(n)] \implies \exists X : [\forall n, n \in X \Leftrightarrow \phi(n)]$$

When the universe of a model is $\mathbb{N}$, then there is a characterization of models of RCA_0.

Definition I.80 (Turing ideal)

A *Turing ideal* $\mathcal{I} \subseteq \mathcal{P}(\mathbb{N})$ is a collection of sets such that for all $A, B \in \mathcal{I}$ we have $A \oplus B \in \mathcal{I}$ and if $C \leq_T A$ then $C \in \mathcal{I}$.

Theorem I.81

A second order structure $(\mathbb{N}, \mathcal{I})$ is a model of RCA_0 if and only if it is a Turing ideal.

Weak Kőnig's Lemma

This principle corresponds to the use of compactness in an argument. The corresponding computational power is those of PA degrees: being able to compute a path on every infinite binary tree.

Definition I.82 (WKL_0)

WKL_0 is the system of axioms consisting of RCA_0 together with "every infinite binary tree has an infinite path".

Definition I.83

A *Scott set* is a Turing ideal $\mathcal{I}$ such that if $A \in \mathcal{I}$, then there exists $B \in \mathcal{I}$ such that B is PA over B. In other words, for every $T \subseteq 2^{<\omega}$, if T is infinite then T has a path in $\mathcal{I}$.

Theorem I.84

A second order structure $(\mathbb{N}, \mathcal{I})$ is a model of WKL_0 if and only if it is a Scott set.

Arithmetical Comprehension Axiom scheme

This principle is used to assert that every arithmetically defined set exists. The corresponding computational power is those of PA degrees: being able to compute a path on every infinite binary tree.

Definition I.85 (ACA_0)

ACA_0 is the system of axioms consisting of RCA_0 together with arithmetic comprehension.

In particular, this implies that the jump of any set must exists in the model, as it is Σ^0_1. In fact, any iteration of the jump must exist, but one proves them all.

Theorem I.86

A second order structure $(\mathbb{N}, \mathcal{S})$ is a model of ACA_0 if and only if it is a Turing ideal and for every $A \in \mathcal{S}$, $A' \in \mathcal{S}$.

Note that even though ACA_0 does not contain the axiom scheme of Σ^0_n-induction, it proves it using induction for sets and Σ^0_n-comprehension. In fact, it proves more than that: transfinite

induction holds for any linear order that is an ordinal in the model. However, it does not prove Σ^0_n-transfinite recursion, as this needs a real quantifier to be proved.

Arithmetic Transfinite Recursion scheme

Arithmetic transfinite recursion is the axiom scheme that allows us to define things by transfinite recursion, using arithmetical steps. Given a well-order on $\mathbb{N}$ and an arithmetical formula, we define the transfinite recursion to be a set where each slice of some elements of the order is obtained using ϕ from the slices of previous elements. The sets of those slices is called a hierarchy.

Definition I.87 (Φ-hierarchy)

Let Φ be any formula (with parameters), $\trianglelefteq$ be a partial order on ω. We say that $H \subseteq \omega$ is a *hierarchy for* Φ *on* $\trianglelefteq$ if for every $j \in \omega$, we have:

$$H_j = \{n \in \mathbb{N} : \Phi(n, \bigoplus_{i \lhd j} H_i)\}$$

where $H_j = \{n : \langle j, n\rangle \in H\}$ and $\bigoplus_{n \in F} X_n = \{\langle n, x\rangle : n \in F \wedge x \in X_n\}$.
We write $\mathsf{Hier}_\Phi(H, \trianglelefteq)$ for "H is a hierarchy for Φ on $\trianglelefteq$". If Φ is Δ^0_n, then $\mathsf{Hier}_\Phi(H, \trianglelefteq)$ is a Π^0_n formula, as it can be expressed by $\forall \langle x, n\rangle \in \mathbb{N}$, $\langle x, n\rangle \in H \Leftrightarrow \Phi(n, \Phi(n, \bigoplus_{i \lhd n} H_i))$

This definition corresponds to the trace of a definition by induction along $\lhd$: H_j is defined from Φ and the previously defined sets. In our definition, we do not require the order to be well-founded: a hierarchy on such an ordering would correspond to a definition by induction, but with no starting point...

Definition I.88 (ATR_0)

ATR_0 is the system of axioms consisting of ACA_0 together with the axiom scheme consisting of:

$$\forall \trianglelefteq \text{ binary relation, } \mathsf{WO}(\trianglelefteq) \implies \exists H, \mathsf{Hier}_\phi(H, \trianglelefteq)$$

for every arithmetical ϕ, where WO stands for the Π^1_1 formula "$\lhd$ is a well-order".

We will see in Chapter IV that neither the set of arithmetic sets nor the set of hyperarithmetic sets is a model of ATR_0. For the first one, it is clear as $\emptyset^{(\omega)}$ is not arithmetic but must be in every model of ATR_0. The second case is less trivial and needs the existence of linear orders with no hyperarithmetic infinite descending sequences (so that in HYP the order appears as well-founded) but with no hyperarithmetic hierarchy. In fact, there is no minimal model for ATR_0 (see [60]).

We allowed arbitrary arithmetical formulas to do the transfinite construction, but a single jump would have sufficed.

Definition I.89 (Jump-Hierarchy)

We say that H is a *Jump-Hierarchy on* $\trianglelefteq$ if H is a Hierarchy for $\Phi(\langle a, b\rangle, X) \equiv b \in X'_a \equiv$

$\varphi_b^{X_a}(b)\downarrow$. In other word, if H is a Jump-Hierarchy for $\trianglelefteq$, then

$$H_a = \bigoplus_{b \lhd a} H'_b.$$

The predicate "H is a Jump-Hierarchy for $\trianglelefteq$" is written $\mathsf{JH}(H, \trianglelefteq)$ and is Π^0_2.

Π^1_1 Comprehension Axiom scheme

The last axiom scheme of the big five is Π^1_1-comprehension. In fact, in the context of RCA_0, Π^1_1-comprehension implies also Σ^1_1-comprehension, so it implies all the previous one. We will not study it in this thesis.

Definition I.90 (ACA_0)

Π^1_1-CA_0 is the system of axioms consisting of RCA_0 together with Π^1_1 comprehension.

Chapter II

Infinite time computations

Gotlib, RàB T2. Le matou matheux

In this chapter, we deal with notions of computation along infinite time. Infinite time computations are the backbone of this thesis, applied to different domains such as randomness and reverse mathematics.

The rough idea of infinite time computation is not to be constrained by having to take decisions after finitely many steps. For instance, given an infinite string, a normal computation cannot check if there is a 1 at some point: If there is no 1 in the input, then it would take infinitely many steps to check all the cells, and there would be no time left for further computation.

Now, using an infinite computation, one should be able to compute $\emptyset'$. Indeed, to check if a particular Turing machine halts, it suffices to run it for each finite step of computation. If it halts at some point, we answer correspondingly, and if it never halts after running for an infinite amount of time, it will never halt. It should also be able to determine if a Turing machine is total, that is if it halts on every input. One way to check this, would be to check if $\phi(n)$ halts for every n. But this suggests "infinitely many" infinite time of computation.

Ordinal numeration appears to be suitable for this settings. For instance, $\omega + \omega$ corresponds to infinitely many steps, followed by infinitely many steps, while ω^2 corresponds to infinitely many times "infinitely many steps". An ω^2 length computation should be able to check the totality of a Turing machine (or even an $\omega + \omega$ length computation can do it, using some parallelization [1]). But this is only informal, and there are still lots of details that need to be provided. For instance, the ordinals are not a set. So should we give a bound on the total number of steps of an infinite computation? If we do not do this, do we risk to end with the collection of computations not being a set?

1. Using an infinite time Turing machine of Definition II.40, even an $\omega + 1$ computation suffices using the $\liminf$ rule.

Also, there is the question of the implementation at limit steps. In this introduction, we said that using infinite time it is possible to check the halting problem for a machine. But how do we actually implement this idea? There are several answers to this question. The first one is that we do *not* implement this: instead, we use the relation between definability and computability, and argue that some definability notions correspond to our intuitive notion of infinite computation.

The first notion of infinite computation that we will study takes this approach. We call it "higher computability" here, although this term sometimes refers to the whole field of infinite computation. It comes from descriptive set theory, and in fact, we have already defined the notion without giving it from the viewpoint of infinite computation.

Contrary to this, Infinite Time Turing Machines are more commonly defined using a model of computation. That is, an Infinite Time Turing Machine behaves like a normal Turing machine at successor steps, and has a behavior at limit steps that allows it to continue the computation, keeping a reasonable memory of what happened before. In this version of infinite computations, the ordinal number of steps is not bounded, we will see that however there is still a bound on the ordinal number of steps it can reach before starting to loop. A suitable restriction of this model also gives a model of the higher computability definitions.

The third notion of infinite computation of this thesis, α-recursion, was first studied in the setting of computability on ordinals instead of integers. In this field, an α-computable set was a subset of the ordinals below α. However, this model still makes sense for the subset of integers, and is very general as it includes in some way the two previous ones. The α in α-recursion corresponds to the fixed bound on infinite computations, which for well chosen ordinals is not a constraint to our intuition. Once again, α-recursion isn't defined with a model of computation. The disadvantages of this is that relativization does not work as well as for instance Infinite Time Turing Machine.

After having defined a computability notion, we will often use informal definitions of algorithm, just as we do with Turing machines. We will always give the same example: the construction of a simple set, a co-infinite c.e. set that intersects every infinite c.e. set. This class of sets was defined by Post when he was trying to solve its so-called Post's problem, as a simple set S can only be non-computable: otherwise, $\mathbb{N} \setminus S$ would be computable, and that would be an infinite c.e. set that does not meet S. We give the proof of the normal case, as a reference for later higher versions:

Theorem II.1

There exists a simple set, i.e. a set S co-infinite and such that $S \cap W \neq \emptyset$ for every infinite c.e. set W.

Proof. Let M be the machine enumerating n if and only if there exists some e such that n is the first element above $2e$ enumerated by W_e (we can suppose that at most one element is enumerated at each step). Formally, n is such that

$$\exists t, \forall m_0 < t, [(m_0 \neq n \wedge m_0 \geq 2e) \Rightarrow m_0 \notin W_{e,t'}] \wedge n \in W_{e,t} \wedge n \geq 2e.$$

Recall that no element above t is enumerated in any $W_{e,t}$.

Call S the set enumerated by M. It is clear that S is c.e. set. Also, for every $e \in \mathbb{N}$, only e numbers can be enumerated below $2e$, so at least e are not in S, and S is co-infinite. If W_e is infinite, then some $n \geq 2e$ is enumerated in W_e, and the first such will be enumerated in S. ∎

Higher computability will be used in Chapter IV, while α-recursion and Infinite Time Turing Machine will be used in Chapter V.

II.1 Higher Computability

A common way to generalize a notion is to generalize one of its characterization. For (a very unrelated) example, the function sin is commonly defined using the unit circle and trigonometric considerations. This definition gives no clue on how to extend this function to imaginary numbers. However, the equality $\sin(x) = \frac{e^{ix}-e^{-ix}}{2i}$ allows to extend sinus to imaginary number in a very straightforward way.

The following notion of computability is defined by modifying a known characterization of classical computability. Let us recall Theorem I.44, that relates definability and computability:

▶ **Theorem.** *Let $A \subseteq \mathbb{N}$ be any set. Then:*

1. *A is computably enumerable if and only if A is Σ^0_1.*
2. *A is co-computably enumerable if and only if A is Π^0_1.*
3. *A is computable if and only if A is Δ^0_1.*

What happens if we replace quantification over integers by quantification over reals, do we still get some kind of computations? The answer is yes, but we need a few material from descriptive set theory to see that.

II.1.1 Lightface analytic and co-analytic sets

We already defined the class of sets that are central for higher recursion, we will study them further in this section. The goal is to show how they can be seen as computations, or enumerations, along ω_1^{CK}. It is separated in two parts: first, we show that every Π^1_1 set can be seen as a uniform enumeration along ω_1^{CK}. Then, we show that defining a set by Δ^1_1 enumeration along ω_1^{CK} yields a Π^1_1 set.

Π^1_1 as uniform enumerations along ω_1^{CK}

This comes from the following important result:

Theorem II.2

WO is Π^1_1-complete: for every Π^1_1 set $A \subseteq \mathbb{N}$, there is a computable function f such that $n \in A$ if and only if $f(n) \in \mathsf{WO}$.
The relativized version also holds: WO^X is $\Pi^1_1(X)$-complete.

Proof. First, it is clear that WO is Π^1_1, as

$$e \in \mathsf{WO} \iff e \in \mathsf{LO} \wedge \forall X \in \omega^\omega,\ \exists n \in \mathbb{N},\ X(n+1) \not\leq_e X(n)$$

where $\leq_e$ is the relation coded by W_e. Being in LO is arithmetic, as the axioms have only integer quantifiers.

Now suppose that $A \subseteq \mathbb{N}$ is Π^1_1. Then $n \in A \Leftrightarrow \forall X \in \omega^\omega,\ \exists t \in \mathbb{N},\ R(X,t,n)$, which is also equivalent to the fact that the c.e. tree $T_n = \{\sigma : \exists \tau \succ \sigma\ \exists t \in \mathbb{N},\ R(\tau,t,n)\}$ contains no infinite path, or seen as a partial order, is well-founded. This last fact is true iff $\mathrm{KB}(T_n)$ is well-founded. Let f be the computable function, mapping any n to the index of a c.e. order corresponding to $\mathrm{KB}(T_n)$. Then, $n \in A$ if and only if $f(n) \in \mathsf{WO}$. ■

▶ **Corollary II.2.1.** *$\mathcal{O}$ is also Π^1_1-complete.*

▶ **Corollary II.2.2.** *For every $A \subseteq \mathbb{N}$ a Π^1_1 set, there exists a uniform family of Δ^1_1 sets such that $A = \bigcup_{\alpha<\omega_1^{\mathrm{CK}}} A_\alpha$ and the family $(A_\alpha)_{\alpha<\omega_1^{\mathrm{CK}}}$ is increasing. By uniform, we mean that given any computable index for α, one can computably obtain a Δ^1_1 code for A_α.*

Proof. As WO is Π^1_1-complete, let f be computable such that $n \in A$ iff $f(n) \in \mathsf{WO}$. Now, let $A_\alpha = \{n : f(n) \in \mathsf{WO}_{<\alpha}\}$. Clearly, we have $A = \bigcup_{\alpha<\omega_1^{\mathrm{CK}}} A_\alpha$. ■

This last Corollary shows how every Π^1_1 set can be seen as an enumeration along ω_1^{CK}.

Uniform enumerations along ω_1^{CK} as Π^1_1 sets

We show that conversely, Π^1_1 sets can be defined by an enumeration of integers along ω_1^{CK}. The enumeration is defined by a Σ^1_1 function, which given the previous stages of the enumeration, answers the newly enumerated integers. This is made precise in the following theorem.

Theorem II.3

Let $F \subseteq \mathbb{N} \times \mathbb{N}^{\mathbb{N}} \times \mathbb{N}$ be Σ^1_1 such that for every $\alpha < \omega_1^{\mathrm{CK}}$ and for every $f : \mathcal{O}_{<\alpha} \to \mathbb{N}$ such that $\forall b \in \mathcal{O}_{<\alpha}, F(b, f_{\restriction \mathcal{O}_{<b}}, f(b))$, there exists a unique n such that for every $a \in \mathcal{O}_{=\alpha}$, we have $F(a, f, n)$.
Then, there exists a unique Π^1_1 function f such that for every $\alpha < \omega_1^{\mathrm{CK}}$ and $a \in \mathcal{O}_{=\alpha}$, we have $F(a, f_{\restriction \mathcal{O}_{<\alpha}}, f(a))$.

The informal corollary of this theorem is that we can define a Π^1_1 set by induction along the computable ordinals. The formula F in the theorem is the definition of a step in the computation: it has access to a code corresponding to the ordinal stage, to a function f that associates previous ordinal stages to the numbers enumerated at these stages, and it defines in a unique manner what number is enumerated at the current stage. Then, the enumerated set is a set A with $n \in A \iff \exists e \in \mathcal{O},\ n = f(e)$ i.e. A is the set of elements added at some stage.

▶ **Example II.4.** There exists a higher simple set. This set cannot be Δ^1_1.

Proof. Let F of Theorem II.3 such that the corresponding enumerated set S is the following: at stage α, S enumerates all n for which there exists some e such that n is the smallest verifying $n \geq 2e$, $n \in P_{e,\alpha}$ but no element of P_e has already been enumerated in S at earlier stage.

Then, the enumerated S is a "higher simple set": it is co-infinite, Π^1_1 and intersects every infinite Π^1_1 set. Therefore, its complement cannot be Π^1_1 so S is not Δ^1_1. ■

▶ **Corollary II.4.1.** *Neither* WO *nor* $\mathcal{O}$ *are* Σ^1_1.

Proof. As they both are Π^1_1-complete, if they were also Σ^1_1 then the higher simple set of the previous example would be Σ^1_1, a contradiction. ■

II.1.2 Properties of higher recursion

In this section, we study the properties of higher recursion, in particular we emphasize the differences with normal computability and how we can still prove many of the results.

The first and good thing is that there exists a universal Π^1_1 set, a set $U = \{(e, n) : n \in P_e\}$ where P_e is the set of Π^1_1 code e. Therefore, in a Π^1_1 enumeration we can act depending on the enumerations of other Π^1_1 sets. We already used this fact in Example II.4.

However, there is a new concern on the bound of transfinite computations: Suppose for instance that $\omega + \omega$ was a bound in place of ω_1^{CK}, then any operation taking ω steps could not be done twice. We shall make sure that this sort of things do not happen.

The requirement we want on the bounding ordinal time, is that if we can do α steps in a computation, and there is α operations we need to do, then we have time to do them all. As an informal example, if we can in ω steps compute the jump of a set, and if we can do α steps, we expect to be able to compute the α-th jump. This requires the bounding ordinal to be closed by sufficiently many operations. It is the case for ω_1^{CK} in the strong way expressed in the following theorem.

Theorem II.5 (Spector's Σ^1_1-boundedness)

Let $A \subseteq \mathsf{WO}$ be a Σ^1_1 set. Then, there exists some computable α such that $A \subseteq \mathsf{WO}_{<\alpha}$. Moreover, a code for α can be found uniformly in a code for A.

Proof. Suppose not. Then, we would have $e \in \mathsf{WO}$ if and only if $\exists a \in A$ such that $e <_0 a$, which is a Σ^1_1 formula. WO would then be Δ^1_1, a contradiction with Corollary II.4.1.

For the uniformity, the idea is to define α to be the supremum over $n \in \mathbb{N}$ of the minimum between the step where n is removed from A and the ordinal coded by n. This is possible, as given two orders where at least one is well-founded, one can define the tree of decreasing sequences of one element from both orders, and take the Kleene-Brouwer ordering of this. This ordinal is recursive and above every ordinal of A. ■

Another important difference between higher computability and regular computability, is that the time of computation, and the size of the sets we compute, have two different length. Sometimes, proofs in the lower case make use of the "homogeneity" of the set of enumerated elements and the time of computation, in this case it is called a "time trick", as first defined in [7, 9].

Sometimes, time tricks are not avoidable, and sometimes there is a way around. One way around is the fact that, even though the computations length are ordinals, they are bounded by a countable ordinal, and more than that there is a Π^1_1 injection from ordinal times to integers.

We first see an example of time trick where the corresponding fact does not hold in higher computability. Then, we will define the projectum function and show an example where it is a way around a time trick.

Recall that an effectively open set is of the form $\bigcup_{\sigma \in W_e} [\sigma]$. However, there might be some redundancy in W_e, in other words two strings $\sigma_0 \prec \sigma_1$ in W_e. A set that contains no redundancy is called a prefix-free set.

▶ **Proposition II.6.** *For every effectively open set A, there exists a prefix-free c.e. set W such that $A = \bigcup_{\sigma \in W} [\sigma]$.*

Proof. Let V be c.e. such that $A = \bigcup_{\sigma \in V} [\sigma]$. We define the enumeration W prefix-free, from the enumeration of V. If σ is enumerated at stage t, then if σ has length t or more we enumerate σ, otherwise we enumerate all extensions of σ of length t. ■

The higher analogue does not hold.

▶ **Proposition II.7.** *There exists a Π^1_1 set of strings V such that for every prefix-free Π^1_1 set W, we have:*

$$\bigcup_{\sigma \in V} [\sigma] \neq \bigcup_{\tau \in W} [\tau].$$

Proof. We first define V_e, a Π^1_1 set of strings such that either the e-th Π^1_1 set P_e is not prefix-free, or it does not define the same effectively open set as V_e. Here is the enumeration of V_e.

At the first stage, V_e enumerates all strings 0^n1 for $n \in \omega$. Then, at all later stages, if $[P_e]^\prec \neq [V_e]^\prec$, then V_e does not enumerate anything. Otherwise, $[P_e]^\prec = [V_e]^\prec$ and V_e enumerates the empty string.

Suppose that in the end, $[P_e]^\prec = [V_e]^\prec$. Then, let α_n be the stage where $[0^n1] \subseteq [P_e]^\prec \neq 2^\omega$. Then, by Spector's Σ^1_1 boundedness Theorem II.5, there exists a bound $\alpha < \omega_1^{\mathrm{CK}}$ above every α_n. But then, at stage $\sup \alpha_n$, V_e enumerates the empty string, so at this stage we have $[P_e]^\prec = 2^\omega \setminus \{O^\omega\}$ and $[P_e]$ cannot equal $[V_e]^\prec = 2^\omega$ while staying prefix-free.

Now, we do a product of the construction: the final set of string V is $\bigcup_{e\in\mathbb{N}} 0^e1^\frown V_{f(e)}$ where $f(e)$ is the Π^1_1-index of $\{\tau \in 2^{<\omega} : 0^e1^\frown\tau \in P_e\}$. ■

Some other time tricks can still be avoided in higher computability, by the use of a projectum. The general notion of projectum will be defined in the next section on α-recursion, but we give the special case of projectum function in higher computability.

Definition II.8

The *projectum function* of ω_1^{CK} is the injective function $p : \omega_1^{\mathrm{CK}} \to \omega$ such that $p(\alpha) = \min(\mathcal{O}_{=\alpha})$.

▶ **Proposition II.9.** *There exists a Π^1_1 function, $\hat{p} : \mathcal{O} \to \omega$ such that if $\alpha < \omega_1^{\mathrm{CK}}$ is of code $a \in \mathcal{O}_{=\alpha}$, then $\hat{p}(a) = p(\alpha)$.*

In other words, the projectum function p is higher computable in terms of code for ordinals. Note that it is coding independent, as required in Theorem II.3 for definition of Π^1_1 sets as enumerations.

We can give the following (trivial) example of a use of the projectum. There exists a c.e. set enumerating a (strictly) new element at each stage of the enumeration. For the basic case, this is trivial: at stage t, just enumerate t. For the higher case, we need the projectum function: at stage α, just enumerate $p(\alpha)$.

Continuing the example of Proposition II.9, we give a more elaborate use of projectum. An ϵ-prefix free is a set of strings such that the measure of the redundancy, when seen as the description of an open set, is bounded by ϵ. A set $W \subseteq 2^{<\omega}$ is ϵ-prefix free if $\sum_{\sigma\in W} \mu([\sigma]) \leq \mu([W]^\prec) + \epsilon$, where μ is the Lebesgue measure of Definition V.5.

▶ **Proposition II.10.** *For every non-zero $\epsilon \in \mathbb{R}^+$ and Π^1_1 set of strings V, there exists an ϵ-prefix-free Π^1_1 set of string W with $[V]^\prec = [W]^\prec$.*

Proof. We can suppose ϵ is rational are those are dense in the reals. Fix $V \subseteq 2^{<\omega}$, we describe the enumeration of an ϵ-prefix free description W of V. At stage α, suppose V enumerates the string σ, and W_α denotes the string already enumerated in W. Then, $[\sigma]\setminus[W_\alpha]^\prec$ is a closed set, so there exists an open set $U \supseteq [\sigma] \setminus [W_\alpha]^\prec$ with $\mu(U) \leq \mu([\sigma] \setminus [W_\alpha]^\prec)$. We have $[\sigma] \subseteq [W_\alpha]^\prec \cup U$, so by compactness it is already true for a finite union of intervals. Now, W enumerates the first finite set of strings found S such that $[\sigma] = [S]^\prec \cup [W_\alpha]^\prec$ and $\mu([S]^\prec \cap [W_\alpha]^\prec) \leq 2^{-p(\alpha)} \times \epsilon$. ■

II.1.3 Higher computable sets

In this section, we prove that the higher computable sets are exactly the hyperarithmetic sets. This is satisfying in the sense that it gives a bound on the complexity of higher computable

sets: those that are computable by the α-th jump, for α a computable ordinal.

We first sketch the proof that all hyperarithmetic sets are Δ^1_1. Then, we sketch the converse: every Δ^1_1 set is hyperarithmetic.

Theorem II.11

For every recursive ordinal α, $\emptyset^{(\alpha)}$ is Δ^1_1.

Proof sketch For any constructive code a of α "$X = \emptyset^{(a)}$ is arithmetic. Therefore, $n \in \emptyset^{(a)}$ if and only if $\forall X \in 2^\omega, X = \emptyset^{(a)} \implies n \in X$ if and only if $\exists X \in 2^\omega, X = \emptyset^{(a)} \implies n \in X$. ■

▶ **Corollary II.11.1.** *Every hyperarithmetic set is* Δ^1_1.

Theorem II.12

Every Δ^1_1 set is hyperarithmetic.

Proof. Let $A \subseteq \mathbb{N}$ be Δ^1_1. By Corollary II.2.2 applied to A and its complement, fix some computable functions f and g such that if $A_\alpha = \{n : f(n) \in \mathsf{WO}_{\leq\alpha}\}$ and $B_\alpha = \{n : g(n) \in \mathsf{WO}_{\leq\alpha}\}$, then $(A_\alpha)_{\alpha\in\omega_1^{\text{CK}}}$ and $(B_\alpha)_{\alpha\in\omega_1^{\text{CK}}}$ are uniform Π^1_1 sequences such that $A = \bigcup_{\alpha<\omega_1^{\text{CK}}} A_\alpha$ and $\mathbb{N} \setminus A = \bigcup_{\alpha<\omega_1^{\text{CK}}} B_\alpha$.

Now, for every n, there exists a smallest computable ordinal α_n such that either $n \in A_{\alpha_n} \cup B_{\alpha_n}$. Its code a_n can be uniformly found, so the set $\{a_n : n \in \mathbb{N}\} \subseteq \mathcal{O}$ is Σ^1_1. By Spector's Σ^1_1-boundedness Theorem II.5, there exists a computable bound α above every α_n.

Then, $A = \bigcup_{\beta\leq\alpha} A_\beta = A_\alpha$. Recall that $A_\alpha = \{n : f(n) \in \mathsf{WO}_{\leq\alpha}\}$ for some computable function f. As $\mathsf{WO}_{\leq\alpha}$ is hyperarithmetic, so is A. ■

II.1.4 Higher recursion with sets of reals

One of the advantages of infinite computation is that it allows to compute sets of reals in a more interesting way than finite computations. We can define computability for subsets of real exactly the same way than for subsets of integers: the characteristic function is computable. In the case of subsets of real, the characteristic function must then be a Turing functional.

So a subset A of the real is decidable if there exists a Turing machine such that if we write $x \in 2^\omega$ in its output tape, it will halt and answer whether $x \in A$ or not. However, as a halting Turing machine can only see a finite part of its input, the decidable sets are all open sets with recursive description, and in the end there is no fundamental difference between decidability of sets of reals, and computability of sets of integers.

In the case of infinite computability, the entire input can be seen, and the higher decidable sets need not to be open. Also, with certain input, ω_1^{CK} might not be a high enough bound on the length of computation. This time, Δ^1_1 subsets of reals can be seen as computations along ω_1. However, an answer on the belonging of X has to be given before the ω_1^X-th step.

Theorem II.13

$\mathbb{WO}$ is also $\mathbf{\Pi^1_1}$-complete, in the sense that for every $\mathbf{\Pi^1_1}$ subset of the reals $\mathcal{A}$, there exists a computable Turing functional Φ such that $x \in \mathcal{A}$ iff $\Phi^x \in \mathbb{WO}$.
WO is Π^1_1-complete for subsets of the reals. More than that, for every Π^1_1 set $A \subseteq 2^\omega$,

there exists an integer e such that

$$\forall X \in 2^\omega,\ X \in A \iff e \in \mathsf{WO}^X.$$

Each Π^1_1 set can be decided by a computation along ω_1^X where X is the input:

Theorem II.14

Let $F \subseteq \mathbb{N}^\mathbb{N} \times \mathbb{N} \times \mathbb{N}^\mathbb{N} \times 2$ be Σ^1_1 such that for every X, $\alpha < \omega_1^X$ and for every $f : \mathcal{O}^X_{<\alpha} \to \mathbb{N}$ such that $\forall b \in \mathcal{O}^X_{<\alpha}, F(X, b, f_{\restriction \mathcal{O}^X_{<b}}, f(b))$, there exists a unique i such that for every $a \in \mathcal{O}^X_{=\alpha}$, we have $F(X, a, f, i)$.
Then, there exists a unique Π^1_1 function $f \subseteq: 2^\omega \times \mathcal{O} \to 2$ such that for every X, $e \mapsto f(X, e)$ is defined only on $\mathcal{O}^X$, and for every $\alpha < \omega_1^X$ and $a \in \mathcal{O}^X_{=\alpha}$, we have $F(X, a, (n \mapsto f(X, n))_{\restriction \mathcal{O}^X_{<\alpha}}, f(a))$. The set defined by the computation F is

$$A_F = \{X \in 2^\omega : \exists a \in \mathcal{O}^X,\ f(X, a) = 1\}$$

We still have some kind of Spector Σ^1_1 boundedness for reals, which gives us that Δ^1_1 sets are Σ^0_α sets for some computable α.

Theorem II.15

If $A \subseteq \mathbf{WO}$ is Σ^1_1, then there exists a computable ordinal α such that $A \subseteq \mathbf{WO}_{\leq\alpha}$.

Proof. Suppose otherwise. Then, WO is also Σ^1_1, as $\trianglelefteq \in \mathsf{WO}$ if and only if $\exists X \in 2^\omega$ and $\exists e \in \mathbb{N}$ such that $(e, X) \in A$ and there exists an order-preserving injection from $\mathrm{dom}(\trianglelefteq)$ and Φ^X_e. ∎

We also have that the higher decidable subsets of the reals are the hyperarithmetic ones.

Theorem II.16

A set $A \subseteq \omega^\omega$ is Δ^1_1 if and only if it is hyperarithmetic.

II.2 α-recursion

In the previous section, we based on Theorem I.44 our generalization of computations to infinite computations. We replaced quantification over integers by quantification over reals, and it turned out that these definitions in some sense yield infinite computations over ω_1^{CK}. However, something unexpected appeared: the counterpart of c.e. sets, which is originally the Σ^0_1-definable sets, is the Π^1_1-definable sets rather than the Σ^1_1-definable sets.

This suggests that the quantifications do not have the same role in the lower and the higher settings. In the proof of Theorem I.44, the existential quantifiers assert the existence of the trace of a halting computation. In the higher case, a trace of computation would be a computable ordinal defining the steps, and "rules for next step", the function F of Theorem II.3. Quantifying over the existence of such a computation is Π^1_1 as $\mathbf{WO}$ is Π^1_1, so we can get some existential quantification as Π^1_1 sentences. The following theorems of Kleene, Spector and Gandy makes

this precise:

Theorem II.17 (Kleene, [59, p.59])

If $A(x, f)$ is arithmetic, then "$\exists f \in \mathrm{HYP},\ A(x, f)$" is a Π^1_1 predicate.

Proof. By the fact that **WO** is Π^1_1-complete and that hyperarithmetic and Δ^1_1 sets are the same, a set $A \subseteq \mathbb{N}$ is hyperarithmetic if and only if it is computable in some recursive iteration of the jump. But then, $\exists f \in \mathrm{HYP},\ A(x, f)$ is equivalent to

$$\exists f, g \text{ total and recursive, } f(n) \notin \mathcal{O} \implies g(n) \in \mathcal{O} \forall A$$

,

■

Theorem II.18 (Spector, Gandy, [59, p.61])

Every Π^1_1 predicate $P(x)$ can be put in the form "$\exists f \in \mathrm{HYP}, A(x, f)$" for some arithmetical A.

We are now able to revisit Theorem II.3.

Theorem II.19

Let $A \subseteq \mathbb{N}$ be any set. Then:

1. A is c.e. iff $\exists \phi \in \Sigma_1$ such that $n \in A \iff \mathbb{N} \models \phi(n)$.
   A is higher c.e. iff $\exists \phi \in \Sigma_1$ such that $n \in A \iff \mathrm{HYP} \models \phi(n)$.
2. A is co-c.e. iff $\exists \phi \in \Pi_1$ such that $n \in A \iff \mathbb{N} \models \phi(n)$.
   A is higher co-c.e. iff $\exists \phi \in \Pi_1$ such that $n \in A \iff \mathrm{HYP} \models \phi(n)$.
3. A is computable iff it is both c.e. and co-c.e.
   A is higher computable iff it is both higher c.e. and higher co-c.e.

This new vision of higher computations allows us to generalize computations even more. Let $\mathcal{A}$ be any set containing at least the integers. Then, a set A would be $\mathcal{A}$-c.e. if and only if it is Σ_1-definable over $\mathcal{A}$. If $\mathcal{A}$ is sufficiently closed and contains sufficiently many ordinals, we expect this definition to be a good definition for infinite computations of length more than ω_1^{CK}.

However, it remains to choose the set $\mathcal{A}$ so that this definition makes sense. Gödel's constructibles, that we will introduce in the next section, make a good candidate for this, when restricted to specific layers.

II.2.1 Godel's constructible universe

Gödel's constructible universe was introduced by Gödel in 1938, in order to prove consistency of the Axiom of Choice and the Generalized Continuum hypothesis. But Gödel's constructible have many more applications. For instance, they form an important hierarchy of complexity of all constructible sets.

The idea behind Gödel's constructible is to add all the sets that can be defined by comprehension, layer by layer. This way, we will get the smallest "closed by comprehension" universe. So, we start with the empty set, and add all the sets that can be defined by comprehension from

the empty set. This will define new sets. Using these new sets, we are able to define new sets. And so forth and so on, we continue to add new definable sets from the previous ones, along the ordinals.

The constructible universe is usually defined starting with $L_\emptyset = \emptyset$. When using some oracle $x \in 2^\omega$, it starts with $L_\emptyset(x) = \mathrm{tc}(x)$ (which equals $\{x\} \cup \omega$ when x is infinite). In order to keep some consistency between the constructible universe defined with and without oracle, we start with $L_\emptyset = \omega$.

Definition II.20

The constructible universe is defined by induction over the ordinals as follow:

- $L_\emptyset = \omega$
- $L_{\alpha^+} = \{X \subseteq L_\alpha : X$ is first order definable in L_α with parameters in $L_\alpha\}$
- $L_{\sup_{\gamma<\alpha}} = \bigcup_{\gamma<\alpha} L_\gamma$

Let $x \in 2^\omega$. The constructible universe starting with x as an oracle is defined by induction over the ordinals as follow:

- $L_\emptyset(x) = \{x\} \cup \omega$
- $L_{\alpha^+}(x) = \{X \subseteq L_\alpha(x) : X$ is first order definable in $L_\alpha(x)$ with parameters in $L_\alpha(x)\}$
- $L_{\sup_{\gamma<\alpha}}(x) = \bigcup_{\gamma<\alpha} L_\gamma(x)$

For $a \in L$, the rank of a, denoted by $\mathrm{rk}(a)$, is the smallest α such that $a \in L_\alpha$.

Our choice of having $L_\emptyset = \omega$ makes the first levels of the constructible peculiar, as α is not the smallest ordinal not in L_α for $\alpha < \omega^2$. However, at the ω^2 layering our definition retrieve this property, and for every $\alpha \geq \omega^2$, we have $\alpha \subseteq L_\alpha$, and $\alpha \in L_{\alpha+1}$ but $\alpha \notin L_\beta$ for any $\beta < \alpha + 1$.

Any limit level has some closure properties, for instance it satisfies Δ_0 comprehension. However, some layers satisfy more than this, and are particularly interesting for our purpose of defining computations inside them. We give two examples, the first of them should ring a bell related to higher recursion.

Theorem II.21 ([25, Theorem 3.6.8])

For every set $A \subseteq \mathbb{N}$, we have A is Δ^1_1 if and only if $A \in L_{\omega_1^{\mathrm{CK}}}$.

This is exactly going in the direction of choosing some layers of the constructible hierarchy to define α-recursion: we have that higher c.e. is equivalent to Σ_1-definable over $L_{\omega_1^{\mathrm{CK}}}$!

We now define a layer of the constructibles above all the ones we will be interested in. This layer contains exactly the sets whose "belonging" relation can be coded into a real (the same way countable ordinals are exactly the only ordinals whose $\leq$ relation can be encoded in a real).

Definition II.22

We define hereditarily countable sets by induction on the membership relation. A set A is *hereditarily countable* if it is countable and all of its members are hereditarily countable.

Theorem II.23

The set L_{ω_1} contains only hereditarily countable sets.

Proof. We show by induction that L_α is hereditarily countable for every $\alpha < \omega_1$. We have that $L_0 = \omega$ is hereditarily countable. If $\alpha = \beta + 1$, then as there are only countably many first order formulas with parameters in a countable set, L_α is countable. As $L_\alpha \subseteq \mathcal{P}(L_\beta)$, L_α contains only hereditarily countable sets, so L_α is hereditarily countable. If α is a limit countable ordinal, and for every $\beta < \alpha$ the set L_β is countable then L_α is countable as a countable union of countable sets. Every element of L_α is in some L_β for $\beta < \alpha$ so is hereditarily countable. Therefore, L_α is hereditarily countable. ■

II.2.2 The definition of α-recursion

After all these preliminaries, the definition of α-recursion comes naturally.

Definition II.24 (α-recursion)

Let $A \subseteq \mathbb{N}$, and α be an ordinal.

— We say that A is α-recursively enumerable if there exists a Σ_1 formula ϕ, with parameters in L_α, such that $n \in A$ if and only if $L_\alpha \models \phi(n)$.

— We say that A is α-co-recursively enumerable if there exists a Π_1 formula ϕ, with parameters in L_α, such that $n \in A$ if and only if $L_\alpha \models \phi(n)$.

— A is α-recursively enumerable if A is both α-r.e. and α-co-r.e.

We have to emphasize the small differences we have with the usual definition of α-recursion. The first difference is that the field of "α-recursion" is usually restricted to special ordinals called "admissible". The study of Σ_1-definability over L_α for other ordinals is called β-recursion. In this thesis, we will write α-recursive according to Definition II.24 without making a distinction on whether the ordinal is admissible or not.

The second difference has a bit more importance. Usually, α-recursion is not restricted to subsets of $\mathbb{N}$, but to subsets of L_α. As we will mostly use α-recursion in the settings of algorithmic randomness, in Chapter V, we are interested in subsets of the Cantor space rather than subsets of 2^α. We still give the corresponding definitions.

Definition II.25

Let α be an ordinal.

1. Any $x \in L_\alpha$ is said to be α-*finite*.
2. A subset $A \subseteq L_\alpha$ is said to be α-r.e. (resp. α-co-r.e., resp. α-recursive) if it is Σ_1-definable (resp. Π_1-definable, resp. Δ_1-definable).
3. A function $f : L_\alpha \to L_\alpha$ is said to be α-recursive if it has an α-r.e. graph.

In order for the previous definition to work as expected, we need several other things. First, similarly to higher computability we need the ordinal bound α never to be reached in an α-finite number of α-finite steps, in order to safely conduct computations. This issue was mainly addressed by Spector's Σ^1_1 boundedness theorem. However, there is a new issue one need to take care in our settings.

We called "time tricks" the arguments in proofs using the homogeneity between "time" and "space". Following this idea, we could call "code tricks" the arguments using the homogeneity between the codes of machines and space. For instance, when defining a simple set S, we enumerate $n \in W_e$ only if $n \geq 2e$, ensuring co-infinity of S. In our definition of α-r.e. sets, the Σ_1 formula have a parameter in L_α. So if there is no way to enumerate all parameters along $\mathbb{N}$,

one cannot build a simple subset of $\mathbb{N}$ (this problem does not occurs in "full" α-recursion where subsets of α are considered). As the codes for machines and for the time are homogeneous, this issue is resolved the same way it is resolved for the "space-time" conflict: by the notion of projectum.

We start by giving some requirements on the ordinal α for α-recursion to behave well. Admissibility provides a safe requirement, however we also need to work with non admissible ordinals, therefore we show that for many recursions, α limit is already enough to conduct them safely.

After that we show that many useful predicates for Gödel's constructibles are α-recursive. We then define and discuss on other properties that allow α-recursion to behave similarly to regular computations for subsets of $\mathbb{N}$. We finally give an example of the construction of an α-r.e. set.

II.2.3 Σ_1-induction for limit and admissible ordinals

In order to safely conduct Σ_1 inductions, we normally need to be in a model of KP: a weakening of set theory in which we have extentionality, pairing, union, Cartesian product, induction over the $\in$ relation (suppose for all a we have $[\forall b \in a\ \Phi(b)] \to \Phi(a)$, then for all a we have $\Phi(a)$), Δ_0-comprehension and Σ_1-collection.

For any α limit, we have that L_α is a model of all these axioms, except Σ_1-collection.

Definition II.26

Let $\mathcal{A} = (A, \epsilon)$ be an $\mathscr{L}$-structure for the language of set theory. We say that $\mathcal{A}$ is a Σ_n-admissible structure if $\mathcal{A}$ is a model of extentionality, pairing, union, Cartesian product, induction over the $\in$ relation, Δ_n-comprehension and Σ_n-collection.

Definition II.27

We say that α is admissible if L_α is a model of KP, that is, if α is limit and L_α is a model of Σ_1-collection and Δ_1-comprehension.
More generally we say that α is Σ_n-admissible if L_α is a Σ_n admissible structure.

Usually, admissibility is stated in the following way: the image of an α-finite set by an α-recursive function is α-finite. This makes sense when we consider subsets of α as our main interest. In our case, we want to emphasize that ordinals are used as time of computation, while we only consider subsets of integers. Therefore, admissibility should be viewed as Spector's Σ^1_1 boundedness theorem: a bound on the computation time of an α-recursive set of α-recursive stages of computation.

In the next section about Infinite Time Turing Machines, we will have to work with ordinals which are not necessarily admissible. We will see for instance that Σ, the smallest non-accidentally writable ordinal of Definition II.50, is not admissible.

Fortunately, we can already define a lot of things in models L_α for α simply limit (and not necessarily admissible). Working with the constructibles involves working constantly with Σ_1-inductive definitions. Whereas these are perfectly safe in L_α for α admissible, some additional care needs to be taken when α is not admissible. Let us determine what we need:

Let $E \in L_\alpha$ and $< \in L_\alpha$ be a well-founded order on elements of E. We define by induction the $<$-rank of elements $a \in E$, denoted by $\mathrm{rk}_<(a)$, to be the smallest ordinal β such that for every $b < a$, b has $<$-rank less than β. Let E_β be the elements of E of $<$-rank strictly smaller than or equal to β, let $E_{<\beta}$ be the elements of E of $<$-rank smaller than β and $E_{=\beta}$ the elements of E of $<$-rank exactly β. Let γ be the supremum of the $<$-rank of elements of E and suppose $\gamma \leq \alpha$.

Suppose we have a Δ_0 formula $F(a, f, r)$ such that for any $a \in E$, with $\mathrm{rk}_<(a) = \beta$ whenever $f \in L_\alpha$ is defined on $E_{<\beta}$, then there is a unique $r \in L_\alpha$ such that $L_\alpha \models F(a, f, r)$. The classical theorem of set theory, that justifies definition by induction, says that we then have a unique function f defined on E and such that the Δ_1 formula $\Phi(\gamma, f)$ is true, where:

$$\Phi(\gamma, f) \equiv \text{For every } \beta < \gamma, \text{ for every } a \in E_{=\beta}, \text{ we have } F(a, f\restriction_{E_{<\beta}}, f(a))$$

Indeed the function f, if it exists, must be unique and Δ_1-recognizable by the formula $\Phi(\gamma, f)$ (using parameter γ). Also by induction one shows that whenever $f\restriction_{E_{<\beta}}$ exists, then $f\restriction_{E_{<\beta+1}}$ must exists as it is Δ_1-definable by F with $f\restriction_{E_{<\beta}}$ as a parameter (see Proposition II.28 below). This uses the axiom of Σ_1 collection: if for all $a \in E_{=\beta}$ there exists a unique $r \in L_\alpha$ such that $F(a, f\restriction_{E_{<\beta}}, r)$, then the corresponding function f' defined on $E_{=\beta}$ must exists. However if the ranks of the r's are unbounded in L_α, the function f' will not exist in L_α. Fortunately most of the time, for simple tasks, the rank will be bounded in L_α by something independent of $a \in E_{=\beta}$, but dependent only on β.

The axiom of Σ_1-collection also needs to be used at a limit step: If for any $\gamma < \beta$, there exists a unique function f_γ defined on E_γ and such that $\Phi(\gamma, f_\gamma)$, then by Σ_1-collection there exists a unique function f_β such that $\Phi(\beta, f_\beta)$ (and the function f_β is simply the union of the functions f_γ). Here again, this argument works within L_α as long as the rank of each function f_γ is bounded in L_α. We sum up in the following proposition conditions in which definitions by induction can be conducted in L_α for α limit:

▶ **Proposition II.28** (Δ_0 Induction with bounded rank replacement)**.** *Let E be a class well-ordered by $<$. Let $f : E \mapsto L$ be Δ_0-definable by induction on $<$, such that for any β there exists $k < \omega$ for which:*

1. *E_β is $\Delta_1^{L_{\beta+k}}$-definable uniformly in β, in particular $E_{<\alpha} \subseteq L_\alpha$ for α limit.*
2. *For any $a \in E_\beta$, $\mathrm{rk}_<(a)$ is $\Delta_1^{L_{\beta+k}}$-definable uniformly in β.*
3. *For any $a \in E_\beta$ we have $\mathrm{rk}(f(a)) < \beta + k$.*

Then f is $\Delta_1^{L_\alpha}$-definable uniformly in any limit ordinal α. By this we mean that there are single Π_1 and Δ_1 formulas that define $f\restriction_{E_{<\alpha}}$ when interpreted in L_α.

Proof. Let $\Phi(\beta, f)$ be the Δ_1 formula defined in the discussion above. We shall show that for any α limit we have:

(a) For any $\beta < \alpha$, the function $f\restriction_{E_\beta}$ belongs, as a set, to $L_{\beta+m}$ for some $m < \omega$.

(b) The function $f\restriction_{E_{<\alpha}}$ is $\Delta_1^{L_\alpha}$-definable by the formulas:

$$\begin{aligned} f\restriction_{E_{<\alpha}}(a) = r \quad &\equiv \quad \exists f\ \Phi(\mathrm{rk}_<(a), f) \wedge f(a) = r \\ &\equiv \quad \forall f\ \Phi(\mathrm{rk}_<(a), f) \rightarrow f(a) = r \end{aligned}$$

It is clear that for any α limit we have (a) implies (b). Suppose now $\alpha = 0$ or α limit and (b) is true for α, and let us show that (a) is true for $\alpha + \omega$. If $\alpha = 0$ we clearly have $f\restriction_{E_{<\alpha}} \in L_{\alpha+1}$. If α is limit and (b) is true for α, thus also it is clear by definition of L that $f\restriction_{E_{<\alpha}} \in L_{\alpha+1}$. Now from $f\restriction_{E_{<\alpha}} \in L_{\alpha+1}$ together with (1) (2) and (3), by iterating inductively the same argument for $n \in \omega$, we easily obtain that $f\restriction_{E_{\alpha+n}}$ is $\Delta_1^{L_{\alpha+(n+1)k}}$-definable and thus belongs to $L_{\alpha+(n+1)k+1} \subseteq L_{\alpha+\omega}$. Thus (b) is true for $\alpha + \omega$.

Suppose now that α is limit of limit and that for any $\beta < \alpha$ limit we have that (a) is true. Thus clearly (a) is true for α, and therefore also (b). This concludes the proposition. ■

We end by one last thing one needs to be careful about when working in L_α for α not admissible. In case α is admissible, formulas of the form $\forall n \in \omega\ \exists \beta\ \Phi(n, \beta)$ where $\Phi(n, \beta)$ is Δ_0, can be considered to be Σ_1-formulas, precisely because if the formula is true in L_α, there must exists $B \in L_\alpha$ such that $\forall n \in \omega\ \exists \beta \in B\ \Phi(n, \beta)$. This is of course not the case for α not admissible, and one has to be careful about keeping Σ_n formulas truly Σ_n.

II.2.4 Some α-recursive predicates

Using induction with bounded rank replacement, it is possible to show that the function $\beta \mapsto L_\beta$ is absolute already in L_α for α limit. This is done formally in [28].

In order to show that the function $\beta \mapsto L_\beta$ is absolute in any model L_α for α limit, the author of [28] uses a bounded rank argument as sketched above. In this case, this requires to be a bit careful with the encoding one uses for ZF formulas by sets (hereditary finite sets in case the formula has no parameter). In particular, it is worth noting that one uses partial function from n to $\{p_1, \ldots, p_n\}$ to encode finite sequences, rather than successive pairing. This way, as long as $P \in L_\alpha$, for any n, a function from n into P has its rank bounded by some $\alpha + k$, where k is an integer independent of n (even in L_ω: recall that we start with $L_\emptyset = \omega$).

Using such an encoding of formulas, we write $\ulcorner \Phi \urcorner$ for the code of Φ. We have the following:

Theorem II.29 (Lemma I.9.10 of [28])

The predicate $M \models \Phi(p_1, \ldots, p_n)$ is $\Delta_1^{L_\alpha}$ uniformly in any α limit, in M, in $\ulcorner \Phi \urcorner$ and in the sequence $\langle p_1, \ldots, p_n \rangle$.

By the above, we formally mean the following: there is a Σ_1 formula $\Phi(M, e, p)$, and a Π_1 formula $\Psi(M, e, p)$, such that for any α limit, as long as we take $M, \langle p_1, \ldots, p_n \rangle$ in L_α, we have:

$$\begin{array}{rl} & M \models \phi(p_1, \ldots, p_n) \\ \Leftrightarrow & L_\alpha \models \Phi(M, \ulcorner \phi \urcorner, \langle p_1, \ldots, p_n \rangle) \\ \Leftrightarrow & L_\alpha \models \Psi(M, \ulcorner \phi \urcorner, \langle p_1, \ldots, p_n \rangle) \end{array}$$

We will also sometimes use the following version of the above: in case Φ is a Δ_0 formula, then Φ is true in L_α iff Φ is true in the model being the transitive closure of all the parameters involved in the formula. Using that such a model can be obtained uniformly and that satisfaction is absolute in any L_α for α limit, we also have:

▶ **Corollary II.29.1.** *The predicate $L_\alpha \models \Phi(p_1, \ldots, p_n)$ is $\Delta_1^{L_\alpha}$ uniformly in any α limit and in the code of any Δ_0 formula $\ulcorner \Phi \urcorner$.*

Using that satisfaction is absolute in any L_α for α limit, we also have:

Theorem II.30 (Lemma II.2.8 of [28])

The function $\beta \mapsto L_\beta$ is $\Delta_1^{L_\alpha}$ uniformly in any α limit.

It is also well-known that L is well-ordered in L, that is, there is a well order $<_L$ on elements of L, which is definable in L. Again, one can show that this order is absolute in any L_α for α limit.

Theorem II.31 (Lemma II.3.5 of [28])

The relation $<_L$ and the function $a \mapsto \{b : b <_L a\}$, are $\Delta_1^{L_\alpha}$, uniformly in any α limit.

We end this subsection by showing that in the special case of Σ_n-admissibility in the constructible hierarchy, only the axiom of Σ_n-collection is needed when α is limit.

▶ **Proposition II.32.** *Suppose L_α is a model of Σ_n-collection for α limit. Then, L_α is a model of Δ_n-comprehension.*

Proof. The proof goes by induction on n. For $n = 0$ as α is limit we always have that L_α is a model of Δ_0-comprehension. Suppose the result is true for n and let us show it is true for $n+1$. Let L_α be model of Σ_{n+1}-collection. Let $\Phi(a,b)$ and $\Psi(a,b)$ be Π_n^0 formulas with parameters in L_α. Let $A \in L_\alpha$ and $E \subseteq A$ be such that:

$$\begin{array}{rcl} a \in E & \Leftrightarrow & L_\alpha \models \exists b\ \Phi(a,b) \\ a \notin E & \Leftrightarrow & L_\alpha \models \exists b\ \Psi(a,b) \end{array}$$

We have in particular that $L_\alpha \models \forall a \in A\ \exists \beta\ \exists b \in L_\beta\ \Phi(a,b) \vee \Psi(a,b)$

By Σ_{n+1}-collection there exists $\beta < \alpha$ such that we have:

$$L_\alpha \models \forall a \in A\ \exists b \in L_\beta\ \Phi(a,b) \vee \Psi(a,b)$$

Note that we then have

$$\begin{array}{rcl} a \in E & \Leftrightarrow & L_\alpha \models \exists b \in L_\beta\ \Phi(a,b) \\ a \notin E & \Leftrightarrow & L_\alpha \models \exists b \in L_\beta\ \Psi(a,b) \end{array}$$

It follows that we have :

$$\begin{array}{rcl} a \notin E & \Leftrightarrow & L_\alpha \models \forall b \in L_\beta\ \neg\Phi(a,b) \\ a \in E & \Leftrightarrow & L_\alpha \models \forall b \in L_\beta\ \neg\Psi(a,b) \end{array}$$

As L_α is a model of Σ_n-collection, formulas $\forall b \in L_\beta\ \neg\Phi(a,b)$ and $\forall b \in L_\beta\ \neg\Psi(a,b)$ are both equivalent in L_α to Σ_n formulas. Therefore E is in fact defined by a Δ_n formula. By induction hypothesis we have that $E \in L_\alpha$. ■

II.2.5 Stability

When lifting up notions of computability to various ordinals, new phenomenons start to appear, one of them central to the study of α-recursion is the notion of stability.

Definition II.33

For $\alpha \leq \beta$ we say that L_α is Σ_n-stable in L_β, and we write $L_\alpha \prec_n L_\beta$ if for every Σ_n formula Φ with parameters in L_α we have $L_\alpha \models \Phi$ iff $L_\beta \models \Phi$. Without confusion, we will also write $\alpha \prec_n \beta$ for $L_\alpha \prec_n L_\beta$.

The notion of n-stability is the same as the notion of elementary substructure for Σ_n formulas in model theory. The following proposition is easy and will be used in various places of the paper:

▶ **Proposition II.34.** *Suppose $L_\alpha \prec_n L_\beta$. Let $\Phi(a_1, \ldots, a_n)$ be a Π_{n+1} formula and let $p_1, \ldots, p_n \in L_\alpha$. If $L_\beta \models \Phi(p_1, \ldots, p_n)$ then $L_\alpha \models \Phi(p_1, \ldots, p_n)$.*

Proof. The formula $\Phi(a_1, \dots, a_n)$ is of the form $\forall x\ \Psi(x, a_1, \dots, a_n)$ for Ψ a Σ_n formula. Also for every $x \in L_\alpha$ we have $L_\beta \models \Psi(x, a_1, \dots, a_n)$ and thus $L_\alpha \models \Psi(x, a_1, \dots, a_n)$ by Σ_n stability. It follows that $L_\alpha \models \forall x\ \Psi(x, a_1, \dots, a_n)$. ■

▶ **Corollary II.34.1.** *$L_\alpha \prec_n L_\beta$ if and only if for every Σ_n formula ϕ with parameters in L_α, we have $L_\beta \models \phi$ implies $L_\alpha \models \phi$.*

Proof. The implication is direct. We prove the reciprocal by induction. The reciprocal is true for $n = 0$. Suppose it is true for n, and for every Σ_{n+1} formula ϕ with parameters in L_α, we have $L_\beta \models \phi$ implies $L_\alpha \models \phi$. In particular, as every Σ_n formula can be seen as a Σ_{n+1} formula, we have $L_\alpha \prec_n L_\beta$. But by Proposition II.34, we get the only missing part for $n+1$-stability: if ψ is Π_{n+1}, then $L_\beta \models \psi$ implies $L_\alpha \models \psi$. ■

▶ **Proposition II.35.** *For β limit and $\alpha < \beta$, the predicate $L_\alpha \prec_n L_\beta$ is $\Pi_n^{L_\beta}$ uniformly in β and α.*

Proof. We start with Σ_1-stability. We have $L_\alpha \prec_1 L_\beta$ iff

$$\begin{array}{rcl} L_\beta & \models & \text{For all } \Delta_0 \text{ formulas } \ulcorner \Phi(b, a_1, \dots, a_k) \urcorner\ \forall p_1, \dots, p_k \in L_\alpha \\ & & \qquad \big[\forall x\ \neg\Phi(x, p_1, \dots, p_k) \text{ or } L_\alpha \models \exists y\ \Phi(y, p_1, \dots, p_k)\big] \end{array}$$

which is $\Pi_1^{L_\beta}$ by Proposition II.29 and II.29.1. Suppose now that the predicate $L_\alpha \prec_n L_\beta$ is $\Pi_n^{L_\beta}$. To show that $L_\alpha \prec_{n+1} L_\beta$ is $\Pi_{n+1}^{L_\beta}$, we write first the formula which says $L_\alpha \prec_n L_\beta$, in order to express that if L_α satisfies a Σ_{n+1} formula, then also L_β satisfies this formula (see Proposition II.34). This formula is $\Pi_n^{L_\beta}$. We then combine it with the following $\Pi_{n+1}^{L_\beta}$ formula, which expresses that if L_β satisfies a Σ_{n+1} formula, then also L_α satisfies this formula:

$$\begin{array}{rcl} L_\beta & \models & \text{For all } \Delta_0 \text{ formulas } \ulcorner \Phi(b_1, \dots, b_{n+1}, a_1, \dots, a_k) \urcorner,\ \forall p_1, \dots, p_k \in L_\alpha, \\ & & \left\{ \begin{array}{l} \forall x_1\ \exists x_2 \cdots Q x_{n+1},\ \neg\Phi(x_1, \dots, x_{n+1}, p_1, \dots, p_k) \\ \text{ or } L_\alpha \models \exists y_1\ \forall y_2 \cdots Q y_{n+1}\ \Phi(y_1, \dots, y_{n+1}, p_1, \dots, p_k) \end{array} \right. \end{array}$$

where $Q \in \{\exists; \forall\}$ depends on the parity of n. This concludes the proof. ■

When dealing with the constructibles, stability presents additional features to the notion of elementary substructures in model theory. For instance, given that α is limit, the set of elements which are Σ_1-definable in L_α with no parameters is necessarily of the form L_β, and β is the smallest such that $L_\beta \prec_1 L_\alpha$ [5, Theorem 7.8]. We also have the following:

Theorem II.36

Suppose $\alpha < \beta$ for β limit, and $L_\alpha \prec_n L_\beta$. Then α is Σ_n-admissible.

Proof. The proof is easy for Σ_1-admissibility, but does not lift straightforwardly to Σ_n-admissibility.

We first show the theorem for Σ_1-admissibility. Suppose α is not Σ_1-admissible. Then there exists $a \in L_\alpha$ and a Σ_1 formula $\Phi(x, y) = \exists z\ \Phi_0(x, y, z)$ with parameters in L_α witnessing the failure of Σ_1-admissibility, that is:

$$\begin{array}{rcccccc} & L_\alpha & \models & & \forall p \in a & \exists r & \exists z & \Phi_0(p, r, z) \\ \text{and} & L_\alpha & \not\models & \exists \gamma & \forall p \in a & \exists r \in L_\gamma & \exists z \in L_\gamma & \Phi_0(p, r, z) \end{array}$$

As $\alpha < \beta$ it is however clear that we have:

$$L_\beta \models \exists \alpha\ \forall p \in a\ \exists r \in L_\alpha\ \exists z \in L_\alpha\ \Phi_0(p, r, z)$$

In particular the above Σ_1 formula is satisfied in L_β but not in L_α, so we do not have $L_\alpha \prec_1 L_\beta$.

We continue by induction: suppose α is not Σ_{n+1}-admissible. Then if L_α is not Σ_n-stable in L_β, it is in particular not Σ_{n+1}-stable in L_β and the proposition is verified. Otherwise we have $L_\alpha \prec_n L_\beta$. Let $a \in L_\alpha$ and let $\Phi_0(x, y, z_1, \dots, z_{n+1})$ be a Δ_0 formula (where $Q \in \{\exists; \forall\}$ depends on the parity of n) such that:

$$\begin{array}{rlllll} & L_\alpha & \models & & \forall p \in a & \exists r & \exists z_1 \forall z_2 \cdots Q z_{n+1} & \Phi_0(p, r, z_1, \dots, z_{n+1}) \\ \text{and} & L_\alpha & \not\models & \exists \gamma & \forall p \in a & \exists r \in L_\gamma & \exists z_1 \forall z_2 \cdots Q z_{n+1} & \Phi_0(p, r, z_1, \dots, z_{n+1}) \end{array}$$

Note that unlike with the Σ_1-case, we cannot necessarily bound the variables $z_1, \dots, z_{n+1}$ by L_γ. Indeed, it might be the case for every p in a there exists some r in L_γ which is Σ_{n+1}-definable in L_γ, even though it is not Σ_{n+1}-definable in L_α. We need to use that $L_\alpha \prec_n L_\beta$. In particular we have:

$$L_\beta \models \exists \alpha \prec_n \beta\ \forall p \in a\ \exists r \in L_\alpha \ \text{ s.t. } L_\alpha \models \exists z_1\ \forall z_2 \dots\ z_{n+1}\ \Phi_0(p, r, z_1, \dots, z_{n+1})$$

First let us note that by Proposition II.35 the above formula is Σ_{n+1}. It is also clear that L_α cannot be a model of this formula, because then, using Proposition II.34, it would also be a model of:

$$\exists \gamma\ \forall p \in a\ \exists r \in L_\gamma\ \exists z_1\ \forall z_2 \dots z_{n+1}\ \Phi_0(p, r, z_1, \dots, z_{n+1})$$

■

II.2.6 Projectibility

Another central notion in α-recursion theory is the notion of projectible ordinal. We are in particular able to lift most of the work done in algorithmic randomness and genericity, in the case α is projectible into ω. The reason for this is that it is a way to recover a weak homogeneity of space, time and machines code. Indeed, it allows for instance to have an enumeration of machines along ω, or to allows some "numeric" errors such that the sum of all errors is finite (such as in Proposition II.10). However, increasing the time stage might not increase its projectum, disallowing the use of some time trick.

Definition II.37 (Projectum)

We say that α is *projectible* in $\beta \leq \alpha$ if there is a one-one function Σ_1-definable (with parameters) in L_α, from α into β. We call *projectum* and write α^* for the smallest ordinal such that α is projectible into α^*. If $\alpha^* < \alpha$ we say that α is projectible. Otherwise we say that α is not projectible.

This notion of projectibility is very useful to lift proofs from lower to higher recursion. This has been done in particular in the hyperarithmetic setting, for instance in [7], using the fact that ω_1^{CK} is projectible into ω. We give a general theorem on projectums. This theorem can be found in a similar form in [5].

Theorem II.38

Let α be admissible. We have that α^* is the smallest ordinal such that L_α is not a model of Σ_1-comprehension for subsets of α^*. If the Σ_1 formula Φ is a witness of this failure, then the projectum is definable with the same parameters as the ones used in Φ.

Proof. We first show that L_α satisfies Σ_1-comprehension for subsets of ordinals smaller than α^*. Let $\delta < \alpha^*$ be an ordinal, and $A \subseteq \delta$ be such that $x \in A \Leftrightarrow L_\alpha \models \exists y\ \Phi(x, y)$ where Φ is Δ_0. Let f be the function defined on A, such that $f(a) = \delta \times \gamma + a$ where γ is the smallest ordinal such that $L_\gamma \models \exists y\ \Phi(a, y)$. Obviously f is 1-1. We then collapse $f[A]$ by defining $g(\gamma)$ to be the first $\beta \in f[A]$ that we find which is not in $\{g(\gamma') : \gamma' < \gamma\}$. Formally, let $\exists y\ \Psi(a, \beta, y)$ with Ψ Δ_0 be the Σ_1 formula defining f. Then we define the function g by $g(\gamma) = \beta$ if there exists η for which $\langle \beta, \eta \rangle$ is the smallest pair such that $L_\eta \models \exists y\ \exists a\ \Psi(a, \beta, y)$ and $\beta \notin \{g(\gamma') : \gamma' < \gamma\}$. We have that $f^{-1} \circ g$ is a Σ_1-definable bijection from an initial segment of α, onto A. Also the domain of $f^{-1} \circ g$ cannot be α otherwise α would be projectible into $\delta < \alpha^*$. Therefore the domain of $f^{-1} \circ g$ is a strict initial segment of α and thus the range of $f^{-1} \circ g$, which is A, is an element of L_α.

We now exhibit a Σ_1-definable subset of α^* which is not in L_α. If p is a projection into α^*, we have that $p[\alpha] = A \subseteq \alpha^*$ is a subset of α^* which is Σ_1 definable in L_α. This subset is not in L_α, as otherwise the function $g : \alpha^* \to \alpha$ defined by $g(\beta) = \sup_{x \in A \wedge x \leq \beta}(p^{-1}(x))$ would contradict the admissibility of α. ■

II.2.7 An α-simple subset of $\mathbb{N}$

We first mention that when α is projectible and $L_\alpha \models$"everything is countable", then there exists a uniform enumeration of Σ_1 formula with parameters, along $\mathbb{N}$.

Theorem II.39

If α is admissible and projectible in ω, and $L_\alpha \models$"everything is countable", then there exists an enumeration $(P_n)_{n \in \omega}$ of Σ_1 formulas with exactly one parameter, and an α-r.e. function U such that for every $n, m \in \omega$, we have $L_\alpha \models U(n, m)$ iff $L_\alpha \models P_n(m)$.

Proof. We first show that there is a partial Σ_1-definable surjection from ω onto L_α. As α is not projectible, there is a Σ_1-definable function $p : \alpha \to \omega$. Note that as $L_\alpha \models$ "everything is countable", for every $\beta < \alpha$ there exists in L_α a bijection between L_β and ω, and the $<_L$-smallest one can be found uniformly. Let p be the projectum of α into ω, if $p(\beta) = n$, then we call f_n the $<_L$-smallest bijection between L_β and ω. The surjection we are looking for is then defined to be $f(\langle n, m \rangle) = f_{p^{-1}(n)}(m)$.

Let $(\theta_m)_{m \in \mathbb{N}}$ be a recursive enumeration of the Σ_1 formulas without parameters, but with two free variables. We now define $U(\langle c, e \rangle, i)$ to be true if $\exists K, K = f(c) \wedge \theta_e(i, K)$.

Let P_e be any Σ^1_1 formula with one parameter $K \in L_\alpha$. Fix e' such that $\theta_{e'}$ is P_e where the parameter K is now a free variable. As f is a surjection from $\mathbb{N}$ to L_α, let c be such that $f(c) = K$. Now, $U(\langle c, e' \rangle)$ is true if and only if $\exists K, K = f(c) \wedge \theta_{e'}(i, K)$ if and only if $\theta_{e'}(i, K)$ if and only if $P_e(i)$. ■

▶ **Corollary II.39.1.** *If α is admissible and projectible in ω, and $L_\alpha \models$"everything is countable", then there exists an α-r.e. set $S \subseteq \mathbb{N}$ which is co-infinite and such that $S \cap W \neq \emptyset$ for every*

infinite α-r.e. W.

Proof. The proof is the same as in the higher and computations, using Theorem II.39. Write $P_e[\beta]$ for the set $\{n : L_\beta \models U(e,n)\}$ (this has to be compared to $W_e[t]$). Enumerate n into S if and only if there is some $e \in \mathbb{N}$ and $\beta < \alpha$ with $n \geq 2e$ and $S \cap P_e[\beta] = \emptyset$, but $n \in P_e[\beta+1]$, and n is the smallest such integer. ■

Note that if α is not projectible in ω, there cannot exists simple subsets of $\mathbb{N}$ as every α-r.r. subset of $\mathbb{N}$ is also in L_α by Theorem II.38 and is therefore α-computable.

II.3 Infinite Time Turing Machines

The idea behind Infinite Time Turing Machine is different from the previous notions of infinite computation. This time, we will define an actual model of computation, a machine, just as we did with Turing machine. It is now easier to be confident that the resulting notion do represent infinite computations.

However, the idea is the same: we index computations by ordinals, instead of integers. In the model, there will be no predetermined bound to the computations, even though bounds appear naturally. Several ordinals will be associated to bound particular features of computations, and we will see that these ordinals have purely set-theoretic definitions, in a surprising but satisfying way.

It turns out that the Infinite Time Turing machines actually give a model to α-recursion for a particular α. But more than that, it gives a much better way to relativize computations to real numbers. In α-recursion, we saw that an admissible α is good for bounding computations to length less than α while still keeping a good behaviour. Yet, when relativized to some set X the computations may need to go further α to be well-behaved: $L_\alpha(X)$ might not be admissible. Infinite Time Turing Machines overcome this difficulty as there is no predetermined bound on the computations.

II.3.1 The Infinite-Time Turing Machines model

We start by introducing the model. Just as we did for regular Turing machines, we need to define two things: the machine, and the computation.

Definition II.40 (Infinite Time Turing Machine)

An *Infinite Time Turing Machine*, often shortened to ITTM, is a Turing machine with 3 tapes called the *input tape*, the *working tape*, and the *output tape*; and a distinguished state called $q_{\lim}$.

This is all for the formal definition of an Infinite Time Turing Machine. The denotation of the tapes already suggests how they will be used in computations. As a special kind of Turing Machine, there exists countably many ITTMs, and they can be given a Gödel code. Let us now explain how they are executed.

Definition II.41 (Infinite Time computations)

If M is an ITTM, we say that c is a *configuration of M* if $c = (q, T, h)$ where q is a state, T is the value of the tapes (merged in one function from $\mathbb{N}$ to 2) and $h \in \mathbb{N}$ is the position

of the head.
A *computation of length* α is a sequence $C = (c_\beta)_{\beta<\alpha}$ where $c_\beta = (q_\beta, T_\beta, h_\beta)$ are configurations of M, and we have the following requirements on the course of computation:

1. q_0 is the initial state of the Turing machine, T_0 three empty tapes and $h_0 = 0$.
2. $q_{\beta+1}$ is given from q_β using the usual rules of regular Turing machines.
3. If λ is a limit ordinal, then $q_\lambda = q_{\lim}$. Moreover, $h_\lambda = 0$ and $T_\lambda(n) = \limsup_{\beta<\lambda} T_\beta(n)$.

If M is an ITTM and α an ordinal such that there exists an α-long computation for M, we write $M[\alpha]$ to denote the value of the output tape at step α. If additionally at the beginning the input tape is filled with x, we write $M(x)[\alpha]$. We write $C_M[\alpha]$ for the whole configuration of M at step α.

In an infinite computation, we want of course the successor steps to be the same as in the case of finite computation, independently of the fact that there may be infinitely many past computations. So the only thing that we had to specify in Definition II.41 is the behaviour at limit steps: the state, the head and the tapes.

At a limit step, the state is always put to the fixed state $q_{\lim}$. Therefore, in a computation we are able to know when we are at limit steps. In modern programming terms (as opposed to Turing machine programming), one could say that there is a special line of code, or function, such that the execution flow is directed to this line at each limit step. Similarly, the reading/writing head of the machine is redirected to the initial position.

However, we want to keep a memory of the past computations, so we cannot erase all the tapes to 0s. If at some point, the value of a cell was stable, then the value at the limit is clear: it should be the stable value. However, sometimes the value in a specific cell alternates when approaching the limit step. In this case, we could choose several behaviours, for instance that the computation diverges. Instead, we define the limit to be 1, making $T_\lambda(n) = \limsup_{\beta<\lambda} T_\beta(n)$. Of course, using $\liminf$ would be completely symmetric. We will see that this is an important design choice of ITTMs.

Definition II.42 (ITTM-computability)

We say that a real $x \in 2^\omega$ is *ITTM-computable*, also called *writable*, if there exists an ITTM whose execution with empty input tape reaches the halting state with x being written on the output tape.
If $y \in 2^\omega$ is any real, we say that $x \in 2^\omega$ is *ITTM-computable in* y, also called *writable in* y, if there exists an ITTM whose execution with y written on the input tape reaches the halting state with x being written on the output tape. In this case, we write $M(y) = x$. We just write $M(y)\downarrow$ to mean it halts.
We say that a class $\mathcal{C} \subseteq 2^\omega$ is *ITTM-decidable* if there exists an ITTM M such that if $x \in 2^\omega$, then $M(x) = \mathcal{C}(x)$ (the ITTM always halts, with 0 on the output tape if $x \notin \mathcal{C}$ and 1 otherwise).
We say that a set $A \subseteq \mathbb{N}$ (resp. $\mathcal{C} \subseteq 2^\omega$) is *ITTM-semidecidable* if there exists an ITTM M such that if $n \in \mathbb{N}$ (resp. $x \in 2^\omega$), then $M(n)\downarrow$ if and only if $n \in A$ (resp. if $x \in 2^\omega$, then $M(x)\downarrow$ if and only if $x \in \mathcal{C}$).

We note that the ITTM model is quite a robust one. Adding or removing tapes (as long as we keep the input, output and working spaces in some way), changing the alphabet, does not

modify the ITTM-computable reals.

There are two fundamental differences between α-recursion and ITTMs. The first one is that there is no fixed bound on the length of computation in ITTMs, while α-recursions are bounded by α. The second is that while α-recursions have access to the entire history of the computation, Infinite Time Turing Machines can only try to code their history on the ω-length tapes. Therefore, it might start to be "lost in time", after a too long computation. As we will see later, there will be a special ordinal after which those machines get confused, and start to loop.

II.3.2 Computational power of ITTMs

We now give a few examples of the ITTM-computable sets. It turns out that the computability power of these machines is very high.

Theorem II.43

The following sets are ITTM-computable:

1. the jump $\emptyset'$,
2. the n-th jump $\emptyset^{(n)}$,
3. the α-th jump $\emptyset^{(\alpha)}$ for any recursive α,
4. the sets WO and $\mathcal{O}$, and more than that the set $\mathbf{WO}$ is ITTM-decidable,
5. if X is any writable set, the set WO^X and $\mathcal{O}^X$. In particular, $\mathcal{O}^{\mathcal{O}}$ and $\mathcal{O}^{\mathcal{O}^{\mathcal{O}}}$ and $\mathcal{O}^{\mathcal{O}^{\mathcal{O}^{\cdot^{\cdot^{\cdot}}}}}$.

Proof. It is clear that item 5 implies all the other items, but we prove them one by one in order to get a better intuition on how to program ITTMs.

We show that $\emptyset'$ is ITTM-computable. Let M be the ITTM that does the following: when run, it simulates in parallel the execution of Turing machines φ_e with input e. Whenever one machine stops, it writes 1 on the output tape, at position e if e is its code. Then, it halts when it reaches a limit state. Clearly, at limit step all cells of the tape converged to the right value.

Computing $\emptyset^{(n)}$ is done by induction. Suppose we can compute $\emptyset^{(n)}$ for some n. Then, we define the machine that first write $\emptyset^{(n)}$ on some working tape, but instead of halting it simulates in parallel the execution of Turing machines φ_e with input e and oracle $\emptyset^{(n)}$. Whenever one machine stops, it writes 1 on the output tape, at position e if e is its code. It halts the next time it reaches the limit state. Clearly, at limit step all cells of the tape converged to the right value.

We won't prove item 3, but it is implied by item 4. We show that $\mathbf{WO}$ is ITTM-decidable. Let x be a real written on the input tape. The fact that x is a code for a linear order is an arithmetical statement, as we proved that we can compute arbitrary jumps, it is ITTM-decidable. So the machine halts with the "no" answer if x is not a code for a linear order. Otherwise, it has to continue to check the well-foundedness of the order coded by x.

The machine works as follow: it find a smallest element of x, removes it, and iterates. If there is no smallest element at some point, then it rejects x, otherwise it reaches a stage where the linear order has empty domain, in this case it accepts x. One of the two events must occur, as any linear order on A is such that $A = B \cup C$ where B is well-ordered and C has no smallest element.

It remains to explain how to find a smallest element. The ITTM will consider all numbers one by one, and stored in some (infinite) part of the tape the current best candidate for being the smallest one. If there is no smallest elements, we will find a new best candidate infinitely

often, and if there is one, we will change our mind only finitely often. In order to know in which case we are, every time we change the best candidate, we also "flash" a particular cell that we call "flag": we make it equal to 1, and then 0 again.

If at the limit step the value of the "flag" cell is 1, then infinitely often we flashed it, and there is no minimum, so x is ill-founded. Otherwise, the reserved tape for the best candidate stabilized at some point, and the minimum is written on it.

Item 5 is clear by relativizing the item 4. ■

The most important thing here is the ability to check that a real is a code for a well-order. This fact used the "lim sup" rule in an unavoidable manner, so the design choice was important here for this model.

Computational weaknesses of ITTMs

For the moment, we only gave examples of ITTM-computable sets, and seen that they go quite high compared to our previously defined hierarchy. We give here a few examples of bounds on ITTM computations.

Definition II.44

We define γ to be the supremum of the halting times of ITTMs.

First, as there are countably many ITTMs, it is clear that γ must be defined, and has cofinality ω. Therefore it cannot be ω_1. We prove that γ is countable.

▶ **Proposition II.45.** *We have $\gamma < \omega_1$.*

Proof. We will prove that if the execution of an ITTM reaches stage ω_1, then it is in a looping process. So suppose M is an ITTM with a computation $C = (c_\alpha)_{\alpha<\omega_1+1}$ reaching stage ω_1. Then, at ω_1 the state is q_{lim} and the head is at position 0. Let us define an ordinal α with the exact same configuration as ω_1, that is $C_\alpha = C_{\omega_1}$.

We say that n is alternating if the sequence $(T_\alpha(n))_{\alpha<\omega_1}$ has no limit, otherwise it is stable. Let $\alpha_{-1} = 0$ and α_n be the smallest ordinal higher than α_{n-1} such that

- If n is stable, the sequence $(T_\alpha(n))_{\alpha<\omega_1}$ has converged at step α_n.
- For every alternating $m < n$, the sequence $(T_\alpha(m))$ has alternated between α_{n-1} and α_n.

Then, let $\alpha = \sup_n \alpha_n$. As each α_n is countable and the union is countable, then $\alpha < \omega_1$. We also argue that $T_\alpha = T_{\omega_1}$: indeed, if n is stable, as $\alpha > \alpha_n$, $T_\alpha(n)$ is already the stable value. If n is alternating, $T_\alpha(n) = \limsup_{\beta<\alpha} T_\beta(n) = 1$ as there is no stable value by definition of the T_{α_n}. So $T_\alpha = T_{\omega_1}$.

As α is limit, its associated state is q_{lim} and the head is at position 0. Then $C_\alpha = C_{\omega_1}$. But contrary to ordinary Turing machine, this solely does not imply that the machine entered an unbounded looping cycle. Indeed, a looping cycle can be stopped after ω many of its iterations, if because of the lim sup rule, one cell stops to be 0. However, in our case, we also have that for all n such that $T_{\omega_1}(n) = 0$, there is no β between α and ω_1 such that $T_\beta(n) = 1$. So the looping cycle cannot be broken, and M will never stop. ■

Theorem II.46

The following are not ITTM-computable:

1. The set $\{e \in \mathbb{N} : M_e(e) \downarrow\}$.

2. Any code for an ordinal $\alpha \geq \gamma$.

Proof. The first item has the exact same proof than the fact that the jump is not computable. We still have the recursion theorem for ITTM, and using this we could devise an ITTM that halts if and only if it does not halt.

Let x be any code for $\alpha \geq \gamma$. In the proof of the ITTM-decidability of **WO**, given x a code for some β, we used at least β steps to answer yes, as we remove one by one the elements of x in the order of β. But then, the machine writing x on a tape, and then checking its well-foundedness would take more than γ steps to halt. This is a contradiction with the definition of γ. ■

Theorem II.47

There exists an ITTM-simple set: an ITTM-semidecidable set $S \subseteq \mathbb{N}$ such that for every infinite ITTM-semidecidable set $W \subseteq \mathbb{N}$, we have $S \cap W \neq \emptyset$.

Proof. We use the fact that there exists a universal Infinite Time Turing Machine, simulating all the other ones. Let us describe a machine M: it simulates on the working tape all machines M_e on every input n. We also keep on the working tape a cell C_e initially 0, for each of machines M_e. We say that M_e is marked if $C_e = 1$. If at some point there is n and e such that $n \geq 2e$, $M_e(n) \downarrow$ and M_e is not marked, then we enumerate n and mark M_e.

The verification is as usual. ■

II.3.3 The three notions of writability

In order to tackle the exact complexity of ITTM-computable sets with regards to the hierarchy of Gödel's constructibles, we need a careful analysis of the halting time of the ITTMs. This goes through an attentive comparison between the halting ordinals of Infinite Time Turing Machines, and the ordinals that have writable codes, as well as two other notion of writability: eventual writability and accidental writability.

Definition II.48 (Hamkins, Lewis [41])

Let $y \in 2^\omega$.

- The real y is *writable* if there is an ITTM M such that $M \downarrow [\alpha] = y$.
- The real y is *eventually writable* if there is an ITTM M and an ordinal α such that $\forall \beta \geq \alpha$ we have $M[\beta] = y$.
- The real y is *accidentally writable* if there is an ITTM M and a stage α such that $M[\alpha] = y$.

For $x \in 2^\omega$ we define the notions of x-writable, x-eventually writable and x-accidentally writable similarly, but with the ITTMs starting with x on their input tape.

Hamkins and Lewis introduced these three analogues of being computable by an ITTM. They used these notions to study the ordinals that are computable by an ITTM, with respect to these definitions. In particular, these definition will be particularly important when applied to codes of ordinals.

Definition II.49 (Hamkins, Lewis [41])

An ordinal α is *writable*, resp. *eventually writable*, resp. *accidentally writable*, if it has a writable code, resp. an eventually writable code, resp. an accidentally writable code. For $x \in 2^\omega$ we define analogously the notion of x-writable, x-eventually writable and x-accidentally writable.

It is clear that the writable, eventually writable, and accidentally writable ordinals, are all initial segments of the ordinals. Hamkins and Lewis showed that the supremum of the writable ordinals was eventually writable and that the supremum of the eventually writable ordinals was accidentally writable.

Definition II.50 (Hamkins, Lewis [41])

We define the following ordinals:
- λ is the supremum of the writable ordinals.
- ζ is the supremum of the eventually writable ordinals.
- Σ is the supremum of the accidentally writable ordinals.

λ^x, ζ^x, Σ^x are defined the same way but relative to x.

Theorem II.51

λ is eventually writable and ζ is accidentally writable.

Proof. The ordinal λ is eventually writable via the machine M, in the following way: M simulates in parallel all ITTMs. At each step, M writes on the output tape the sum of all ordinals written on the output tape of halted simulated machines. When the simulation of all halting machines have halted, the output tape of M will never change, and contains an ordinal higher than all writable ordinals. Therefore, as the set of eventually writable ordinals is closed downward, λ is eventually writable.

We now prove that ζ is accidentally writable, via the following machine M. The execution of M once again simulates all ITTMs, but this times it writes on the output tape the sum of ordinals written on the output tape of any machine, halted or not. Let α be an ordinal where all eventually writable ordinals have settled in the simulations. Then, the ordinal written in the output tape at this stage α will be greater than any eventually writable ordinal. By the downward closure of the accidentally writable, ζ is accidentally writable. ■

Hamkins and Lewis also defined the clockable ordinals: an ordinal α is clockable if it is the halting time of some ITTM M. It is clear that the supremum of the clockable ordinals (we already defined it as γ) is at least λ: if an ordinal α is writable, one can design the machine that writes α and then "counts down α" in at least α steps just as we did in Theorem II.46. The question of equality between λ and the supremum of the clockable ordinals was one of the main question in Hamkins and Lewis [41]. It was later solved by Welch:

Theorem II.52 (Welch [65])

Let M be an ITTM. Write $T_M[\alpha]$ for the content of the tapes at stage α.

1. If $(T_M[\alpha](n))_{\alpha<\lambda}$ converges to $i \in \{0,1\}$, then $T_M[\alpha](n) = i$ for every $\alpha \geq \lambda$.
2. If $(T_M[\alpha](n))_{\alpha<\zeta}$ converges to $i \in \{0,1\}$, then $T_M[\alpha](n) = i$ for every $\alpha \geq \zeta$.

3. $(T_M[\alpha](n))_{\alpha<\zeta}$ diverges if and only if $(T_M[\alpha](n))_{\alpha<\Sigma}$ diverges.

We have in particular $C_M[\zeta] = C_M[\Sigma]$. Also ζ, Σ is the lexicographically smallest pair of ordinals such that $C_U[\zeta] = C_U[\Sigma]$ for the universal machine U.

Proof. First, suppose that the result is true for $\alpha < \Sigma$. Then, We still have that $C_M[\zeta] = C_M[\Sigma]$ and more than that, the 0 valued cells never take value 1 between stage ζ and Σ. The machine thus enters a loop, and the converging cells never take a new value. Therefore, we only need to prove item 1 and 2 for $\alpha < \Sigma$.

We start by proving item 1. Suppose that $(T_M[\alpha](n))_{\alpha<\lambda}$ converges to $i \in \{0, 1\}$, but there exists α with $\lambda \leq \alpha \leq \Sigma$ and $T_M[\alpha](n) = 1 - i$. Let $\beta_0 < \lambda$ be a stage where $(T_M[\beta])_{\beta<\lambda}$ has converged. Let M' be the machine that searches for an accidentally writable β with $\beta > \beta_0$ but $T_M[\beta] = 1 - i$, writes it on the tape and halts. This machine will halt, as there exists such an accidentally writable ordinal. But then this ordinal satisfies $\beta_0 < \beta < \lambda$ but $T_M[\beta](n) = 1 - i$, a contradiction with the definition of β_0.

We now prove item 2 with very similar method, but using eventually writability instead of writability. Once again, we let $\beta_0 < \zeta$ be an eventually writable ordinal such that between β_0 and ζ, the value of $T_M(n)$ is constant and equal to i. We suppose that for some accidentally writable ordinal β above ζ, we have $T_M[\beta](n) = 1-i$. Then, we define the following ITTM M', which does the following: it finds the first accidentally writable β such that $T_M[\alpha](n) = 1 - i$ that is above the approximation of β_0, and writes it to the tape. Whenever the approximation of β_0 changes, it may change the ordinal written on the output tape.

First, because there exists one, M' will always find an ordinal to write on the tape. This ordinal written on the tape will eventually settle, as the approximation of β_0 will settle, and therefore it is an eventually writable ordinal $\beta < \zeta$. But $T_M[\beta](n) = 1 - i$ and $\beta_0 < \beta < \zeta$, a contradiction.

It remains to prove item 3. The reverse implication is a weakening of item 2, so we only have to prove the forward implication. Suppose that there exists an accidentally writable $\alpha < \Sigma$ such for $\alpha < \beta < \Sigma$, $T_M[\beta](n)$ is constant equal to i (we take α as the smallest such). Then, we define the following ITTM M': the machine M' searches for all accidentally writable ordinals β such that $T_M[\beta](n) = i$. Whenever it finds such a β, it first checks that it is greater that the one, currently β_0, written on the output tape. If so, it checks that between β and β_0, there exists an ordinal δ with $T_M[\delta] = 1 - i$. If so, it replaces on the output tape the ordinal β_0 by the new ordinal β.

We argue that the machine M' eventually writes an ordinal above α. Indeed, the first accidentally writable found above α will be written on the output tape, and never be replaced. Therefore, $\alpha < \zeta$ and $(T_M[\alpha](n))_{\alpha<\zeta}$ must converge. ■

▶ **Corollary II.52.1** (Welch [65]). *The ordinal λ is also the supremum of the ITTMs' halting time.*

Proof. By Theorem II.52, we have that if an ITTM M has not halted before stage Σ, then it will never halt, because the configuration of an ITTM at stage Σ is the same as the configuration of an ITTM at stage ζ, and every 1 in the tape at stage ζ will stay a 1 at every stage between ζ and Σ. Thus the computation will loop forever, and if an ITTM halts it must halt before stage Σ. We can then run an ITTM which looks for all the accidentally writable ordinals α (using some universal ITTM) and for each of them, which runs M for α steps. When the machine finds an accidentally writable ordinal α such that $M[\alpha]\downarrow$, then it writes α

and halts. By hypothesis on M our ITTM will write α and halt at some point. Thus α is a writable ordinal, which implies that M halts at a writable step. ■

▶ **Corollary II.52.2** (Welch [65]). *The ordinal ζ is equal to the supremum of the ITTMs' converging time.*

Proof. By Theorem II.52, we have that if an ITTM M has not converged before stage Σ, then it will never converge, because the configuration of an ITTM at stage Σ is the same as the configuration of an ITTM at stage ζ, and thus the computation will loop forever with at least one change in some cell in each loop. Thus if an ITTM converge it must converge before stage Σ. But then, by Theorem II.52 each cell must actually have converged at time ζ, never to be changed again. So if M has a converging output tape, it must be converging before ζ.

Now, let us show that the time of converging computations are cofinal below ζ. Let $\alpha < \zeta$ be an eventually writable ordinal. Then, the following machine M has a converging output tape, that reaches a stable value after at least α steps: M computes an approximation of α, on its working tape. Each time the approximation changes, M verify if it codes an ordinal using the algorithm of Theorem II.43, item 4, while at the same time "messing" with the output tape: flipping one of its bit for every step of the algorithm. But given an ordinal γ, the algorithm takes at least γ steps to finish. So, the first time that the approximation of α reaches its final state, the output tape will not be converging for at least α more steps, and settle down after that. ■

▶ **Corollary II.52.3** (Welch [65]). *The ordinal Σ is equal to the supremum of the times where new accidentally reals appear.*

Proof. As any ITTM starts to loop after it has reached Σ, then no new reals can appear after stage Σ. It remains to show that some new reals appear cofinally below Σ. First, as ζ is accidentally writable but not eventually writable, it cannot appear at a stage below ζ. Now, suppose that no new real appears at stage $\alpha > \zeta$ that we consider limit. Then, the universal machine M_u at stage α has the exact same configuration as at some ordinal $\beta < \alpha$. Let δ such that $\beta + \delta = \alpha$. Then, by Theorem II.52, all cells containing a 0 do not change of value between stage α and $\alpha\delta$. Therefore, the universal machine starts looping from stage α to $\alpha+\delta$, and as by Theorem II.52 the pair (ζ, Σ) is the smallest initiating such a loop, we must have $\alpha + \delta \geq \Sigma$, so $\alpha \geq \Sigma$ as Σ is closed by sum. ■

Welch's theorem and proof provided a clear understanding of ITTMs allowing us, as we will see it soon, to cut ourselves off the machine model, and to reason within the constructible hierarchy.

II.3.4 The λ-ζ-Σ theorem

We state in this section results regarding the three ordinals λ, ζ, Σ, and their relative versions λ^x, ζ^x and Σ^x. In particular, as promised we establish a clear connection between ITTMs and constructibles, summed up in the two following theorems.

We introduce a coding of hereditarily countable sets defined in Definition II.23, into elements of the Cantor set.

Definition II.53

A *code for an hereditarily countable set* H, is a real $x \in 2^\omega$ such that $x = \langle \mathrm{dom}(x), \in_x \rangle$ and there is an isomorphism between $(\mathrm{dom}(x), \in_x)$ and $(\mathrm{tc}(\{H\}), \in)$.

We can prove that every hereditarily countable set has a code. Indeed we prove by induction over the membership relation $\in$ that for every hereditarily countable set H, the transitive closure of H is countable. This is clear by induction, because $\mathrm{tc}(\{H\}) = \{H\} \cup \{\mathrm{tc}(A) : A \in H\}$ and H is countable. So if $(A_n)_{n\in\mathbb{N}}$ is an enumeration of $\mathrm{tc}(\{H\})$, then $x \in 2^\omega$ defined by $\mathrm{dom}(x) = \mathbb{N}$ and $n \in_x m$ if and only if $A_n \in A_m$, is a code for H.

Also note that from a code x for H it is possible to recover the particular $n \in \mathrm{dom}(x)$ which is linked to H in the isomorphism: it is the only n such that there is no m with $n \in_x m$.

Theorem II.54 (Welch [65])

The set L_λ (resp. L_ζ, resp. L_Σ) is the set of all sets with a writable (resp. eventually writable, resp. accidentally writable) code.

Proof. Let us call $H(\lambda)$ (respectively $H(\zeta)$, $H(\Sigma)$) the set of hereditarily countable sets with a writable (respectively eventually writable, accidentally writable) code. Our goal is to prove that $H(\eta) = L_\eta$ for η being λ, ζ and Σ.

Before that, we argue that ITTMs can manipulate freely the codes for hereditarily countable sets. If a and b are two codes, then it is ITTM-decidable if they both code for the same set: indeed, this fact is Σ^1_1 as it can be restated as "there exists an isomorphism between the two relations". Given a code b for a set B, the elements of the transitive closure of B are the one with code b_n, where $n \in \mathrm{dom}(b)$ and b_n consists of b with domain restricted to the m such that there exists a finite sequences $(m_i)_{i\leq k}$ with $m = m_0$, $m_k = n$ and $m_i \in_a m_{i+1}$. Given two codes a and b for sets A and B, an ITTM can also decide if $A \in B$: it is the case if there exists an n such that a and b_n code for the same set. Therefore, all Δ_0 formulas are decidable, and as the quantification in a code ranges over integers, the validity of any first order formula in some hereditarily countable set is ITTM-decidable in the code.

For clarity we separate the proof in the two steps.

▶ **Claim II.54.1.** *$L_\eta \subseteq H(\eta)$ for η being λ, ζ and Σ.*

Proof. First, we will prove that $L_\lambda \subseteq H(\lambda)$, in other words, $L_\alpha \subseteq H(\lambda)$ for every writable α. As $\alpha + 1$ is a well order on $\mathbb{N}$, an ITTM can use it to split the working tape into countably many tapes $(T_\beta)_{\beta<\alpha+1}$. Then, we argue that the construction of L_α can be done by an ITTM, by having a code for L_β on the tape T_α.

A code for L_0 is clearly ITTM-computable. Given a code for L_β, we have to show that a code for $L_{\beta+1}$ is ITTM-writable. The first order formulas with parameters from a code c are recursively enumerable in c. Then, an ITTM can check their validity . Therefore, an ITTM is able to write a code for $L_{\beta+1}$.

If β is limit, then, a code for L_β should be a code for the union of L_γ for $\gamma < \beta$. All those codes are accessible on the tapes T_γ, so it suffices to create a code for the union. The code $x_\cup$ of the union will have $\mathrm{dom}(x_\cup)$ split into β set, where the γ-th subset contains the elements of $L_{\gamma+1} \setminus L_\gamma$. As we fill the γ-th subset, we also complete the membership relation.

Using the previous construction, we are able to create a code for every L_α for α writable. Therefore, every set in L_λ has a writable code, and $L_\lambda \subseteq H(\lambda)$. If α is eventually writable, we are able to do the same construction but restarting from the beginning every time the approximation of α changes. So $L_\zeta \subseteq H(\zeta)$. It is also possible to enumerate all accidentally writable ordinals, so we can do the construction for each of them, and accidentally write all the codes obtained, so $L_\Sigma \subseteq H(\Sigma)$. ■

▶ **Claim II.54.2.** *$H(\eta) \subseteq L_\eta$ for η being λ, ζ and Σ.*

Proof. We argue that any computation (halting or not) of length α is in $L_{\alpha+n}$ for some n. Indeed, if $c = (c_\beta)_{\beta<\alpha+1}$ is a computation of length α, there is a unique and definable way to extend it to a computation $\hat{c}$ of length $\alpha+1$, so if $c \in L_{\alpha+n}$ then $\hat{c} \in L_{\alpha+n+m}$ for some $m \geq 1$ (depending on the way we code the computation). If there is a computation of length α in $L_{\alpha+n_\alpha}$ for every $\alpha < \lambda$, then there is a computation of length λ in $L_{\lambda+n}$ for some n, defined from the union of computations of length less than λ and the limit rule.

Therefore, if a computation halts before α, then the result of the computation must be in $L_{\alpha+n}$ for some n. By Corollary II.52.1, $\gamma = \lambda$. Let $A_x \in H(\lambda)$ with x being a code for A_x such that there is a computation of length $\alpha < \lambda$ writing x. Then, we have $x \in L_\lambda$, and we need to argue that A_x is in L_λ. This is because we can define the Mostowski collapse of x, which yields A: Suppose $x \in L_\alpha$. Then, constructing by induction the function f with domain $\mathrm{dom}(x)$ verifying $f(n) = \{f(m) : m \in_x n\}$. The induction is of length β bounded by $\mathrm{KB}(\in_x)$, which is a writable ordinal as x is writable, so at $L_{\alpha+\beta} \subseteq L_\lambda$ there is a function from $\mathrm{dom}(x)$ to A_x, and then $A_x \in L_\lambda$.

In order to show that $H(\zeta) \subseteq L_\zeta$, this time we use a relation between ζ and converging times for ITTMs. Indeed, by Corollary II.52.2, ζ is also the supremum of the converging time of converging computations. So, if x is an eventually writable code of a set A_x, and x is the converging value of the output tape of an ITTM, then the output tape must have converged at step $\alpha < \zeta$. Then, x appears in some computation of length $\alpha < \zeta$, so $x \in L_\alpha$ and $A_x \in L_\zeta$.

Finally, we deal with $H(\Sigma) \subseteq L_\Sigma$. If x is accidentally writable, by Corollary II.52.3 it must appear at some time before Σ. But then $x \in L_\alpha$ for $\alpha < \Sigma$, and $A_x \in L_\Sigma$. ■

The following theorem is similar to Theorem II.52, but with the constructible hierarchy in place of ITTM's tapes.

Theorem II.55 (Welch [65])

The triplet of ordinals (λ, ζ, Σ) is the lexicographically smallest triplet such that

$$L_\lambda \prec_1 L_\zeta \prec_2 L_\Sigma$$

By relativization, $(\lambda^x, \zeta^x, \Sigma^x)$ is the lexicographically smallest triplet such that

$$L_{\lambda^x}(x) \prec_1 L_{\zeta^x}(x) \prec_2 L_{\Sigma^x}(x)$$

Proof. Let us first show that $L_\lambda \prec_1 L_\Sigma$. Let Φ be a Σ_1 formula, with parameters in L_α for α-writable. Thus, the parameter has a writable code. Let M be the machine that searches for an accidentally writable β such that Φ is true in L_β, then writes β on the output tape and halts. If $L_\Sigma \models \Phi$, then $L_\lambda \models \Phi$, so $L_\lambda \prec_1 L_\Sigma$.

By Corollary II.34.1, we only need to prove that if L_Σ is a model of a Σ_2 formula, then so is L_ζ. Let $\Phi \equiv \exists a\, \forall b\, \phi(a, b, p)$ be a Σ_2 formula with a parameter p in L_ζ. Then, let M be the machine which does the following: M searches for the first accidentally writable α it can find so that for all the accidentally writable ordinal β to come, we have $\exists a \in L_\alpha, \forall b \in L_\beta, \Phi(a, b, p)$ (where p is approximated). It writes the current such ordinal α on the output tape when it finds one, and replaces it only when a strictly bigger is needed.

We argue that M eventually writes an ordinal on its output tape: indeed, as we have

$L_\Sigma \models \exists a\ \forall b\ \phi(a,b,p)$, there exists some $\alpha < \Sigma$ such that $\exists a \in L_\alpha\ \forall b \in L_\Sigma\ \phi(a,b,p)$, so when the approximation of p has converged and such an α is found it will be written on the tape and never be replaced. But then, $\alpha < \zeta$ and so $L_\zeta \models \exists a\ \forall b\ \phi(a,b,p)$.

We now argue that this is the smallest pair. Suppose there exists a smaller pair (α,β,δ) with $L_\alpha \prec_1 L_\beta \prec_2 L_\delta$. Then as we can do all computations of length less than η in L_η for η limit, every converging cells at step δ actually converged at step β. But then, the universal machine (in the sense that it simulates all the others) have reached a looping cycle at stage β, repeating at stage δ. This is so for $\delta = \Sigma$ and $\beta = \zeta$ by Theorem II.52. If an Infinite Time Turing Machine has halted, then it has halted at step $\beta = \zeta$, so by stability it has also halted at stage α. Therefore, α is at least the supremum of halting machines, that is $\alpha = \lambda$ by Corollary II.52.1. ■

▶ **Corollary II.55.1.** *The ordinal ζ is Σ_2-admissible, and there are cofinally many eventually writable Σ_2-admissible ordinals below ζ.*

Proof. As $L_\zeta \prec_2 L_\Sigma$, we deduce from Theorem II.36 that ζ is Σ_2-admissible. In particular for any eventually writable α we have that L_Σ is a model of $\Phi \equiv$"there exists $\beta > \alpha$ which is Σ_2-admissible". It follows that L_ζ is also a model of Φ and thus that there are cofinally many eventually writable Σ_2-admissible ordinals. ■

▶ **Corollary II.55.2** (Hamkins, Lewis [41])**.** *The ordinal λ is Σ_1-admissible, and there are cofinally many writable Σ_2-admissible ordinals below λ.*

Proof. For any writable α we have that L_Σ is a model of "there exists $\beta > \alpha$ which is Σ_2-admissible". It follows that L_λ is also a model of that and thus that there are cofinally many writable Σ_2-admissible ordinals. ■

An important question of [41] was to determine whether Σ was admissible or not. Welch's proof that ITTMs halt only at ordinals smaller than λ provides insight on the way ITTMs work, and helped to solve the question. The proof can also be found in [66].

Theorem II.56 (Welch)

There is a function $f : \omega \mapsto \Sigma$ which is Σ_1-definable in L_Σ with ζ as a parameter and such that $\sup_n f(n) = \Sigma$.

Proof. Let U be the universal ITTM, which simulates every other ITTM. In particular we have by Theorem II.52 that Σ is the smallest ordinal greater than ζ such that $C_U[\zeta] = C_U[\Sigma]$. For every n let us define the function f_n such that $f_n(0) = \zeta$ and $f_n(m+1)$ is the smallest ordinal bigger than $f_n(m)$ such that $\forall i \leq n$ for which $\{C_U(i)[\beta]\}_{\beta<\zeta}$ does not converge, we have that $C_U(i)$ has changed at least once in the interval $[f_n(m), f_n(m+1)]$.

If there was some n such that $\sup_m f_n(m) = \Sigma$, this would prove the theorem already. It is actually possible to show, by combining Σ_2-stability of L_ζ in L_Σ, together with admissibility of L_ζ, that this cannot happen for any n. Let us then define the function f as follow: $f(n)$ is the smallest ordinal α greater than ζ such that $C_U[\zeta]\restriction n = C_U[\alpha]\restriction n$. As for every m we have $\sup_m f_n(m) < \Sigma$, then $f(n) < \Sigma$ and thus f is Σ_1-definable in L_Σ with ζ as a parameter. It is clear that $f(n) \leq f(n+1)$. It is also clear that $f(n) < \sup_n f(n)$ as otherwise we would have $C_U[\zeta] = C_U[\alpha]$ for some $\alpha < \Sigma$.

Let $\alpha = \sup_n f(n)$ and let us show $\alpha = \Sigma$. Let $n \in \omega$. If $\{C_U(n)[\beta]\}_{\beta<\zeta}$ converges then by (2) of Theorem II.52 we have $C_U(n)[\zeta] = C_U(n)[\alpha]$. If we have that $\{C_U(n)[\beta]\}_{\beta<\zeta}$ diverges

then $C_U(n)[\zeta] = 0$. Then either $\{C_U(n)[\beta]\}_{\beta<\alpha}$ converges to $C_U(n)[\zeta] = 0$ or $\{C_U(n)[\beta]\}_{\beta<\alpha}$ diverges and then $C_U(n)[\alpha] = 0$. In both cases we have $C_U(n)[\alpha] = C_U(n)[\zeta]$. This implies that $C_U[\alpha] = C_U[\zeta]$ which implies $\alpha = \Sigma$. ■

▶ **Corollary II.56.1** (Welch). *The ordinal Σ is not admissible.*

The function of Theorem II.56 will be used in various places of this thesis. We also have the following:

Theorem II.57 (Welch)

The ordinal Σ is a limit of admissible ordinals.

Proof. By Lemma II.55.1, ζ is a limit of admissible ordinals. By Σ_2-stability, Σ must also be a limit of admissible ordinals. ■

We now study what effect the increase of one of the three main ordinals has on the others.

Theorem II.58

The following are equivalent:

1. $\zeta^x > \zeta$
2. $\Sigma^x > \Sigma$
3. $\lambda^x > \Sigma$

Proof. Suppose $\zeta^x > \zeta$. In particular ζ is eventually writable in x. Let $\{\zeta_s\}_s$ be the successive approximations of ζ using an ITTM that eventually writes ζ using x. We can run an ITTM $M(x)$ which does the following: at step s, it uses ζ_s as a parameter in the function $f : \omega \mapsto \Sigma$ of Theorem II.56, and whenever it has found values for every $f(n)$ (and no new version of ζ_s has arrived so far), it writes $\Sigma_s = \sup_n f(n)$ on the output tape. At some stage s we will have $\zeta_s = \zeta$ and thus $\Sigma_s = \Sigma$ will be on the output tape. It follows that $\Sigma^x > \Sigma$.

Suppose now that we have $\Sigma^x > \Sigma$. We can run the ITTM $M(x)$ which search for two x-accidentally writable ordinals $\alpha < \beta$ such that $L_\alpha \prec_2 L_\beta$, then writes β and halts. As $\zeta < \Sigma$ is the smallest such pair of ordinals and as $\Sigma^x > \Sigma$, the ITTM will write an ordinal equal to or bigger than Σ at some point and halt. We then have $\lambda^x > \Sigma$.

Finally if $\lambda^x > \Sigma$ it is clear that $\zeta^x > \zeta$. ■

Theorem II.59

For every $\lambda \leq \alpha < \zeta$, there exists $x \in \omega$ such that $\alpha \leq \lambda^x$, such that $\zeta^x = \zeta$ and $\Sigma^x = \Sigma$.

Proof. Let α be such that $\lambda \leq \alpha < \zeta$. Let $x \in \omega$ be an eventually writable code for α. It is clear that $\alpha \leq \lambda^x$. Let us show $\zeta^x = \zeta$. Let α be any x-eventually writable ordinal, via some ITTM M. Let N be the ITTM which starts to eventually write x and in the same time uses the current version x_s of x to run $M(x_s)$ and copy at every time the output tape of M on the output tape of N. There is some stage s such that for every stage $t \geq s$ we will have $x_s = x_t = x$ together with $M(x)[s] = M(x)[t] = \alpha$. This implies also $N[s] = N[t] = \alpha$. Thus α is eventually writable.

From Theorem II.58 we have $\Sigma^x = \Sigma$, as $\zeta^x = \zeta$. ■

As a last theorem about the properties of reals which collapse λ, we give a bound on the Borel complexity of this set. This uses a result by Friedman and Welch that is not covered in this thesis, but can be found in [34]. It states that there is a "Theory Machine" that is able to write on its output tape the theory of $(J_\alpha, \in)$ at stage $\omega^2 \times (\alpha+1)$, where J_α is the α-th level of Jensen's hierarchy, another version of the constructibles. We have $\alpha = \omega\alpha$ implies that $L_\alpha = J_\alpha$, so $L_\eta = J_\eta$ for η being λ, ζ or Σ.

Theorem II.60

The two sets $\{x \in 2^\omega : \lambda^x > \lambda\}$ and $\{x \in 2^\omega : \Sigma^x > \Sigma\}$ are both $\mathbf{\Sigma}^{\mathbf{0}}_{\omega^{\mathbf{2}}(\mathbf{\Sigma}+\mathbf{2})+\mathbf{1}}$.

Proof. Let us show that for any $x \in 2^\omega$, we have that

$$\lambda^x > \lambda \quad \iff \quad \exists e \in \mathbb{N}(\forall \alpha < \lambda, M_e(x) \uparrow [\alpha]) \wedge M_e(x) \downarrow [\omega^2(\Sigma+2)]$$

where $(M_e)_{e\in\mathbb{N}}$ is an enumeration of ITTMs.

For the reverse implication, it is clear as if there exists a computation that does not halt in time less than λ, but still halts when given more time, then $\lambda^x > \lambda$. So, the non trivial part is to find a bound on the smallest halting time strictly above λ, for all x such that there exists one.

If $\Sigma^x = \Sigma$ and $\lambda^x > \lambda$, then there is a computation halting after λ but below Σ. Otherwise, suppose $\Sigma^x > \Sigma$. Then, we use [34, Corollary 3] to show that there exists a computation halting between Σ and $\omega^2(\Sigma+2)$. We transform the "Theory Machine" that writes the theory of $J_\alpha[x]$ for x-accidentally writable ordinals, to make it halt whenever $J_\alpha[x] \models \exists \alpha, \beta$ such that $L_\alpha \prec_1 L_\beta$. Let M_e be this machine.

Then, the set $\{x \in 2^\omega : \lambda^x > \lambda\}$ is also the set:

$$\{x \in 2^\omega : L_{\omega^2(\Sigma+2)}(x) \models \exists e \in \mathbb{N}(\forall \alpha < \lambda, M_e(x) \uparrow [\alpha]) \wedge M_e(x) \downarrow\}$$

and by Theorem V.20 [a]this set is $\mathbf{\Sigma}^{\mathbf{0}}_{\omega^{\mathbf{2}}(\mathbf{\Sigma}+\mathbf{2})+\mathbf{1}}$.

Similarly, the set $\{x \in 2^\omega : \Sigma^x > \Sigma\}$ is also the set:

$$\{x \in 2^\omega : L_{\omega^2(\Sigma+2)}(x) \models \exists e \in \mathbb{N}(\forall \alpha < \Sigma, M_e(x) \uparrow [\alpha]) \wedge M_e(x) \downarrow\}$$

and this set is also $\mathbf{\Sigma}^{\mathbf{0}}_{\omega^{\mathbf{2}}(\mathbf{\Sigma}+\mathbf{2})+\mathbf{1}}$.

a. This theorem will be seen in the last chapter of this thesis. It says that given a formula Δ^0 formula Φ, the set $\{x \in 2^\omega : L_\beta(x) \models \Phi\}$ is Borel of complexity $\mathbf{\Sigma}^{\mathbf{0}}_\beta$, with a code that is β-recursive in a code for Φ. We refer the reader to section V.2.3 for a more detailed presentation.

We now study the projectibility of the three ordinals λ, ζ and Σ. Intuitively λ is projectible into ω, by the function which to $\alpha < \lambda$ associates the code of the first ITTM which is witnessed to write α. Such a thing is of course not possible to achieve with ζ, which indeed is not projectible.

Theorem II.61

1. λ is projectible into ω without parameters.
2. ζ is not projectible.

Proof. A direct proof of (1) would be possible. It is also a direct consequence of Theorem II.38: it is well known that the set $\{e \in \omega : \text{the } e\text{-th ITTM halts}\}$ is not writable (by a standard diagonalization, see for instance [41]) and thus does not belong to L_λ. It is however Σ_1-definable in L_λ and thus L_λ is not a model of Σ_1-comprehension for subsets of ω. It follows that $\lambda^* = \omega$, with no parameters.

To prove (2), we will show that L_ζ satisfies Σ_1-comprehension for any set in L_ζ. We shall first argue that for every $\alpha < \zeta$, there exists $\beta \geq \alpha$ such that $L_\beta \prec_1 L_\zeta$. For every $\alpha < \zeta$, there exists by Theorem II.59 some $x \in 2^\omega$ such that $\lambda^x > \alpha$ and such that $\zeta^x = \zeta$ and $\Sigma^x = \Sigma$. In particular we have $L_{\lambda^x} \prec_1 L_{\zeta^x} = L_\zeta$. Now suppose that for $\alpha < \zeta$ we have that $A \subseteq L_\alpha$ is Σ_1-definable in L_ζ with parameters in L_α. Let $\beta \geq \alpha$ be such that $L_\beta \prec_1 L_\zeta$. In particular $A \subseteq L_\alpha$ is Σ_1-definable in L_β. It follows that $A \in L_\zeta$. ■

We now study the projectibility of L_Σ. We will show that it is projectible into ω with parameter ζ, in a strong sense, that is, with a bijection. To do so we first need to argue that L_Σ is a model of "everything is countable". It is clear intuitively: if x belongs to L_Σ then it has an accidentally writable code, and this code gives the bijection between x and ω. Friedman showed a bit more:

▶ **Lemma II.62** (Friedman [34]). *Let α be limit. Suppose there exists $x \in L_\alpha$ such that $L_\alpha \models$ "x is uncountable". Then there exists $\gamma < \delta < \alpha$ such that $L_\gamma \prec L_\delta$ (that is, $L_\gamma \prec_n L_\delta$ for every n).*

Proof. Let us first argue that there must be a limit ordinal $\delta < \alpha$ such that $L_\alpha \models$ "L_δ is uncountable". If α is limit of limit this is clear, because there must be a limit ordinal δ such that L_δ contains an x which is uncountable in L_α. As L_δ is transitive, it must be itself uncountable in L_α. If α is not limit of limit, let δ be limit such that $\alpha = \delta + \omega$. Suppose that L_δ is countable in L_α. Thus also by definition of L, every element of $L_{\delta+1}$ must be countable in L_α. We can continue inductively to show that every element of $L_{\delta+\omega} = L_\alpha$, must be countable in L_α, contradicting our hypothesis.

Thus there must be a limit ordinal $\delta < \alpha$ such that $L_\alpha \models$ "L_δ is uncountable". We then conduct within L_α the Löwenheim-Skolem proof to find a countable set $A \subseteq L_\delta$ such that $A \prec L_\delta$. The Mostowsky collapse A' of A is transitive, as $A' \prec L_\delta$ and as L_δ is a model of "everything is constructible" together with "for all β the set L_β exists"[a], then A' must be of the form L_γ for some $\gamma \leq \delta$. As L_γ is countable in L_α we must have $\gamma < \delta$. ■

a. Note that this is where we use that δ is limit, using Theorem II.30

▶ **Corollary II.62.1.** *For any limit ordinal $\alpha \leq \Sigma$, we have that $L_\alpha \models$ "everything is countable".*

Proof. It is immediate using that Σ is the smallest ordinal such that $L_\alpha \prec_2 L_\Sigma$ for some $\alpha < \Sigma$. ■

Theorem II.63

Suppose $L_\alpha \models$ "everything is countable" and α is not admissible. Then there is a bijection $b : \omega \to L_\alpha$ which is Σ_1-definable in L_α with the same parameters than the ones used in a witness of non-admissibility of α. In particular α is projectible into ω, with these parameters (using b^{-1} restricted to ordinals).

Proof. We first show that there is a Σ_1-definable surjection from ω onto L_α. As α is not admissible, there is a set $a \in L_\alpha$ and a function $g : a \mapsto \alpha$ which is Σ_1-definable

over L_α with some parameter $p \in L_\alpha$, and such that $\sup_{x\in a} g(x) = \alpha$. Note that as $L_\alpha \models$ "everything is countable", there is a bijection in L_α between a and ω. Using this bijection there is then a function $f : \omega \to \alpha$ which is Σ_1-definable over L_α with parameter p, a, and such that $\sup_{n\in\omega} f(n) = \alpha$ (the bijection does not need to be a parameter, as it can be detected directly with a Σ_1-formula, similarly to the way it is done below). Let $\Psi(n, \beta)$ be the Σ_1 functional formula with parameter p, a, which defines f.

We now define a Σ_1 formula (with parameter p, a) $\Phi(n, m, z)$ such that for every n, m there is a unique $z \in L_\alpha$ for which $L_\alpha \models \Phi(n, m, z)$, and such that for every $z \in L_\alpha$, there exists n, m such that $L_\alpha \models \Phi(n, m, z)$. We define:

$$\begin{aligned}\Phi(n, m, z) \quad \equiv \quad & \exists g\ \exists \beta \ \text{ s.t.} \\ & \Psi(n, \beta) \text{ and} \\ & g \text{ is a bijection between } \omega \text{ and } L_\beta \text{ s.t. } g(m) = z \text{ and} \\ & \text{every } g' <_L g \text{ is not a bijection between } \omega \text{ and } L_\beta\end{aligned}$$

Recall that $<_L$ is the order of Theorem II.31. Note that the quantification $\forall g' <_L g$ can formally be replaced by a quantification bounded by the set $\{g' <_L g\}$, and that this set can be obtained with an existential quantification at the beginning of the formula Φ. In particular Φ is Σ_1. It is clear that for every n, m, there is at most one z such that $\Phi(n, m, z)$. The fact that every $z \in L_\alpha$ is defined by Φ for some n, m is clear because $L_\alpha \models$"everything is countable".

It follows that there is a surjection f from ω onto L_α, which is Σ_1-definable in L_α with parameters p, a. To obtain a bijection, we define the function $h : \omega \to \omega$ such that $h(0) = 0$ and $h(n + 1) = \min\{m \in \omega \ : \ \forall n' \leq n \ f(h(n')) \neq f(m)\}$. Note that h is defined by Σ_1-induction. As α is not admissible, we should make sure we can do so. This relies on the fact that h is defined only on integers: we can then essentially rely on the admissibility of ω. Indeed, to decide $h(n + 1) = m$, we only need the finite function $h \restriction_n$ and the finite function $f \restriction_m$. In particular only finitely many witnesses for values of f are needed and they then all belongs to some L_β for $\beta < \alpha$. Formally we can define h in L_α as follow:

$$\begin{aligned}h(n) = m \quad \equiv \quad & \exists \beta\ \exists h' \restriction_n \ \ \forall k < n \\ & h'(k + 1) > h'(k) \wedge L_\beta \models \forall i < k \ f(h'(k)) \neq f(i) \wedge \\ & \forall j \text{ with } h'(k) < j < h'(k + 1) \ L_\beta \models \exists i < j \ f(h'(j)) = f(i) \\ & \text{and } h'(n) = m\end{aligned}$$

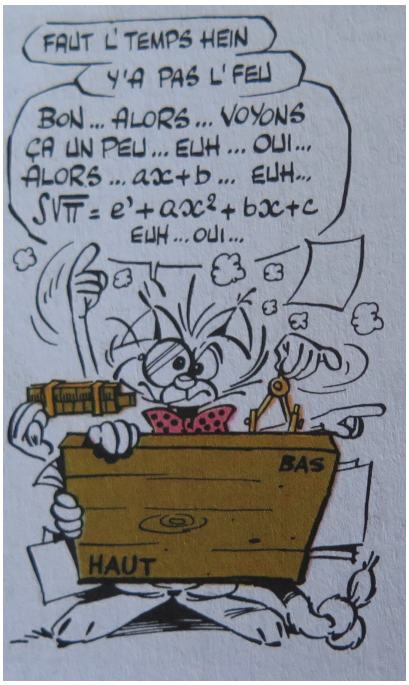
The bijection is then given by $b(n) = f(h(n))$. ■

▶ **Corollary II.63.1.** *There is a bijection $b : \omega \to L_\Sigma$ which is Σ_1-definable in L_Σ with ζ as a parameter. In particular Σ is projectible into ω, with parameter ζ.*

Proof. From Theorem II.56 there is a function $f : \omega \mapsto \Sigma$ which is Σ_1-definable over L_Σ with parameter ζ and such that $\sup_n f(n) = \Sigma$. From Corollary II.62.1: we have that $L_\Sigma \models$"everything is countable". The result follows. ■

Chapter III

The Reverse Math of Hindman's Theorem



Gotlib, RàB T2. Le matou matheux

We have seen that many theorems of usual mathematics are equivalent to one of the big five axiomatic systems, known as RCA_0, WKL_0, ACA_0, ATR_0 and $\Pi^1_1\text{-}\mathrm{CA}_0$. There exists also many axiomatic systems that are not equivalent to any of these big five. But what about natural theorems? What kind of theorems are an exception to the empirical phenomenon of the big five?

Most answers to this question come from combinatorics, and especially Ramsey Theory. Ramsey Theory is a branch of mathematics that studies how some order must appear in sufficiently large structures. The first and founding theorem of this branch is Ramsey's theorem, which states that for any n, there exists a monochromatic clique of size n in every sufficiently large complete graph with finitely colored edges.

If there are only two colors, one of the popular way of saying it is: In a party, and for any n, if there are sufficiently many people, then either there exists n people who never saw each others, or there exists n people who all know each others. For instance, in a rather small party of six people, there must exists three guests who never saw each others, or who all know each others.

There are generalizations of this theorem, where instead of coloring edges, that is a set of two elements, we color sets of n elements for a fixed n. We are interested here in an infinite version of this: If we color the sets of cardinality n with finitely many colors, then there exists an infinite set whose all subsets of cardinality n have the same color. The infinite version for 2 colors and 2 elements, RT^2_2, is between RCA_0 and ACA_0, but incomparable to WKL_0.

Among all the theorems studied in Ramsey Theory, which is the branch of mathematics that study orders emerging from chaos, RT^2_2 became the key-stone of natural theorems that are not in the big five.

In this chapter, we study one theorem from Ramsey Theory whose belonging to the big five is still unknown. While most theorems outside of the big five are below ACA_0, this one is known to imply it.

III.1 Hindman's Theorem

Hindman's Theorem is an answer to a conjecture by Graham and Rotschild [37] in 1971. Neil Hindman solved this conjecture positively in [43], showing that any finite coloring has a color containing the finite sums of an infinite set. But this is only the beginning of the study of this theorem: since Hindman's article, there has been many other proofs, and especially a proof by Galvin and Glazer (first appeared in [27]) starting a fruitful interaction between combinatorics and ultrafilters.

Hindman's Theorem has also been studied intensively by reverse mathematicians. Its first analysis was done by Blass, Hirst and Simspon in [1], showing that the principle must be between ACA_0 and ACA_0^+. There has been many attempts and improvements as in [64, 21], however the exact position of HT in the reverse mathematics zoo is still unknown. There has also been an extensive study of the various weakening of Hindman's theorem as in [21, 20, 30].

We start by giving all the details on Hindman's Theorem with no concerns on reverse mathematical aspects.

III.1.1 Statement

Definition III.1

Let $A \subseteq \mathbb{N}$. We denote by $\mathrm{FS}(A)$ the set of finite sums of distinct elements from A, that is:

$$\mathrm{FS}(A) = \left\{ \sum_{n \in F} n : F \subseteq_{\mathrm{fin}} A \right\}$$

We say that $B \subseteq \mathbb{N}$ is an IP *set* if there exists an infinite set $A \subseteq \mathbb{N}$ such that $\mathrm{FS}(A) \subseteq B$. We write $A \subseteq_{\mathrm{FS}} B$ for $\mathrm{FS}(A) \subseteq \mathrm{FS}(B)$.

As a part of Ramsey Theory, Hindman's Theorem share the same general form with Ramsey's theorem. However, instead of coloring tuples of fixed size, we color arbitrary sums of distinct elements.

Theorem III.2 (Hindman's Theorem, [43], $\mathrm{HT}_{\mathrm{FS}}$)

Let $c : \mathbb{N} \to k$ be a finite coloring of the integers. Then, there exists a monochromatic IP set.

The fact that it is a sum and not another operation on the integers is not crucial in the combinatorics of the proof. For instance, it could be adapted with no changes for multiplication. The precise requirement on the operation is that it forms a "right-moving semigroup", as defined in [36] (see also [44]). We will stick to the version of Hindman's theorem with finite sums, together with a version involving unions of finite sets that we will detail.

We often distinguish these two versions by calling them the "finite sum theorem" and the "finite union theorem". We will show that they are equivalent in terms of reverse mathematics.

Definition III.3

Let $U \subseteq \mathcal{P}_{\text{fin}}(\mathbb{N})$, then, $\text{FU}(U)$ is the set of finite unions of elements from U, that is:

$$\text{FU}(U) = \left\{ \bigcup_{A \in F} A : F \subseteq_{\text{fin}} U \right\}$$

Given two finite sets F and G, we write $F < G$ for $\max F < \min G$. We say that F and G are *apart* if $F < G$ or $G < F$, in other words if $\max F < \min G$ or $\max G < \min F$. A set U of finite sets has the *apartness property* if any two sets $F, G \in U$ are apart.
We say that S is an $\text{IP}_\cup$ *set* if there exists an infinite set $U \subseteq \mathcal{P}_{\text{fin}}(\mathbb{N})$ with the apartness property such that $\text{FU}(U) \subseteq S$.

One of the advantages of the Finite Union Theorem III.4 is that we don't really have to deal with the "preservation of finite sums" (in symbol $\subseteq_{\text{FS}}$): There exists some infinite set $A \subseteq \mathbb{N}$ such that $\text{FS}(A)$ is not closed under finite sums; an example is $A = \{1\} \cup 3\mathbb{N}$ where $\text{FS}(A) = 3\mathbb{N} \cup 3\mathbb{N}+1$, but $\text{FS}(\text{FS}(A)) = \mathbb{N} \setminus \{2\}$. Contrary to this, if U has the apartness property, then U is closed under finite union and $\text{FS}(\text{FS}(U)) = \text{FS}(U)$.

Theorem III.4 (HT_{FU})

Let $c : \mathcal{P}_{\text{fin}}(\mathbb{N}) \to k$ be a finite coloring of the finite set of integers. Then, there exists an $\text{IP}_\cup$ monochromatic set.

We will prove that over RCA_0, these two formulations are equivalent. We introduce the key link between these two theorems:

Definition III.5

We write $t_2 : \mathcal{P}_{\text{fin}}(\mathbb{N}) \to \mathbb{N}$ for the bijection $F \mapsto \sum_{i \in F} 2^i$. It draws a correspondence between the binary expansion of an integer and a finite set.
By extension, we say that a set of integers A has the apartness property if $t_2^{-1}[A]$ has the apartness property. In this chapter, we write $\lambda : \mathbb{N} \to \mathbb{N}$ for $\min \circ t_2^{-1}$ and $\mu : \mathbb{N} \to \mathbb{N}$ for $\max \circ t_2^{-1}$

Theorem III.6 (Lemma 2.2, [43])

We have $\text{HT}_{\text{FU}} \equiv_0 \text{HT}_{\text{FS}}$.

Proof. We start with the easy implication: the finite union theorem proves the finite sum theorem. Given c a coloring of the integer, we define the coloring $c' : F \mapsto c(t_2(F))$. By HT_{FU}, let $U \subseteq \mathcal{P}_{\text{fin}}(\mathbb{N})$ be with the apartness property, such that $\text{FU}(U)$ is monochromatic for c'. Then, $S = t_2[U]$ is an infinite set, such that $\text{FS}(S)$ is monochromatic for c: indeed, if $a_0, \cdots a_n \in S$, with corresponding $F_0, \cdots, F_n$ in U, then $a_0 + \cdots + a_n = t(F_0 \cup \cdots \cup F_n)$ by the apartness property, and therefore it must have the same c-color as a_0.

Now, let us deal with the other implication. We cannot only use the inverse of t_2, as for F, G apart we do have $t(F \cup G) = t(F) + t(G)$, but HT_{FS} does not imply that $\mu(n) < \lambda(m)$ for n, m in a solution for c. Therefore, we have to thin out the solution set.

More formally, let $c : \mathcal{P}_{\text{fin}} \to k$ be a coloring. Define $c' = c \circ t^{-1} : \mathbb{N} \to k$ and use HT_{FS} to

get a S such that $\mathrm{FS}(S)$ is monochromatic. Our goal is to find a sequence of elements $(a_n)_{n\in\mathbb{N}}$ with $\mu(a_n) < \lambda(a_{n+1})$ for all n.

We prove by induction on n that if $A \subseteq \mathbb{N}$ is any infinite set, then there exists $a \in \mathrm{FS}(A)$ such that $\lambda(a) \geq n$. The initial case is trivial. Suppose it is true for n, then let $F_0 \subseteq_{\mathrm{fin}} A$ and $a_0 = \sum_{b\in F_0} b$ such that $\lambda(a_0) \geq n$. Again, apply the induction hypothesis for $A \setminus F_0$ to get $F_1 \subseteq_{\mathrm{fin}} A \setminus F_0$ and $a_1 = \sum_{b\in F_1} b$ with $\lambda(a_1) \geq n$. If $\lambda(a_i) \geq n$ for some $i < 2$ we are done, otherwise $\lambda(a_i) = n$ for all $i < 2$ and therefore $\lambda(a_0 + a_1) \geq n + 1$.

We are therefore able to build the sequence $(a_n)_{n\in\mathbb{N}}$ with the apartness property and such that $\mathrm{FS}(\{a_n : n \in \mathbb{N}\}) \subseteq \mathrm{FS}(S)$: at each step, we search some a_{n+1} in the finite sums of elements of S not already used in a previous sum, with $\lambda(a_{n+1}) \geq \mu(a_n)$. The set $T = t_2^{-1}[\{a_n : n \in \mathbb{N}\}]$ is such that $\mathrm{FU}(T) = t_2^{-1}[\mathrm{FS}(\{a_n : n \in \mathbb{N}\})]$ and therefore is a solution to c for the finite union theorem. ■

The proof even show $\mathrm{HT}_{\mathrm{FU}} \equiv_{\mathrm{W}} \mathrm{HT}_{\mathrm{FS}}$ as defined later in Definition IV.4.

III.2 Multiple proofs

One of the striking phenomenons with regards to Hindman's Theorem is the variety of its proof. These proofs are truly different, and therefore make a good case of study for reverse mathematics. We will only present some of them, following the chronological order in which they appeared: Hindman first gave a pretty complex proof in [43], then Baumgartner simplified it in [6], and finally Galvin and Glazer found a very simple and powerful proof using ultrafilters, which first appeared in [27].

The amusing fact is that the simpler the proof is, the more it complex are the axioms used, therefore the reverse mathematics order of complexity of these proof is reversed [1].

▶ **Notation III.7.** Let $c : X \to k$ be a coloring of a set X. Then, we write $C_i^c = \{x \in X : c(x) = i\}$ the set of elements of X of color i. When there is no ambiguity we omit the coloring and write C_i.

III.2.1 The most effective proof

The most effective proof that we give here is almost the original one given by Hindman in 1974. In his paper [43], he had one very uneffective argument that was not really needed. In [1], Blast, Hirst and Simpson studied this proof in terms of reverse mathematics, and removed the uneffective bit.

Later, Towsner [64] gave a very elegant version of Hindman's proof by clarifying the intermediate combinatorial objects and steps of the proof. This is what we present here.

Definition III.8 ([64])

Let $F \subseteq \mathbb{N}$ be a finite set, and $S \subseteq \mathbb{N}$ be an infinite set with $F < S$, c a k-coloring defined on $\mathbb{N}$ and $i < k$ a color.

- We say that (F, S) is a *right-match for color* i if $F < S$ and for every $x \in \mathrm{FS}(S)$ of color i, there exists an $a \in F$ such that $c(x) = c(x + a)$,
- We say that (F, S) is a *left-match for color* i if $F < S$ and for every $x \in \mathrm{FS}(S)$ of color i, there exists an $a \in F$ such that $c(a) = c(x + a)$,
- We say that (F, S) is a *full-match for color* i if $F < S$ and for every $x \in \mathrm{FS}(S)$ of

1. as in "reverse" mathematics...

color i, there exists an $a \in F$ such that $c(a) = c(x) = c(x+a)$.

We say that (F, S) is a *right-match* (respectively *left-match*, *full-match*) if it is a right-match (respectively left-match, full-match) for every color.

Let us give a property on these objects, that explains why they are of interest for finding a witness to Hindman's Theorem.

▶ **Proposition III.9.** *Assume Hindman's Theorem. If (F, S) is a full-match then for every $T_0 \subseteq_{\mathrm{FS}} S$, there exists a set T_1 with $T_1 \setminus F \subseteq_{\mathrm{FS}} T_0$, $\mathrm{FS}(T_1)$ monochromatic and $F \cap T_1 \neq \emptyset$.*

In other word, no matter how we shrink S, there will always be a solution containing an element of F.

Proof. Suppose (F, S) is a full-match and $T_0 \subseteq_{\mathrm{FS}} S$. Then, let c_0 be the coloring $k \mapsto \langle c(k), a\rangle$ with $a \in F$ such that $c(k) = c(a+k) = c(a)$. By Hindman's Theorem, let $\widetilde{T_1} \subseteq_{\mathrm{FS}} T_0$ be such that $\mathrm{FS}(\widetilde{T_1})$ is monochromatic for c_0. It is clear that $T_1 = \{a\} \cup \widetilde{T_1}$ prove the result. ■

This theorem does not help to prove Hindman's Theorem, it only helps to understand the notion of full-match, and why we build a solution by picking elements from full-matches.

We have the same kind of result for right-matches, suggesting that right-matches will be helpful to build full-matches:

▶ **Proposition III.10.** *Assume that every coloring has a full-match inside any IP set. If (F, S) is a right-match then for every $T_0 \subseteq_{\mathrm{FS}} S$, there exists a finite set G and an infinite set T_1 such that $(F+G, T_1)$ is a full-match and $(F+G) \cup T_1 \subseteq_{\mathrm{FS}} T_0$.*

In other word, no matter how we shrink S, there will always be a full-match whose finite set contains only elements that are obtained by a finite sum using an element of F.

In order to better understand, let S be such that every $x \in \mathrm{FS}(S)$ is the result of a unique sum (for instance $\{2^i : i \in \mathbb{N}\}$ is such a set). Then, if (G, S) is a right-match, every $b \in G$ has some $a \in F$ in its unique sum decomposition.

Proof. Suppose (F, S) is a right-match and $T_0 \subseteq_{\mathrm{FS}} S$. Then, let c_0 be the coloring $k \mapsto \langle c(k), a\rangle$ with $a \in F$ such that $c(k) = c(a+k)$. By the existence of full-matches inside any IP set, let (G_0, T_1) be a full-match with $G_0 \cup T_1 \subseteq_{\mathrm{FS}} T_0$. Then, let $x \in \mathrm{FS}(T_1)$, there must exist $b \in G_0$ such that $c_0(b) = c_0(x) = c_0(x+b)$, that is there exists $a \in F$ such that $c(b) = c(b+a)$ and $c(x) = c(x+a)$ and $c(x+b) = c(x+b+a)$. We conclude that $c(a+b) = c(x) = c(a+b+x)$, and $(F+G, T_1)$ is a full-match as well. ■

▶ **Corollary III.10.1.** *If (F, S) is a right-match then for every $T_0 \subseteq_{\mathrm{FS}} S \cup F$, there exists an infinite set $T_1 \subseteq_{\mathrm{FS}} T_0$ and an element $x \in T_1$ such that $x \in F + \mathrm{FS}(S)$.*

In the "more effective" proof of Hindman, we will be interested only in right-matches and full-matches, which were called half-matches and full-matches in [64]. Therefore we do not give a suggestion of how left-matches can be used, but they will be seen to be of great importance in the reverse mathematical study of Hindman's Theorem, in section III.3.3.

The first step: constructing right-matches

We build right-matches by first explaining how to build one for some particular color, and then iterate the construction for every color.

Theorem III.11 ([43, 64])

Let $S \subseteq \mathbb{N}$ be any infinite set and c a finite coloring. Then, there exists a right-match (F, T) with $F \cup T \subseteq_{\mathrm{FS}} S$.

Proof. We first prove a restricted version of the result to a specific color.

▶ **Lemma III.12.** *Let $S \subseteq \mathbb{N}$ be any infinite set, c a coloring and i a color. Then, there exists a right-match (F, T) for color i with $F \cup T \subseteq_{\mathrm{FS}} S$.*

Proof. We will try to find a right-match (F, T) by adding into F one by one the elements of T that witness that (F, T) is not a right-match. If we never find a right-match, then we will be able to build one from the witnesses of our failure at each step.

More formally, we build a sequence of integers $(a_j)_{j\in\mathbb{N}}$. We write S_j for $S \cap (a_{j-1}, \infty)$ (and $S_0 = S$) and $F_j = \{a_k : k < j\}$. We will ensure that $a_j \in \mathrm{FS}(S_j)$, so that $\mathrm{FS}(\{a_j : j \in \mathbb{N}\}) \subseteq \mathrm{FS}(S)$.

If $(a_k)_{k<j}$ has been constructed, we split into two cases:

1. If $(\mathrm{FS}(F_j), S_j)$ is a right-match for color i, then we are done.
2. Otherwise, there must exist some $\hat{a}_j \in \mathrm{FS}(S_j)$ such that $c(\hat{a}_j) = i \neq c(a + \hat{a}_j)$ for all $a \in \mathrm{FS}(F_j)$. The smallest such defines a_j.

If at some point the construction stops, then we have found a right-match for color i. Otherwise, $(a_j)_{j\in\mathbb{N}}$ is defined, and by construction it is clear that $(\mathrm{FS}(\{a_j : j \in \mathbb{N}\}) \setminus \{a_j : j \in \mathbb{N}\}) \cap C_i = \emptyset$. Therefore, $T = \{a_{2j} + a_{2j+1} : j \in \mathbb{N}\}$ is such that $\mathrm{FS}(S) \cap C_i = \emptyset$. So $(\emptyset, T)$ is a right-match for color i. ■

We iterate the use of the lemma for every color. Let c be a k-coloring of the integers. We define (F_i, S_i) for $i \in [-1, k-1]$, starting with $F_{-1} = \emptyset$ and $S_{-1} = A$, and getting (F_i, S_i) by applying Lemma III.12 to S_{i-1} and color i.

We have $A \supseteq_{\mathrm{FS}} S_0 \supseteq_{\mathrm{FS}} \cdots \supseteq_{\mathrm{FS}} S_{k-1}$, and $(\bigcup_{i<k} F_i, S_{k-1})$ is a half-match: Let $x \in \mathrm{FS}(S_{k-1})$ and $i = c(x)$, then as $x \in \mathrm{FS}(S_i)$ there exists $a \in F_i$ such that $c(x) = c(a + x)$. ■

The second step: constructing full-matches

The goal of the second step is to iterate the construction of right-matches to get a full-match. Remember from Proposition III.10 that given a right-match, we can safely assume that one should sum elements from it to obtain a full-match.

Theorem III.13 ([43, 64])

Let $S \subseteq \mathbb{N}$ be any infinite set and c a finite coloring. Then, there exists a full-match (F, T) with $F \cup T \subseteq_{\mathrm{FS}} S$.

Proof. Again, we start with the color-restricted version of the theorem.

▶ **Lemma III.14.** *Let $S \subseteq \mathbb{N}$ be any infinite set, c a finite coloring and i a color. Then, there exists a full-match (F, T) with $F \cup T \subseteq_{\mathrm{FS}} S$.*

Proof. In the proof of Lemma III.12, we created a sequence of couples (F_j, S_j), each corresponding to a potential right-match. If none worked, we were able to build a right-match from our failure. We will prove this lemma the same way, trying to build potential full-matches. If none work, we will be able to conclude by extracting a full-match from the previous failure.

We simultaneously build a sequence of couple $(F_j, S_j)_{j\in\mathbb{N}}$ and a sequence of coloring $(c_j)_{j\in\mathbb{N}\cup\{-1\}}$ such that

1. c_j is a coloring of $\mathrm{FS}(S_j)$,
2. (F_{j+1}, S_{j+1}) is a right-match for c_j,
3. $F_{j+1} \cup S_{j+1} \subseteq_{\mathrm{FS}} S_j$
4. c_{j+1} is a refinement of c_j (in the sense that $c_{j+1}(a) = c_{j+1}(b)$ implies $c_j(a) = c_j(b)$).

Start with $c_{-1} = c$ and $S_{-1} = S$. Now, suppose that c_k has been defined for every $k < j$. We use Theorem III.11 to define (F_j, S_j), a right-match for c_{j-1}, with $F_j \cup S_j \subseteq_{\mathrm{FS}} S_{j-1}$. We now split into two cases:

— If $(\bigcup_{j' \leq j} F_{j'}, S_j)$ is a full-match for color i for c, then we are done.

— Otherwise, define c_j to be $x \mapsto \langle c_{j-1}(x), a\rangle$, where $a \in F_j$ is the smallest such that $c_j(x) = c_j(a + x)$.

If at some point the construction stops, then we have found a full-match for color i for c. Otherwise, before going further, we explain what is a color for c_j. Writing $\langle\!\langle t_0, t_1, \ldots, t_n\rangle\!\rangle$ as a shortcut for $\langle \cdots \langle t_0, t_1\rangle, t_2\rangle \cdots \rangle, t_n\rangle$, if $x \in \mathrm{FS}(S_j)$, we have $c_j(x) = \langle\!\langle a_j, a_{j-1}, \cdots, a_0, i\rangle\!\rangle$; where $c(x) = i$ and $a_k \in F_k$ is the smallest such that $c_{k-1}(x) = c_{k-1}(a_k + x)$. In other words, $c_j(x)$ is the list for $k \leq j$ of the smallest witness of F_k being a half-match for c_{k-1} at x, plus the original color. In the following, we write $a_k(x)$ for the integer such that $c_k(x) = \langle\!\langle a_k(x), \cdots, a_0(x), i\rangle\!\rangle$.

We argue that for $x \in \mathrm{FS}(S_j)$, the set $x + \mathrm{FS}(\{a_k(x) : k \leq j\})$ is monochromatic of color $c(x)$: We proceed by induction on the length of the sum. For sums of length 0, the result is trivial. Now suppose the result is true for all $x \in \mathrm{FS}(S_j)$ and k long sums, fix $x \in \mathrm{FS}(S_j)$, we write a_j for $a_j(x)$ to shorten notations. Let $a = a_{j_1} + \cdots + a_{j_k}$ with $j_0 < \cdots < j_k$. By definition of a_{j_k}, we have $c_{j_k-1}(a_{j_k}+x) = c_{j_k-1}(x) = \langle\!\langle a_{j_k-1}, \cdots, a_0, i\rangle\!\rangle$, so $a_{k'}(a_{j_k} + x) = a_{k'}$ for $k' < k$. By these equalities, and the induction hypothesis on $a_{j_k} + x$ and the k long sum $a_0 + \cdots + a_{j_{k-1}}$, we have $c(a_0 + \cdots + a_{j_{k-1}} + (a_{j_k} + x)) = c(a_{j_k} + x)$. But $c(a_{j_k} + x) = c(x)$, so we have the result on the $k + 1$ long sum.

As the construction never stopped, we know that at each level j there is some $x \in S_j$ of color i witness of $(\bigcup_{j' \leq j} F_{j'}, S_j)$ not being a full-match for c. Let $(a_{j'})_{j' \leq j}$ with $c_j(x) = \langle\!\langle a_j, \cdots, j_0, i\rangle\!\rangle$, we know that $x + \mathrm{FS}(\{a_{j'} : j' \leq j\})$ is monochromatic of color $c(x) = i$, therefore there cannot be an element of $\mathrm{FS}(\{a_{j'} : j' \leq j\})$ of color i: if such element a existed, we would have $c(a) = c(x) = c(a + x)$, a contradiction with our choice of x.

We then know that for each level j, there exists a sequence $(a^j_{j'})_{j' \leq j}$ with $a^j_{j'} \in F_{j'}$ and $\mathrm{FS}(\{a^j_{j'} : j' < j\}) \cap C_i = \emptyset$. As every F_j is finite, by compactness there exists an infinite sequence $(a_j)_{j \in \mathbb{N}}$ with $\mathrm{FS}(\{a_j : j \in \mathbb{N}\}) \cap C_i = \emptyset$. But then, $(\emptyset, \{a_j : j \in \mathbb{N}\})$ is a full-match for color i. ■

We iterate the use of the lemma for every color. Let c be a k-coloring of the integers. We define (F_i, S_i) for $i \in [-1, k-1]$, starting with $F_{-1} = \emptyset$ and $S_{-1} = S$, and getting (F_i, S_i) by applying Lemma III.14 to S_{i-1} and color i.

We have $S \supseteq_{\mathrm{FS}} S_0 \supseteq_{\mathrm{FS}} \cdots \supseteq_{\mathrm{FS}} S_{k-1}$, and $(\bigcup_{i<k} F_i, S_{k-1})$ is a full-match: Let $x \in \mathrm{FS}(S_{k-1})$ and $i = c(x)$, then as $x \in \mathrm{FS}(S_i)$ there exists $a \in F_i$ such that $c(x) = c(a + x)$. ■

The third step: constructing monochromatic sets

Once again, the key to build a solution for a coloring (a monochromatic IP set) is to iterate Theorem III.13.

Theorem III.15 ([64])

Let $S \subseteq \mathbb{N}$ be any infinite set and c a finite coloring. Then, there exists an infinite set $T \subseteq_{\mathrm{FS}} S$ with $\mathrm{FS}(T)$ monochromatic.

Proof. We build sequences $(F_j, S_j)_{j\in\mathbb{N}}$ and $(c_j)_{j\in\mathbb{N}\cup\{-1\}}$ such that

1. c_j is a coloring of $\mathrm{FS}(S_j)$
2. (F_{j+1}, S_{j+1}) is a full-match for c_j
3. $F_{j+1} \cup S_{j+1} \subseteq_{\mathrm{FS}} S_j$,
4. c_{j+1} is a refinement of c_j.

We start with $S_{-1} = S$ and $c_{-1} = c$, and at each step j, (F_j, S_j) are given by Theorem III.13 with S_{j-1} and c_{j-1}. The coloring c_j is defined by $c_j(x) = \langle c_{j-1}(x), a\rangle$ where $a \in F_j$ is the smallest such that $c_{j-1}(a) = c_{j-1}(x) = c_{j-1}(a + x)$.

Just as in the proof of Lemma III.14, we make the coloring c_j more explicit: for any $x \in \mathrm{FS}(S_j)$, we have $c_j(x) = \langle\langle a_j, \cdots, a_0, i\rangle\rangle$ with $c(x) = i$, $a_k \in F_k$ and $c_{k-1}(a_k) = c_{k-1}(x) = c_{k-1}(a_k + x)$ for any $k \leq j$. We use the same notation $a_j(x)$ as in the proof of Lemma III.14.

This time we claim that for any $x \in \mathrm{FS}(S_j)$, the set $\mathrm{FS}(\{a_k(x) : k \leq j\} \cup \{x\})$ is monochromatic. We prove it by induction on j. We already have that $x + \mathrm{FS}(\{a_k(x) : k \leq j\})$ is monochromatic of color $c(x)$ by the proof of Lemma III.14. As $c_{j-1}(x) = c_{j-1}(a_j(x))$, we have $a_k(x) = a_k(a_j)$, so by this equality and the induction hypothesis on $a_j(x) \in \mathrm{FS}(S_{j-1})$, we have $\mathrm{FS}(\{a_k(x) : k < j\} \cup \{a_j\})$ is monochromatic of color $c(a_j) = c(x)$. Combining al together, we get that $\mathrm{FS}(\{a_k(x) : k < j\} \cup \{a_j\} \cup \{x\})$, concluding the induction.

For any level l, there is a sequence $(a_k)_{k\leq l}$ such that $\mathrm{FS}(\{a_k : k \leq l\})$ is monochromatic, with $a_k \in F_k$ and F_k finite. Therefore, by compactness there must exist an infinite sequence $(a_k)_{k\in\mathbb{N}}$ with $\mathrm{FS}(\{a_k : k \in \mathbb{N}\})$ monochromatic, and $\{a_k : k \in \mathbb{N}\} \subseteq_{\mathrm{FS}} S$. ■

III.2.2 The less effective proof

The "less effective" proof is due to Baumgartner [6]. It appeared in 1974 just after Hindman's original proof, and it is an attempt to make it simpler, in the sense of being shorter and easier to understand.

In the previous proof we make some guess on what will be the elements of the solution, but still consider integers that won't be in the final solution. For instance, in the construction of a solution in the proof of Theorem III.15, we have some kind of tree made by full-matches. In the end, a path in the tree will consist of a solution, therefore it will take exactly one element of the finite set of each full-match. All the work on the other potential path is not useful. The same is valid for the construction of full-matches: the finite set of a right-match consists of a finite set of candidate to sum with others for the future full-match.

In Baumgartner's proof, the idea is to "guess" in advance an integer of a full-match that can be part of a solution, and shrink the reservoir so that is can be part of every solution. We know that if Hindman's Theorem is true there must exist one by Proposition III.9, although we cannot use this proposition in proving Hindman's Theorem. Instead, we use a largeness notion.

Definition III.16 ([6])

If $C \subseteq \mathbb{N}$ and $S \subseteq \mathbb{N}$ is infinite, then C *is large for* S if for any infinite set $T \subseteq_{\mathrm{FS}} S$, $\mathrm{FS}(T) \cap C \neq \emptyset$.
If $c : X \to k$ is a coloring, we say that a color i is large for S if $C_i = \{x \in X : c(x) = i\}$ is large for S.

In other word, a color is large for an IP set if one cannot avoid it, while staying in the IP set. Therefore, it has to be the only possible color of a solution inside the IP set.

We first show that this does correspond to a kind of largeness notion:

▶ **Proposition III.17** ([6]). *If C is large for S and $C = C_0 \cup C_1$, then there is an infinite $T \subseteq_{\mathrm{FS}} S$ such that either C_0 or C_1 is large for T.*

Proof. Suppose not. Then, C_0 is not large for S, so let $T_0 \subseteq_{\mathrm{FS}} S$ such that $\mathrm{FS}(T_0) \cap C_0 = \emptyset$. Applying the contraposition of the proposition on the C_1 side with T_0, we have that C_1 is not large for T_0, so there exists $T_1 \subseteq_{\mathrm{FS}} T$ with $\mathrm{FS}(T_1) \cap C_1 = \emptyset$.

But then, $T_1 \subseteq_{\mathrm{FS}} S$ and $\mathrm{FS}(T_1) \cap (C_0 \cup C_1) = \emptyset$, contradicting the largeness of C for S. ■

▶ **Corollary III.17.1.** *If S is any infinite set and $c : \mathbb{N} \to k$ is a coloring, then there exists a large color $i < k$ for some $T \subseteq_{\mathrm{FS}} S$.*

Proof. By repeated application of Proposition III.17, using the fact that $\mathbb{N}$ is large for any set. ■

Corollary III.17.1 says that for every coloring of the integers, there exists an infinite set on which a color is large. Therefore, it suffices to show Hindman's Theorem for a large set, that is: For any $C, S \subseteq \mathbb{N}$ such that C is large for S, there exists $T \subseteq_{\mathrm{FS}} S$ such that $\mathrm{FS}(T) \subseteq C$. Note that starting from an S such that color i is large for S, we can only build solutions for color i.

The fact that there always exists a large color allows us to know in which direction to do the construction. The part on the construction of a right-match is exactly the same, however this time we are able to pick one element of the finite part of the right-match that we know can be extended to a full-match.

Theorem III.18 ([6])

Suppose c is a coloring and i is a large color for S. Then, there exists $a \in \mathrm{FS}(S)$ and $T \subseteq_{\mathrm{FS}} S$ such that $\{x \in \mathbb{N} : c(x) = i = c(a + x)\}$ is large for T.

Proof. By Lemma III.12 with color i, there exists (F, T_0) a right-match for color i such that $F \cup T_0 \subseteq_{\mathrm{FS}} S$. For any $x \in \mathrm{FS}(T_0)$ of color i, there exists some $a \in F$ such that $a + x \in C_i$. Therefore, we are able to define a coloring c_1 such that $c_1(x)$ is the first $a \in F$ such that $c(x) = i = c(a + x)$ if $x \in C_i \cap \mathrm{FS}(T_0)$, and $c_1(x) = -1$ otherwise. It is clear that color -1 cannot be large in any $T_1 \subseteq_{\mathrm{FS}} T_0$, so by Corollary III.17.1 there must exist some $a \in F \subseteq_{\mathrm{FS}} S$ and $T_1 \subseteq_{\mathrm{FS}} T_0$ with color a for c_1 being large, that is: $\{x \in \mathbb{N} : c(x) = i = c(a + x)\}$ is large for T_1, concluding the proof. ■

In the statement of the previous theorem, the set $\{x \in \mathbb{N} : c(x) = i = c(a + x)\}$ has to be considered as a new large color, for the refinement c_1 of c, so that we can iterate. The following theorem is to be considered as the choice of a particular element of a full-match. However, the several elements in the finite set of full-matches in Theorem III.13 comes from the several elements in the finite set of right-matches. As in Theorem III.18 we got a right-match whose finite set is a singleton, we won't need the use of Proposition III.17 to also get a full-match whose finite set is a singleton.

Theorem III.19 ([6])

Suppose c is a coloring and i is a large color for S. Then, there exists $a \in \mathrm{FS}(S)$ with $c(a) = i$ and $T \subseteq_{\mathrm{FS}} S$ such that $\{x \in \mathbb{N} : c(a) = c(x) = i = c(a + x)\}$ is large for T.

Proof. We build a sequence $(a_n)_{n\in\mathbb{N}}$ and $(S_n, C_n)_{n\in\mathbb{N}\cup\{-1\}}$ such that

1. $S_{n+1} \subseteq_{\mathrm{FS}} S_n$,
2. $a_{n+1} \in \mathrm{FS}(S_n)$,
3. $C_{n+1} = \{x \in \mathbb{N} : x \in C_n \text{ and } a_{n+1} + x \in C_n\}$ large for S_{n+1}.

Start with $S_{-1} = S$ and $C_{-1} = \{x \in \mathbb{N} : c(x) = i\}$. Given S_n and C_n we use Theorem III.18 with c_n being the characteristic function of C_n, and large color i, to define a_{n+1}, S_{n+1} and $C_{n+1} = \{x \in \mathbb{N} : x \in C_n \text{ and } a_{n+1} + x \in C_n\}$ large for S_{n+1}.

We will prove by induction that if $a \in \mathrm{FS}(\{a_k : k \leq j\})$ then $x \in C_j$ implies $c(x) = c(x+a) = i$ (this is the same argument that we have done several times now). Suppose $a = a_{j_0} + \cdots + a_{j_k}$ with $j_0 < \cdots < j_k$. Then $a_{j_k} + x \in C_{j_k - 1} \subseteq C_{j_{k-1}}$. By induction with $a_{j_k} + x$ and the sequence $a_{j_0} + \cdots + a_{j_{k-1}}$, we have $i = c(a_{j_0} + \cdots + a_{j_{k-1}} + (a_{j_k} + x)) = c(a_{j_k} + x) = c(x)$, concluding the induction.

As color i is large for S we know that $\mathrm{FS}(\{a_n : n \in \mathbb{N}\})$ cannot avoid color i, so there must be some $a = a_{j_0} + \cdots + a_{j_k}$ such that $c(a) = i$. By the prior remark, we also have $c(x) = c(a + x) = i = c(a)$ for any $x \in \mathrm{FS}(S_{j_k})$, concluding the proof of the theorem. ■

It is now possible to iterate Theorem III.19 to get a solution to Hindman's Theorem.

Theorem III.20 ([6])

Suppose c is a coloring on X and i is a large color for S. Then, there exists $T \subseteq_{\mathrm{FS}} S$ such that $\mathrm{FS}(T)$ is monochromatic of color i.

Proof. Iterating Theorem III.19, we are now able to build a sequence (a_n, S_n, C_n) with the additional property that $a_n \in C_{n-1}$:

1. $S_{n+1} \subseteq_{\mathrm{FS}} S_n$,
2. $a_{n+1} \in \mathrm{FS}(S_n)$,
3. $C_{n+1} = \{x \in \mathbb{N} : x \in C_n \text{ and } a_{n+1} + x \in C_n\}$ large for S_{n+1}.
4. $a_{n+1} \in C_n$.

We start with $S_{-1} = S$ and $C_{-1} = \{x \in \mathbb{N} : c(x) = i\}$. By the proof of Theorem III.19, we have that if $a \in \mathrm{FS}(\{a_k : k \leq j\})$ then $x \in C_j$ implies $c(x) = c(x + a) = i$. Let $a = a_{j_0} + \cdots + a_{j_n}$, then as $a_{j_n} \in C_{j_{n-1}}$, $c(a) = i$. Take $T = \{a_n : n \in \mathbb{N}\} \subseteq_{\mathrm{FS}} S$, we have that $\mathrm{FS}(T)$ is monochromatic. ■

III.2.3 The ultrafilter proof

We include the ultrafilter proof of Hindman's Theorem in this thesis only as a curiosity. The interesting thing in that proof is that it is an example where manipulating objects of very high complexity such as ultrafilters [2] can simplify a lot the argument. The beauty of this proof is also that it seems quite magical, in the sense that we never dive into complicated combinatorial argument, yet it proves a complicated combinatorial result. More on the interaction between combinatorics and ultrafilters can be found in [44].

2. Though, there has been modification of the proof to make it fit in second order arithmetic, as in [63]

Ultrafilters

Definition III.21

A *filter* $\mathcal{F}$ on X is a subset of $\mathcal{P}(X)$ such that:

1. $X \in \mathcal{F}$ and $\emptyset \notin \mathcal{F}$,
2. If $A \in \mathcal{F}$ and $A \subseteq B \subseteq \mathbb{N}$, then $B \in \mathcal{F}$,
3. If $A, B \in \mathcal{F}$, then $A \cap B \in \mathcal{F}$.

An *ultrafilter* is a filter $\mathcal{F}$ that is maximal for inclusion, that is there is no filter $\mathcal{F}' \supsetneq \mathcal{F}$. Write $\mathbb{U}(X)$ for the set of ultrafilters on X.

In informal ways, a filter measures the largeness of a set. The total set X is large, $\emptyset$ is not, a superset of a large set is large and the intersection of two large sets is large. This extends many other notions of largeness such as co-null sets, co-meager sets, in areas where they would not work such as $\mathbb{N}$.

The ultrafilters correspond to the filters where the union of two small sets cannot be large. In some sense, it can be seen as a notion of largeness where each set is either small or large, compared to filters where a set can be neither small nor large.

▶ **Proposition III.22.** *A filter $\mathcal{F}$ is an ultrafilter on an arbitrary set X if and only if for every $A \subseteq X$, either A or $X \setminus A$ is in $\mathcal{F}$.*

Proof. Suppose $\mathcal{F}$ is a filter, and there exists $A \cup (X \setminus A) = X \in \mathcal{F}$, but none of them is in $\mathcal{F}$. We can easily verify that $\mathcal{F} \cup \{\widetilde{A} \cap B : \widetilde{A} \supseteq A \text{ and } B \in \mathcal{F}\}$ is a filter that strictly contains $\mathcal{F}$, so $\mathcal{F}$ is not an ultrafilter.

Now suppose $\mathcal{F}$ is not an ultrafilter. Let $\mathcal{F}' \supseteq \mathcal{F}$ be a filter and $A \in \mathcal{F}' \setminus \mathcal{F}$. Then, $X \setminus A \notin \mathcal{F}$ otherwise $\emptyset \in \mathcal{F}'$. ■

We continue with some general background on ultrafilters. Using the product topology, we get a topology on the families of sets generated by the basic open sets $\{\mathcal{F} \subseteq \mathcal{P}(X) : A \in \mathcal{F}\}$ and $\{\mathcal{F} \subseteq \mathcal{P}(X) : A \notin \mathcal{F}\}$ for any $A \subseteq X$.

▶ **Proposition III.23.** *The set $\mathbb{U}(X) \subseteq 2^{2^X}$ is compact for the product topology on $2^{2^X} = \prod_{x \in 2^X} 2$.*

Proof. The set $\prod_{x \in 2^X} 2$ is compact by Tychonoff's Theorem. Moreover, by the axioms of Definition III.21 its subset $\mathbb{U}(X)$ is closed, and therefore compact. ■

Ultrafilters and semi-groups

We define an operation on $\mathbb{U}(\mathbb{N})$.

Definition III.24

If $\mathcal{U}, \mathcal{V} \in \mathbb{U}(\mathbb{N})$, then we define $\mathcal{U} \oplus \mathcal{V}$ by

$$A \in \mathcal{U} \oplus \mathcal{V} \quad \text{iff} \quad \{n : A - n \in \mathcal{V}\} \in \mathcal{U}$$

where $A - n = \{m \in \mathbb{N} : m + n \in A\}$.

Not only this operation is internal in $\mathbb{U}(\mathbb{N})$, but it also defines a semi-group.

▶ **Proposition III.25.** $(\mathbb{U}(\mathbb{N}), \oplus)$ *is a semi-group with* $\oplus$ *being right-continuous.*

Proof. We first prove that the operation is internal, that is, for any $\mathcal{U}, \mathcal{V} \in \mathbb{U}(\mathbb{N})$, $\mathcal{U} \oplus \mathcal{V}$ is an ultrafilter.

For any set $A \subseteq \mathbb{N}$, write $A^\star = \{n \in \mathbb{N} : A - n \in \mathcal{V}\}$. It is clear that $\emptyset \notin \mathcal{U} \oplus \mathcal{V}$ as $\emptyset^\star = \emptyset$, and that $\mathbb{N} \in \mathcal{U} \oplus \mathcal{V}$ as $\mathbb{N}^\star = \mathbb{N}$.

Let $A \in \mathcal{U} \oplus \mathcal{V}$, and $A \subseteq B \subseteq \mathbb{N}$. For every n we have $A - n \subseteq B - n$ therefore $A^\star \subseteq B^\star$. It follows that $B \in \mathcal{U} \oplus \mathcal{V}$.

Let $A, B \in \mathcal{U} \oplus \mathcal{V}$. If $n \in A^\star \cap B^\star$, then both $A - n$ and $B - n$ are in $\mathcal{V}$, therefore $A - n \cap B - n = (A \cap B) - n \in \mathcal{V}$ and $n \in (A \cap B)^\star$. In other words, $A^\star \cap B^\star \subseteq (A \cap B)^\star$. As $A^\star, B^\star \in \mathcal{U}$, we have $(A \cap B)^\star \in \mathcal{U}$ and $A \cap B \in \mathcal{U} \oplus \mathcal{V}$.

We proved that $\mathcal{U} \oplus \mathcal{V}$ is a filter, using only the fact that $\mathcal{U}$ and $\mathcal{V}$ are filters. It remains to prove the "ultrafilter" part, using the characterization of Proposition III.22. Let $A \subseteq \mathbb{N}$ be any set. For any n, $A - n$ and $B - n$ are disjoint union, so exactly one of them is in $\mathcal{V}$. Therefore, $A^\star \cup (\mathbb{N} \setminus A)^\star = \mathbb{N}$ and one of them must be in $\mathcal{U}$, in other word either A or $\mathbb{N} \setminus A$ is in $\mathcal{U} \oplus \mathcal{V}$.

The associativity is clear as

$$\begin{aligned} A \in \mathcal{U} \oplus (\mathcal{V} \oplus \mathcal{W}) &\iff \{n : A - n \in (\mathcal{V} \oplus \mathcal{W})\} \in \mathcal{U} \\ &\iff \{n : \{m : A - n - m \in \mathcal{W}\} \in \mathcal{V}\} \in \mathcal{U} \\ A \in (\mathcal{U} \oplus \mathcal{V}) \oplus \mathcal{W} &\iff \{m : A - m \in (\mathcal{W} \oplus \mathcal{W})\} \in (\mathcal{U} \oplus \mathcal{V}). \end{aligned}$$

It only remains to prove the right-continuity. Let $\mathcal{V} \in \mathbb{U}(\mathbb{N})$ and $A \subseteq \mathbb{N}$, for any $\mathcal{U} \in \mathbb{U}(\mathbb{N})$ we have $A \in \mathcal{U} \oplus \mathcal{V}$ iff $\{n : A - n \in \mathcal{V}\} \in \mathcal{U}$ iff $\mathcal{U} \in \{\mathcal{W} : \{n : A - n \in \mathcal{V}\} \in \mathcal{W}\}$ a basic clopen set. ■

The operation $\oplus$, seen as a largeness notion, say that a set A is large if there is a $\mathcal{U}$-large set of elements x such that there is a $\mathcal{V}$-large set of elements y such that $x + y \in A$. In other words, to get an element in A as a sum of two elements, we have at least a $\mathcal{U}$-large choice for the first and a $\mathcal{V}$-large choice for the second. Given the fact that we are interested in finding sums that end in some particular color, it is clear that this operation can be useful. However, we need to deal with sums of arbitrary length.

Theorem III.26

There exists an ultrafilter $\mathcal{U} \in \mathbb{U}(\mathbb{N})$ such that

$$\mathcal{U} \oplus \mathcal{U} = \mathcal{U}.$$

Proof. This is just an application of the Idempotent Theorem III.27, using Proposition III.25. ■

Theorem III.27 (The Idempotent Theorem [38])

Let E be a compact semi-group for which right-multiplication ψ_g is continuous for all $g \in E$. Then, there exists $g \in E$ such that $g^2 = g$.

Proof. We do a construction by induction. Start with $E_0 = E$. If E_α is defined, let $g \in E_\alpha$. If $g \notin E_\alpha g = \{fg : f \in E_\alpha\}$, then we take this set as $E_{\alpha+1}$ as by right-continuity it is compact. Otherwise, $B = \{f \in E_\alpha : fg = g\}$ is non empty. If $g \in B$, then $g^2 = g$ and we have

found our idempotent element, we set $E_\beta = E_\alpha$ for every $\beta > \alpha$. Otherwise, we let $E_{\alpha+1} = B$ as it is a semi-group (if $f_0, f_1 \in B$ then $f_0 f_1 g = f_0 g = g$) and compact by right-continuity.

If E_β has been defined for every $\beta < \alpha$, then we define $E_\alpha = \bigcap_{\beta<\alpha} E_\beta$. This set cannot be empty as every E_β is compact.

Let α be a stage where $E_\alpha = E_{\alpha+1}$, which must happen by cardinality reasons. We must have found a g such that $g^2 = g$, as the only way it can happen in the construction is if we find an idempotent g. ■

In [38] (Section 6.1, Theorem 9), the authors give a version of the proof by just taking a smallest non-empty compact subgroup which must exist by Zorn's Lemma. We chose to present the proof this way as it gives an example of a transfinite recursion with no bound known in advance (as discussed in section IV.5.1, "the unbounded construction").

Proving Hindman's Theorem with ultrafilters

In the previous results about ultrafilters, we did not do anything especially related to Hindman's Theorem, except for the definition of $\oplus$ in Definition III.24 and for Proposition III.25, whose proof is just a verification of the definition. Moreover, nothing was combinatorially complex. Still, we will be able to derive Hindman's Theorem as a corollary in a very simple manner.

Theorem III.28 ([27])

For every coloring of the integers, there exists an infinite set A such that $\mathrm{FS}(A)$ is monochromatic.

Proof. By Theorem III.26, let $\mathcal{U}$ be an ultrafilter such that $\mathcal{U} = \mathcal{U} \oplus \mathcal{U}$. For any set $A \subseteq \mathbb{N}$, write $A^\star = \{n : A - n \in \mathcal{U}\}$. We have $A \in \mathcal{U}$ iff $A^\star \in \mathcal{U}$. Also let i be a color such that $\{a : c(a) = i\} \in \mathcal{U}$, by repetitive application of Proposition III.22.

We construct a sequence of sets $C_n \in \mathcal{U}$ and $a_n \in C_n \cap C_n^\star$. We start with $C_0 = \{a : c(a) = i\}$, and $a_0 \in C_0 \cap C_0^\star$ (there exists such an a_0 as $C_0 \cap C_0^\star \in \mathcal{U}$). Given $C_n \in \mathcal{U}$ and a_n such that $C_n - a_n \in \mathcal{U}$, let $C_{n+1} = C_n \cap (C_n - a_n) \setminus \{a_n\} \in \mathcal{U}$, a_{n+1} any element of $C_{n+1} \cap C_{n+1}^\star$, which exists as $C_{n+1} \cap C_{n+1}^\star \in \mathcal{U}$.

We prove by induction that $C_n \subseteq C_0 \cap C_0 - \mathrm{FS}(\{a_i : i < n\})$. It is clear for $n = 0$. If it is true for n, then

$$\begin{aligned} C_{n+1} &= C_n \cap C_n - a_n \setminus \{a_n\} \\ &\subseteq C_0 \cap C_0 - \mathrm{FS}(\{a_i : i < n\}) \cap C_0 - a_n \cap C_0 - a_n - \mathrm{FS}(\{a_i : i < n\}) \\ &\subseteq C_0 \cap C_0 - \mathrm{FS}(\{a_i : i < n+1\}). \end{aligned}$$

As $a_n \in C_n$, we get the result. ■

Picking an element in C_n ensures that the finite sums stay in C_0, picking it also in $C_n^\star$ ensures that we can continue the construction. The ultrafilters take care of everything else.

III.3 The Reverse Mathematics of HT

We are now pretty convinced that Hindman's Theorem is true, as we have three proofs of it. So we can start its reverse mathematical study. We will begin with the analysis of the axiomatic systems used in the "most effective" proof, to get an upper bounds on its strength. More precisely,

we will benefit from the structure of the proof in "steps", to give an upper bound on the axioms needed to build the intermediate combinatorial objects.

The other proofs take place in more powerful axiomatic systems, in fact the ultrafilter proof does not fit as is into second-order arithmetic. It is still possible to modify it a bit to place the argument in second order arithmetic [63]. We won't study these proofs in terms of reverse mathematics.

After giving the upper bounds, the natural thing is to give the lower bounds. Again, one would wish to give lower bounds also to the intermediate combinatorial objects. However, we will only be able to give lower bounds to right-matches, left-matches of 2-colors and solution to Hindman's Theorem. It is not yet known whether there always exists computable full-matches, or not.

The way we give a lower bound is by using combinatorics and computability, instead of proof-theory. We exhibit a coloring such that every solution to this coloring computes a particular set, such as the halting problem of the coloring.

The main contribution in this section is the reverse mathematical study of full-matches and left-matches. In particular, we show that there always exists computable left-matches for 2 colors, invalidating a proof from [64]. We are also able to prove the existence of computable full-matches in many particular cases. Finding computable full-matches for every coloring would result in an equivalence between HT and ACA_0, solving the most important open question related to Hindman's Theorem.

All of this section is joint work with Ludovic Patey and Benoît Monin.

III.3.1 Upper bound

Recall that the formal system ACA_0, which asserts the existence of every arithmetical set, is equivalent to the existence of the jump of any set. Indeed, we are then able to iterate the jump up to any finite (in the meta-theory) level. However, we are not able to conduct a construction that requires an unbounded amount of jumps. We will see that the "more effective" proof of Hindman's theorem contains such a construction, as therefore does not hold in ACA_0. We need to define an axiomatic system under which those constructions can happen.

Definition III.29 (ACA_0^+)

We write ACA_0^+ for the axiomatic system that consists of RCA_0 and:

$$\forall A \subseteq \mathbb{N}, \exists Y \text{ s.t } Y = A^{(\omega)}.$$

We first exhibit the axiomatic system in which the proof of Lemma III.12 and therefore Theorem III.11 takes place.

Upper bounds on right-matches

We prove that right-matches do not need more than the base theory of reverse mathematics: there always exists computable right-matches. However, the computation is not computably uniform in the coloring.

Theorem III.30

Given a finite coloring $c : \mathbb{N} \to k$, an infinite set S and a color $i < k$,

1. Either there exists (F, T), a right-match for color i computable in (c, S), with $T \subseteq_{\mathrm{FS}}$

S

2. Or, there exists $T \subseteq_{\mathrm{FS}} S$ computable in (c, S) with $\mathrm{FS}(T) \cap C_i = \emptyset$.

The disjunction is decidable uniformly in $(c, S)''$.

Proof. We consider the construction in the proof of Lemma III.12. It is split depending on the following:

$$\exists n, (\mathrm{FS}(S \cap [0; n]), S \cap (n; \infty)) \text{ is a right-match for color } i \qquad (\mathrm{RMP}_i)$$

If (RMP_i) is true, which is decidable in $(c, S)''$, then one can easily find, using $(c, S)''$ again, a code for a (c, S)-computable right-match for color i.

Otherwise, the construction in Lemma III.12 will never end. This is a computable construction of an infinite set $T \subseteq_{\mathrm{FS}} S$ with $\mathrm{FS}(T) \cap C_i = \emptyset$. ■

▶ **Corollary III.30.1.** *Given a coloring c and an infinite set S, there exists a right-match (F, T) with $T \subseteq_{\mathrm{FS}} S$ that is computable in (c, S) uniformly in $(c, S)''$.*

▶ **Corollary III.30.2.** *Theorem III.11 is provable in* RCA_0.

Upper bounds on full-matches

We continue with an upper bound on the construction of full-matches. This time, because of the non-uniformity of the construction of right-matches, the full-matches are not known to be computable. Indeed, the construction consists of an infinite iteration of a right-match construction.

Theorem III.31

Given a coloring c, an infinite set S and a color i,

1. Either there exists (F, T) a full-match for color i, computable in (c, S) with $T \subseteq_{\mathrm{FS}} S$
2. Or, there exists $T \subseteq_{\mathrm{FS}} S$ computable in $(c, S)^{(3)}$, with $\mathrm{FS}(T) \cap C_i = \emptyset$.

The disjunction is decidable uniformly in $(c, S)^{(3)}$.

Proof. We use Corollary III.30.1 to study the complexity of the proof of Theorem III.13. In the construction of the full-match, we build a succession of right-matches. Each of them is defined from the previous one. By Corollary III.30.1, each of these right-matches is computable, but for the construction to be effective it needs to be uniform, so the construction itself is computable in $(c, S)''$.

Now there are two cases: either there exists a computable full-match for color i in the sequence, in this case one can find it uniformly using $(c, S)''$; or there is no such full-match. It is decidable whether we are in the first case or not, using $(c, S)^{(3)}$. If we are in the first case an index for the computable full-match can be found in $(c, S)''$.

In the second case, the construction goes on forever and defines an infinite sequence of colorings and right-matches that are not full-matches for color i. From this construction, which is effective in $(c, S)''$, we can compute the set of witnesses that the right-matches are not full-matches for color i, and their "refined" color defines a tree, as described in the proof of Theorem III.13.

The solution is any path of the $(c, S)''$-computable tree. So, given $(c, S)^{(3)}$ we are able to uniformly compute a path on this tree, which corresponds to a solution as explained in the proof of Theorem III.13. ■

▶ **Corollary III.31.1.** *Given a coloring $c : \mathbb{N} \to k$ and an infinite set S, there exists a full-match (F, T) for c with $T \subseteq_{\mathrm{FS}} S$ and (F, T) is computable in $(c, S)^{(3(k-1))}$ uniformly.*

Proof. By iterating Theorem III.31. One cannot always be in Case 2., as otherwise we get a set avoiding all colors. ■

▶ **Corollary III.31.2.** *Theorem III.13 is provable in* ACA_0.

Upper bounds on Hindman's Theorem

With Corollary III.31.1, it is now easy to tackle the implication of the proof of Hindman's Theorem III.15. The proof consists of an iteration of a construction of full-matches, which are arithmetically defined. However, the construction of a full-match depends on the previously constructed one, so there is no bound in the arithmetic power needed by the iteration.

Theorem III.32

Given a coloring c and an infinite set S, there exists an infinite set $T \subseteq_{\mathrm{FS}} S$ with $\mathrm{FS}(T)$ monochromatic and T uniformly computable in $(c, S)^{(\omega+1)}$.
Theorem III.15 is provable in ACA_0^+.

Proof. In the proof of Theorem III.15, we built a sequence of full-matches (F_n, S_n), and by Corollary III.31.1 each of them can be taken computable in a finite amount of jumps of the previous one. This means that we need $(c, S)^{(\omega)}$ to do the construction. When the construction is achieved, it defines a tree on which we take a path using one more iteration of the jump. The path is a solution, and is computable in $(c, S)^{(\omega+1)}$. ■

III.3.2 Lower bounds

In this part, we prove lower bounds on Hindman's Theorem. We work with the finite union theorem, as it is more convenient for our purpose and equivalent to the finite sum theorem.

In practice, we will devise particular computable colorings of the finite subsets of the integers, such that any monochromatic $\mathrm{IP}_\cup$ set computes the halting set. This shows that HT implies ACA_0 as the latter is equivalent to the existence of the jump of any set.

We first define the two colorings we are interested in, as we will mention them outside of the proof of HT implies ACA_0.

Definition III.33 (Gaps in finite sets [1])

— Let $F \subseteq \mathbb{N}$ be a set, and $m < n$ be elements of F. We say that (m, n) is a *gap in* F if there is no $k \in F$ with $m < k < n$.
— Let (m, n) be a gap in F. We say that it is a *small gap for* k if

$$\emptyset'_{\restriction m}[n] \neq \emptyset'_{\restriction m}[k].$$

We say that it is a *small gap* if it is a small gap for some sufficiently large k.

— Given F finite, we denote by $\mathrm{SG}_k(F)$ the number of small gaps for k in F. In particular, we denote $\mathrm{SG}(F) = \mathrm{SG}_\infty(F)$ the number of small gaps in F, and $\mathrm{VSG}(F)$ the number $\mathrm{SG}_{\max F}(F)$, where VSG stands for "very small gaps".
— We denote by $c_{\mathrm{vsg}} : \mathcal{P}_{\mathrm{fin}}(\mathbb{N}) \to 2$ the coloring $c_{\mathrm{vsg}} : F \mapsto \mathrm{VSG}(F) \mod 2$. Note that this is a computable coloring.

Let us justify a bit the notation of small gaps. If (m, n) is a small gap and $m < p < n$, then both (p, n) and (m, p) are small gaps. Indeed, $\emptyset'_{\restriction m}[n] \neq \emptyset'_{\restriction m}$ imply both $\emptyset'_{\restriction p}[n] \neq \emptyset'_{\restriction p}$ and $\emptyset'_{\restriction m}[p] \neq \emptyset'_{\restriction m}$.

Theorem III.34 ([1])

There exists a computable coloring of the integers such that for all $A \subseteq \mathbb{N}$ with $\mathrm{FS}(A)$ monochromatic, we have $A \geq_T \emptyset'$ uniformly.

Proof of Theorem III.34 using c_{vsg}. We will show that c_{vsg} has the desired property. Let S be an $\mathrm{IP}_\cup$ set, such that $\mathrm{FU}(S)$ is monochromatic for c_{vsg}. We prove two claims to get the result.

▶ **Claim III.34.1.** *If $F \in \mathrm{FS}(S)$ then $\mathrm{SG}(F) = 0 \mod 2$.*

Proof. Fix an $F \in \mathrm{FS}(S)$. Let t be big enough so that $\emptyset_{\restriction \max F}[t] = \emptyset_{\restriction \max F}$, in other words any step t such that $\emptyset'$ has converged on its prefix of size $\max F$. Let $G \in S$ with $\min G > t, \max F$, this is possible since S in an $\mathrm{IP}_\cup$ set. We know that $[\mathrm{VSG}(F) = \mathrm{VSG}(F \cup G) = \mathrm{VSG}(G)] \mod 2$. But

$$\mathrm{VSG}(F \cup G) = \mathrm{SG}_{\max G}(F) + \mathrm{SG}_{\max G}(\{\max F; \min G\}) + \mathrm{VSG}(G),$$

and by our choice of G,

$$\mathrm{SG}_{\max G}(F) = \mathrm{SG}(F) \quad \text{and} \quad \mathrm{SG}_{\max G}(\{\max F; \min G\}) = 0.$$

We get that $\mathrm{VSG}(G) = \mathrm{SG}(F) + \mathrm{VSG}(G) \mod 2$ so $\mathrm{SG}(F)$ must be even. ■

▶ **Claim III.34.2.** *For every $F, G \in \mathrm{FU}(S)$ with $F < G$, the gap $(\max F, \min G)$ cannot be a small gap.*

Proof. Fix such F and G in $\mathrm{FU}(S)$, we need to show that $\mathrm{SG}(\{\max F; \min G\}) = 0$. We have that $\mathrm{SG}(F \cup G) = \mathrm{SG}(F) + \mathrm{SG}(\{\max F; \min G\}) + \mathrm{SG}(G)$. As all of F, G and $F \cup G$ are in $\mathrm{FU}(S)$, by Claim III.34.1 they have an even number of short gaps so we have $0 = 0 + \mathrm{SG}(\{\max F; \min G\}) + 0 \mod 2$.

As $\mathrm{SG}(\{\max F; \min G\})$ is even and either 0 or 1, it is 0 and $\{\max F; \min G\}$ is not a short gap. ■

We just proved that in between two elements $F < G$ of S, there must be a large gap, in other words the approximation of $\emptyset'$ already converged on its initial segment of size $\max F$ at step $\min G$. But then, given S, we are clearly able to compute $\emptyset'$. ■

▶ **Corollary III.34.1.** HT *implies* ACA_0 *over* RCA_0.

Remark that in the proof, we only needed unions of three elements, and 2 colors. So it proves the stronger result that the finite union theorem with unions of at most three elements and two colors implies ACA_0 over RCA_0. The next proof strengthen this result to unions of two elements.

Definition III.35 (Important gaps, [21])

— Let F be a finite set. We say that a gap (m, n) in F is *important for* k if something is enumerated in the approximation of $\emptyset_{\restriction k}$ between stage m and stage n. We simply say that it is *important* if it is important for $\min F$.
— We write $\mathrm{IMP}_k(F)$ for the number of gaps in F that are important for k. We write $\mathrm{IMP}(F) = \mathrm{IMP}_{\min F}(F)$ for the number of important gaps in F.
— We write $c_{\mathrm{imp}} : \mathcal{P}_{\mathrm{fin}}(\mathbb{N}) \to 2$ for the function $F \mapsto \mathrm{IMP}(F) \mod 2$.

We are now able to give the proof of Theorem III.34 using c_{imp}, given in [21], that only requires unions of two and one elements.

Proof of Theorem III.34 using c_{imp}. We will show that c_{imp} has the desired property. Let S be an $\mathrm{IP}_\cup$ set such that $\mathrm{FU}(S)$ is monochromatic. We will prove that for any $F \in S$, we have that $\emptyset_{\restriction \min F}[\max F] = \emptyset_{\restriction \min F}$.

Indeed, let $F \in S$ be any set. Let t be big enough so that the approximation of $\emptyset'$ has converged in its $\min F$ first elements, and because S is an $\mathrm{IP}_\cup$ set, let $G \in S$ be with $\min G > t$.

Now, $c_{\mathrm{imp}}(F) = c_{\mathrm{imp}}(F \cup G)$. We also have:

$$\mathrm{IMP}(F \cup G) = \mathrm{IMP}(F) + \mathrm{IMP}_{\min F}(\{\max F; \min G\}) + \mathrm{IMP}_{\min F}(G).$$

By our choice of G with a sufficiently big minimum, $\mathrm{IMP}_{\min F}(G) = 0$, so we get:

$$\mathrm{IMP}_{\min F}(\{\max F; \min G\}) = 0 \mod 2$$

and therefore it is equal to 0. But that means that nothing is enumerated between $\min F$ and $\max G$, so $\emptyset'_{\restriction \min F}$ has already converged at step $\max F$.

This suggests the algorithm to compute $\emptyset'$ from S: to compute $\emptyset'$ up to n, find $F \in S$ with $\min F > n$ and approximate $\emptyset'$ up to $\max F$. ■

There is still quite an important gap between our lower and upper bounds: we know that we can force all solutions to compute the jump, and we know that there always exists a solution computable in the ω+1-th jump. But what is in between? Does there always exist an arithmetical solution? Or does there exist a computable coloring with all solutions computing the ω-th jump?

▶ **Question III.36** ([56]). Which one of the following is true:
— HT is equivalent to ACA_0 over RCA_0?
— HT is strictly between ACA_0 and ACA_0^+ over RCA_0?
— HT is equivalent to ACA_0^+ over RCA_0?

Let us show that neither c_{imp} nor c_{vsg} answer the question.

Definition III.37

We say that a coloring is *stable in* S if there exists a computable $T \subseteq_{\mathrm{FS}} S$, such that $\forall n \in \mathrm{FS}(T)$, $\exists m \in \mathrm{FS}(T \setminus [0, n])$ such that for all $p \in \mathrm{FS}(T \setminus [0, m + n])$, we have $c(n + m) = c(n + m + p)$. Such a color is called the *limit color* of n.

Theorem III.38

If c is stable, then there exists a c'-computable infinite set S with $\mathrm{FS}(S)$ monochromatic, uniformly in $c^{(3)}$.

Proof. Let S be such that $\forall n \in \mathrm{FS}(S)$, $\exists m \in \mathrm{FS}(S \setminus [0,n])$ such that for all $p \in \mathrm{FS}(S \setminus [0, m+n])$, $c(n+m) = c(n+m+p)$. Define $c_{\lim}(n)$ to be the color of $c(n+m_n)$ for m_n being the first to witness the stability property. There must be a limit color which appear infinitely often in S, fix it to be i (using the triple jump of c we can find such a color). Now, define a sequence $(x_n)_{n\in\mathbb{N}}$ such that: $c_{\lim}(x_0) = i$, and x_{n+1} being the first in S such that $c_{\lim}(x_{n+1}) = i$ and $x_{n+1} > x_n + m_{x_n}$.

Any finite sum in $\{x_n + m_{x_n} : n \in \mathbb{N}\}$ is of the form $x_{n_0} + m_{x_{n_0}} + z$ where $z > x_{n_0} + m_{x_{n_0}}$, and therefore is of color i. ■

▶ **Corollary III.38.1.** *The coloring c_{imp} admits an arithmetical monochromatic IP set.*

Proof. Using Theorem III.38, it is sufficient to show that c_{imp} is stable. We use the "finite union theorem" terminology. Let A be a finite subset of integers, and fix 2^N big enough so that the first $\min A$ bits of $\emptyset'$ have converged at step N. Then, let B be such that $\max B > S$. Then $c_{\mathrm{imp}}(A \cup B) = c_{\mathrm{imp}}(A \cup B \cup C)$ for every C with $C > B$. ■

Theorem III.39

Let S be any computable $\mathrm{IP}_\cup$ set. Then, c_{vsg} admits an infinite $\mathrm{IP}_\cup$ set T with $\mathrm{FU}(T) \subseteq \mathrm{FU}(S)$, computable in $\emptyset'$ uniformly in $\emptyset^{(3)}$.

Proof. Toward a contradiction, suppose that for some n, if $F \in \mathrm{FU}(S)$ is such that $n < F$ then $\mathrm{SG}(F) = 1 \mod 2$. Then, let $A, B \in \mathrm{FU}(S)$ with $n < A < B$ and $(\max A, \min B)$ is a large gap. Then, $\mathrm{SG}(A \cup B) = \mathrm{SG}(A) + \mathrm{SG}(B) = 0 \mod 2$, a contradiction.

Therefore, there exists a $\emptyset'$-computable set T_0 with $\mathrm{FU}(T_0) \subseteq \mathrm{FU}(S)$ and every set $F \in T_0$ has $\mathrm{SG}(F) = 0 \mod 2$. Using $\emptyset^{(3)}$, find a color i for c_{vsg} that appear infinitely many often in T_0.

Now, let T be a shrink of T_0 where every set has color i and if A, B are in T then they are apart in a large gap. Then, any union $F = \bigcup_{i<n} F_i$ in $\mathrm{FU}(T)$ with $F_i < F_{i+1}$ is such that $\mathrm{VSG}(F) = \mathrm{SG}(F_0) + \cdots + \mathrm{SG}(F_{k-2}) + \mathrm{VSG}(F_{k-1}) = \mathrm{VSG}(F_{k-1}) = i \mod 2$. ■

III.3.3 Right-match, Left-match and Full-match

In order to answer Question III.36, we investigate where the construction can be improved, and where it cannot, to unveil which steps we should focus on. The so-called steps are the constructions of right-match, full-match and solution.

First, the construction of the right-match is optimal in some sense as the obtained right-match is computable in the coloring. The only room for improvement is the uniformity of the solution, however there can be no hope to get a computable uniformity.

Right-matches

Theorem III.40

There exists no continuous functional with as input a coloring c, and with output a right-match (F, S) for c.

Proof. Towards a contradiction, let Φ be such a functional. Let σ be a sufficiently big prefix of some coloring, such that Φ^σ converge to some finite F. Let $m = \max F$. We extend σ to the coloring c such that for all $n \notin \mathrm{dom}(\sigma)$, $c(n) = 1$ iff $n = 0 \mod m+1$. Suppose that there exists a right-match (F, S) for some S. Let $x \in \mathrm{FS}(S)$ with $x > |\sigma|$ such that $x \in (m+1)\mathbb{N}$. We have $c(x) = 1$ but $c(x+a) = 0$ for every $a \leq m$, a contradiction. ■

We proved that if m is any integer, then there exists a computable coloring with no right-match whose finite set F is such that $F < m$. In [64], Towsner proves a different result: given k a bound on the *size of the finite set*, there exists a computable coloring with no *computably enumerable* right-match whose finite set has size less than k.

The result is the same: we cannot hope to improve the construction of Theorem III.11 to a uniform one.

Left-matches

In his paper [64], where he introduced the formulation of the proof using right-matches and full-matches, Towsner also had the approach of studying the reverse mathematics of every step, trying to pin those that can be improved.

However, his conclusions were erroneous: he made a flawed proof that there exists a computable coloring with no computably enumerable full-match. Therefore, he concluded that we cannot improve the construction of a full-match to computable, and the efforts in improving the upper bounds should be made toward improving the last construction in the proof of Theorem III.15: the construction of monochromatic IP sets.

In this thesis not only we discovered the flaw in Towsner's lower bound for full-matches, but we also show that his proof cannot be patched. Indeed, it was actually "proving" the stronger result that there exists a 2-coloring with no c.e. left-match. We prove here that this is not possible.

Theorem III.41

For any 2-coloring c and infinite set S, there exists a left-match (F, T) with $T \subseteq_{\mathrm{FS}} S$, T being computable uniformly in $(c, S)''$.

Proof. We write $S_{\leq N}$ for $S \cap [0; n]$ and $S_{>N}$ for $S \cap (n; \infty)$. We split the construction into two cases.

If there exists some N such that $(\mathrm{FS}(S_{\leq N}), S_{>N})$ is a left-match. Then, the result is clear.

Otherwise, we do the following construction. Let $x_0 \in S$ be any element. If the x_i are defined for $i < n$, then we define x_n to be the first element in $\mathrm{FS}(S_{>x_{n-1}})$ such that for all $a \in \mathrm{FS}(\{x_i : i < n\})$ we have $c(a + x_n) \neq c(x_n)$, which exists because we are in case 2.

Now, let $S_0 = \{x_i : i \in \mathbb{N}\}$. We prove by induction on the length of the sum that $c(x_{i_0} + x_{i_1} + \cdots + x_{i_n}) = c(x_{i_0}) + n \mod 2$. If $n = 0$, then the result is clear. If we suppose that the result is true for n, let $x = x_{i_0} + x_{i_1} + \cdots + x_{i_{n+1}}$, then we know by construction that $c(x_{i_0} + x_{i_1} + \cdots + x_{i_{n+1}}) \neq c(x_{i_0} + x_{i_1} + \cdots + x_{i_n})$. By the induction hypothesis $c(x_{i_0} + x_{i_1} +$

$\cdots + x_{i_{n+1}}) \neq c(x_{i_0}) + n \mod 2$ and by the fact that there are only two colors, $c(x_{i_0} + x_{i_1} + \cdots + x_{i_{n+1}}) = c(x_{i_0}) + n + 1 \mod 2$.

By the pigeon-hole principle, let $S_1 = \{x_{i_0}; x_{i_1}; \cdots\}$ be a monochromatic subset of S_0 of color say i. Let $T = \{x_{i_0} + x_{i_1}; x_{i_2} + x_{i_3}; \cdots\}$. Then, T is monochromatic of color i: indeed, any finite sum of elements from T is composed of an even number of x_i with the minimum of those being of color i. ■

In Corollary III.51.1 we improve a bit Theorem III.41 by showing that the finite set of the left-match to be monochromatic.

Note that the fact that the coloring has only two colors is used here, when concluding from $c(a) \neq c(b) \neq c(d)$ that $c(a) = c(d)$. This is important, compared to the case of right-matches and full-matches. Indeed, a construction for a right-match (respectively full-match) for 2-colorings can be iterated to work on colorings with an arbitrary number of colors.

Definition III.42

- FM is the principle: for every k, and k-coloring $c : \mathbb{N} \to k$ and infinite $S \subseteq \mathbb{N}$, there exists a full-match (F, T) with $F \cup T \subseteq S$. FM_k is the analogue for a fixed k.
- LM is the principle: for every k, and k-coloring $c : \mathbb{N} \to k$ and infinite $S \subseteq \mathbb{N}$, there exists a left-match (F, T) with $F \cup T \subseteq S$. LM_k is the analogue for a fixed k.
- RM is the principle: for every k, and k-coloring $c : \mathbb{N} \to k$ and infinite $S \subseteq \mathbb{N}$, there exists a right-match (F, T) with $F \cup T \subseteq S$. RM_k is the analogue for a fixed k.

▶ **Proposition III.43.** *We have that* $\mathrm{HT}_2 \implies \mathrm{HT}$ *over* RCA_0.

Proof. This is clear, by induction on k. If c is a $k+1$-coloring, then applying HT_2 to the 2-coloring c_0 defined by $c_0(x) = 0$ iff $c(x) = 0$ either gives us a solution to c, or an IP set on which c is a k-coloring, and we can apply the induction hypothesis. ■

▶ **Proposition III.44.** *We have that* $\mathrm{FM}_2 \implies \mathrm{FM}$ *and* $\mathrm{RM}_2 \implies \mathrm{RM}$ *over* RCA_0.

Proof. Let $k \in \omega$. We only prove that $\mathrm{RM}_2 \implies \mathrm{RM}$ over RCA_0, the other implication having exactly the same proof. Suppose RM_2. Let c be a k-coloring of the integers. For every $i < k$, let c_i be the 2-coloring such that $c_i(n) = 1$ if and only if $c(n) = i$, the characteristic function of the integers of color i. We let X_{-1} be $\mathbb{N}$, and for $0 \leq i < k$, we let $X_i \subseteq X_{i-1}$ and F_i be given by RM_2 on c_i. This construction is a meta-induction outside of RCA_0 which is possible as k is an integer of the meta-theory. The induction hypothesis at stage i is that RCA_0 proves the existence of such an X_i.

We claim that $\bigcup_{i<k} F_i$ together with the IP set X_{k-1} is a right-match. Let $x \in X_{k-1}$. Letting $i < k$ be the color of x, as $x \in X_i$, let $a \in F_i$ be such that $c_i(a) = c_i(x + a)$. By definition of c_i and i, this implies $c(a) = c(x + a)$, which concludes the proof. ■

▶ **Question III.45.** Does $\mathrm{LM}_2 \implies \mathrm{LM}_k$ over RCA_0 for some k?

Suppose we do the same construction as in Proposition III.44 to solve Question III.45. To make it simpler, consider a 3-coloring of the integers c. Define c_0, c_1 and c_2 to be the characteristic functions of color respectively 0, 1 and 2. By Theorem III.41, we can create (F_0, S_0) a left-match for coloring c_0, (F_1, S_1) a left-match for coloring c_1, and (F_2, S_2) a left-match for coloring c_2 with $S_2 \subseteq_{\mathrm{FS}} S_1 \subseteq_{\mathrm{FS}} S_0$. However, there is no guarantee that $(F_0 \cup F_1 \cup F_2, S_2)$ is a left-match for c:

indeed, given $x \in \mathrm{FS}(S_2)$, we can have $c_i(a) = c_i(x+a) = 0$ but still $i \neq c(a) \neq c(x+a) \neq i$ for all $i < 3$ and $a \in F_i$. In particular, we don't know the answer for $k = 3$:

▶ **Question III.46.** Does there exists a computable 3-coloring with no computable left-match?

Full-matches

We try to prove that every computable 2-coloring of the integers admits a computable full-match. We are not able to tackle the general case, however we are able to pin many cases of a disjunction, leaving only computable coloring of a particular form to be investigated.

We first add a property to the (right, left, full)-matches in order to explain more easily the disjunction:

Definition III.47

Fix a coloring $c : \omega \to k$. Let $F \subseteq \omega$ be a finite set, and $S \subseteq \omega$ be a set such that $\max F < \min X$. We say that (F, S) is a *middle match for color* i if $\forall x \in \mathrm{FS}(S)$, there exists $a \in F$ such that $c(a + x) = i$.

Now, we are able to express our first property used in disjunction.

Definition III.48

Let c be a 2-coloring of the integers. Let $i, j < 2$, we say that an infinite set S satisfies $(\mathrm{MM}_{i,j})$ if:

$$\exists F \subseteq C_i : (F, S) \text{ is a middle match for color } j \qquad (\mathrm{MM}_{i,j})$$

We are now able to treat the cases depending on the truth of $(\mathrm{MM}_{0,0})$, $(\mathrm{MM}_{1,1})$, $(\mathrm{MM}_{0,1})$ and $(\mathrm{MM}_{1,0})$.

When we do not have $(\mathrm{MM}_{i,1-i})$

We start with the case when there is some i such that we have $(\neg\mathrm{MM}_{i,1-i})$. In this case, we are able to find a computable monochromatic IP set.

Theorem III.49

Suppose there is some $i < 2$ such that $(\mathrm{MM}_{i,1-i})$ does not hold within an infinite computable set $X \subseteq \omega$. Then there is an infinite computable set $Y \subseteq X$ such that $\mathrm{FS}(Y)$ is homogeneous.

Proof. We prove the result for $i = 0$. We suppose that there is no computable homogeneous set for color 1, as otherwise the proof is complete. We will now describe how to build a computable IP set for color 0. For that, we define a computable sequence (x_n), a sequence (X_n) of uniformly computable infinite sets, such that:

1. $x_n \in \mathrm{FS}(X_n) \cap C_0$,
2. For all $x \in \mathrm{FS}(\{x_i : i \leq n\})$, $x + \mathrm{FS}(X_{n+1}) \subseteq C_0$,

We first prove that if we can construct such sequences, then $X = \mathrm{FS}(\{x_n : n \in \omega\})$ is a monochromatic IP set. Let $x \in \mathrm{FS}(X)$, and $i_0 < \ldots < i_N \in \mathbb{N}$ be such that $x = \sum_{0 \leq n \leq N} x_{i_n}$. If $N = 0$ then by Item 1 $c(x) = 0$. If $N > 0$, then $\sum_{0 \leq n < N} x_{i_n} \in \mathrm{FS}(\{x_i : i < i_n\})$ and $x_{i_n} \in \mathrm{FS}(X_{i_n})$ so by Item 2, $c(x) = 0$.

Now, we explain how to do the construction. We take $x_0 \in C_0$ be any element of C_0, and $X_0 = \mathbb{N}$. Now, suppose that we have defined $(x_i)_{i<n}$ and $(X_i)_{i<n}$. We let $X_n = \{y_i : i \in \omega\}$, where the y_i are defined as follows: for any i, $y_i \in \mathbb{N}$ is the smallest element greater than y_{i-1} (when defined) such that $\{x_j : j < n\} + \mathrm{FS}(\{y_0; \dots; y_{i-1}; y_i\}) \subseteq C_0$. Note that by $(\neg\mathrm{MM}_{0,1})$ applied with $F = \{x_i : i < n\} + \mathrm{FS}(\{0; y_0; \dots; y_{i-1}\})$, such a y_i always exists, and X_n is infinite and computable. It is clear by construction that Item 2 is ensured.

Now, as we have supposed that no computable IP set is monochromatic of color 1, it cannot be the case that $\mathrm{FS}(X_n) \subseteq C_1$. Using this we define x_n to be the smallest element of $\mathrm{FS}(X_n) \cap C_0$, ensuring Item 1. ■

We continue the case disjunction, this time focusing on $(\mathrm{MM}_{i,i})$.

When $(\mathrm{MM}_{0,0})$ and $(\mathrm{MM}_{1,1})$ have the same truth value

Theorem III.50

If $(\mathrm{MM}_{0,0})$ and $(\mathrm{MM}_{1,1})$ hold in some S, then there exists a computable full-match.

Proof. Let F_0 and F_1 be given by $(\mathrm{MM}_{0,0})$ and $(\mathrm{MM}_{1,1})$ on S. Then, we claim that $(F_0 \cup F_1, S)$ is a full-match: let $x \in \mathrm{FS}(S)$, and $c(x) = i$ Let $a \in F_i$ such that $c(x + a) = i$. Then, we have $c(a) = c(x + a) = c(x)$. ■

Theorem III.51

If $(\neg\mathrm{MM}_{0,0})$ and $(\neg\mathrm{MM}_{1,1})$ hold on some S, then there exists a computable monochromatic IP set.

Proof. We prove that in this case, we must have $(\neg\mathrm{MM}_{0,1})$ on S. Let $F \subseteq C_0$. Then, by $(\neg\mathrm{MM}_{0,0})$ let x such that $x + F \subseteq C_1$. By $(\neg\mathrm{MM}_{1,1})$, let y be such that $y + x + F \subseteq C_0$. Then, $x + y$ is a witness of $(\neg\mathrm{MM}_{0,1})$.

We conclude using Theorem III.49. ■

Note that, at this point, we have as a consequence the existence of left-match. We even have the stronger result that the finite part of the left-match can be considered monochromatic:

▶ **Corollary III.51.1.** *For every 2-coloring c and infinite set S, there exists a left-match (F, T) with $F \cup T \subseteq_{\mathrm{FS}} S$ and F monochromatic.*

Proof. If $(\neg\mathrm{MM}_{i,i})$ for all i on some S, then by Theorem III.51 we have a monochromatic IP set and therefore a monochromatic left-match. Otherwise, there must exist some $i < 2$ such that $(\mathrm{MM}_{i,i})$ holds, that is there is some $F \subseteq C_i$ such that we have $\forall x \in S$, there exists $a \in F$ with $c(x + a) = i = c(a)$, which is therefore a left-match with monochromatic finite set. ■

The remaining case

There is only one remaining case: When we have all the following:

1. For every infinite computable $S \subseteq \mathbb{N}$, we both have $(\mathrm{MM}_{0,1})$ and $(\mathrm{MM}_{1,0})$,
2. For every infinite computable $S \subseteq \mathbb{N}$, $(\mathrm{MM}_{0,0})$ for S if and only if $(\neg\mathrm{MM}_{1,1})$ for S.
3. The coloring is not stable.

We call this case the *remaining case*.

By symmetry, suppose that $(\mathrm{MM}_{0,0})$ holds on $\mathbb{N}$. We must have that for every S, $(\neg\mathrm{MM}_{1,1})$ holds for S. Note that by $(\mathrm{MM}_{0,0})$, there already exists a full-match for color 0. We only need to build a full-match for color 1 in order to conclude.

We investigate a bit on the implications of our hypothesis.

▶ **Lemma III.52.** *Suppose we are in the remaining case. If $F_0 \subseteq C_0$ is such that $(F_0, \mathbb{N})$ is a middle-match for color 1, then it is also a middle match for color 0.*

Proof. Suppose this is not true, let x be such that $F_0 + x \subseteq C_1$. Then, $F_0 + x$ is a middle-match for color 1: Indeed, let $y \in \mathbb{N}$. As $y + x \in \mathbb{N}$ and F_0 is a middle-match for color 1, there exists $a \in F_0$ such that $c(a + x + y) = 1$. This is a contradiction with $(\neg\mathrm{MM}_{1,1})$. ■

▶ **Lemma III.53.** *Suppose we are in the remaining case, and there is no computable full-match. Then, for every $F_1 \subseteq C_1$, there exists a finite $H > F_1$ such that $(F_1, H \cap C_1)$ is a middle-match for color 1, but for every $x > H$, $(F_1, (\{x\} \cup (H + x)) \cap C_1)$ is not a middle match for color 1.*

Proof. Suppose not. Then, let $F_1 \subseteq C_1$ be given as a witness of failure of the lemma. We build a computable sequence $(x_n)_{n\in\mathbb{N}}$ such that for every n, $(F_1, \mathrm{FS}(\{x_i : i < n\}) \cap C_1)$ is a middle match for color 1. But then, F_1 is a full-match for color 1. By $((\mathrm{MM}_{0,0}))$, there is a full-match for color 0, so we get a full-match. ■

▶ **Question III.54.** Does every computable coloring has a computable full-match?

The special case of c_{imp} and c_{vsg}

Recall that c_{imp} and c_{vsg} are defined on the finite sets of integers. We show that, despite having no computable solution, they both have computable full-matches.

▶ **Proposition III.55.** *For every $\mathrm{IP}_\cup$ set S, c_{imp} satisfy $(\mathrm{MM}_{0,0})$ and $(\mathrm{MM}_{1,1})$ in T with $T \subseteq_{\mathrm{FU}} S$. Therefore, c_{imp} must have a computable full-match (F, T) with $F \cup T \subseteq_{\mathrm{FU}} S$.*

Proof. We now show the result for the coloring c_{imp}. Let A_0 and A_1 in $\mathrm{FS}(S)$ be such that:

1. $\emptyset' \upharpoonright \min A_i$ has converged at step $\max A_i$, for $i = 0, 1$.
2. $c_{\mathrm{imp}}(A_i) = i$ for $i = 0, 1$.

Then, let $T = \{X \in S : X > \max(A_0 \cup A_1)\}$. Then, $(\{A_0; A_1\}, T)$ is a full-match: let $X \in \mathrm{FS}(T)$, and $i = c_{\mathrm{imp}}(X)$. Then, $c_{\mathrm{imp}}(A_i) = i$ and $c_{\mathrm{imp}}(A_i \cup X) = c_{\mathrm{imp}}(A_i) = i$ as no gap after $\max A_i$ can be important as the $\min A_i$ first bits of $\emptyset'$ has already converged. $\{A_0\}$ and $\{A_1\}$ are witness of $(\mathrm{MM}_{0,0})$ and $(\mathrm{MM}_{1,1})$.

Now, let us argue that we can find such A_0 and A_1. For every i, there must exist some X_i such that $c_{\mathrm{imp}}(X) = i$ and Item 1. is not true. Indeed, otherwise we would be able to compute the n first bits of $\emptyset'$ by just finding some $X \in \mathrm{FS}(S)$ with $c_{\mathrm{imp}}(X) = i$ and $n < X$, which is possible as a computable S cannot have $\mathrm{FS}(S)$ monochromatic, and then, output $\emptyset' \upharpoonright n[\max X]$.

Set $A_i = X_{1-i} \cup Y_i$ where $Y_i \in \mathrm{FS}(S)$ and $\min Y_i$ has made $\emptyset'$ converged on its $\min X_{1-i}$ first bits. Then, $c_{\mathrm{imp}}(A_i) = c_{\mathrm{imp}}(X_i) + 1 = i$, and A_i satisfy 1. ■

We prove a slightly weaker statement for c_{vsg}.

▶ **Proposition III.56.** *The coloring c_{vsg} has a computable full-match (F, T).*

Proof. We start with c_{vsg}. We claim that there exists A_0, A_1 in $\text{FS}(S)$ such that

1. $\text{VSG}(A_i) = i \mod 2$, and
2. $\text{SG}(A_i) = 0 \mod 2$.

In this case, let N be big enough so that the approximation of $\emptyset'$ has converged on its $\max A_0$ and $\max A_1$ first bits, at step N. Let $T = \{E \in S : E > N\}$. Then $(\{A_0; A_1\}, T)$ is a full-match: Let $X \in \text{FS}(T)$. If $c_{\text{imp}}(X) = i$, then $c_{\text{imp}}(A_i) = i$ and $c_{\text{imp}}(A_i \cup X) = \text{SG}(A_i) + \text{SG}(\{\max A_i; \min X\}) + \text{VSG}(A_i) \mod 2 = 0 + 0 + \text{VSG}(A_i) = c_{\text{imp}}(A_i)$ by our choice of N and $\min X > N$.

The existence of such A_0 and A_1 is clear if we can take any finite subset of the integers. Take $A_0 = F \cup \{N\}$, where $\text{SG}(F) = 0 \mod 2$ and $(\max F, N)$ is a large gap. Take any small gap (x, y) that is not a very small gap, and set $A_1 = \{x; y\}$. ■

III.4 Ordinal Analysis of Hindman's proof

Ordinal analysis of theories is an important part of proof theory. Given a theory T, the *proof-theoretic ordinal* associated with a theory T is the smallest ordinal α such that T proves the well-foundedness of α, that is: There exists a primitive recursive order $\prec$ of order-type α, and T does not prove that $\prec$ is well-founded. The associated ordinal-theoretic usually yields a nice well-ordering on theories, as theories with higher ordinal-theoretic tends to prove the coherence of the smaller ones. For instance, Gentzen proved the coherence of Peano Arithmetic using only "finitistic reasoning" and an induction along $\epsilon_0 = \omega^{\omega^{\cdot^{\cdot^{\cdot}}}}$, the ordinal-theoretic of PA. Any theory with ordinal-theoretic above ϵ_0 therefore prove the coherence of PA.

Comparing this ordinal-theoretic to other ordinals associated with combinatorial statements allows to prove their independence from a theory. For instance, Ketonen and Solovay were able to prove Paris-Harrington Theorem, and as a consequence the independence of the strengthened finite Ramsey theorem from Peano Arithmetic. They did so using a model-theoretic method, compared to the usual syntactic methods of proof-theory, such as Gentzen's cut-elimination.

The strengthened finite Ramsey Theorem is a version of the finite Ramsey Theorem where the size of the solution is at least its minimum. In other words, while the finite Ramsey theorem states that there exist solutions of arbitrary cardinality given sufficiently many integers, the strengthened version adds another notion of largeness. A set A is ω-large if $|A| > \min A$, and the strengthened Ramsey Theorem states that for every a, n, k, there exists some N such that for every coloring $c : [N]^n \to k$, there exists some c-homogeneous set H with $|H| > \min H \geq a$.

As suggested by the notation "ω-largeness", there exists a whole family of largeness notion for finite sets, indexed by ordinals. These notions of largeness for finite sets are linked to ordinal analysis of theories. The first introduction of such largeness notion and ordinal analysis were conducted by Ketonen and Solovay in [46]. Given a combinatorial statement P, such as the strengthened Ramsey Theorem, one can associate an operation on ordinal: From α, define $P(\alpha)$ to be the smallest ordinal such that any $P(\alpha)$-large instance has an α-large solution. Comparing the first ordinal closed by the operation P to the ordinal-theoretic of PA allowed Paris and Harrington to conclude, using Gentzen result.

Ordinal analysis is often linked with syntactic method, as cut-elimination. One can look at the ordinal-size of a proof seen as a tree, after eliminating all the cuts in the proof. However, here we will use a more semantic method, where we actually build models to show independence. This model-theoretic approach to ordinal analysis is nicely introduced by Jeremy Avigad and Richard Sommer in [3].

We first give a formal definition of the largeness notions we are interested in. Then, we give an account on how we would be able to benefit from a satisfying ordinal analysis of Hindman's theorem. This gives an example of construction of models from the model-theoretic approach to ordinal analysis of [3]. Finally we perform the analysis of Hindman's proof in terms of largeness notion: in particular, we study how large the coloring can be to ensure the existence of large sets of solutions.

This part is a joint work with Keita Yokoyama.

III.4.1 Largeness notions

Recall that the largeness notions we are about to define are a generalization of cardinality, and are linked to ordinal analysis. We start by defining the general axioms for a notion of largeness. Then, we explicit a particular case of α-largeness, for α being below ε_0. We end by defining notions of largeness related to Hindman's Theorem.

Definition III.57

We say that a subset Γ of $\mathcal{P}(\mathbb{N})$ is a *notion of largeness* if:

1. any infinite set has a finite subset in Γ,
2. Γ is closed under superset,
3. for any two pointwise different sequence $(x_i)_{i\in\mathbb{N}}$ and $(y_i)_{i\in\mathbb{N}}$, if $\{x_i : i < n\}$ is large and $\forall i < n, y_i < x_i$, then $\{y_i : i < n\}$ is large. (We sometimes refer to this as left-preservation of largeness)

where in this context X is large if $X \in \Gamma$.

The left-preservation of largeness makes it resemble to a density notion: not only the size is important, but also how much the members are stacked to the beginning of the integers. This notion is even sometimes referred to as "density", or "regular largeness notion".

We now define the notion of α-largeness for some ordinal α. The idea of α-largeness is that one can "count-down" the elements of a set, starting from α and reaching 0. For $\alpha = k \in \mathbb{N}$, it is clear that counting down from k the element of a set E, will reach 0 if and only if there are at least k elements in E.

However, it is not even clear how to count down from any limit ordinal, as it has no predecessor. We have to choose one smaller ordinal, and we will do so depending on the integer we count.

Definition III.58

Let $\alpha < \epsilon_0$. We inductively define the Cantor normal form of α to be the decomposition of α into $\omega^{\beta_0} + \omega^{\beta_1} + \cdots + \omega^{\beta_k}$ where $\beta_0 \geq \beta_1 \geq \cdots \beta_k$ are ordinals in Cantor normal form.

The above definition works for all ordinals below ϵ_0, since if $\alpha = \omega^{\beta_0} + \omega^{\beta_1} + \cdots + \omega^{\beta_k}$, then every $\beta_i < \alpha$. This is of course not the case for ϵ_0, where $\epsilon_0 = \omega^{\epsilon_0}$.

Definition III.59

Let $\alpha < \epsilon_0$, and $n \in \mathbb{N}$. We define $\alpha[n]$ as follows:

1. $0[n] = 0$.

2. $\omega^0 = 0$ and $\omega^\beta[n] = n \cdot \omega^{\beta[n]}$.
3. If the Cantor normal form of α is $\omega^{\beta_0} + \omega^{\beta_1} + \cdots + \omega^{\beta_k}$, then $\alpha[n] = \omega^{\beta_0} + \omega^{\beta_1} + \cdots + \omega^{\beta_{k-1}} + (\omega^{\beta_k}[n])$.

If $A = \{a_0; \cdots; a_n\}$ is a finite set with $a_k < a_{k+1}$, then we write $\alpha[A]$ for $\alpha[a_0][a_1]\cdots[a_n]$, where the precedence is given as follows: $\alpha[n_1][n_2] = (\alpha[n_1])[n_2]$.

We stated earlier that this definition should be a kind of generalization of the predecessor in a particular class of ordinals. For successor ordinals, it is clear that $(\alpha+1)[n] = \alpha$. For a limit ordinal λ, we can prove that $\lambda[n] \to \lambda$ when $n \to \infty$. Therefore the "predecessor" chosen as $\lambda[n]$ in the countdown is just some particular ordinal below λ, but closer and closer as n is bigger.

Note that we could have extended this to ordinals higher than ε_0, by using a suitable function, which to any countable ordinal α, associate a family $(\alpha_n)_{n\in\mathbb{N}}$ with $\lim_{n\to\omega} \alpha_n = \alpha$.

Here are a few examples of the previous definition for limit ordinals:

1. We have $\omega^l \cdot k[n] = \omega^l \cdot (k-1) + \omega^{l-1} \cdot n$ for $l \in \omega$,
2. $\omega^\omega[n] = n \cdot \omega^n$
3. $\omega^{\omega^\omega}[n] = n \cdot \omega^{n \cdot \omega^n}$

Definition III.60 ([40])

Let $\alpha < \epsilon_0$ be an ordinal, and $A \subseteq \mathbb{N}$ be a set. We say that A is *α-large* if A is infinite or $\alpha[A] = 0$.

▶ **Example III.61.** For any $n \in \omega$, A is n-large if and only if $|A| \geq n$. A is ω-large if $A \setminus \{\min A\}$ is $\min A$-large, that is $|A| > \min A$.

The notions of α-largeness still lack some expressiveness. For instance, the largeness notion corresponding to "$\min(A) + 1 < |A|$" is not an α-largeness notion for any α [3]. Yet, it is included in $\omega + 1$-largeness. We give a few operations on largeness notions to palliate this lack:

Definition III.62 (Operation on largeness notions)

Let Γ, Γ_0 and Γ_1 be notions of largeness.

1. We say that Z is $\Gamma_0 \oplus \Gamma_1$*-large* if there exists a partitioning of $Z = Z_1 \sqcup Z_0$ such that Z_0 is Γ_0-large, Z_1 is Γ_1-large and $Z_1 < Z_0$ [a].
2. We say that Z is $\Gamma_0 \otimes \Gamma_1$*-large* if there exists a partitioning of $Z = Z_0 \sqcup \cdots \sqcup Z_{k-1}$ such that
   (a) For every $i < k-1$, $Z_i < Z_{i+1}$,
   (b) For every $i < k$, Z_i is Γ_0-large,
   (c) For every $(x_i)_{i<k} \in \prod_{i<k} Z_i$, $\{x_i : i < k\}$ is Γ_1-large.
3. We write $\Gamma^{\{k\}}$-largeness for $\Gamma \otimes (\Gamma^{\{k-1\}})$-largeness. The notation with curly braces is to distinguish it with exponentiation.
4. We write ${}^{\{k\}}\Gamma$-largeness for $({}^{\{k-1\}}\Gamma) \otimes \Gamma$-largeness.

a. The reason Z is split in this order will appear clear in Proposition III.64

3. ω-largess corresponds to $|A| > \min A$, while $\omega + 1$-largeness corresponds to $|A| > \min(A \setminus \{\min A\})$. Both are clearly different.

The notation $\oplus$ and $\otimes$ is useful to distinguish it with ordinal sum in the case of α-largeness: note that for instance $(1 \oplus \omega)$-largeness, which is "$\min A + 1 < |A|$" does not correspond to $(1 + \omega)$-largeness which is ω-largeness. However, $\oplus$ and $+$ correspond for well-chosen ordinals.

Definition III.63

Let $\alpha_0 = \sum_{i<k_0} \omega^{\nu_i^0}$ and $\alpha_1 = \sum_{i<k_1} \omega^{\nu_i^1}$ be two ordinals below ϵ_0, in their Cantor normal form. We say that (α_0, α_1) is *compatible for addition* if $\nu_{k_0-1}^0 \geq \nu_0^1$.

▶ **Proposition III.64.** *Let $\alpha, \beta < \epsilon_0$ be ordinals. If (α, β) is compatible for addition, then A is $\alpha + \beta$-large if and only if A is $\alpha \oplus \beta$-large.*

Proof. Suppose A is $\alpha + \beta$-large. As α and β are compatible for addition, the Cantor normal of $\alpha + \beta$ is just the concatenation of the two Cantor normal forms. Therefore, there exists a prefix A_0 of A such that $(\alpha + \beta)[A_0] = \alpha$, $\beta[A_0] = 0$. But $(\alpha + \beta)[A] = \alpha[A \setminus A_0] = 0$, so $A_1 = A \setminus A_0$ is α-large. We found $A_0 < A_1$ witnessing that A is $\alpha \oplus \beta$-large.

Conversely, if A is $\alpha \oplus \beta$-large, let $A = A_0 \sqcup A_1$ be a witness of this. Then, $(\alpha + \beta)[A_0] \leq \alpha$ as $\beta[A_0] = 0$ and as (α, β) is compatible for addition. Therefore $(\alpha + \beta)[A] \leq \alpha[A_1] = 0$, and A is $\alpha + \beta$-large. ■

▶ **Question III.65.** Is there a good notion of compatibility for product, for which a proposition similar to Proposition III.64 holds? What would be the smallest γ such that γ-largeness implies $\alpha \otimes \beta$-largeness?

▶ **Remark III.66.**
- A set Z is $\omega^{\{2\}}$-large if Z can be partitioned in $Z_0 < \cdots < Z_{k-1}$ where for each $i < k$, $\min Z_i < |Z_i|$ and $k = \max Z_0$.
- $\Gamma \oplus \Gamma = \Gamma \otimes 2$ (it may not be equal to $2 \otimes \Gamma$)
- The operations $\oplus$ and $\otimes$ are not commutative (like addition and multiplication of ordinals).
- The condition (c) of the definition of $\otimes$ (Definition III.62, Item 2.) is equivalent to: $\{\max Z_i : i < k\}$ is Γ_0-large.

We give the following generalization of the pigeon-hole principle, as an example of the kind of things we can do with these operations:

Theorem III.67

If Z is $\Gamma^{\{k\}}$-large, and Z is k-colored, then there exists $Y \subseteq Z$ monochromatic and Γ-large.

Proof. By induction on k. When $k = 1$, the result is obvious. Suppose the result for k, and let Z be $\Gamma \otimes \Gamma^{\{k\}}$-large. Decompose $Z = Z_0 \sqcup \cdots \sqcup Z_l$ as given by the largeness.
- If $\forall i < l$, there is $x_i \in Z_i$ of color 0, then $\{x_i : i < l\}$ is Γ-large and monochromatic.
- Otherwise, let $i < l$ be such that Z_i has no member of color 0. Then Z_i is $\Gamma^{\{k\}}$-large and k colored, so we can apply the induction hypothesis. ■

III.4.2 The strategy

As it was done for proving the Paris-Harrington theorem, we associate an ordinal to the combinatorial statement that interests us.

Definition III.68

Let $X \subseteq \mathbb{N}$ be any set.
- We say X is 0-HT-*large* if it is ω-large.
- We say that X is $n+1$-HT-*large* if it is (n-HT) $\otimes\, \omega$-large, and for every coloring of $\mathrm{FS}(X)$, there exists a finite F such that $\mathrm{FS}(F) \subseteq \mathrm{FS}(X)$ is monochromatic and F is n-HT-large.

Definition III.69

The statement n-FinHT is

$$\forall X \in [\omega]^\omega, \exists F \in [\omega]^{<\omega} : F \text{ is } n+1\text{-HT-large}.$$

The idea to separate HT from ACA_0^+ is to use the following two facts:

Theorem III.70

There exists an arithmetical formula Φ such that $\mathrm{ACA}_0^+ \vdash \Phi$ but $\mathrm{ACA} \nvdash \Phi$.

Proof. The formula Φ is $\Phi \equiv \neg\mathrm{Con}(\mathrm{ACA})$. By Gödel's incompleteness Theorem, this formula cannot be proven in ACA. It remains to show that the consistence of ACA in ACA_0^+, and we will do so by building a model of ACA using only the axioms of ACA_0^+.

We define the model to be $(\mathbb{N}, \{A \in \mathcal{P}(\mathbb{N}) : A \leq_T \emptyset^{(w)}\})$. Clearly, it is a model ACA_0. It remains to show that this model satisfies induction for arbitrary formula. Let ϕ be an arbitrary formula of second order arithmetic. We replace quantification over sets by quantification over sets below $\emptyset^{(\omega)}$, which is really a number quantification. The obtained formula is arithmetical. As ACA_0^+ proves arithmetical induction, $(\mathbb{N}, \{A \in \mathcal{P}(\mathbb{N}) : A \leq_T \emptyset^{(w)}\})$ satisfies induction for ϕ. So it is a model of ACA. ■

Theorem III.71 (Patey, Yokoyama [58])

HT is Π^1_1 conservative over $\mathrm{ACA}_0 + \{n\text{-FinHT} : n \in \omega\}$.

The following corollary of the previous two facts shows that if we are able to prove a certain amount of finite versions of Hindman's Theorem using only ACA, then we would have the separation.

▶ **Corollary III.71.1.** *We have that* ($\forall n \in \omega$, $\mathrm{ACA} \vdash n$-FinHT) *implies that* $\mathrm{HT} + \mathrm{ACA}_0 \nvdash \mathrm{ACA}_0^+$.

Proof. Suppose, towards a contradiction, that $\forall n \in \omega$, $\mathrm{ACA} \vdash n$-FinHT and $\mathrm{HT} \vdash \mathrm{ACA}_0^+$.

Now, let ϕ be given by Theorem III.70. As $\mathrm{HT} \vdash \mathrm{ACA}_0^+$ we also have $HT \vdash \phi$. But by Theorem III.71, $\mathrm{RCA}_0 + \{n\text{-FinHT} : n \in \omega\} \vdash \phi$ already by Π^1_1-conservativity.

As $\mathrm{ACA} \vdash n$-FinHT for all n, we have that $\mathrm{ACA} \vdash \phi$, a contradiction with Theorem III.70. ■

So, according to Corollary III.71.1, it remains to prove n-FinHT for every n using only ACA to achieve our goal.

III.4.3 Proof of Theorem III.71

We want to show that Hindman's Theorem is Π^1_1-conservative over $\mathrm{ACA}_0 + \{n\text{-FinHT} : n \in \omega\}$. To do this, we prove that any Π^1_1 formula ϕ that is not provable in $\mathrm{RCA}_0 + \{n\text{-FinHT} : n \in \omega\}$, is still not provable in $\mathrm{HT} + \phi$. As we will use the equivalence between unprovability and the existence of a model, we start with a model of $\mathrm{RCA}_0 + \{n\text{-FinHT} : n \in \omega\} + \neg\phi$ to build a model of $\mathrm{HT} + \neg\phi$.

We need a few tools to build these models. We will use non-standard models of arithmetic, that is models where the universe is not the integers. If $\mathcal{M}$ is a non-standard model, then $\mathbb{N}$ must inject itself in $\mathcal{M}$, as to every element n of $\mathbb{N}$ corresponds a term $1 + 1 + \cdots + 1$. So every non-standard model contains $\mathbb{N}$ as an initial segment, we call these the *standard* numbers and *non-standard* the other numbers of the model.

If $\mathcal{M}$ is a non-standard model of Σ^0_n-induction, then there cannot be a Σ^0_n definition of $\mathbb{N}$ in $\mathcal{M}$, as otherwise we would have a ϕ such that $\mathcal{M} \models \phi(0)$ and $\forall n,\ \phi(n) \implies \phi(n+1)$, but $\mathcal{M} \not\models \forall n,\ \phi(n)$. So, if ϕ is any Σ^0_n property that is true on the integers, it must also be true for some non-standard number. This fact is called *overspill*. This is of particular importance, as a non-standard element is above all integers, it can encode infinite information but behave inside the model as something finite.

Definition III.72

Let M be some structure on the language of first order arithmetic. Call M-*finite* the finite subsets of M, coded by elements of M (usually taking the binary expansion). Let I be a subset of M. We write $\mathrm{Cod}(M/I)$ for the set $\{A \cap I : A \text{ is } M\text{-finite}\}$.

In other words, $\mathrm{Cod}(M/I)$ are subsets that may be infinite in I, but are coded as finite subsets in M (though we need I to decode it). We use this notion and the following results to build models. Lemma III.74 allows us to build models of WKL_0.

Definition III.73

Let $\mathcal{M}$ be a model of arithmetic. A *cut* is an initial segment that is closed under successor. A *semi-regular* cut is a cut $\mathcal{I}$ such that for every M-finite set X such that $\mathrm{card}_M(X) \in \mathcal{I}$, $X \cap \mathcal{I}$ is bounded in $\mathcal{I}$.

▶ **Lemma III.74** (Lemma IX.3.11 in [60]). *Let $\mathcal{M}$ be a model of RCA_0 and $\mathcal{I}$ be a proper semi-regular cut of $\mathcal{I}$. Then, $(M, \mathrm{Cod}(M/\mathcal{I}))$ is a model of WKL_0.*

▶ **Lemma III.75** (Lemma 3.2 [58]). *Given a countable non-standard model M of $\mathrm{I}\Sigma_1$, and an M-finite set $Z \subseteq M$ which is a-HT-large for some $a \in M \setminus \mathbb{N}$, there exists an initial segment $\mathcal{I}$ of M such that $(\mathcal{I}, \mathrm{Cod}(M/\mathcal{I})) \models \mathrm{WKL}_0 + \mathrm{HT}$ and $\mathcal{I} \cap \mathrm{FS}(Z)$ is unbounded in $\mathcal{I}$.*

Proof. We will build a cut of M, ensuring that it is semi-regular and satisfies HT. Let $(E_i)_{i\in\mathbb{N}}$ be an enumeration of the M-finite sets, and $(c_i)_{i\in\mathbb{N}}$ an enumeration of the M-finite 2-colorings. We suppose that each M-finite set appears infinitely often in the enumeration.

Start with $S_0 = Z$ the a-HT-large set. We define a sequence $(S_n)_{n\in\mathbb{N}}$ with S_n being a $(a-n)$-HT-large set. The resulting cut will be $\{e \in M : \exists n \in \mathbb{N}, e < \min S_n\}$. Here is the construction:

1. If S_{3i} is defined, then it is $(a-3i)$-HT-large. Then, let $S_{3i+1} \subseteq_{\mathrm{FS}} S_{3i}$ be a $(a-3i-1)$-HT-large set such that $\mathrm{FS}(S_{3i+1})$ is monochromatic for c_i.
2. If S_{3i+1} is defined, we ensure that the resulting cut will be semi-regular. If $|E_i| \geq \min S_{3i+1}$, then $S_{3i+2} = S_{3i+1}$. Otherwise, $E_i = \{e_0; \cdots; e_{l-1}\}$ with $l < \min S_{3i+1}$. For

$k \in [-1, l-1]$, we define $W_k = S_{3i+1} \cap [e_k, e_{k+1}[$ where we extend $e_{-1} = 0$ and $e_l = \infty$. Then $S_i = \bigsqcup_{-1 \leq k \leq l} W_k$. As S_{3i+1} is $(a-3i-1)$-HT-large, it is $(a-3i-2)$-HT$\otimes\omega$-large, so let k be such that W_k is $(a-3i-2)$-HT-large. We define S_{3i+2} to be W_k.

3. Define S_{3i+3} to be $S_{3i+2} \setminus \min S_{3i+2}$, which is a $(a-3i-3)$-HT-large set as $n+1$-HT-largeness clearly implies $1 \oplus (n\text{-HT})$-largeness.

Now, let us show that the set $\mathcal{I} = \{e \in M : \exists n \in \mathbb{N}, e < \min S_n\}$ proves the theorem. First, it clearly is an initial segment of M. The step 3. where we remove the minimum of S_{3i+2} ensures that $\mathcal{I}$ is closed by successor, so it is a cut. Let E be an M-finite set with $|M| \in \mathcal{I}$. Then, at some point we have $E_i = E$ and $|E_i| < \min S_{3i+1}$. By step 2. we shrink the set to S_{3i+2} so that there is no element of E in $]\min S_{3i+2}, \max S_{3i+2}]$. Therefore, $E \cap \mathcal{I}$ is bounded in $\mathcal{I}$.

It remains to show that $(\mathcal{I}, \mathrm{Cod}(M/\mathcal{I})) \models \mathrm{HT}$. Let c be an M-finite 2-coloring of $\mathcal{I}$. As c is M-finite, there exists an i such that $c = c_i$. We have that S_{3i+1} is such that $\mathrm{FS}(S_{3i+1})$ is monochromatic for c. Therefore $S_{3i+1} \cap \mathcal{I} \in \mathrm{Cod}(M/\mathcal{I})$ is a solution to c which is unbounded in $\mathcal{I}$, again by the action of step 3. ■

We are now able to prove Theorem III.71, that we recall here.

▶ **Theorem** (Patey, Yokoyama [58]). HT *is* Π^1_1 *conservative over* $\mathrm{ACA}_0 + \{n\text{-FinHT} : n \in \omega\}$.

Proof. First, we proved that HT implies ACA_0 over RCA_0 in Theorem III.34. Therefore, by compactness, $\mathrm{HT} + \mathrm{RCA}_0$ proves n-FinHT for every $n \in \mathbb{N}$, so we only have to prove that $\mathrm{HT} + \mathrm{RCA}_0$ do not prove any additional Π^1_1 statement.

Let Φ be a Π^1_1 formula such that Φ is not provable in $\mathrm{ACA}_0 + \{n\text{-FinHT} : n \in \omega\}$; say of the form $\forall X \in \mathcal{P}(\mathbb{N}), \exists i, \forall j, \varphi(i, j, X)$, where φ is Δ^0_1 (here, we used ACA_0 to be able to put Φ in such a form). By the completeness theorem, let $\mathcal{M}$ be a model of $\mathrm{ACA}_0 + \{n\text{-FinHT} : n \in \omega\} + \neg\Phi$. Fix some $X \in \mathcal{M}$ such that $\mathcal{M} \models \forall i, \exists j \neg\varphi(i, j, X)$. Define in $\mathcal{M}$ the following sequence recursive in X:

1. $x_0 = 0$, and
2. x_{n+1} is the smallest integer strictly greater than x_n such that $\forall i \leq x_n, \exists j \leq x,\ \neg\varphi(i, j, X)$.

Let $X = \{x_i : i \in M\}$. By RCA_0, $X \in \mathcal{M}$, and it is unbounded as (x_i) is strictly increasing and therefore $x_i \geq i$. The set $\{a \in M :$ there is an $\mathcal{M}$-finite a-HT-large subset of $X\}$ is Σ^0_1-definable and contains all the integers, therefore by overspill there must exist some non-standard a and an M-finite set $A \subseteq X$ with A being a-HT-large.

By Lemma III.75, let $(\mathcal{I}, \mathrm{Cod}(M/\mathcal{I}))$ be a model of $\mathrm{HT} + \mathrm{RCA}_0$ with A being infinite in $\mathcal{I}$. As $A \subseteq X$ is unbounded in $\mathcal{I}$, $\mathcal{I}$ must be a model of $\exists X, \forall i, \exists j,\ \neg\varphi(X, i, j)$. Therefore, $\mathrm{HT} + \mathrm{RCA}_0$ cannot prove Φ. ■

III.4.4 Ordinal analysis of n-FinHT

In this section, we analyze how large a set must be to contain a large set of monochromatic finite sum, for a given notion of largeness. The idea would be to bound n-FinHT by some ordinal below the ordinal-theoretic of ACA, which is ϵ_{ϵ_0} ([4]). By Corollary III.71.1, this would separate HT from ACA^+_0. In practice, we will bound n-FinHT with iteration of $\oplus$ and $\otimes$ rather than α-largeness for a specific ordinal, however an ordinal could be recovered using Proposition III.64.

▶ **Remark III.76** ("$\mathbb{N}$ is sparse"). We consider from now on that any set $X \subseteq_{\mathrm{FS}} \mathbb{N}$ is such that $\forall a \in X$, we have $(\sum_{b \in X, b < a} b) < a$.

Note that we are now interested in finite sets, therefore in this section we weaken our notion of right-match and full-match to finite sets.

Constructing finite right-matches

In this section, we want the same kind of result as Theorem III.67, for the construction of right-matches. However, if A is some large set, we will not look for a large right-match S with $S \subseteq A$, but rather with $S \subseteq_{\mathrm{FS}} A$. Therefore, we need this first result, dealing with finite sums.

Theorem III.77

If Γ is a largeness notion and $Z = Z_0 \sqcup Z_1 \sqcup \cdots \sqcup Z_{k-1}$ is a witness of $\Delta \otimes (\Gamma \oplus 1)$-largeness, then any $(x_i) \in \prod_{i<k} \mathrm{FS}(Z_i)$ is such that $\{x_i : i < k\}$ is Γ-large.

Proof. Let $(y_i)_{i<k} \in \prod_{i<k} Z_i$ and $(x_i)_{i<k} \in \prod_{i<k} \mathrm{FS}(Z_i)$. We have that $\{y_i : i < k\}$ is $\Gamma \oplus 1$-large, therefore $\{y_i : 0 < i < k\}$ must be Γ-large. As $x_i < y_{i+1}$ for every $i < k-1$, the set $\{x_i : i < k-1\}$ is Γ-large, which implies the result. ■

We use this to tackle the "sum by two" drop of largeness:

▶ **Corollary III.77.1.** *If Γ is a largeness notion and $Z = \{x_i : i < 2k\}$ is $2 \otimes (\Gamma \oplus 1)$-large, then $\{x_{2i} + x_{2i+1} : i < k\}$ is Γ-large.*

We define the exact largeness that we need in the proof of Corollary III.30.1 for a set to have a Γ-large right-match.

Definition III.78

Let Γ be a largeness notion.
- We define RM_0-Γ to be Γ.
- We define RM_{k+1}-Γ to be $(\mathrm{RM}_k\text{-}\Gamma) \otimes [2 \otimes ((\mathrm{RM}_k\text{-}\Gamma) \oplus 1) \oplus 1]$.
- Finally, we say that Z is RM-Γ-large if Z is $\mathrm{RM}_{\min Z}$-Γ-large (we will always have less colors than $\min Z$)

Theorem III.79

Let Γ be a largeness notion. If $\mathrm{FS}(Z)$ is k-colored and $\mathrm{RM}_k(\Gamma)$-large, then
- Either there exists $F, Y \subseteq_{\mathrm{FS}} Z$ a right-match such that Y is Γ-large,
- or there exists $Y \subseteq_{\mathrm{FS}} Z$, a Γ-large set such that $\mathrm{FS}(Y)$ is monochromatic.

Proof. We prove the result by the following induction on k: When Z is only 1-colored, the result is trivial.

Suppose the result for k colors, and Z is $(\mathrm{RM}_k\text{-}\Gamma) \otimes [2 \otimes ((\mathrm{RM}_k\text{-}\Gamma) \oplus 1) \oplus 1]$-large and $k+1$-colored. By the largeness of Z, it can be decomposed into $Z_0 \sqcup \cdots \sqcup Z_k$. We build a sequence of elements from $z_i \in Z_i$ (almost) avoiding one color, such that if at some point we cannot continue, then we already have a sufficiently large right-match. Suppose $z_0, \cdots, z_{i-1}$ has been defined. We split the construction in two cases:
- If $\mathrm{FS}(\{z_a : a < i\})$ together with Z_i make a right-match for color 0. As Z_i is RM_k-Γ-large, by induction hypothesis [a], there exists F, Y a right-match for all other colors, with $Y \subseteq_{\mathrm{FS}} Z_i$ Γ-large. But then, $F \cup \mathrm{FS}(\{z_a : a < i\})$ is a right-match.
- Otherwise, let $z_i \in \mathrm{FS}(Z_i)$ of color 0 be such that $\forall a \in \mathrm{FS}(\{z_a : a < i\})$, $c(z_i + a) \neq 0$, and continue the construction.

If the construction continues until the z_i's are defined for $i \leq k$, then by Theorem III.77,

$\{z_i : i \leq k\}$ is $2 \otimes (\mathrm{RM}_k\text{-}\Gamma \oplus 1)$-large, and by Corollary III.77.1, $Y_0 = \{z_{2i} + z_{2i+1} : i < k/2\}$ is RM_k-Γ-large. But as $\mathrm{FS}(Y_0)$ never gets color 0, it is k-colored and we can apply the induction hypothesis.

a. There is a small abuse here, as Z_i is $k+1$-colored. To be accurate we would have to separate the total number of colors to the number of colors we want to have a right-match on, but we omit this for the sake of concision! ■

Constructing finite full-matches

Now that we can construct large right-matches, we need to iterate the construction to get full-matches. For example, if we have a set that is RM-(RM-($\cdots$-RM-k)$\cdots$)-large where there is k-many "RM", then we can iterate the construction of k right-matches, and thus obtain a full-match of size k (or if it fails at some point, a solution of size at least k).

However, instead of a fixed number of iterations, that gives us a full-match of a fixed size, we need a largeness notion. For example, to obtain a full-match that is ω-large, we roughly need Z to be $\mathrm{RM}^{\min Z}$-ω-large.

For this to work with a more general largeness notion than ω-largeness, we need to define the exponentiation of two largeness notions.

An exponentiation of largeness notions

We start by defining $\Gamma^{\{\Delta\}}$, our goal being to define RM^{Δ}-Γ. Going from the first to the second will be straightforward, following the same idea.

Remember that $\Gamma^{\{k\}}$ is $\Gamma \otimes (\Gamma^{\{k-1\}})$. We need to replace k by Δ. For example, replacing k with ω seems easy: the number of times that we split is $\min Z$, that is a set Z is $\Gamma^{\{\omega\}}$-large if it is $\Gamma^{\{\min Z\}}$-large.

To do this for an arbitrary given largeness notion Δ, we define a well-ordering on finite sequences of integers.

Definition III.80

Let Γ be a largeness notion, and $\bar{x}$ an increasing sequence of integers. We say that Z is $\Gamma - \bar{x}$-large if $Z \cup \{x : x \in \bar{x}\}$ is Γ-large.

This notation is not necessary, but motivates the notation of Theorem III.83. We have to be careful that $\Gamma - \bar{x}$ may not be a largeness notion.

Definition III.81 (The order $<_\Gamma$)

The order $<_\Gamma$ is the order on finite increasing sequences, generated by $\bar{x} >_\Gamma \bar{y}$ if $\bar{y}$ strictly extends $\bar{x}$ as sequences, and $\{x : x \in \bar{x}\}$ is not Γ-large.

▶ **Proposition III.82.** *Let Γ be a largeness notion. Then $<_\Gamma$ is well-founded.*

Proof. Suppose there is an infinite decreasing sequence for $<_\Gamma$. This defines an infinite set, by properties of largeness it must be large and some finite part must also be large. But then, this finite part cannot have any extension that is $<_\Gamma$ smaller. ■

Using this proposition, we are able to define the exponentiation by induction.

Definition III.83

Let Δ and Γ be largeness notions, and $\bar{x}$ be an increasing sequence. We define the $\Gamma^{\{\Delta-\bar{x}\}}$*-largeness* by induction on $<_\Delta$, using Proposition III.82:

— If $\bar{x}$ is a minimal element for $<_\Delta$ (that is, $\{x : x \in \bar{x}\}$ is Δ-large), then every set is $\Gamma^{\{\Delta-\bar{x}\}}$-large.

— Otherwise, we say that Z is $\Gamma^{\{\Delta-\bar{x}\}}$-large if $Z \setminus \min Z$ is $\Gamma \times \Gamma^{\{\Delta-\bar{x}^\frown \min Z\}}$-large, where $\bar{x}^\frown \min Z$ is the addition of $\min Z$ in the end of the sequence $\bar{x}$.

— Finally $\Gamma^{\{\Delta\}}$-largeness is $\Gamma^{\{\Delta-\epsilon\}}$-largeness.

We can do the same with the "operation" RM:

Definition III.84

Let Δ and Γ be largeness notions, and $\bar{x}$ an increasing sequence. We define the $\mathrm{RM}^{\Delta-\bar{x}}$-$\Gamma$-largeness by induction on $<_\Delta$, using Proposition III.82:

— If $\bar{x}$ is a minimal element for $<_\Delta$ (that is $\bar{x}$ is Δ-large) then $\mathrm{RM}^{\Delta-\bar{x}}$-$\Gamma$-largeness is Γ-largeness.

— Otherwise, we say that Z is $\mathrm{RM}^{\Delta-\bar{x}}$-$\Gamma$-large if $Z \setminus \min Z$ is RM-$\mathrm{RM}^{\Delta-\bar{x}\min Z}$-$\Gamma$-large.

— Finally RM^{Δ}-Γ-largeness is $\mathrm{RM}^{\Delta-\epsilon}$-largeness.

▶ **Fact III.85.** 1. For every largeness notions Γ and Δ, $\Gamma^{\{\Delta\}}$ and RM^{Δ}-Γ are largeness notions.

2. $\Gamma^{\{k\}}$ as defined in Definition III.62 is implied by $\Gamma^{\{k\}}$ as defined in Definition III.83 (where k is interpreted as the k-largeness notion).

   Both notions are not equivalent as at each iteration of the decomposition into Γ subsets, we remove one element. In fact, $\Gamma^{\{k\}}$ would be equivalent to the definition by induction $\Gamma^{\sim k+1} = (\Gamma^{\sim k} + 1) \otimes \Gamma$.

3. RM^{k}-Γ as in Definition III.84 implies RM-RM-$\cdots$-Γ, where there is k times RM.

Proof. We only prove the items for the exponentiation, as the same results for RM follow exactly the same proof.

We prove that $\Gamma^{\{\Delta\}}$ is a largeness notion by induction, using the well-ordering $<_\Delta$ and Proposition III.82. For the base case, a notion where all sets are large is a largeness notion.

Let us show the successor case (there is no limit point in $<_\Delta$). First, by another induction, $\Gamma^{\{\Delta-\bar{x}\}}$ is included in $\Gamma^{\{\Delta-\bar{y}\}}$ when there is an injection f from $\bar{x}$ to $\bar{y}$ such that $f(x) \leq x$. Therefore, $\Gamma^{\{\Delta-\bar{x}\}}$ is stable under superset and also satisfies left-preservation.

Item 2. is clear by induction, as Z is Γ^k-large by definition if $Z \setminus \min Z$ is $\Gamma \otimes \Gamma^{\{k-1\}}$-large (recall that a set is k-large if and only if it has k elements). ■

The finite full-match construction

Using the tools from the previous section, we are able to give a sufficiently powerful largeness notion for constructing Γ-large full-matches.

Definition III.86

Let Γ be a largeness notion. We define:

1. FM_0-Γ to be Γ, and FM_1-Γ to be $\mathrm{RM}^{\Gamma\oplus 1}$-$\Gamma$
2. For $k \geq 1$, FM_{k+1}-Γ to be FM_1-(FM_k-Γ).

We say that Z is FM-Γ-large if it is $\mathrm{FM}_{\min Z}$-Γ-large.

And now the theorem:

Theorem III.87

Let Γ be a largeness notion, $Z \subseteq \mathbb{N}$ an FM-Γ-large set and c a k-coloring set with $k < \min Z$. Then, there exists $F, Y \subseteq_{\mathrm{FS}} Z$ a Γ-large full-match.

Proof. We should be comfortable with the construction of full-matches given in the proof of Theorem III.13 and studied in Theorem III.31. Therefore we only detail the analysis of largeness. As usual, we first prove existence of full-match for one specific color.

▶ **Lemma III.88.** *Suppose that Γ is a largeness notion, and Z is a k-colored FM_1-Γ-large set. Then, there exists $F, Y \subseteq_{\mathrm{FS}} Z$ a full-match for color 0, where Y is Γ-large.*

Proof. We use Theorem III.79 to construct sequences $F_0, F_1, \cdots$ and $Z_0 \supseteq_{\mathrm{FS}} Z_1 \supseteq_{\mathrm{FS}} \cdots$ and $c_0, c_1, \cdots$ such that F_i, Z_i is a right-match for c_i, c_{i+1} is the refinement of c_i corresponding to the right-match F_i, and Z_i is $\mathrm{RM}^{\Gamma\oplus 1-\langle \min Z_j : j<i\rangle}$-$\Gamma$-large. Note that we can always apply Theorem III.79 as the number of colors of c_i is less than $\min Z_i$.

If at some point, $(\mathrm{FS}(F_i), Z_i)$ is a full-match for color 0, we have proven the theorem as every Z_i is at least Γ-large. Otherwise, we continue until, $\{\min Z_i : i < N\}$ is $\Gamma \oplus 1$-large, and $(\mathrm{FS}(F_N), Z_N)$ is still not a full-match for color 0. But in this case there must exist a sequence $\langle x_0, \cdots, x_N\rangle$ with $x_i \in F_i$ with $\mathrm{FS}(\{x_i\})$ avoiding color 0, just as in the infinite case. As $F_i < \min Z_{i+1}$ and by left-preservation, the path must be Γ-large . ■

The theorem is clear from the lemma. ■

Constructing finite Hindman's solution

The construction of large solutions to Hindman's Theorem is very similar to the construction of large full-matches, except that the base step is the construction of a full-match and not of a right-match. Therefore, this section and the previous one are very similar.

Definition III.89

Let Δ and Γ be largeness notions, and $\bar{x}$ be an increasing sequence. We define the $\mathrm{FM}^{\Delta-\bar{x}}$-$\Gamma$-*largeness* by induction on $<_\Delta$, using Proposition III.82:

— If $\bar{x}$ is a minimal element for $<_\Delta$ (that is $\bar{x}$ is Δ-large) then $\mathrm{FM}^{\Delta-\bar{x}}$-$\Gamma$-largeness is Γ-largeness.

— Otherwise, we say that Z is $\mathrm{FM}^{\Delta-\bar{x}}$-$\Gamma$-large if $Z \setminus \min Z$ is FM-$\mathrm{FM}^{\Delta-\bar{x}\min Z}$-$\Gamma$-large.

— Finally FM^{Δ}-Γ-largeness is $\Gamma^{\Delta-\epsilon}$-largeness.

Definition III.90

Let Γ be a largeness notion. We define HT-Γ-largeness to be $\mathrm{FM}^{\Gamma\oplus 1}$-1-largeness.

And now the theorem:

Theorem III.91

Let Γ be a largeness notion, $Z \subseteq \mathbb{N}$ an HT-Γ-large set and c a k-coloring set with $k < \min Z$. Then, there exists $Y \subseteq_{\mathrm{FS}} Z$ a Γ-large set with $\mathrm{FS}(Y)$ monochromatic.

Proof. We should be comfortable with the construction of monochromatic IP sets given in the proof of Theorem III.15 and studied in Theorem III.32. Therefore we only detail the analysis of largeness.

We use Theorem III.87 to construct sequences $F_0, F_1, \cdots$ and $Z_0 \supseteq_{\mathrm{FS}} Z_1 \supseteq_{\mathrm{FS}} \cdots$ and $c_0, c_1, \cdots$ such that F_i, Z_i is a full-match for c_i, c_{i+1} is the refinement of c_i corresponding to the full-match F_i, and Z_i is $\mathrm{FM}^{\Gamma\oplus 1-\langle \min Z_j : j<i\rangle}$-1-large. Note that we can always apply Theorem III.87 as the number of colors of c_i is less than $\min Z_i$.

We continue until $\{\min Z_i : i < N\}$ is $\Gamma \oplus 1$-large. As in the infinite case, there must exist a sequence $\langle x_0, \cdots, x_N\rangle$ with $x_i \in F_i$ and $\mathrm{FS}(\{x_i\})$ monochromatic. As $F_i < \min Z_{i+1}$ and by left-preservation, the path must be Γ-large . ■

III.4.5 Conclusion on the ordinal analysis

It is clear using Theorem III.91 that n-FinHT is bounded by HT-(HT-($\cdots$-(HT-ω)))-largeness, where we iterated n times the HT operation on largeness. However, it remains to show what ordinal-theoretic is needed to prove the existence of these notions, and find whether is it below ϵ_{ϵ_0} or not.

Informally, the construction suggests that the ordinal associated with the construction of right-matches is ω^ω. Indeed, HM-Γ-largeness corresponds to an unbounded iteration of sums and product from Γ. Therefore, most likely when going back to ordinal largeness notions, HM-α-largeness is bounded by α^ω.

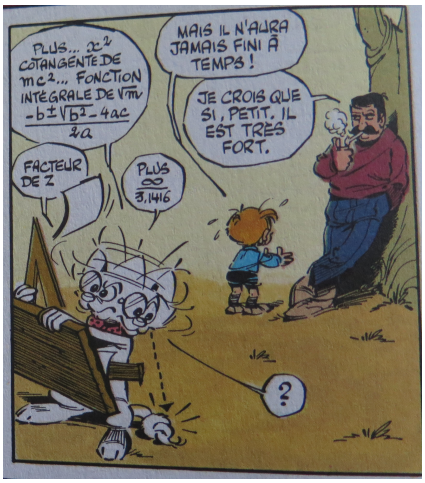
Similarly, the construction of full-matches is associated with ϵ_0, as it is the unbounded iteration of HM. Therefore, we should get that FM-ω-largeness is bounded by ϵ_0.

The construction of solutions to Hindman's Theorem following the same pattern, it should require ϵ_{ϵ_0}-largeness to bound n-FinHT. This ordinal is exactly the ordinal-theoretic of ACA, and is therefore too high to leave us a hope to use Corollary III.71.1 to separate HT from ACA_0^+. An important modification of the construction would be needed to conclude using these techniques.

Although the proof-theoretic aspects is not complete, in particular the bounding of $\oplus, \otimes$-constructed notions, as well as exponent-constructed notions, we believe that the ordinal analysis of Hindman's Theorem in this setting is still of interest.

Chapter IV

AC and ATR in Weihrauch lattice



Gotlib, RàB T2. Le matou matheux

Despite its importance in the community of computability theorists, the "implication over RCA_0" is really a notion of reduction from proof theory. It is used to compare theorems in terms of "provability strength", and to find the minimal axioms needed for a proof. This "provability reduction" is linked to computability as RCA_0 consists exactly of the axioms needed for computations to exist in a theory, and comparing computational implication of two principles is useful to separate them over RCA_0. For instance, using the theory of PA degrees is a way to separate WKL_0 from ACA_0, by creating a model of WKL_0 containing only low degrees.

Conversely, the implication over RCA_0 is not a good way to compare the computational content of principles. There are several reasons for this. We mention three of them:

1. A finite amount of non-uniformity is allowed in proofs, as when using the law of excluded middle. For instance, it is provable using WKL_0 that for any tree T, either there exists a maximum depth for T, or there exists a path for T. However, the computational strength usually associated to WKL_0 is those of PA degrees, and incomplete PA degrees cannot decide ill-foundedness of binary trees.
2. We want the computational strength of a principle to be independent from the underlying theory, that is from the model. Sometimes, it may be the case that the interpretation of a principle truly depends on the model in an unexpected way. For instance, we have seen in Theorem II.18 that if $\exists X : \Phi(X)$ is a Σ^1_1 formula, then the formula $\exists X \in \mathrm{HYP} : \Phi(X)$ is Π^1_1. This justifies some differences on analytical principles between provability reductions and computable reductions, as seen in Theorem IV.80 and in the paragraph following this theorem.
3. In order to be fine-grained in our comparison on computational power, we don't want the principle to be used multiple times. For instance, we want to make a distinction between the computational power of solving RT^2_2 once and the computational power of solving RT^2_2 twice. It is not obvious to quantify the number of times a principle is used in a proof: for

instance, a principle may appear only once in a proof, but in an induction, suggesting the principle is in fact used many times.

Therefore, in order to compare computational strength of theorems, we need to devise other reductions. One of the most successful is Weihrauch reduction, which has been actively studied for instance in [16, 14, 10, 12, 11] as a continuation to the reverse mathematics program. We start this chapter by presenting many examples and tools around Weihrauch reductions. We focus on the problem of choosing a particular object in an analytic set. We then study the computational power of ATR_0, introducing two new Weihrauch problems, inspired by the arithmetical transfinite recursion and its method of pseudo-hierarchy. We show how all these principles are linked with the Axiom of Choice. In particular, we use the principle ATR_2 to separate the analytical versions of the axioms of choice and dependent choice, solving a question of [47, 13].

IV.1 Reducibilities on multivalued function

Turing reducibility is very convenient in order to computably compare elements of the Cantor or the Baire space, using a Turing Machine with an oracle. However, it is not suitable to compare higher order objects, such as subsets of the Cantor or the Baire space, as they do not fit in the tape. One way to counter this misfitting is to consider the "decidability" version: a set $B \subseteq \omega^\omega$ is decidable by a Turing machine if there exists one that answers the question of whether its infinite input is in the set or not.

But, even if this notion of decidability can be view as a version of being computable for subsets of the Baire space, it does not relativize to subsets of the Baire space by using an oracle tape for querying the oracle: if we had an oracle tape solving the decidability problem of a set, it would take an infinite amount of time just to write the question! So there is no easy way to define a notion of "$\mathcal{A}$ is decidable relative to $\mathcal{B}$" for subsets of the Baire space. Note that using infinite time Turing machines, it is easy to define such a notion.

There still exists reducibilities that can compare computational strength of subsets of Cantor or Baire space: Muchnik reducibility, and its uniform counterpart, Medvedev reducibility. The idea is to use a representation that fits in the tape: a set of reals is represented by any of its member. Therefore for $\mathcal{A} \subseteq \omega^\omega$ to compute $\mathcal{B} \subseteq \omega^\omega$, the input may be any representation of $\mathcal{A}$, that is, an element of $\mathcal{A}$, and the output must be a representation of $\mathcal{B}$, that is an element of $\mathcal{B}$. Another point of view is to see a set as a problem, and its members as the solutions to the problem. A problem is easier than another if any solution of the latter computes a solution to the former.

IV.1.1 Medvedev and Muchnik reducibilities

Definition IV.1

Let $A, B \subseteq \omega^\omega$. We say that A is *Muchnik-reducible* to B, written $A \leq_w B$ if for every $y \in B$, there exists $x \in A$ such that $x \leq_T y$.
We say that A is *Medvedev-reducible* to B, written $A \leq_s B$ if there exists a single computable functional Φ such that for every $y \in B$, $\Psi^y \in A$.

Clearly Medvedev reducibility is stronger than Muchnik one, justifying the notation: w stands for "weak" while s is for "strong".

▶ **Proposition IV.2.** *Both Medvedev and Muchnik reducibilities are preorders on subsets of Baire space. The degree structures have a minimum and a maximum element.*

Proof. It is clear that they are reflexive, by the identity function. If every element of C compute (uniformly) and element of B, which itself (uniformly) computes an element of A, then every element of C (uniformly) computes an element of A.

The minimum element is the degree of the set of the Baire space that contains a computable element, while the maximal is the empty set. ■

Note that the degree structure of both reducibilities embed the Turing degrees, by considering the singletons.

These reducibilities allow to compare subsets of Baire or Cantor space. An example of an interesting degree is MLR, the set of Martin-Löf Randoms (see Definition V.7): the degrees below it are the sets such that any source of (Martin-Löf) randomness allows to compute one member. No Turing degree is below MLR by a Theorem of Sacks [57].

However, these reducibilities are of less interest in the case of the Reverse Mathematical program. Indeed, the only theorems that can be naturally embedded into such degrees are theorems of the Σ^1_1 form: If $\exists X, \Psi(X)$ is a true Σ^1_1 sentence, then $\{X \in 2^\omega : \Psi(X)\}$ is a non-empty set with a Medvedev and Muchnik degree.

The theorems with a Σ^1_1 form are a rather small class of interesting theorems, however many interesting theorems are of the Π^1_2 form, such as all the big fives and Hindman's theorem. To these theorems correspond a multivalued function: If $\forall X \exists Y, \Psi(X, Y)$, is a true statement, then the natural corresponding multivalued function is $X \mapsto \{Y \in 2^\omega : \Psi(X, Y)\}$. Instead of subsets of Baire space, we need a way to compare multivalued functions. The best candidate for our purpose, Weihrauch reduction, has recently been given focus by reverse mathematicians who wanted a more computational approach.

IV.1.2 Weihrauch Reduction

All the reductions of this subsection are on multivalued functions. We see multivalued functions as problems, where the input are instances of the problem and the output are solutions to the instances. Therefore, we will often use this terminology in the definition, even by using "problems" and "multivalued functions" interchangeably.

We start with defining the strong Weihrauch reduction. This reduction, from a problem P to a problem Q is quite simple and proceed in two steps: Given an instance I of P, it computes an instance I' of Q, and given a solution S' to I', it computes a solution S to I. More formally, here is the definition:

Definition IV.3

Let $P, Q \subseteq: 2^\omega \to \mathcal{P}(2^\omega)$ be partial multivalued functions. Then we say that P is strongly Weihrauch reducible to Q, in symbols

$$P \leq_{\mathrm{sW}} Q$$

if there exists f, g partial computable functions such that

$$\forall x \in \mathrm{dom}(P), g(x) \in \mathrm{dom}(Q) \wedge \forall y \in Q(g(x)), f(y) \in P(x)$$

In other words,

$$\forall x \in \mathrm{dom}(P), f[Q(g(x))] \subseteq P(x)$$

We can see this reduction as: if we had a solver for a problem Q, we could solve P by computably modifying the input and the output of the solver. However, this reduction has some

unnatural features. For instance, one would expect that the constant 0 function and the identity function have the same degree, as their computational difficulty are the same: the computable ones. However, this is not the case, as in strong Weihrauch reduction, the input of the initial problem is forgotten when it comes to compute a solution.

Another way of seeing this is: we would have a Turing Machine that gets an input, computes a question to ask to the solver. Then, everything is removed from its tape before giving the answer from the solver. It seems more natural to leave the initial input to the second stage of the reduction, as in Weihrauch reduction.

Definition IV.4

Let $P, Q \subseteq: 2^\omega \to \mathcal{P}(2^\omega)$ be partial multivalued functions. Then we say that P is Weihrauch reducible to Q, in symbols

$$P \leq_{\mathrm{W}} Q$$

if there exists f, g partial computable functions such that

$$\forall x \in \mathrm{dom}(P), g(x) \in \mathrm{dom}(Q) \wedge \forall y \in Q(g(x)), f(y, x) \in P(x)$$

In natural language, given any instance x of the problem P, we are able to compute an instance $g(x)$ of Q such that for any solution y of $g(x)$, then $f(y, x)$ is a solution for x.

Note that there also exists versions of $\leq_{\mathrm{sW}}$ and $\leq_{\mathrm{W}}$ which are non-uniform: the two functionnals f and g depends on the input and the solution. These reductions are called computable and strongly computable reductions. However, we will never use them in this Thesis.

We show that this reduction is sound, and we are able to speak about Weihrauch degree by considering the usual equivalence classes:

▶ **Proposition IV.5.** *$\leq_{\mathrm{W}}$ and $\leq_{\mathrm{sW}}$ are preorders.*

Proof. We only show the result for $\leq_{\mathrm{W}}$, as it is even easier for $\leq_{\mathrm{sW}}$. We need to show reflexivity and transitivity. If F is any multivalued function, then clearly identity and the first projection are witnesses of $F \leq_{\mathrm{W}} F$.

Now suppose $F \leq_{\mathrm{W}} G$ and $G \leq_{\mathrm{W}} H$, via witnesses (f_0, g_0) for the first one and (f_1, g_1) for the second. Then, we claim that $F \leq_{\mathrm{W}} H$ via (f, g) where $f = f_1 \circ f_0$ and $g(y, x) = g_0(g_1(y, f_0(x)), x)$. Indeed, suppose x is an instance of F. Then $f_0(x)$ is an instance of G and $f_1(f_0(x))$ is an instance of H. So suppose y is a solution to the instance $f_1 \circ f_0(x)$ of problem H. Then, $g_1(y, f_0(x))$ is a solution to the instance $f_0(x)$ of problem G. But then, $g_0(g_1(y, f_0(x)), x)$ is a solution to the instance x of the problem F. ■

In the beginning of this chapter, we argued that the "implication over RCA_0" is not suitable to compare computational implication of theorems, and we gave three reasons for this. The first one was the use of the law of excluded middle in proofs, which introduced non uniformity. The Weihrauch reduction is uniform (but not the computable reduction).

The second thing was that computational implications should be independent of the base theory. This is the case with Weihrauch reduction, which takes place in the ambient model of mathematics: We can use very complex axioms to show that a reduction exists, but the reduction itself has to be simple.

The third complaint was the resource insensitivity of $\leq_0$. In a proof, it is very unclear how to quantify the number of times an axiom is used. In the case of Weihrauch reduction, it is very clear and only one application of the solver is used. We will see that there exists ways to consider principles where multiple applications are allowed.

We now give a few examples of multivalued functions and degrees that are of interest for Weihrauch reducibility:

▶ **Examples IV.6.**

1. If $A \subseteq \omega^\omega$ is any set, we write f_A for the multivalued function defined only on 0^ω and such that $f_A(0^\omega) = A$. If $A, B \subseteq \omega$, then $A \leq_s B$ if and only if $f_A \leq_{\mathrm{W}} f_B$: This show that Medvedev degrees are embedded into Weihrauch degrees.
2. Any f is single valued and computable is in the degree of computable functions, written id. Note that this degree of computable functions does not exists in the strong Weihrauch reduction.
3. The multivalued function IVT, which given a continuous function with $f(0) \times f(1) < 0$, returns the elements $x \in [0; 1]$ such that $f(x) = 0$. There must exist such an x by the Intermediate Value Theorem, justifying the name.
4. The multivalued function WKL which to any infinite binary tree T associates the elements of $[T]$. The degree of this function is considered to be the Weihrauch equivalent of the WKL_0 principle of reverse mathematics.
5. The single valued function lim, which to any converging element of ω^ω, return the limit. The degree of this function is considered to be the Weihrauch counterpart of the ACA_0 principle from reverse mathematics.
6. The single valued function Π^1_1-CA which to any Π^1_1 formula ϕ, associates the set $\{n \in \mathbb{N} : \phi(n)\}$. The formula is allowed to have parameters.

In the examples we gave the Weihrauch degrees equivalent of four principles out of the Big Five: The Arithmetical Transfinite Induction is missing. We will discuss more on the equivalent of ATR_0 in the Weihrauch lattice later.

IV.1.3 Algebraic structure of the Weihrauch lattice

Due to its fine-grained nature, the Weihrauch reducibility contains many natural ways to combine degrees, making the Weihrauch quasi-order an interesting algebraic structure.

Definition IV.7

Let P, Q be a multivalued functions. Then $P \times Q$ is the problem where instances are couples $(i_0, i_1) \in \mathrm{dom}(P) \times \mathrm{dom}(Q)$ with solutions $(s_0, s_1) \in P(i_0) \times Q(i_1)$.

Note that as the Weihrauch reduction is resource sensitive, $P \times P$ may not be equal to P. When $P \times P \equiv_{\mathrm{W}} P$, then we say that P is *idempotent*.

For instance, WKL is idempotent: given two infinite binary trees T_0 and T_1, one can build an infinite binary tree where the even levels correspond to T_0 and the odd ones to T_1. Formally, $\sigma \in T$ if $\sigma \circ (i \mapsto 2i) \in T_0$ and $\sigma \circ (i \mapsto 2i + 1) \in T_1$. Given a path on this binary tree T, it is easy to recover a path from both binary tree T_0 and T_1.

The degree $P \times P$ allows two independent queries for an answer to P. To express an arbitrary number of independent queries, or an infinite number of independent queries, to the same problem P, then we need the following:

Definition IV.8

Let P be a multivalued function. Then P^* is the finite parallelization of P: an instance is a finite sequence $\langle i_0, \ldots, i_n \rangle$ where $i_k \in \mathrm{dom} P$ for all $k \leq n$, and a solution is a product of solution for every i_k.

$\widehat{P}$ is the parallelization of P: an instance is an infinite sequence $(i_k)_{k\in\mathbb{N}}$ where $i_k \in \mathrm{dom}P$ for all $k \in \mathbb{N}$, and a solution is a product of solutions for every i_k. When $P \equiv_{\mathrm{W}} \widehat{P}$ we say that P is *parallelizable*.

For instance, for every P we have $\widehat{P} \equiv_{\mathrm{W}} \widehat{\widehat{P}}$, that is $\widehat{P}$ is parallelizable: It is easy to "flatten" a sequence of sequences to a unique sequence, with the same elements.

In all the previous constructions, multiple queries must be independent from each others. However, proofs of usual mathematics are usually more sequential than parallel: we often apply a theorem of a Π^1_2 form by instantiating it with something depending on the application of another Π^1_2 theorem: in this case, the instance of one query must depend on the answer to another.

In order to define this sequentiallity of two Weihrauch degrees, one cannot just compose the multivalued functions, as the output of the first one might need to be modified before being given as input to the second one. This is reflected by the following construction:

Definition IV.9

Let P, Q be multivalued functions. Then $P * Q$ is the compositional product defined by $P * q(i, e) = (\mathrm{id} \times P) \circ \Phi_e \circ Q$.

There are many more algebraic properties and construction in the lattice of Weihrauch degrees, but in this chapter we won't use them. An interested reader can refer to [15].

IV.1.4 Arithmetical Weihrauch reduction

In this thesis, we are mainly interested by objects of high complexity. Sometimes, Weihrauch complexity is a little bit too fine-grained: some degrees that are conceptually close can be distinct for Weihrauch reduction, but equivalent when given a few more power to the function modifying the input and the output. Therefore, we define the arithmetical Weihrauch reduction, where the functionnals are arithmetic instead of being recursive.

Definition IV.10

Let $P, Q \subseteq: 2^\omega \to \mathcal{P}(2^\omega)$ be partial multivalued functions. Then we say that P is arithmetically Weihrauch reducible to Q, in symbols

$$P \leq^{\mathrm{a}}_{\mathrm{W}} Q$$

if there exists f, g partial functions that are arithmetically definable, and such that

$$\forall x \in \mathrm{dom}(P), g(x) \in \mathrm{dom}(Q) \wedge \forall y \in Q(g(x)), f(y, x) \in P(x)$$

We now go on to define the most studied classes of Weihrauch degrees, the degrees corresponding to a choice in a definable set.

IV.2 Choices in the Weihrauch lattice

IV.2.1 Single Choices

Although Weihrauch degrees have been introduced in this chapter to compare computational strength of theorems, inspired by the reverse mathematical program, Weihrauch reduction has been defined independently of this goal. It compares the difficulty of solving problems that sometimes cannot be translated into a natural theorem. One class of such problems is the "choice problem": given a code for a non empty set, output a member of the set. A corresponding theorem would be: "for every non empty set, the set has a member", an uninteresting theorem but an interesting degree in the Weihrauch hierarchy of computational power.

Definition IV.11

Let $\mathscr{X}$ be a recursively presented topological space, and $\mathcal{A} = (A_p)_{p\in 2^\omega}$ be a presentation of sets by elements of the Cantor space. Then $\mathcal{A}\text{-C}_{\mathscr{X}}$ is the multivalued function such that:

1. $\mathrm{dom}(\mathcal{A}\text{-C}_{\mathscr{X}})$ is the set $\{p \in 2^\omega : A_p \neq \emptyset\}$.
2. $\mathcal{A}\text{-C}_{\mathscr{X}}(p) = A_p$.

In other words, $\mathcal{A}\text{-C}_{\mathscr{X}}$ is the multivalued function which given a presentation p of a set A_p, outputs the elements of this set. When omitted, $\mathcal{A}$ will be the standard presentation of closed sets: A sequence (o_n) of code for the basic open sets that are removed from the closed set. In the case of Cantor and Baire space, this corresponds to the tree presentation, where a tree codes for its infinite path. We will call Σ^1_1-presentation a presentation of a closed set $A \subseteq \mathscr{X} \times \omega^\omega$, and this presentation will code the set $p(A)$ where p is the projection on the first coordinate.

Let us give a few examples:

▶ **Example IV.12.**

1. C_2 is the task of, given a co-enumeration of a non-empty set $D \subseteq 2$, to find an element of D. C_ω is the same with a co-enumeration of a non-empty set of integers.
2. C_{2^ω} is the task of, given an ill-founded tree $T \subseteq 2^{<\omega}$, to find an element of $[T]$. $\mathrm{C}_{\omega^\omega}$ is the task of, given an ill-founded tree $T \subseteq \omega^{<\omega}$, to find an element of $[T]$.
3. $\Sigma^1_1\text{-C}_{\mathscr{X}}$ is the task of, given a Σ^1_1 presentation of a set, to find an element of the set.
   It corresponds to the task of finding an element of $p(A)$, given a presentation of a closed set $A \subseteq \mathscr{X} \times \omega^\omega$, to find an element of $p(A)$, where p is the projection along the first coordinate. When $\mathscr{X} = \omega^\omega$ (respectively $\mathscr{X} = 2^\omega$), the input can equivalently be an ill-founded tree $T \subseteq (\omega \times \omega)^{<\omega}$ (respectively $T \subseteq (2 \times \omega)^{<\omega}$) and the output an element of $p([T])$.
4. $\Sigma^1_1\text{-tree-C}_{\mathscr{X}}$ is the task of, given a Σ^1_1-presentation of an ill-founded Σ^1_1 tree $T \subseteq \mathscr{X}^{<\omega}$, to find an element of $[T]$ [1]. We also write Σ^1_1-WKL for $\Sigma^1_1\text{-tree-C}_2$.

The next few theorems show how these principles relate under Weihrauch reducibility.

Theorem IV.13

$\mathrm{C}_{\omega^\omega} \equiv_{\mathrm{W}} \Sigma^1_1\text{-tree-C}_{\omega^\omega} \equiv_{\mathrm{W}} \Sigma^1_1\text{-C}_{\omega^\omega} \equiv_{\mathrm{W}} \Sigma^1_1\text{-C}_{2^\omega}$.

1. Note that a Σ^1_1-presentation is a presentation of a closed set, and therefore is a tree. So a Σ^1_1 presentation of a tree is a tree T such that its projection $p([T])$ is itself a tree.

Proof. It is clear that $\Sigma^1_1\text{-C}_{2^\omega} \leq_{\mathrm{W}} \Sigma^1_1\text{-C}_{\omega^\omega}$. Let c be the usual injection from ω^ω to 2^ω. Given a Σ^1_1 set $A \subseteq \omega^\omega$, then its image by c is a Σ^1_1 set of the Cantor space, such that any element from it uniformly computes an element of A just by using c^{-1}, so $\Sigma^1_1\text{-C}_{\omega^\omega} \leq_{\mathrm{W}} \Sigma^1_1\text{-C}_{2^\omega}$.

We continue by showing that $\Sigma^1_1\text{-C}_{\omega^\omega} =_{\mathrm{W}} \Sigma^1_1\text{-tree-C}_{\omega^\omega}$. If $T \subseteq \omega^{<\omega}$ is a Σ^1_1 tree, then the set $[T]$ is also Σ^1_1, as $X \in [T]$ if and only if $\forall n, X_{\restriction n} \in T$. Moreover, given a Σ^1_1 presentation of T, one can compute a Σ^1_1 presentation of $[T]$. So $\Sigma^1_1\text{-tree-C}_{\omega^\omega} \leq_{\mathrm{W}} \Sigma^1_1\text{-C}_{\omega^\omega}$.

Conversely, if $A \subseteq \omega^\omega$ is Σ^1_1, then let $T \subseteq \omega^{<\omega}$ be computable such that $A = p([T])$. T is also Σ^1_1, so let $\widetilde{T}$ be a presentation of T as a Σ^1_1 set. Given a path y of T by $\Sigma^1_1\text{-tree-C}_{\omega^\omega}$, one can obtain $x = p(y) \in A$. So $\Sigma^1_1\text{-C}_{\omega^\omega} \leq_{\mathrm{W}} \Sigma^1_1\text{-tree-C}_{\omega^\omega}$.

It is obvious that $\Sigma^1_1\text{-C}_{\omega^\omega} \geq_{\mathrm{W}} \mathrm{C}_{\omega^\omega}$. It only remains to prove $\Sigma^1_1\text{-C}_{\omega^\omega} \leq_{\mathrm{W}} \mathrm{C}_{\omega^\omega}$. Let A be a Σ^1_1 set, and let T be such that $p([T])$. Then, given any path $y \in [T]$ by $\mathrm{C}_{\omega^\omega}$, one can find an element of A just taking the projection $p(y) \in A$. ■

We now show that the only gap in the choice principles mentioned above happens between Π^0_1 and Π^0_2 choices. These version of choice are defined from Definition IV.11 using the usual coding of Π^0_n sets.

Theorem IV.14

For all $n \geq 2$, we have

$$\mathrm{C}_{2^\omega} \equiv_{\mathrm{W}} \Pi^0_1\text{-C}_{2^\omega} <_{\mathrm{W}} \Pi^0_n\text{-C}_{2^\omega} \equiv_{\mathrm{W}} \Sigma^1_1\text{-C}_{2^\omega}$$

Proof. We have $\mathrm{C}_{2^\omega} \equiv_{\mathrm{W}} \Pi^0_1\text{-C}_{2^\omega}$ just by definition. It is clear that $\Pi^0_n\text{-C}_{2^\omega} \leq_{\mathrm{W}} \Sigma^1_1\text{-C}_{2^\omega}$. By Theorem IV.13, for the right part we only have to show $\mathrm{C}_{\omega^\omega} \leq_{\mathrm{W}} \Pi^0_n\text{-C}_{2^\omega}$.

Let $T \subseteq \omega^{<\omega}$ be an ill-founded tree. We fix a computable bijection $(\sigma_n)_{n\in\mathbb{N}}$ from $\mathbb{N}$ to $\omega^{<\omega}$. Let e be a code for the set of codes for pruned subtrees of T. Formally, e is a code for the formula $\phi(C)$ being $\forall n, n \in C \Rightarrow \sigma_n \in T$ (inclusion in T) and $\forall n \in C, \forall m, \sigma_m \prec \sigma_n \Rightarrow m \in C$ (C is a tree) and $\forall n \in C, \exists m \in C : \sigma_n \prec \sigma_m \wedge \sigma_n \neq \sigma_m$ (C is pruned).

This formula is Π^0_2. Because T is ill founded, $\{C \in 2^\omega : \phi(C)\} \neq \emptyset$. Let $C \in 2^\omega$ be any such that $\phi(C)$. As C codes for a pruned tree, it computes its leftmost path. But as it codes a subtree of T, this path is an element of $[T]$.

Therefore, $C_{\omega^\omega} \equiv_W \Pi^0_2\text{-}C_{2^\omega}$.

It remains to prove the separation. We do so by comparing C_{2^ω} with C_{ω^ω}. There exists a recursive tree $T \subseteq \omega^{<\omega}$ with a non empty set of paths, but no hyperarithmetic path. Finding a path in such a tree cannot be reduced to finding a path in a tree of $2^{<\omega}$, as any infinite tree in $2^{<\omega}$ has a low path. ■

▶ **Corollary IV.14.1.** *For all $n, m \geq 2$, $\Pi^0_n\text{-C}_{2^\omega} \equiv_{\mathrm{W}} \Pi^0_m\text{-C}_{2^\omega}$*

▶ **Corollary IV.14.2.** *For all $n \geq 2$, $\Pi^0_n\text{-C}_{2^\omega} \equiv_{\mathrm{W}} \Pi^0_n\text{-C}_{\omega^\omega}$*

Recall that Σ^1_1-WKL is defined as $\Sigma^1_1\text{-tree-C}_{2^\omega}$. In contrast with Theorem IV.13, we have the following:

Theorem IV.15

$\mathrm{C}_{2^\omega} <_{\mathrm{W}} \Sigma^1_1\text{-WKL} <_{\mathrm{W}} \Sigma^1_1\text{-C}_{2^\omega}$.

Proof. The tree with a single path corresponding to $\emptyset'$ is Σ^1_1, while any recursive tree in $2^{<\omega}$ has a low path, establishing the separation $\mathrm{C}_{2^\omega} <_{\mathrm{W}} \Sigma^1_1\text{-WKL}$.

It is clear that Σ^1_1-WKL $\leq_{\mathrm{W}}$ Σ^1_1-C_{2^ω}, as if T is Σ^1_1, then $[T]$ is Σ^1_1 as well uniformly in T. We now have to show that Σ^1_1-WKL $\not\leq_W$ Σ^1_1-tree-$\mathrm{C}_{\omega^\omega}$, and conclude using Theorem IV.13.

We start by remarking the following: there exists a Σ^1_1 tree $T_0 \subseteq \omega^{<\omega}$ and a computable function g_0 such that for every $X \in [T_0]$, g_0^X is total and majorizes every Δ^1_1 function. Indeed, one can take the Σ^1_1 tree $T_0 = \{\sigma : \forall n < \mathrm{lh}(\sigma), \forall i \leq n, \sigma(n) \geq f_i(n) \text{ whenever } f_i(n) \downarrow\}$ where $(f_i)_{i\in\omega}$ is an enumeration of partial Π^1_1 functions. Then, $g_0^X : n \mapsto X(n)$ majorizes every Δ^1_1 function whenever $X \in [T_0]$, up to finite values.

Contrary to this fact, for every Σ^1_1 tree $T \subseteq 2^\omega$ and computable functional f total on $[T]$, there exists a Δ^1_1 function g majorizing every f^X for $X \in [T]$. Indeed, let T be Σ^1_1, we define $g(n)$ using the following procedure: First, wait for an ordinal stage s such that $f_n : X \mapsto f^X(n)$ is total on T_s, the co-enumeration of T at stage s. Then, output the maximum of the values taken by $f_n + 1$ on this set. This must happen, as by compactness there is a finite level of T already witnessing totality, and the co-enumeration must be exact on this finite level at some computable ordinal stage. Moreover, only finitely many values can be taken, so the maximum must exist.

Now suppose that Σ^1_1-WKL $\geq_{\mathrm{W}}$ Σ^1_1-tree-$\mathrm{C}_{\omega^\omega}$. Then, there must exist some Σ^1_1 tree $S \subseteq 2^{<\omega}$, such that every path of S uniformly computes a path of T_0, which itself uniformly computes a Δ^1_1-majoring function, a contradiction. ■

IV.2.2 Restrictions on single choice

In this section, we investigate on several restrictions of choice in analytic sets. In [15], the author lists several variations on the principle of single choice on closed set, corresponding mainly to restrictions on the cardinality of the inputs. The following can be considered as an extension of his work, by considering the same kinds of restrictions but for Σ^1_1 sets instead of closed sets.

Unique and Compact choice

Definition IV.16

Let $\mathcal{A}$ and $\mathscr{X}$ be as in Definition IV.11. Then, $\mathcal{A}$-$\mathrm{UC}_{\mathscr{X}}$ is $\mathcal{A}$-$\mathrm{C}_{\mathscr{X}}$ restricted to singletons, while $\mathcal{A}$-$\mathrm{KC}_{\mathscr{X}}$ is $\mathcal{A}$-$\mathrm{C}_{\mathscr{X}}$ restricted to compact sets.

We first characterizes how these restrictions work at the Σ^1_1 level, using the degrees already defined.

Theorem IV.17

Σ^1_1-$\mathrm{UC}_{\omega^\omega}$ $<_{\mathrm{W}}$ Σ^1_1-$\mathrm{KC}_{\omega^\omega}$ $\equiv_{\mathrm{W}}$ Σ^1_1-WKL.

Proof. It is clear that a Σ^1_1 singleton is a Δ^1_1 element. However, there exists compact subsets of ω^ω with no hyperarithmetic elements, showing Σ^1_1-$\mathrm{UC}_{\omega^\omega}$ $<_{\mathrm{W}}$ Σ^1_1-$\mathrm{KC}_{\omega^\omega}$.

It is clear that any Σ^1_1 tree of the Cantor space induces a compact Σ^1_1 subset of the Baire space. Conversely, let A be a compact subset of Baire space. As a compact set, it must be closed, so the Σ^1_1 tree $T = \{\sigma \in \omega^{<\omega} : \exists X \in A, X \succ \sigma\}$ is such that $[T] = A$. It is clear that each node in T has finitely many direct child, as $[T]$ is compact. Therefore, the map defined on T by the function $n_0 n_1 \cdots n_k \mapsto 0^{n_0}10^{n_1}10^{n_2}1\cdots 0^{n_k}$ give a tree $T' \subseteq 2^{<\omega}$ with no other path but the image of elements of $[T]$, and any path of T' computes a path of T. ■

Next, we show that the compact Σ^1_1-choice principle is also Weihrauch equivalent to the following principles:

— The principle Π^1_1-Tot_2, the *totalization problem for partial* Π^1_1 *two-valued functions*, is the partial multivalued function which, given a partial function $\varphi\colon\subseteq\omega\to 2$ which is Π^1_1 relative to a given parameter, chooses a total extension $f\colon\omega\to 2$ of φ.

— The principle Π^1_1-DNC_2, the *problem of finding a two-valued diagonally non-*Π^1_1 *function*, is the partial multivalued function which, given a sequence of partial functions $(\varphi_e)_{e\in\omega}$ which are Π^1_1 relative to a given parameter, chooses a total function $f\colon\omega\to 2$ diagonalizing the sequence, that is, $f(e)\neq\varphi_e(e)$ whenever $\varphi_e(e)$ is defined.

The latter notion has also been studied by Kihara-Marcone-Pauly [47].

▶ **Proposition IV.18.** $\widehat{\Sigma^1_1\text{-}\mathsf{C}_2}\equiv_{\mathrm{W}}\Pi^1_1\text{-}\mathsf{Tot}_2\equiv_{\mathrm{W}}\Pi^1_1\text{-}\mathsf{DNC}_2$.

Proof. Π^1_1-$\mathsf{Tot}_2\leq_{\mathrm{W}}\Pi^1_1$-$\mathsf{DNC}_2$: Given a partial function $\varphi\colon\subseteq\omega\to 2$, define $\psi_e(e)=1-\varphi(e)$. If g diagonalizes $(\psi_e)_{e\in\omega}$, then $g(e)=1-\psi_e(e)=\varphi(e)$ whenever $\varphi(e)$ is defined. Therefore, g is a totalization of φ.

Π^1_1-$\mathsf{DNC}_2\leq_{\mathrm{W}}\widehat{\Sigma^1_1\text{-}\mathsf{C}_2}$: Define $S_e=\{a:\varphi_e(e)\downarrow<2\ \to\ a\neq\varphi_e(e)\}$ is uniformly Σ^1_1. Moreover, the choice for $(S_e)_{e\in\omega}$ clearly diagonalizes $(\varphi_e)_{e\in\omega}$.

$\widehat{\Sigma^1_1\text{-}\mathsf{C}_2}\leq_{\mathrm{W}}\Pi^1_1$-$\mathsf{Tot}_2$: Given a Σ^1_1 set $S_n\subseteq 2$, wait for S_n to become a singleton, say $S_n=\{s_n\}$. It is easy to find an index of a partial Π^1_1 function f such that $f(n)=s_n$ whenever $S_n=\{s_n\}$. Then, any total extension of f is a choice for $(S_n)_{n\in\omega}$. ■

Other restrictions

Then, we define:

$$\begin{aligned}
\Sigma^1_1\text{-}\mathsf{UC}_X &= \Sigma^1_1\text{-}\mathsf{C}_X\restriction_{\{A\subseteq X:|A|=1\}},\\
\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{fin}}_X &= \Sigma^1_1\text{-}\mathsf{C}_X\restriction_{\{A\subseteq X:A\text{ is finite}\}},\\
\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{cof}}_X &= \Sigma^1_1\text{-}\mathsf{C}_X\restriction_{\{A\subseteq X:A\text{ is cofinite}\}},\\
\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{foc}}_X &= \Sigma^1_1\text{-}\mathsf{C}_X\restriction_{\{A\subseteq X:A\text{ is finite or cofinite}\}},\\
\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aof}}_X &= \Sigma^1_1\text{-}\mathsf{C}_X\restriction_{\{A\subseteq X:A=X\text{ or }A\text{ is finite}\}},\\
\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aou}}_X &= \Sigma^1_1\text{-}\mathsf{C}_X\restriction_{\{A\subseteq X:A=X\text{ or }|A|=1\}}.
\end{aligned}$$

Note that the all-or-unique choice is often denoted by AoUC_X instead of $\mathsf{C}^{\mathsf{aou}}_X$, cf. [48]. Among others, we see that the all-or-unique choice Σ^1_1-$\mathsf{C}^{\mathsf{aou}}_{\mathbb{N}}$ is quite robust. Recall from Proposition IV.18 that the Π^1_1-totalization principle Π^1_1-Tot_2 and the Π^1_1-diagonalization principle Π^1_1-DNC_2 restricted to *two valued functions* are equivalent to the Σ^1_1 compact choice principle. We now consider the ω-valued versions of the totalization and the diagonalization principles:

— The principle Π^1_1-$\mathsf{Tot}_{\mathbb{N}}$, the *totalization problem for partial* Π^1_1 *functions*, is the partial multivalued function which, given a partial function $\varphi\colon\subseteq\omega\to\omega$ which is Π^1_1 relative to a given parameter, return the total extensions of φ.

— The principle Π^1_1-$\mathsf{DNC}_{\mathbb{N}}$, the *problem of finding a diagonally non-*Π^1_1 *function*, is the partial multivalued function which, given a sequence of partial functions $(\varphi_e)_{e\in\omega}$ which are Π^1_1 relative to a given parameter, returns the total functions $f:\omega\to\omega$ diagonalizing the sequence.

It is clear that Π^1_1-$\mathsf{DNC}_{\mathbb{N}}\leq_{\mathsf{W}}\Pi^1_1$-$\mathsf{DNC}_2\equiv_{\mathsf{W}}\Pi^1_1$-$\mathsf{Tot}_2\leq_W\Pi^1_1$-$\mathsf{Tot}_{\mathbb{N}}$. One can easily see the following.

▶ **Proposition IV.19.** $\widehat{\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aou}}_{\mathbb{N}}} \equiv_{\mathsf{W}} \Pi^1_1\text{-}\mathsf{Tot}_{\mathbb{N}}$.

Proof. The argument is almost the same as Proposition IV.18. Given a partial function φ, define $S_n = \{a : \varphi(n) \downarrow \to \ a = \varphi(n)\}$, which is uniformly Σ^1_1. Clearly, either $S_n = \mathbb{N}$ or S_n is a singleton. Hence, the all-or-unique choice principle chooses an element of S_n, which produces a totalization of φ.

Conversely, given a Σ^1_1 set $S_n \subseteq \mathbb{N}$, wait until S_n becomes a singleton, say $S_n = \{s_n\}$. It is easy to find an index of partial Π^1_1 function f such that $f(n) = s_n$ whenever $S_n = \{s_n\}$. Then, any total extension of f is a choice for $(S_n)_{n\in\omega}$. ■

IV.2.3 Axioms of Choices

The previous subsection dealt with single choice. This principle has no equivalent in the reverse mathematical zoo. However, as uniformity matters in proofs, single choice does not imply infinite choice. The Axiom of Choice is a very important principle in mathematical logic, from reverse mathematics to set theory.

Of course, any kind of axiom of choice on $\mathscr{X}$ can be seen as a single choice on $\mathscr{X}^{\mathbb{N}}$, but we still make a difference, in particular when $\mathscr{X}$ is countable and therefore of a different nature than $\mathscr{X}^{\mathbb{N}}$.

Definition IV.20 ($\Sigma^1_1\text{-AC}_{\mathscr{X}}$)

Let $\mathscr{X}$ be a recursively presented topological space. We define $\Sigma^1_1\text{-AC}_{\mathscr{X}}$ to be the multivalued function with $\mathrm{dom}(\Sigma^1_1\text{-AC}_{\mathscr{X}})$ being the set of all families $(c_n)_{n\in\mathbb{N}}$ of countably many codes for Σ^1_1 non empty sets A_n, and with solutions being elements of the product $\prod_{n\in\mathbb{N}} A_n$.

It is clear that, by design, we have $\Sigma^1_1\text{-AC}_{\mathscr{X}} \equiv_{\mathrm{W}} \widehat{\Sigma^1_1\text{-C}_{\mathscr{X}}}$. Dependent choice is when the set of choices depends on the previous choices. Therefore, the input looks like a tree without leaves.

Definition IV.21 ($\Sigma^1_1\text{-DC}_{\mathscr{X}}$)

Let $\mathscr{X}$ be a recursively presented topological space. We define $\Sigma^1_1\text{-AC}_{\mathscr{X}}$ to be the multivalued function with $\mathrm{dom}(\Sigma^1_1\text{-DC}_{\mathscr{X}})$ being the set of all families $(c_\sigma)_{\sigma\in\mathbb{N}^{<\mathbb{N}}}$ of countably many codes for Σ^1_1 sets A_n, with $A_\sigma \neq \emptyset$ iff $\forall n < |\sigma|$, $\sigma(n) \in A_{\sigma\restriction n}$. Its solutions are the infinite sequences $x \in \mathscr{X}^{\mathbb{N}}$ with $x(n) \in A_{x\restriction n}$.

As mentioned in the introduction, we have that $\Sigma^1_1\text{-DC}_{\mathscr{X}} \leq_{\mathrm{W}} \Sigma^1_1\text{-C}_{\mathscr{X}^{\mathbb{N}}}$, however more precisely we have the following:

▶ **Proposition IV.22.** *Let $\mathscr{X}$ be a recursively presented topological space. We have* $\Sigma^1_1\text{-DC}_{\mathscr{X}} \equiv_{\mathrm{W}} \Sigma^1_1\text{-tree-C}_{\mathscr{X}}$.

Proof. Between the two principles, only the way the input are given differ. Given $(c_\sigma)_{\sigma\in 2^{<\mathbb{N}}}$ Σ^1_1 codes for sets A_σ, one can clearly compute an index for the Σ^1_1 tree $\{\sigma \in \mathscr{X}^{<\mathbb{N}} : \forall n < |\sigma|, \sigma(n) \in A_{\sigma\restriction n}\}$. Conversely, given a index of a Σ^1_1 tree $T \subseteq 2^{<\mathbb{N}}$, one can uniformly compute indices for the sets $A_\sigma = \{i \in \mathscr{X} : \sigma^\frown i \in T\}$. ■

▶ **Corollary IV.22.1.** *We have both* $\Sigma^1_1\text{-DC}_2 \equiv_{\mathrm{W}} \Sigma^1_1\text{-WKL}$, *and* $\Sigma^1_1\text{-DC}_{\mathbb{N}} \equiv_{\mathrm{W}} \Sigma^1_1\text{-tree-C}_{\mathbb{N}}$.

We will often see dependent choices as trees. In order to see independent choices as particular instances of dependent choice, we make the following definition:

Definition IV.23

A tree $T \subseteq \mathscr{X}^{<\mathbb{N}}$ is homogeneous if $A_\sigma = \{n \in \mathscr{X} : \sigma \frown n \in T\}$ only depends on the length of σ. We also say that a set $X \subseteq \mathscr{X}^{\mathbb{N}}$ is homogeneous if it corresponds to the path of a homogeneous tree.

Theorem IV.24

We have that

$$\Sigma^1_1\text{-AC}_2 \equiv_{\mathrm{W}} \Sigma^1_1\text{-DC}_2 <_{\mathrm{W}} \Sigma^1_1\text{-C}_{2^\omega}$$

but

$$\Sigma^1_1\text{-AC}_\omega \leq_{\mathrm{W}} \Sigma^1_1\text{-DC}_\omega \equiv_{\mathrm{W}} \Sigma^1_1\text{-C}_{\omega^\omega}$$

Proof. We start with the corollaries of already proven reductions and separations, leaving the proof of $\Sigma^1_1\text{-DC}_2 \equiv_{\mathrm{W}} \Sigma^1_1\text{-AC}_2$ at the end.

The fact that $\Sigma^1_1\text{-AC}_{\mathscr{X}} \leq_{\mathrm{W}} \Sigma^1_1\text{-DC}_{\mathscr{X}}$ where $\mathscr{X} = 2$ or $\mathscr{X} = \omega$ is clear, by considering the tree $\{\sigma \in \mathscr{X}^{<\omega} : \forall i < |\sigma|, \sigma(i) \in A_i\}$ given an instance $\Pi_n A_n$ of $\Sigma^1_1\text{-AC}_{\mathscr{X}}$.

The fact that $\Sigma^1_1\text{-DC}_2 <_{\mathrm{W}} \Sigma^1_1\text{-C}_{2^\omega}$ is due to the fact that $\Sigma^1_1\text{-DC}_2 \equiv_{\mathrm{W}} \Sigma^1_1\text{-WKL}$ and Theorem IV.15. The fact that $\Sigma^1_1\text{-DC}_\omega \equiv_{\mathrm{W}} \Sigma^1_1\text{-C}_{\omega^\omega}$ is due to the fact that $\Sigma^1_1\text{-DC}_\omega \equiv_{\mathrm{W}} \Sigma^1_1\text{-tree-C}_\omega$ and Theorem IV.13.

Let us now show that $\Sigma^1_1\text{-DC}_2 \equiv_{\mathrm{W}} \Sigma^1_1\text{-AC}_2$. By Corollary IV.22.1, it is sufficient to show $\Sigma^1_1\text{-WKL} \equiv_{\mathrm{W}} \Sigma^1_1\text{-AC}_2$, so let $T \subseteq 2^{<\omega}$ be a Σ^1_1 tree. As $\{\sigma \in T : \exists Y \succ \sigma,\ Y \in [T]\}$ is also Σ^1_1, we can suppose T with no dead end. Now, we define $A_\sigma = \{i \in 2 : \sigma^\frown i \in T\}$, a collection of Σ^1_1 subsets of 2. However, $(A_\sigma)_{\sigma \in 2^{<\omega}}$ is not always non empty: for $\sigma \notin T$, we will certainly have $A_\sigma = \emptyset$, so it is not an instance of $\Sigma^1_1\text{-AC}_2$.

In order to make $(A_\sigma)_{\sigma \in 2^{<\omega}}$ into an instance of $\Sigma^1_1\text{-AC}_2$, we uniformly modify A_σ into the non-empty Σ^1_1 set $\widehat{A}_\sigma \subseteq 2$ such that $A_\sigma \neq \emptyset$ implies $\widehat{A}_\sigma = A_\sigma$. We define $\widehat{A}_\sigma$ using the view of Σ^1_1 sets as co-enumeration along ω_1^{CK}: $\widehat{A}_\sigma$ is defined to follow the co-enumeration of A_σ until one element is removed from A_σ. At any stage after the first removal, $\widehat{A}_\sigma$ stays the same.

Now, let $(b_\sigma)_{\sigma \in 2^{<\omega}} \in \prod_{\sigma \in 2^{<\omega}} \widehat{A}_\sigma$. We claim that $(b_\sigma)_{\sigma \in 2^{<\omega}}$ uniformly computes a path of T: Let $a_0 = b_\epsilon$. As T is non empty, $A_\epsilon \neq \emptyset$ so $A_\epsilon = \widehat{A}_\epsilon$, and therefore $\langle a_O \rangle \in T$. Now suppose $\langle a_0, \cdots, a_n \rangle \in T$ has been defined. As T has no dead end and $\langle a_0, \cdots, a_n \rangle \in T$, we must have $A_{\langle a_0, \cdots, a_n \rangle} \neq \emptyset$, so $b_{\langle a_0, \cdots, a_n \rangle} \in A_{\langle a_0, \cdots, a_n \rangle}$. Define $a_{n+1} = b_{\langle a_0, \cdots, a_n \rangle}$, we have that $\langle a_0, \cdots, a_{n+1} \rangle \in T$.

As the sequence $(a_n)_{n \in \mathbb{N}}$ is uniformly computable in $(b_\sigma)_{\sigma \in 2^{<\omega}}$, and a code for $(\widehat{A}_\sigma)_{\sigma \in 2^{<\omega}}$ is uniformly computable from a code for $(A_\sigma)_{\sigma \in 2^{<\omega}}$, we have $\Sigma^1_1\text{-WKL} \leq_{\mathrm{W}} \Sigma^1_1\text{-AC}_2$. ■

▸ **Corollary IV.24.1.** $\widehat{\Sigma^1_1\text{-C}_2} \equiv_{\mathrm{W}} \Sigma^1_1\text{-WKL}$.

The $\Sigma^1_1\text{-DC}_2 \equiv_{\mathrm{W}} \Sigma^1_1\text{-AC}_2$ part of the previous proof is important. Its core is that in some case it is possible to prevent a Σ^1_1 co-enumeration from removing the last element of a set. However, when co-enumerating an arbitrary subset of $\mathbb{N}$ we are not able to achieve this: the set can become empty without a last element removed. This implies that there is no version of this proof for axioms of choice in all $\mathbb{N}$, raising the question whether dependent and independent choice are equivalent when the choices are made on natural numbers. This question has been asked in [13]

and [47] in the form of $\widehat{\Sigma^1_1\text{-}\mathrm{C}_{\mathbb{N}}}$ versus $\mathrm{C}_{\mathbb{N}^{\mathbb{N}}}$.

▶ **Question IV.25** ([13, 47])**.** Do we have $\Sigma^1_1\text{-}\mathrm{AC}_{\mathbb{N}} <_{\mathrm{W}} \Sigma^1_1\text{-}\mathrm{DC}_{\mathbb{N}}$?

We will answer later this question by the affirmative, however the proof need another principle, ATR_2, that we will introduce only in section IV.5.1. Therefore, we post-pone its resolution. Before starting with the study of restricted axioms of choice, we show another example of a use of the trick where we prevent the removal of the last element, to show that this can be done in other settings than the finite sets.

Definition IV.26

The totalization of a multivalued function P with $\mathrm{dom}(P) \subseteq \omega^\omega$ is the principle P^{tot} of domain the full Baire space; such that if $x \in \mathrm{dom}(P)$ then $P^{\mathsf{tot}}(x) = P(x)$, and whenever $x \notin \mathrm{dom}(P)$, $P(x) = \omega^\omega$.

In other words, all inputs become admissible, but in the case where there is no output in the original principle, any output is a solution.

▶ **Proposition IV.27.** *Let X be a Δ^1_1 subset of $\mathbb{N}$. Then, $\Sigma^1_1\text{-}\mathsf{UC}^{\mathsf{tot}}_X \equiv_{\mathsf{W}} \Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aou}}_X$.*

Proof. $\Sigma^1_1\text{-}\mathsf{UC}^{\mathsf{tot}}_X \leq_{\mathsf{W}} \Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aou}}_X$: Given a Σ^1_1 set S, wait until S becomes a singleton at some ordinal stage. If it happens, let $R = S$; otherwise keep $R = X$. One can effectively find a Σ^1_1-index of R, and either $R = X$ or R is a singleton.

$\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aou}}_X \leq_{\mathsf{W}} \Sigma^1_1\text{-}\mathsf{UC}^{\mathsf{tot}}_X$: Trivial. ■

In particular, the totalization of two-valued unique choice is equivalent to the compact choice.

▶ **Corollary IV.27.1.** $\widehat{\Sigma^1_1\text{-}\mathsf{UC}^{\mathsf{tot}}_2} \equiv_{\mathsf{W}} \Sigma^1_1\text{-}\mathrm{WKL}$.

Proof. It is clear that $\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aou}}_2 \equiv_{\mathsf{W}} \Sigma^1_1\text{-}\mathsf{C}_2$. Thus, the assertion follows from Corollary IV.24.1. ■

IV.3 Restricted Analytical Axioms of Choice

Just as we did in section IV.2.2 for single choice, we investigate the restrictions of the axiom of choice to several properties. In particular, we are interested in the structure of different (semi-)sublattices of the Medvedev degrees corresponding to instances of a restricted axiom of choice.

Given a Weihrauch problem P, we define the Medvedev lattice of P by the lattice of Medvedev degrees of $P(x)$ for all computable instances $x \in \mathrm{dom}(P)$. One reason we are interested in upward density of Medvedev lattices of those various Weihrauch degrees, is that it can be used to Weihrauch-separate two of them. Suppose that $P \leq_{\mathrm{W}} Q$ and the Medvedev lattice of Q is upward dense while the Medvedev lattice of P is not. Then, we have $P <_{\mathrm{W}} Q$: Let $x \in \mathrm{dom}(P)$ be any computable instance realizing a maximal P-Medvedev degree, and take $y \in \mathrm{dom}(Q)$ such that $P(x) \leq_s Q(y)$ (as $P \leq_{\mathrm{W}} Q$). By upward density, let $z \in \mathrm{dom}(Q)$ be such that $Q(z) <_s Q(y)$. Then, it cannot be that there is $t \in \mathrm{dom}(P)$ such that $P(t) \geq_s Q(z)$, as it would contradict maximality of x. Therefore, z is a witness that $P <_W Q$.

We now define the restricted axioms of choice studied in this section.

Definition IV.28

Recall that we already have defined several versions of single choice where the set we have to choose from are restricted to special kinds: For $\star \in \{\mathsf{fin}, \mathsf{cof}, \mathsf{foc}, \mathsf{aof}, \mathsf{aou}\}$, we defined:

$$\Sigma^1_1\text{-}\mathsf{C}^\star_{\mathbb{N}}$$

to be $\Sigma^1_1\text{-}\mathrm{C}_{\mathbb{N}}$ where the domain is restricted to be respectively finite, cofinite, finite or cofinite, all or finite and all or unique. We define this single choice to correspond to a single step in a restricted axiom of choice:

$$\Sigma^1_1\text{-}\mathsf{AC}^\star_{\mathbb{N}} = \widehat{\Sigma^1_1\text{-}\mathsf{C}^\star_{\mathbb{N}}}$$

We will also consider the Dependent Choice where the restriction is also dependent on the previous choices:

$$\Sigma^1_1\text{-}\mathsf{DC}^\star_{\mathbb{N}} = \Sigma^1_1\text{-}\mathrm{DC}_{\mathbb{N}} \restriction_{\{(A_\sigma)_\sigma : A_\sigma \text{ is } \star \text{ or empty}\}} .$$

where $\star \in \{\mathsf{fin}, \mathsf{cof}, \mathsf{foc}, \mathsf{aof}, \mathsf{aou}\}$ has the same meaning.

Before going further, we mention that under Medvedev reducibility, AC and DC are always different, as there exists products of two homogeneous sets that are never Medvedev equivalent to a homogeneous set.

▶ **Proposition IV.29.** *For every $\star \in \{\mathsf{fin}, \mathsf{cof}, \mathsf{foc}, \mathsf{aof}\}$, there exists $A \in \Sigma^1_1\text{-}\mathsf{DC}^\star_{\mathbb{N}}$ such that there is no $B \in \Sigma^1_1\text{-}\mathsf{AC}^\star_{\mathbb{N}}$ with $A \equiv_s B$.*

Proof. Simply take A_0 and A_1 in $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$ which are not Medvedev equivalent, and consider $C = 0^\frown A_0 \cup 1^\frown A_1$, which is in $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}}$. Now, toward a contradiction, suppose also that there exists H in $\Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$ (actually there is no need for H to be Σ^1_1) such that $C \equiv_s H$. Let ϕ and ψ be witnesses of this, i.e ϕ (resp. ψ) is total on C (resp. H) and its image is included in H (resp. C).

Now, we describe a way for some A_i to Medvedev compute A_{1-i}: Pick $i \in 2$ and σ be extensible in H such that $\psi^\sigma(0) = 1 - i$. Given $x \in A_i$, apply ϕ on $i^\frown x$ to obtain an element y of H. Replace the beginning of y by σ and apply ϕ: by homogeneity, y with σ as beginning is still in H, and the result has to be in $(1-i)^\frown A_{1-i}$.

For other values of $\star$, the proof is very similar. ■

Note that the above proof used the fact that there exists infinum of any two elements in $\Sigma^1_1\text{-}\mathsf{DC}^\star_{\mathbb{N}}$ while this is not clear in $\Sigma^1_1\text{-}\mathsf{AC}^\star_{\mathbb{N}}$. However, these differences on the point of view of the Medvedev lattice do not always imply a difference in the corresponding Weihrauch degrees. For instance, the Weihrauch degrees $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}}$ and $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$ coincide.

IV.3.1 Axiom of finite Choice

Theorem IV.30

$\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}} \equiv_W \Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}}$

Proof. It is clear by Corollary IV.24.1 that we have $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}} \leq_W \Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}} \leq_W \Sigma^1_1\text{-}\mathsf{WKL} \leq_W \widehat{\Sigma^1_1\text{-}\mathsf{C}_2} \leq_W \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$. ■

However, we are interested in a finer analysis of $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$ and $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}}$, using Medvedev reducibility. In particular, we show that upward density does not hold in both these lattices: Indeed, we show that there is a single non-empty compact homogeneous Σ^1_1 set coding all information of non-empty compact Σ^1_1 sets.

Theorem IV.31

There exists $A \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$ such that for every $B \in \Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}}$, $B \leq_s A$.

Proof. To construct a greatest element in $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}}$, we only need to enumerate all non-empty Σ^1_1 sets $S_e \subseteq 2^\omega$, as every compact subset of ω^ω is below a closed subset of 2^ω. Consider an upper Δ^1_1 approximation $(S_{e,\alpha})_{\alpha<\omega_1^{\mathrm{CK}}}$ of S_e. Note that emptiness of S_e is a Π^1_1-property, and therefore, if $S_e = \emptyset$, then it is witnessed at some stage $\alpha < \omega_1^{\mathrm{CK}}$. Let α be the least ordinal such that $S_{e,\alpha}$ is empty. By compactness of S_e, such an α must be a successor ordinal.

Now we construct a uniform sequence $(T_e)_{e\in\omega}$ of non-empty Σ^1_1 sets such that if $S_e \neq \emptyset$ then $S_e = T_e$. Define $T_{e,0} = 2^{\mathbb{N}}$, and for any $\alpha > 0$, $T_{e,\alpha} = S_{e,\alpha}$ if $S_{e,\alpha} \neq \emptyset$. If $\alpha > 0$ is the first stage such that $S_{e,\alpha} = \emptyset$, then α is a successor ordinal, say $\alpha = \beta + 1$, and define $T_{e,\gamma} = T_{e,\beta}$ for any $\gamma \geq \alpha$, and ends the construction. It is not hard to check that the sequence $(T_e)_{e\in\omega}$ has the desired property.

As a maximal element, it suffices to take the product T of all T_e, defined by $x \in T$ if and only if $\sigma \circ (n \mapsto \langle e, n\rangle) \in T_e$ for every $e \in \mathbb{N}$. This is not yet an element of $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$, but by Theorem IV.30 we have that $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}} \equiv_W \Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}}$, so there exists an element in the domain of $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$ which is above T for Medvedev reduction, making it above every element of the domain of $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}}$. ■

Even if lattices of dependent and independent choice share a common maximum, they still have structural differences. The most evident one is the existence of infimums: Given two Σ^1_1 trees T_1 and T_2, it is easy to create a tree T such that $[T]$ is the infinum of $[T_0]$ and $[T_1]$, by considering for example $0^\frown T_0 \cup 1^\frown T_1$ [2]. However, this is not possible when the trees are homogeneous as in the independent choice.

As a special property of Σ^1_1 compact sets, we have the following analog of the hyperimmune-free basis theorem. For $p, q \in \mathbb{N}^{\mathbb{N}}$ we say that p is *higher Turing reducible to* q (written $p \leq_{hT} q$) if there is a partial Π^1_1-continuous function $\Phi\colon\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ such that $\Phi(q) = p$ (see Bienvenu-Greenberg-Monin [8] for more details).

▶ **Lemma IV.32.** *For any Σ^1_1 compact set $K \subseteq \mathbb{N}^{\mathbb{N}}$ there is an element $p \in K$ such that every $f \leq_{hT} p$ is majorized by a Δ^1_1 function.*

Proof. Let (ψ_e) be a list of higher Turing reductions. Let $K_0 = K$. For each e, let $Q_{e,n} = \{x \in \mathbb{N}^{\mathbb{N}} : \psi_e^x(n)\uparrow\}$. Then $Q_{e,n}$ is a Σ^1_1 closed set. If $K_e \cap Q_{e,n}$ is non-empty for some n, define $K_{e+1} = K_e \cap Q_{e,n}$ for such n; otherwise define $K_{e+1} = K_e$. Note that if $K_e \cap Q_{e,n}$ is non-empty for some n, then ψ_e^x is undefined for any $x \in K_{e+1}$. If $K_e \cap Q_{e,n}$ is empty for all n, then ψ_e is total on the Σ^1_1 compact set K_e, one can find a Δ^1_1 function majorizing ψ_e^x for all $x \in K_e$ (cf. [47]). Define $K_\infty = \bigcap_n K_n$, which is non-empty. Then, for any $p \in K_\infty$, every $f \leq_{hT} p$ is majorized by a Δ^1_1 function. ■

2. or $2\mathbb{N}^\frown T_0 \cup (2\mathbb{N}+1)^\frown T_1$ when finite sets are restricted such as in $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{cof}}_{\mathbb{N}}$.

Note that continuity of higher Turing reduction is essential in the above proof. Indeed, one can show the following:

▶ **Proposition IV.33.** *There is a non-empty Σ^1_1 compact set $K \subseteq \mathbb{N}^{\mathbb{N}}$ such that for any $p \in K$, there is $f \leq_T p'$ that dominates all Δ^1_1 functions.*

Proof. Let (φ_e) be an effective enumeration of all partial Π^1_1 functions $\varphi_e :\subseteq \omega \to 2$. As in the argument in Proposition IV.18 or Proposition IV.19, one can see that the set S_e of all two-valued totalizations of the partial Π^1_1 function φ_e is non-empty and Σ^1_1. Then the product $K = \prod_e S_e = \{\langle e, n\rangle : n \in S_e\}$ is also a non-empty Σ^1_1 subset of 2^ω. It is clear that every $p \in K$ (non-uniformly) computes any total Δ^1_1 function on ω. Let f be a total p'-computable function which dominates all p-computable functions. In particular, $f \leq_T p'$ dominates all Δ^1_1 functions. ■

IV.3.2 Axiom of all-or-finite Choice

We now discuss the axiom of choice, when the sets from which we choose can be either everything, or finite. We will show that under the Weihrauch scope, this principle is a robust one that is strictly above $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}}$. It also shares with the latter that dependent or independent choice does not matter, and the existence of a maximal element containing all the information, with a very similar proof as for $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}}$.

In Proposition IV.19, we showed that $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aou}}_{\mathbb{N}}$ is robust. We give two other evidences of this in the following theorems.

Theorem IV.34

For any $A \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$, there exists $B \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aou}}_{\mathbb{N}}$ such that $A \leq_s B$.

Proof. Let $A = \prod_n A_n \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$. We define $B = \prod_{\langle m,n\rangle} B^m_n$ such that $A \leq_s B$. We will ensure that there exists a single computable function Φ such that for any m and $X \in \prod_n B^m_n$ we have $\Phi(X) \in A_m$.

We first describe the co-enumeration of B^m_n. Let $(A_{m,\alpha})_{\alpha<\omega_1^{\mathrm{CK}}}$ be an approximation of A_m. First, wait for the first stage where A_m is finite. If it happens, wait for exactly n additional elements to be removed from A_m. If this happens, remove from B^m_n all elements but $c \in \mathbb{N}$, the integer coding for the finite set A_m at this stage, say at stage α_n. More formally, let D_e be the finite set coded by e, and set $B^m_n = \{c\}$ with $D_c = A_m[\alpha_n]$.

Now, we describe the function Φ. Given X, find the first i such that we do not have the following: $X(i+1)$ viewed as coding a non-empty finite set consists of elements from $X(i)$ with exactly one element removed. Note that $X(0)$ codes a finite set, so the length of chains $D_{X(0)} \supsetneq D_{X(1)} \supsetneq \dots$ has to be finite. Therefore, there exists such an i. Then, output any element from $D_{X(i)}$. Whenever we reach stage α_n, we have $D_{X(n)} = A_m[\alpha_n]$, and thus $i \geq n$. This implies that the chosen element $\Phi(X)$ is contained in A_m, as required. ■

We have seen in Proposition IV.18 that $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$ is Weihrauch equivalent to $\Pi^1_1\text{-}\mathsf{Tot}_2$ and $\Pi^1_1\text{-}\mathsf{DNC}_2$. Moreover, we have also shown in Proposition IV.19 that $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aou}}_{\mathbb{N}}$ is Weihrauch equivalent to $\Pi^1_1\text{-}\mathsf{Tot}_{\mathbb{N}}$. Recall the Π^1_1-diagonalization principle $\Pi^1_1\text{-}\mathsf{DNC}_{\mathbb{N}}$ we introduced in Section IV.2.2, which is a special case of the cofinite (indeed, co-singleton) Σ^1_1-choice principle. In particular, at first we know a bound on the number of elements removed from a cofinite set. We now consider the following principle for a bound $\ell \in \mathbb{N}$:

$$\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{cof}\restriction\ell}_X = \Sigma^1_1\text{-}\mathsf{C}_X \restriction_{\{A\subseteq X : |X\setminus A| \leq \ell\}} .$$

We call the union of $(\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{cof}\restriction\ell}_X)_{\ell\in\mathbb{N}}$ the *strongly-cofinite choice* on X, and write $\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{cof}\restriction *}_X$. Later we will show that the cofinite choice $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}}$ is not Medvedev or Weihrauch reducible to the all-or-finite choice $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$; however we will see that the strong cofinite choice is Medvedev or Weihrauch reducible to $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$.

Even more generally, we consider the *finite-or-strongly-cofinite choice*, denoted $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fosc}}_{\mathbb{N}}$, which accepts an input of the form (p,ψ), where for all $n\in\mathbb{N}$, $p(n)$ is a code of a Σ^1_1 subset $S_{p(n)}$ of $\mathbb{N}$ such that either $S_{p(n)}$ is non-empty and finite, or $|\mathbb{N}\setminus S_{p(n)}|\leq\psi(n)$. If (p,ψ) is an acceptable input, then $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fosc}}_{\mathbb{N}}$ chooses one element from $\prod_n S_{p(n)}$.

We show that the all-or-unique choice is already strong enough to compute the finite-or-strongly-cofinite choice:

▶ **Proposition IV.35.** *For any $A\in\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fosc}}_{\mathbb{N}}$, there exists $B\in\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$ such that $A\leq_s B$.*

Proof. Let $A=\prod_n A_n$ with a bound ψ is given. We will construct a uniformly Σ^1_1 sequence $(B^n_m)_{m\leq\psi(n)}$ of subsets of $\mathbb{N}$. We use $B^n_0,B^n_1,\dots,B^n_{\psi(n)-1}$ to code which element is removed from A_n whenever A_n is cofinite, and use $B^n_{\psi(n)}$ to code full information of A_n whenever A_n is finite. If a_0 is the first element removed from A_n, then put $B^n_0=\{a_0\}$, and if a_1 is the second element removed from A_n, then put $B^n_1=\{a_1\}$, and so on. If A_n becomes a finite set, then $B_{\psi(n)}$ just copies A_n. One can easily ensure that for any $n\in\mathbb{N}$ and $m<\psi(n)$, if A_n is finite, then B^n_m is a singleton, which is not contained in A_n; and if A_n stays infinite then $B^n_m=\mathbb{N}$. Moreover, we can also see that either $B^n_{\psi(n)}$ is non-empty and finite or $B^n_{\psi(n)}=\mathbb{N}$.

Now, assume that $X\in\prod_{n,m}B^n_m$ is given. If $X(n,\psi(n))\notin\{X(n,i):i<\psi(n)\}$, then put $Y(n)=X(n,\psi(n))$. Otherwise, choose $Y(n)\notin\{X(n,i):i<\psi(n)\}$. Clearly, the construction of Y from X is uniformly computable.

If A_n becomes a finite set, the first case happens, as every $X(n,i)$ is the only element of B^n_i, which is an element removed from A_n, and as A_n is finite it is equal to $B^n_{\psi(n)}$. Then, $Y(n)=X(n,\psi(n))\in B^n_{\psi(n)}=A_n$. If A_n remains cofinite, it is easy to see that $\mathbb{N}\setminus A_n\subseteq\{X(n,i):i<\psi(n)\}$, and therefore $Y(n)\in A_n$. Consequently, $Y\in A$. ■

▶ **Corollary IV.35.1.** $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aou}}_{\mathbb{N}}\equiv_{\mathsf{W}}\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}\equiv_{\mathsf{W}}\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fosc}}_{\mathbb{N}}$.

In the following we will only consider all-or-finite choice, by convenience. We now prove that dependent choice does not add any power, and the existence of a maximal instance that already codes all the other instances, with very similar proofs as in the $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}}$ case.

Theorem IV.36

For every $A\in\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{aof}}_{\mathbb{N}}$ there exists $B\in\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$ such that $A\leq_s B$.

Proof. The argument is similar as in Theorem IV.31. If T is a Σ^1_1 tree, define T_σ by the following Σ^1_1 procedure: First, wait for $\{n:\sigma^\frown n\in T\}$ to be finite but non-empty. If this happens, at every stage define T_σ to be $\{n:\sigma^\frown n\in T\}$ except if this one becomes empty. Note that if $\{n:\sigma^\frown n\in T\}$ becomes a finite set at some stage α_0, but an empty set at a later stage α_1, then the least such stage α_1 must be a successor ordinal, and therefore we can keep T_σ being non-empty (see also the proof of Theorem IV.31). Clearly, T_σ is either finite or $\mathbb{N}$ and $\prod_{\sigma\in\omega^{<\omega}}[T_\sigma]\geq_s[T]$. ■

▶ **Corollary IV.36.1.** $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}\equiv_W\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{aof}}_{\mathbb{N}}$.

Proof. By uniformity of the precedent proof. ■

The upward density of the axiom of choice on "all-or-finite" sets would allow us to Weihrauch separate it from its "finite" version. However, $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$ does also has a maximum element.

Theorem IV.37

There exists a single maximum Medvedev degree in $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$ and $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{aof}}_{\mathbb{N}}$.

Proof. The argument is similar as the one of Theorem IV.31, even though we have no compactness assumption. By the fact that $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}} \equiv_W \Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{aof}}_{\mathbb{N}}$, it suffices to prove the result for one, let's say $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$. Let $A_e = \prod_n S^e_n$ be the e-th Σ^1_1 homogeneous set. We set $\widehat{A_e} = \prod_n \widehat{S^e_n}$ to be defined by the following Σ^1_1 procedure: First, wait for some S^e_n to become finite and non-empty. If this happens, define $\widehat{S^e_n} = S^e_n$ until it removes its last element. At this point, leaves $\widehat{S^e_n}$ non-empty, which is possible since it can happen only at a successor stage (see also the proof of Theorem IV.36).

Then $(\widehat{A_e})_{e\in\mathbb{N}}$ is an enumeration of all non-empty elements of $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$. Define the maximum to simply be $\prod_e \prod_n \widehat{S^e_n}$. ■

We now prove that the relaxed constraint on the sets that allows them to be full does increase the power of the choice principle, making $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$ strictly above $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$. We use the fact that the lattice of $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$ has a maximal element, and we show that it must be strictly below some instance of $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$.

Theorem IV.38

For every $A \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$, there exists $B \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$ such that $A <_s B$.

Proof. We will find $C = \prod_n C_n \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$ such that $C \not\leq_s A$. Then, $A \times C$ will witness the theorem.

Now, let us describe the co-enumeration of C_n. First, wait for $\Phi_n(\cdot; n)$ to be total on A, where Φ_n is the n-th partial computable function. Then, wait for it to take only finitely many values, which will happen by compactness. At this point, remove everything from C_n except $\max \Phi_n(A; n) + 1$.

We have that C_n is either $\mathbb{N}$ if the co-enumeration is stuck waiting for $\Phi_n(\cdot; n)$ to be total, or a singleton otherwise. Also, it is clear that for any n, Φ_n cannot be a witness that $C \leq_s A$, so $C \not\leq_s A$. ■

▶ **Corollary IV.38.1.** *We have* $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}} <_W \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$.

Proof. By Theorem IV.38 and Theorem IV.31. ■

One can also use the domination property to separate the all-or-finite choice principle and the compact principle.

▶ **Proposition IV.39.** *There exists* $A \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$ *such that every element* $p \in A$ *computes a function which dominates all* Δ^1_1 *functions.*

Proof. Let $(\varphi_e)_{e\in\mathbb{N}}$ be an effective enumeration of all partial Π^1_1 functions on ω. Put $s(e) = \sum_{n\leq e} n$. Define $A_{s(e)+k} \subseteq \mathbb{N}$ for $k \leq e$ as follows. Begin with $A_{s(e)+k} = \mathbb{N}$. Wait

until we see $\varphi_e(k)\downarrow$. If it happens, set $A_{s(e)+k} = \{\varphi_e(k)\}$. Define $A = \prod_n A_n$. Then define $\Psi(p;n) = \sum_{k\leq e} p(k)$, which is clearly computable in p. It is easy to see that $\Psi(p)$ dominates all Δ^1_1 functions whenever $p \in A$. Indeed, since Ψ is total, every $p \in A$ tt-computes a function which dominates all Δ^1_1 functions. ■

IV.3.3 Axiom of cofinite Choice

The choice problem when all sets are cofinite is quite different from the other restricted choices we study. It is the only one that does not contain $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$.

Let us fix an instance $A = \prod_n A_n$ of $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}}$. For every n, A_n is cofinite, so there exists a_n such that for any $i \geq a_n$, we have $i \in A_n$. Now, call $f : n \mapsto a_n$. We have that $f \in A$, and for every g pointwise above f, we must have $g \in A$. So we clearly have $A \leq_{\mathsf{W}} \{g \in \omega^\omega : \forall i, f(i) \leq g(i)\} = A_f$. This essential property of $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}}$ prevents an instance to have more computational power than A_f for some $f \in \omega^\omega$.

The cofiniteness still allows some power, as we will prove in this section that $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}}$ is Weihrauch incomparable with both $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$ and $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$.

Theorem IV.40

There exists $A \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}}$ such that for any $B \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$ $A \not\leq_s B$.

Proof. We use the existence of a maximal all-or-finite degree of Theorem IV.37 to actually only prove

$$\forall B \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}, \exists A \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}} : A \not\leq_s B.$$

Fix a $B = \prod_{n\in\mathbb{N}} B_n$, with $B_n \subseteq \mathbb{N}$ being either $\mathbb{N}$ or finite. We will construct $A = \prod_{e\in\mathbb{N}} S_e$, and use S_e to diagonalize against Φ_e being a witness for the reduction, by ensuring that either Φ_e is not total on B, or $\exists k \in \mathbb{N}, \sigma \in \prod_{n<k} B_n$ with $\Phi^\sigma_e(e)\downarrow \notin S_e$. Here is a description of the construction of S_e, along with sequences of string (σ_n) and (τ_n):

1. First of all, wait for a stage where $B \subseteq \mathrm{dom}(\Phi_e)$, that is Φ_e is total on the current approximation of B. If it happens, it must happen before stage ω_1^{CK} as the totality of a recursive function on a Σ^1_1 set is a Π^1_1 property: Φ_e is total on B if and only if for all $X \in \omega^\omega$, either $X \notin B$ or Φ_e is total on B. Define $\sigma_0 = \epsilon = \tau_0$.
2. Let n be the maximum such that τ_n is defined. Find $\sigma_{n+1} \succ \tau_n$ such that $\Phi^{\sigma_{n+1}}_e(e)\downarrow \in S_e$. Take σ_{n+1} to be the least such, and remove $\Phi^{\sigma_{n+1}}_e(e)$ from S_e.
3. Wait for some stage where $\Phi^B_e(e) \subseteq S_e$. If it happens, it must happen before ω_1^{CK}, as it is a Π^1_1-property: $\forall X$, either $X \notin B$ or $\Phi^B_e(e) \in S_e$ (during the wait, S_e is a fixed set). Then, wait again for the current approximation of B to be "all or finite", which will happen. Take τ_{n+1} to be the greatest prefix of σ_{n+1} still in B, and return to step (2).

Let us prove that S_e is cofinite. If the co-enumeration of S_e stays at step (1), then $S_e = \mathbb{N}$ is cofinite. Otherwise, let us prove that there can only be finitely many τ_n defined.

Suppose infinitely many (τ_n) are defined. Then, the sequence (τ_n) must converge to a real: By induction, let l be a level such that $(\tau_n(l'))_n$ stabilizes for all $l' < l$. Start from a stage where they have stabilized. From this stage, if $\tau_n(l)$ changes, it must have been removed from B_l. But then, by step 3 B_l will become finite before the algorithm continue, and $(\tau_n(l))$ can only take value from B_l and never twice the same. Therefore, $(\tau_n(l))$ becomes constant at some point, concluding the induction: For every l, $(\tau_n(l))$ becomes constant at some point.

If there are only finitely many τ_n, then only finitely many things are removed from S_e which is cofinite. It remains to prove that $A \not\leq_s B$. Suppose Φ_e is a potential witness for the

inequality. Either Φ_e is not total on B, or we get stuck at some step in the co-enumeration of S_e, waiting for $\Phi_e^B(e) \in S_e$ to never happen, leaving us with $\Phi_e^B(e) \not\subseteq S_e$. ■

Theorem IV.41

For any $A \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$ and $B \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}}$, if $A \leq_s B$, then A contains a Δ^1_1 path.

Proof. Assume that $A \leq_s B$ via some functional Φ, and A and B are of the forms $\prod_n A_n$ and $\prod_n B_n$, respectively. We describe the Δ^1_1 procedure to define C:

Given n, in parallel, wait for n to be enumerated in one of those two Π^1_1 sets:

1. If n is enumerated in $\{n : \exists k \in \mathbb{N}, \forall f \in \omega^\omega, \exists \sigma \prec f,\ \Phi(\sigma; n) = k\}$, define $C(n)$ to be one of these k.
2. If n is enumerated in $\{n : \forall f \in \omega^\omega, \forall k, \exists k' > k, \exists \sigma \prec f$ such that $\Phi(\sigma; n) = k'\}$ then define $C(n) = 0$.

Here, $\sigma \geq f$ denotes the pointwise domination order, that is, $\sigma(n) \geq f(n)$ for all $n < |\sigma|$. It is clear that one of the two options will happen. Let $f \in \omega^\omega$ be such that $\forall k \geq f(n), k \in B_n$. In case (1), it is clear that $C(n) \in A_n$. In case (2), it is clear that A_n is infinite, therefore it is equal to $\mathbb{N}$ and $C(n) \in A_n$. So $C \in A$. ■

▶ **Corollary IV.41.1.** *We have both* $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}} \not\leq_W \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$ *and* $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}} \not\leq_W \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}}$.

Proof. The first part is implied by Theorem IV.40. The second part is implied by Theorem IV.41 and the fact that there exists Σ^1_1 finitely branching homogeneous trees with no Δ^1_1 member. ■

We now show upper density of $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}}$, using a similar proof than the one of Theorem IV.40.

Theorem IV.42

The Medvedev degrees of $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}}$ are upward dense.

Proof. Fix a $B = \prod_{n\in\mathbb{N}} B_n$, with $B_n \subseteq \mathbb{N}$ being cofinite. We will construct $A = \prod_{e\in\mathbb{N}} S_e$, and use S_e to diagonalize against Φ_e being a witness for the reduction, by ensuring that either Φ_e is not total on B, or $\exists k \in \mathbb{N}, \sigma \in \prod_{n<k} B_n$ with $\Phi_e^\sigma(e) \downarrow \notin S_e$. Here is a description of the construction of S_e, along with sequences of string (σ_n) and (τ_n):

1. First of all, wait for a stage where $B \subseteq \mathrm{dom}(\Phi_e)$, that is Φ_e is total on the current approximation of B. As before, if this happens it must be before ω_1^{CK} as the property of being total on a Σ^1_1 set in Π^1_1. Define $\sigma_0 = \epsilon = \tau_0$.
2. Let n be the maximum such that τ_n is defined. Find $\sigma_{n+1} \succ \tau_n$ such that $\Phi_e^{\sigma_{n+1}}(e) \downarrow \in S_e$. Take σ_{n+1} to be the least such, and remove $\Phi_e^{\sigma_{n+1}}(e)$ from S_e.
3. Wait for some stage where $\Phi_e^B(e) \subseteq S_e$. Take τ_{n+1} to be the greatest prefix of σ_{n+1} still in B, and return to step (2).

Let us prove that S_e is cofinite. If the co-enumeration of S_e stays at step (1), then $S_e = \mathbb{N}$ is cofinite. Otherwise, let us prove that there can only be finitely many τ_n defined.

Suppose infinitely many (τ_n) are defined. Then, Then, the sequence (τ_n) must converge to a real: By induction, let l be a level such that $(\tau_n(l'))_n$ stabilizes for all $l' < l$. Start from a stage where they have stabilized. From this stage, if $\tau_n(l)$ changes, it must have been removed from B_l. But that can happen only finitely many times, as B_l is cofinite. Therefore, $(\tau_n(l))$

becomes constant at some point.

If there are only finitely many τ_n, then only finitely many things are removed from S_e which is cofinite. It remains to prove that $A \not\leq_s B$. Suppose Φ_e is a potential witness for the inequality. Either Φ_e is not total on B, or we get stuck at some step in the co-enumeration of S_e, waiting for $\Phi_e^B(e) \in S_e$ to never happen, leaving us with $\Phi_e^B(e) \not\subseteq S_e$. By taking $C = A \times B = \prod_n C_n$ where $C_{2n} = B_n$ and $C_{2n+1} = S_n$, we get the result. ■

IV.3.4 Axiom of finite-or-cofinite Choice

In this part, we study the weakened restriction to sets that are either finite, or cofinite. This restriction allows any instance from the stronger restrictions, thus $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}}$, $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}}$, and $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}}$ are Weihrauch reducible to $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{foc}}_{\mathbb{N}}$ (and similarly for dependent choice). It is the weakest form of restriction other than "no restriction at all" that we will consider. However, we don't know if this restriction does remove some power and is strictly below $\Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$ or not, as asked in Question IV.44.

In the following, we will show upper density for both $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{foc}}_{\mathbb{N}}$, $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{foc}}_{\mathbb{N}}$ and $\Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$, $\Sigma^1_1\text{-}\mathsf{DC}_{\mathbb{N}}$. We will give several different proofs of this result. Theorem IV.43 has a weaker conclusion, but is an attempt to answer Question IV.44. This attempt fails, by being not effective enough to make a diagonalization out of it.

Theorem IV.43

For every $A \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{foc}}_{\mathbb{N}}$, there exists $B \in \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$ such that $B \not\leq_s A$.

Proof. We will build $B = \prod_e B_e \in \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$ by defining B_e in a uniform Σ^1_1 way, such that if Φ_e is total on A, then $\Phi_e(A; e) \notin B_e$.

Fix $e \in \mathbb{N}$, and $A = \prod_n A_n \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{foc}}_{\mathbb{N}}$. In our definition of the co-enumeration of B_e along the ordinals, there will be two main steps in the co-enumeration: The first one forces that if $\Phi_e(A; e) \subseteq B_e$, then for every l, $|\Phi_e(A_{\restriction \leq l})| < \omega$ where $A_{\restriction \leq l} = \{\sigma \in \omega^{\leq l} : [\sigma] \cap A \neq \emptyset\}$. The second step will force that if $\Phi_e(A; e) \subseteq B_e$, then A is empty or Φ_e is not total on A.

In order to conduct all these steps, we will need to remove several times an element of B_e, but we do not want it to become empty. This is why in parallel of removing elements from B_e, we also mark some as "saved for later", so we know that even after infinitely many removal, B_e is still infinite.

We now describe the first part of the co-enumeration. For clarity, we use the formalism of

an infinite time algorithm, that could easily be translated into a Σ^1_1 formula.

```
for l ∈ ω do
    Mark a new element of B_e as saved;
    while Φ_e(A_{↾≤l}; e) is infinite do
        for i ∈ ω do
            Mark a new element of B_e as saved;
            Remove from B_e the first element of Φ_e(A_{↾≤l}; e) that is not saved, if it
              exists. Otherwise, exit the loop;
            Wait for Φ_e(A; e) ⊆ B_e;
        end
        Wait for every A_n to be finite or cofinite;
        Unmark the elements marked as saved by the "for i ∈ ω" loop;
    end
end
```

Let us first argue that for a fixed l, the "while" part can only be executed a finite number of times. At every execution of the "for $i \in \omega$" loop, either one element of $A_{\restriction \leq l}$ is removed, or $\Phi_e(A_{\restriction \leq l}; e)$ is finite and we exit the while loop (this is because at every step, only finitely many elements are marked as saved). But this means that if a "for" loop loops infinitely many times, by the pigeon hole principle there must exists a specific level $l_0 \leq l$ such that A_{l_0} went from cofinite to finite. But this can happen only $l+1$ times, and the "while" loop can only run $l+1$ many times.

Let us now argue that at every stage of the co-enumeration, including its end, B_e is infinite. Fix a level l, and suppose that at the beginning of a "while" loop, B_e is infinite. As after every loop of the "for $i \in \omega$" loop one element is saved, it means that after all these infinitely many loops, B_e contains infinitely many elements. This will happen during only finitely many loops of the "while" loop, so at the beginning of level $l+1$, B_e is infinite. A similar argument with the elements saved by the first "for $l \in \omega$" loop shows that if the first part of the co-enumeration ends, B_e is still infinite.

Now we split into two cases. If the first part of the co-enumeration never stops, as the "while" loop is in fact bounded, it means that the co-enumeration is forever stuck waiting for $\Phi_e(A; e) \subseteq B_e$. But as this never happens, B_e has the required property. Otherwise, the first part of the co-enumeration ends, and we are at a stage where for every l, $\Phi_e(A_{\restriction \leq l}; e)$ is finite, but B_e is infinite. We now continue to the second part of the co-enumeration of B_e:

```
for l ∈ ω do
    Remove from B_e all the elements of Φ_e(A_{↾≤l}; e);
    Wait for Φ_e(A; e) ⊆ B_e;
end
```

We argue that this co-enumeration never finishes. Let $x \in A$, and $\sigma \prec x$ such that $\Phi_e(\sigma; e) \downarrow= k$. The co-enumeration will never reach the stage where $l = |\sigma + 1|$, as it cannot go through $l = |\sigma|$: If it reaches such a stage, it will remove k from B_e and never have $\Phi_e(A; e) \subseteq B_e$. So, the co-enumeration has to stop at some step of the "for" loop, waiting for $\Phi_e(A; e) \subseteq B_e$ never happening. As B_e is infinite, it has the required property. ■

In order to Weihrauch-separate $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{foc}}_{\mathbb{N}}$ from the unrestricted $\Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$, one would need a stronger result with a single $B \in \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$ not Medvedev reducible to any $A \in \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{foc}}_{\mathbb{N}}$. We could try to apply the same argument to define $\prod_{\langle n,e\rangle} B_{\langle n,e\rangle}$, this time diagonalizing against an enumeration $(S^e)_{e\in\mathbb{N}}$ of $S^e = \prod_n S^e_n \in \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$. If S^e is not in $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{foc}}_{\mathbb{N}}$, the co-enumeration will be stuck somewhere in the co-enumeration of some level, with no harm to the global diagonalization.

However, if some particular S^e is empty, we could end up with some $B_{\langle n,e\rangle} = \emptyset$, making B empty. Indeed, suppose we reach the second part of the co-enumeration. Then, the malicious S^e can make sure that every step of the second loop are achieved, by removing from S^e all strings σ such that $\Phi_e(\sigma; e) \downarrow \notin B_{\langle n,e\rangle}$, at every stage of the co-enumeration. As a result, both S^e and $B_{\langle n,e\rangle}$ will become empty.

▶ **Question IV.44.** Do we have $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{foc}}_{\mathbb{N}} <_{\mathsf{W}} \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$?

We now give a stronger result with a much simpler, but not effective, proof. As a corollary, we will obtain the upper density of $\Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$ and $\Sigma^1_1\text{-}\mathsf{DC}_{\mathbb{N}}$.

Theorem IV.45

For every $A \in \Sigma^1_1\text{-}\mathsf{DC}_{\mathbb{N}}$, there exists $B \in \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$ such that $B \not\leq_s A$.

Proof. We first claim that there is no enumeration of all non-empty elements of $\Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$. More than that, we will prove that there is no $\prod_{n,e\in\mathbb{N}} S^e_n \in \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$ uniformly Σ^1_1 such that for every $B = \prod_n B_n \in \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$, there exists an e such that $\prod_n S^e_n \subseteq B$. Let $(S^e_n)_{n,e\in\mathbb{N}}$ be any uniformly Σ^1_1 enumeration. We construct $(B_e)_{e\in\mathbb{N}}$, a witness that this enumeration is not a counter-example to our claim. We define B_e by stage: At stage α, B_e is equal to the interval $]\min(S^e_e), \infty[$, where $\min(S^e_e)$ is computed up to stage α. This defines a Σ^1_1 set. We have $\prod_n B_n \not\supseteq \prod_n S^e_n$ for every $e \in \mathbb{N}$ and the claim is proven.

Now, suppose that there exists $A \in \Sigma^1_1\text{-}\mathsf{DC}_{\mathbb{N}}$ such that for every $B \in \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$, we have $B \leq_s A$. Let us define S^e_n by

$$m \in S^e_n \Leftrightarrow \exists X \in A : \Phi_e(X; n) \downarrow = m \text{ or } \Phi_e \text{ is not total on } A.$$

Given any $B \in \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$, as $B \leq_s A$, fix a witness Φ_e. We have $\Phi_e(A) \subseteq B$, and as B is homogeneous we also have $\prod_n S^e_n \subseteq B$. Then, $(S^e_n)_{e,n\in\mathbb{N}}$ would be a contradiction to our first claim. ■

▶ **Corollary IV.45.1.** *We have upward density for* $\Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$ *and* $\Sigma^1_1\text{-}\mathsf{DC}_{\mathbb{N}}$.

In [22], Cenzer and Hinman showed that the lattice of Π^0_1 classes in Cantor space is dense. Here we already showed upward density, we now prove downward density:

Theorem IV.46

$\Sigma^1_1\text{-}\mathsf{DC}_{\mathbb{N}}$ is downward dense. In other words, for every $A \in \Sigma^1_1\text{-}\mathsf{DC}_{\mathbb{N}}$ with no computable member, there exists $B >_s \omega^\omega$ in $\Sigma^1_1\text{-}\mathsf{DC}_{\mathbb{N}}$ such that

$$\omega^\omega <_s A \cup B <_s A.$$

Proof. We first reduce the problem to finding a non-computable hyperarithmetical real G such that A contains no G-computable point. Indeed, assume such a G exists, then we have $\omega^\omega <_s A \cup \{G\} <_s A$.

It suffices to show that $\Phi_e(G) \notin A$ for any e, and $\emptyset <_T G$. To show that any 1-generic satisfy this, fix a pruned Σ^1_1 tree T_A such that $[T_A] = A$. There are two ways for Φ_e to not be a witness that A has no G-computable element: either $\Phi_e(\sigma) \notin T_A$ for some $\sigma \prec G$, or $G \notin \mathrm{dom}(\Phi_e)$. Let us argue that we have the following: For any $e \in \mathbb{N}$ and $\sigma \in \omega^{<\omega}$ there

$$
\begin{tikzcd}
 & \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}} \arrow[r, phantom, "<_{\mathrm{W}}"] \arrow[d, phantom, "\lneq_{\mathrm{W}}"] & \Sigma^1_1\text{-}\mathsf{DC}_{\mathbb{N}} \arrow[d, phantom, "\lneq_{\mathrm{W}}"] & \\
 & \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{foc}}_{\mathbb{N}} \arrow[r, phantom, "\leq_{\mathrm{W}}"] \arrow[d, phantom, "\lneq_{\mathrm{W}}"] & \Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{foc}}_{\mathbb{N}} \arrow[d, phantom, "\lneq_{\mathrm{W}}"] & \\
\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{aof}}_{\mathbb{N}} \arrow[r, phantom, "\equiv_{\mathrm{W}}"] \arrow[d, phantom, "\lneq_{\mathrm{W}}"] & \Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{aof}}_{\mathbb{N}} \arrow[r, phantom, "\not\geq_{W}"] & \Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}} \arrow[r, phantom, "\leq_{\mathrm{W}}"] & \Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{cof}}_{\mathbb{N}} \\
\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{fin}}_{\mathbb{N}} \arrow[r, phantom, "\equiv_{\mathrm{W}}"] & \Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{fin}}_{\mathbb{N}} \arrow[ur, phantom, "\not\geq_{\mathrm{W}}"] & &
\end{tikzcd}
$$

Figure IV.2 – The Weihrauch hierarchy of analytical axioms of choice

exists a finite string τ extending σ such that

$$\text{either } \Phi_e(\tau) \notin T_A \text{ or } [\tau] \cap \mathrm{dom}(\Phi_e) = \emptyset \tag{IV.1}$$

Indeed, if it was not the case for some $e \in \omega$, we would have a string σ such that for every τ, $\Phi_e(\tau) \in T_A$ and there exists an extension $\rho \succ \tau$ such that $\Phi_e(\rho)$ strictly extends $\Phi_e(\tau)$, allowing us to compute a path of T_A, which is impossible as $A >_s \omega^\omega$.

Using (IV.1), it is clear that any 1-generic real has the desired property, and $B = \{G\}$ proves the theorem. ■

IV.3.5 Conclusion on the restrictions

We gather the results obtained so far in the Weihrauch hierarchy of restricted analytical axioms of choice in Figure IV.2. In particular, We proved that for choices that are restricted to finite sets, and all-or-finite sets, we have that dependent and independent are Weihrauch equivalent. We will prove in Theorem IV.100 that for unrestricted subsets of the integers, dependent choice is strictly above axiom of choice.

The question whether independent choice and dependent choice are in the same Weihrauch degree has been solved when there are strong restrictions (such as finite and all-or-finite), and will be solved in the absence of restrictions. However, for the other kinds of restrictions, the question is still open.

▶ **Question IV.47.** Does $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{foc}}_{\mathbb{N}} \equiv_{\mathrm{W}} \Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{foc}}_{\mathbb{N}}$?

▶ **Question IV.48.** Does $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{cof}}_{\mathbb{N}} \equiv_{\mathrm{W}} \Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{cof}}_{\mathbb{N}}$?

The remaining question is whether "finite-or-cofinite" is a strict restriction:

▶ **Question IV.49.** Does $\Sigma^1_1\text{-}\mathsf{AC}^{\mathsf{foc}}_{\mathbb{N}} <_{\mathrm{W}} \Sigma^1_1\text{-}\mathsf{AC}_{\mathbb{N}}$? And $\Sigma^1_1\text{-}\mathsf{DC}^{\mathsf{foc}}_{\mathbb{N}} <_{\mathrm{W}} \Sigma^1_1\text{-}\mathsf{DC}_{\mathbb{N}}$?

IV.4 More on ATR_0

In definition I.88, we introduced the definition of Arithmetical Transfinite Recursion, as part of the Big Five: the five robust axiomatic systems such that most theorems of mathematics are

equivalent to one of them, over RCA_0. In this chapter we will study the ATR principle in the Weihrauch lattice, however before doing so we review some properties of ATR in the context of reverse mathematics. In particular, we see some theorems that are equivalent to it over RCA_0. One of them, the Perfect Tree theorem, needs an original method to get around the limitation of second order arithmetic: the method of pseudo-hierarchy. This method, and the underlying concepts, is important to understand why ATR is so different in the Weihrauch context.

Most of this section can also be found from Simpson's book [60], which has a very detailed presentation of ATR_0 in the context of reverse mathematics. We recall Definition I.88 of ATR already given in Chapter I.

▶ **Definition** (Φ-hierarchy)**.** Let Φ be any formula (with parameters), $\trianglelefteq$ be a partial order on ω. We say that $H \subseteq \omega$ is a *hierarchy for* Φ *on* $\trianglelefteq$ if for every $j \in \omega$, we have:

$$H_j = \{n \in \mathbb{N} : \Phi(n, \bigoplus_{i \lhd j} H_i)\}$$

where $H_j = \{n : \langle j, n\rangle \in H\}$ and the operation $\bigoplus$ is defined by $\bigoplus_{n\in F} X_n = \{\langle n, x\rangle : n \in F \wedge x \in X_n\}$ for any (X_n).

We write $\mathsf{Hier}_\Phi(H, \trianglelefteq)$ for "H is a hierarchy for Φ on $\trianglelefteq$". If Φ is Δ^0_n, then $\mathsf{Hier}_\Phi(H, \trianglelefteq)$ is a Π^0_n formula, as it can be expressed by $\forall\langle x, n\rangle \in \mathbb{N}$, $\langle x, n\rangle \in H \Leftrightarrow \Phi(n, \Phi(n, \bigoplus_{i\lhd n} H_i))$

This definition corresponds as a trace of a definition by induction along $\lhd$: H_j is defined as the jump of the previous defined sets. In our definition, we do not require the order to be well-founded: a hierarchy on such an ordering would correspond to a definition by induction, but with no starting point...

▶ **Definition** (ATR_0)**.** ATR_0 is the system of axioms consisting of ACA_0 together with the axiom scheme consisting of:

$$\forall \trianglelefteq \text{ binary relation}, \ \trianglelefteq \in \mathbb{WO} \implies \exists H, \mathsf{Hier}_\Phi(H, \trianglelefteq)$$

for every arithmetical Φ.

IV.4.1 Comparability of Well-Ordering

The ATR principle stipulates that construction by arithmetical transfinite recursion is possible. Well-orders are well-suited for proofs by induction, so ATR should prove most existential statements about well-orders. And indeed, it proves that two well-order must be comparable, that is given any two well-orders there must exists a function from one into an initial segment of the other.

Theorem IV.50

ATR_0 proves that if $X, Y \in \mathbb{WO}$, then there exists an embedding either from X into an initial segment of Y, or from Y into an initial segment of X.

Proof. We use ATR_0 to build the embedding. The arithmetic formula we use is the following: $\Phi(n, H)$ is true if $H \subseteq \mathbb{N} \times \mathbb{N}$ is an embedding from an initial segment I_0 of X to an initial segment I_1 of Y, and $n = \langle a, b\rangle$ where a is the smallest in $X \setminus I_0$ and b is the smallest in $Y \setminus I_1$.

If H is such that $\mathsf{Hier}_\Phi(H, X)$, then either H is the graph of an embedding from X to an initial segment of Y, or H^{-1} (the inverse of the function of which H is the graph) is an

embedding from Y to an initial segment of X. ■

In fact, the two statements are equivalent over RCA_0.

Theorem IV.51 ([60])

The axiomatic system consisting of RCA_0 and "for every two well-ordering, there exists an embedding from one to the initial segment of the other" proves ATR_0.

IV.4.2 Σ^1_1 separation

Another theorem that is equivalent to ATR_0 over RCA_0 is the Σ^1_1 separation theorem. It is a very important result by Lusin, stating that given two disjoint $\mathbf{\Sigma^1_1}$ subsets of Baire space A and B, there exists a Borel set C separating the two sets, in the sense that $A \subseteq C \subseteq \omega^\omega \setminus B$. Note that a direct consequence of this is that every $\mathbf{\Delta^1_1}$ set A is also Borel: A and $\omega^\omega \setminus A$ are two disjoint $\mathbf{\Sigma^1_1}$ sets so there exists a Borel set C with $a \subseteq C \subseteq \omega^\omega \setminus (\omega^\omega \setminus A)$ so $A = C$.

However, being able to speak about Borel codes, and membership in a Borel code, in second order arithmetic already needs quite complex axioms such as ATR_0. So it is quite complex to show equivalence between ATR_0 and this version of $\mathbf{\Sigma^1_1}$-separation, as the notion of the latter needs ATR_0 to be defined. So, instead, we turn to a version of separation for subsets of $\mathbb{N}$.

Definition IV.52 (Σ^1_1-Sep)

Σ^1_1-Sep is the axiomatic system consisting of :

$$(\forall n, \phi(n) \Rightarrow \neg\psi(n)) \implies \exists Z, \forall n, \phi(n) \Rightarrow n \in Z \wedge \psi(n) \Rightarrow n \notin Z$$

for every Σ^1_1 formula with parameters.

Note that contrary to the comparability of well-ordering and the next example, the perfect tree theorem, this is not a theorem from natural mathematics. Indeed, proving it only require an instance of the axiom of comprehension, and the axiomatic system can be considered as a weakening of Σ^1_1-comprehension. The real Σ^1_1-separation theorem for subsets of $\mathbb{N}$ would be that the separating set Z is hyperarithmetic.

Theorem IV.53

Σ^1_1-Sep is equivalent to ATR_0 over RCA_0.

Proof. We first prove that ATR_0 implies Σ^1_1-Sep. We use our intuition of Σ^1_1 sets as co-enumerations along ω_1^{CK}. Let $A, B \subseteq \mathbb{N}$ be two disjoint sets and let α_n be the first stage where n has been enumerated either out of A or out of B. Then, let α be a bound on the α_n, and let $\hat{A}$ and $\hat{B}$ be the co-enumeration of A and B up to α. By ATR_0 it is possible to build those sets, and $\hat{A}$ or $\mathbb{N} \setminus \hat{B}$ provide a solution to the separation.

However, the proof should be conducted in second order arithmetic, so we detail a bit more. As ϕ_A and ϕ_B are Σ^1_1 formulas, let T_A and T_B be two functions such that $\phi_A(n)$ iff $T_A(n)$ is ill-founded, and $\phi_B(n)$ iff $T_B(n)$ is ill-founded. As $\phi_A(n)$ implies $\neg\phi_B(n)$, for every n at least one of $T_A(n)$ and $T_B(n)$ is well-founded. So let $\alpha = \sup_{n\in\mathbb{N}}(\min(\mathrm{KB}(T_A(n)), \mathrm{KB}(T_B(n))))$. By a simultaneous transfinite recursion along α, let H be a comparison $\mathrm{KB}(T_A(n))$ and $\mathrm{KB}(T_B(n))$ for every n. Then, the set $\{n \in \mathbb{N} : H \text{ says } \mathrm{KB}(T_A) \text{ is greater than } \alpha\}$ is a solution.

Now, we prove ATR_0 from Σ^1_1-Sep and RCA_0. As every arithmetical formula is also Σ^1_1, we already have arithmetical comprehension, and therefore we can prove the principle of transfinite induction. Let $\trianglelefteq$ be a well-ordering, and Φ be a formula. Then, the following formulas are Σ^1_1: $\phi_0(n) \equiv \exists H, \mathsf{Hier}_\Phi(H, \trianglelefteq) \wedge n \in H$ and $\phi_1(n) \equiv \exists H, \mathsf{Hier}_\Phi(H, \trianglelefteq) \wedge n \notin H$. Moreover, using a transfinite induction along $\trianglelefteq$, there cannot exists two different sets H_0 and H_1 with $\mathsf{Hier}_\Phi(H_0, \trianglelefteq)$ and $\mathsf{Hier}_\Phi(H_1, \trianglelefteq)$, so $\phi_0(n) \implies \neg\phi_1(n)$, and we can apply Σ^1_1-Sep to get a separating set W. And then, using another transfinite induction along $\trianglelefteq$, W is such that $\mathsf{Hier}_\phi(\trianglelefteq, W)$. ■

IV.4.3 Perfect Set Theorem

Here, we present yet another theorem that is equivalent to ATR_0 over RCA_0. However, this time we won't prove the equivalence, which is a bit more involved and refer the reader to [60]. We will only be interested in one implication: the proof of the perfect tree theorem in the context of second order arithmetic and ATR_0. It involves a surprising technique, called the method of pseudo-hierarchy, to get around a lack of expressiveness: the usual proof that was given in the context of descriptive set theory uses the ordinal ω_1, which is inaccessible in second order arithmetic.

The perfect tree theorem is a version of the continuum hypothesis. In its full version, it says that a $\mathbf{\Sigma^1_1}$ set is either countable, or of cardinality $2^{\aleph_0}$. However, the core result is that every closed set of Baire space has this property.

Definition IV.54

A *perfect set* is a closed set $A \subseteq \omega^\omega$ with no isolated point. A *perfect tree* $T \subseteq \omega^{<\omega}$ is a tree such that for every $\sigma \in T$, there exists incomparable $\tau_0, \tau_1 \in T$ extending σ.

It is clear that $A = [T]$ for a perfect tree is equivalent to being perfect.

Theorem IV.55 (Perfect Tree Theorem)

For every tree $T \subseteq \omega^{<\omega}$, either $[T]$ is countable, or there exists a perfect subtree of T.

Proof. Let $T \subseteq \omega^{<\omega}$ be a tree. For every α countable, we define by induction a tree T_α by removing at each step all the paths that are not splitting. More formally, we use the following induction:

— $T_0 = T$,

— Write $T_{<\alpha} = \bigcap_{\beta<\alpha} T_\beta$. Then,

$$T_\alpha = \{\sigma \in T_\alpha : \exists \tau_0, \tau_1 \in T_{<\alpha} \text{ incomparable and } \sigma \prec \tau_0, \tau_1\}.$$

Let us argue that there must be a countable ordinal α such that $T_\alpha = T_{\alpha+1}$. Indeed, suppose the construction strictly decrease the tree until ω_1. Then, there is a surjection from the set of removed strings $R = T \setminus T_{\omega_1}$ to ω_1: to any element σ, associate the ordinal α such that $\sigma \in R_\alpha = T_{<\alpha} \setminus T_\alpha$, the set of elements removed at time α. This is impossible, as $R \subseteq \omega^{<\omega}$ is countable, whereas ω_1 is not. Therefore, let α_0 be such that $T_{\alpha_0} = T_{\alpha_0+1}$. Now, we split the analysis in two cases.

The first is when we have $T_{\alpha_0} = T_{\alpha_0+1} \neq \emptyset$. Then, it is clear that T_{α_0} is a perfect set: any element of T_{α_0} is an element of T_{α_0+1} and therefore has two extensions in T_{α_0}.

The other case is when $T_{\alpha_0} = \emptyset$. We have that $[T] = \bigcup_{\alpha<\alpha_0}[T_{<\alpha}] \setminus [T_\alpha]$: the reverse

inclusion is clear as $T_{<\alpha} \subseteq T$, and if $X \in [T]$ then let β be the smallest such that $X \notin [T_\beta]$, we have $\beta \leq \alpha_0$. Then, $X \in [T_{<\beta}] \setminus [T_\beta]$. However, to each $\sigma \in T_{<\alpha} \setminus T_\alpha$ corresponds at most one element of $[T_{<\alpha}] \setminus [T_\alpha]$, so any of these set is countable. As α_0 is countable, we have a countable union of countable set: $[T]$ is countable. ■

It is clear in this proof how the Arithmetical Transfinite Recursion is used. However, the proof does not fit in the Second Order Arithmetic: indeed, we mention the ordinal ω_1 which cannot be encoded in second order arithmetic, and therefore we cannot conduct induction of this length. Therefore we are not directly able to conclude: it would be possible that $T_\alpha \neq \emptyset$ for every α countable, but we never have $T_\alpha = T_{\alpha+1}$. We need to use another technique to fill this lack of expressiveness.

Theorem IV.56

ATR_0 proves the perfect tree theorem IV.55.

Proof. Using the same argument as before, if there exists a countable α_0 such that $T_{\alpha_0+1} = T_{\alpha_0}$, then we are done. So, suppose we are not in this case. We claim that there exists an ill-founded linear order $\trianglelefteq$ and some $(T_s)_{s\in\mathrm{dom}(\trianglelefteq)}$ such that $\mathsf{Hier}_\Phi((T_s), \trianglelefteq)$ where Φ defines the construction of the proof of Theorem IV.55, with $T_s \neq T_{s+1}$ at any stage $s+1 \in \mathrm{dom}(\trianglelefteq)$.

Indeed, otherwise it would contradict Corollary II.4.1, as $\mathsf{WO}(\trianglelefteq)$ would be equivalent to $\exists (T_s)_{s\in\mathrm{dom}(\trianglelefteq)}$ such that $\mathsf{Hier}_\Phi((T_s), \trianglelefteq)$ and $T_s \neq T_{s+1}$ for all $s+1 \in \mathrm{dom}(\trianglelefteq)$.

Therefore, fix an ill-founded linear order $\trianglelefteq$, with $(s_n)_{n\in\mathbb{N}}$ an infinite decreasing sequence for $\trianglelefteq$ and $(T_s)_{s\in\mathrm{dom}(\trianglelefteq)}$ defined by induction. Let us build a perfect tree inside T.

Start with $L_0 = \{\epsilon\}$. We have $L_0 \subseteq T_{s_0}$ as $T_{s_0} \neq \emptyset$. If $L_n \subseteq T_{s_n}$ has been defined, then we define L_{n+1} by adding in this set, for every $\sigma \in L_n$, two incomparable strings $\tau_0, \tau_1 \in T_{s_{n+1}}$ extending σ. This is possible, as $s_{n+1} < s_n$, and by construction of T_α.

The tree $\{\sigma \in \omega^{<\omega} : \exists n \in \mathbb{N}, \tau \in L_n, \sigma \prec \tau\}$ generated by the union of L_n is included in T. Moreover it is by construction a perfect tree. ■

The method used in this proof is called by Simpson in [60] the method of pseudo-hierarchy. In the next section, we exhibit the general principle of this method, and show that it is equivalent to ATR_0.

IV.4.4 The Method of Pseudo-Hierarchies

The issue in the proof of Theorem IV.55 is that we need to do an induction until it reaches a particular state (in our case, $T_\alpha = T_{\alpha+1}$), without knowing in advance a countable bound on the required length of the induction. In the context of set theory, this is not a problem as we can consider ω_1-long induction, and argue afterward that the induction must have reached the particular state at a countable stage.

However, the ATR_0 principle only allows inductions for countably many steps, that we should give in advance. The method of pseudo-hierarchy uses the fact that $\mathbb{WO}$ is not Σ^1_1 to tackle this problem.

Definition IV.57 (Method of Pseudo-Hierarchy: MPH)

MPH is the axiomatic system, containing the following axiom for every Π^1_1 formula $\Psi(\trianglelefteq, H)$ and arithmetical formula $\Phi(n, \trianglelefteq, H)$:

1. Either there exists a well-order α and a hierarchy H for the formula $\hat{\Phi}(n, H) \equiv$

$\Phi(n, \alpha, H)$ on α such that $\Psi(\alpha, H)$,

2. or there exists an ill-founded $\trianglelefteq$ and a hierarchy H for the formula $\hat{\Phi}(n, H) \equiv \Phi(n, \trianglelefteq, H)$ on $\trianglelefteq$ such that $\neg\Psi(\trianglelefteq, H)$.

In the second, we call such a hierarchy a *pseudo-hierarchy*.

The formula Φ has to be considered as the inductive definition, while the formula Ψ has to be considered as a "halting condition": either there exists a countable ordinal reaching the halting condition, or there exists an ill-founded order with an induction along this order never finding the halting condition. As the order is not fixed in advance, we feed the formula with it so the inductive definition has access to the order.

Theorem IV.58

ATR_0 proves MPH over RCA_0.

Proof. Suppose that for some arithmetical Φ and Π^1_1 Ψ, we do not have the first possibility of MPH: For all $\trianglelefteq \in \mathbb{WO}$, there exists a hierarchy H for $\hat{\Phi}(n, H) \equiv \Phi(n, \trianglelefteq, H)$, so we must have $\neg\Psi(\trianglelefteq, H)$. Then, the Σ^1_1 formula $\theta(\trianglelefteq)$ being:

$$\exists H, \mathsf{Hier}_{\hat{\Phi}}(H, \trianglelefteq) \wedge \neg\Psi(\trianglelefteq, H)$$

is true on all well-order. As $\mathbb{WO}$ is not Σ^1_1 and θ is Σ^1_1, it cannot be an equivalence and there must exists an ill-founded order satisfying θ, which is exactly the second possibility of MPH. ■

We now show the reverse.

Theorem IV.59

MPH proves ATR_0 over RCA_0.

Proof. First, it is obvious that MPH implies ACA_0 by choosing $\Phi(n, \trianglelefteq, H) \equiv \varphi_n(n)\downarrow$, and $\Psi(\trianglelefteq, H) \equiv \mathrm{dom}(\trianglelefteq) \neq \emptyset$.

We will prove that MPH proves CWO over RCA_0. So, let α and β be two well-orders. Define $\Phi(\langle a, b\rangle, \trianglelefteq, H)$ to be true if and only if $H = \bigoplus_{a \in N} H_n$, where H_n is either empty or $H_n = \{\langle a_n, b_n\}$, and $a = \min\{x \in \mathrm{dom}(\alpha) : \forall n, x \geq_\alpha a_n\}$ and $b = \min\{x \in \mathrm{dom}(\beta) : \forall n, x \geq_\beta b_n\}$.

Let $\Psi(\trianglelefteq, H)$ be true if and only if one of the following is true, where $H_n = \{x \in \mathbb{N} : \langle n, x\rangle \in H\}$:

1. for some $n \in \mathrm{dom}(\trianglelefteq)$, H_n is not a singleton,
2. for all $n \in \mathrm{dom}(\trianglelefteq)$, H_n is the singleton $\langle a_n, b_n\rangle$; and there exists $m \lhd n$ such that $a_m >_\alpha a_n$.

By MPH for Φ and Ψ, there exists either a well-founded $\trianglelefteq$ and a hierarchy for Φ along $\trianglelefteq$ such that $\Psi(\trianglelefteq, H)$, or an ill-founded $\trianglelefteq$ and a hierarchy for Φ along $\trianglelefteq$ such that $\neg\Psi(\trianglelefteq, H)$.

Suppose $\trianglelefteq$ is the latter possibility. Then, as $\neg\Psi(\trianglelefteq, H)$, by $\neg 1$ we have that all H_n are singletons for $n \in \mathrm{dom}(\trianglelefteq)$. By $\neg 2$, if $m \lhd n$, then $a_m <_\alpha a_n$. Therefore, an infinite descending sequence in $\trianglelefteq$ implies an infinite descending sequence in α, a contradiction.

So, $\trianglelefteq$ must be well-founded, and H must verify Ψ. By arithmetical transfinite induction

along $\trianglelefteq$, at each step $s \in \mathrm{dom}(\trianglelefteq)$, $p_s = \{a : \langle s, a\rangle \in H\}$ is an embedding from an initial segment of α to an initial segment of β. Therefore, 2 must be false, and because Ψ is true, 1. must be true. But if for some s, H_s is empty, it means that there were no minimum either to $\mathrm{dom}(\alpha) \setminus \mathrm{dom}(p_s)$ or to $\mathrm{dom}(\beta) \setminus \mathrm{Im}(p_s)$, and therefore p_s is a comparison map. ■

IV.5 ATR in the Weihrauch lattice

We are now ready to study the ATR principle in the context of Weihrauch reduction. As a part of the big five, we saw that this principle is one of the most important of reverse mathematics, however it has been studied in the scope of Weihrauch reduction only recently, by Takayuki et al in [47], and Jun Le Goh in [51]. Unlike other principles, the exact Weihrauch degree candidate for ATR_0 is not so clear. In particular, the reverse mathematics equivalence with many theorems do not translate, as the techniques used in their proof might include non-constructive methods such as the pseudo-hierarchy method. Therefore, the first step of the study of ATR_0 in the Weihrauch scope is to exhibit the degrees that are natural candidates for being an analogue of ATR_0.

IV.5.1 The Weihrauch degrees of ATR

The ATR degree

We recall that ATR_0 is RCA_0 plus the principle that says that if $\trianglelefteq$ is a well order and Φ is arithmetic, then there exists a hierarchy for Φ on $\trianglelefteq$. It is also equivalent to the existence of Jump-hierarchies for every well-order $\trianglelefteq$. This principle therefore has the Π_2 form that allows us to naively turn it into a multi-valued function.

Definition IV.60 (ATR)

ATR is the multivalued function, with input any well order $\trianglelefteq$ on the integers and arithmetical formula Φ, and as output any H such that H is a Hierarchy for Φ on $\trianglelefteq$.

We need to emphasize that, just as in the settings of RCA_0, this principle is robust. In particular, Goh showed in [51] that creating Jump-hierarchies suffices to have the full power, and that we can always suppose we have access to the knowledge of which elements of the order are successor, and which are their predecessors.

Theorem IV.61 ([51])

The following versions of ATR are all Weihrauch equivalent:

1. **Input**: A couple $(\trianglelefteq, A)$ where $\trianglelefteq$ is a well-order.
   Output: A jump hierarchy relativized to A.
2. **Input**: A couple $(\trianglelefteq, A, \Phi)$ where $\trianglelefteq$ is a well-order and Φ is arithmetic.
   Output: A hierarchy for Φ on $\trianglelefteq$.
3. **Input**: A couple $(\trianglelefteq, A, S, p, \Phi)$ where $\trianglelefteq$ is a well-order, S is the successor elements of $\trianglelefteq$, p the predecessor function defined on S and Φ is arithmetic.
   Output: A hierarchy for Φ on $\trianglelefteq$.

The ATR_2 degree

Although ATR is shown robust by Theorem IV.61, its set of instances is very complex: only well-orders can be given as input. This limit its Weihrauch power, as for any P to be below ATR, an instance of P must compute an instance of ATR. However, when a linear order is not an instance of ATR because it is ill-founded, this can be witnessed by an infinite decreasing sequence. In other words, we can expand the logical form of ATR to another Π_2 form with more possible inputs. The standard way of writing ATR is:

$$\forall \lhd \in \mathbf{WO}, \exists H_{\lhd} \text{ a Jump Hierarchy on } \lhd,$$

but as, **WO** is defined by a Π^1_1 formula, it can also be written

$$\forall \lhd \in \mathbf{LO}, \exists (x_n)_{n\in\mathbb{N}} : (\forall n \in \mathbb{N}, x_{n+1} \lhd x_n) \vee (\exists H_{\lhd} \text{ a Jump Hierarchy for } \lhd)$$

This suggests the following variants of ATR, introduced by Jun Le Goh in [51].

Definition IV.62 (ATR_2, **[51]**)

ATR_2 is the multivalued function, with input any linear order $\trianglelefteq$ on the integers, and as output either an infinite decreasing sequence for $\trianglelefteq$, or any H such that H is a Jump-Hierarchy on $\trianglelefteq$.

One more time, Goh showed in [51] the robustness of this principle:

Theorem IV.63 (**[51]**)

The following versions of ATR_2 are all Weihrauch equivalent:

- **Input**: A couple $(\trianglelefteq, A)$ where $\trianglelefteq$ is a linear order.
  Output: Either a jump hierarchy on $\trianglelefteq$ relativized to A, or an infinite decreasing sequence for $\trianglelefteq$.
- **Input**: A tuple $(\trianglelefteq, A, \Phi)$ where $\trianglelefteq$ is a linear order and Φ is arithmetic.
  Output: A hierarchy for Φ on $\trianglelefteq$, or an infinite decreasing sequence for $\trianglelefteq$.
- **Input**: A tuple $(\trianglelefteq, A, S, p, \Phi)$ where $\trianglelefteq$ is a linear order, S is the successor elements of $\trianglelefteq$, p the predecessor function defined on S and Φ is arithmetic.
  Output: A hierarchy for Φ on $\trianglelefteq$, or an infinite decreasing sequence for $\trianglelefteq$.

The introduction of the principle ATR_2 has also some repercussion on the study of the Weihrauch lattice outside of the reverse mathematics point of view. In particular, it allows to answer Question IV.25, in Corollary IV.100.1. This question was first asked by Pauly and Brattka in [13] and later in [47].

The $\mathrm{ATR}_{2'}$ degree

We know by Theorem IV.87, that there exists ill-founded orders that still support Jump-Hierarchies. Therefore, for such orders, the solution for ATR_2 can be either an infinite decreasing sequence, or a Jump Hierarchy. The interest of having a Jump Hierarchy on an ill-founded order is in practice often relevant only on the well-founded initial segment of the ill-founded order. Therefore, we can weaken ATR_2 a bit by allowing more answers to it.

Definition IV.64 ($\mathrm{ATR}_{2'}$, **[2]**)

$\mathrm{ATR}_{2'}$ is the multivalued function, with input any linear order $\trianglelefteq$ on the integers, and as output either an infinite decreasing sequence for $\trianglelefteq$, or any H such that H is a Jump-Hierarchy on an initial segment of $\trianglelefteq$ that contains its well-founded part.

Seemingly, $\mathsf{ATR}_{2'}$ is completely unrelated to any other choice principles. Surprisingly, however, we will see that (the parallelization of) $\mathsf{ATR}_{2'}$ is arithmetically equivalent to the choice principle for Σ^1_1-compact sets, which is also equivalent to the Π^1_1 separation principle.

Theorem IV.65

$\widehat{\mathrm{ATR}_{2'}} \equiv^{\mathsf{a}}_{\mathsf{W}} \Sigma^1_1\text{-}\mathsf{KC}_{\mathbb{N}^{\mathbb{N}}} \equiv^{\mathsf{a}}_{\mathsf{W}} \widehat{\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aou}}_{\mathbb{N}}}$.

We divide the proof of Theorem IV.65 into two lemmas.

▶ **Lemma IV.66.** $\mathsf{ATR}_{2'} \leq^{a}_{\mathsf{W}} \widehat{\Sigma^1_1\text{-}\mathsf{UC}^{\mathsf{tot}}_2}$.

Proof. Fix $x \in \mathbf{LO}$. Given $n \in \mathbb{N}$, let JH_n be the set of jump hierarchies for $\prec_x\restriction_n$. Note that JH is an arithmetical relation. For $a, k \in \mathbb{N}$, if $a \prec_x n$ then let $S^n_{a,k}$ be the set of all $i < 2$ such that for some jump hierarchy $H \in JH_n$, the k-th value of the a-th rank of H is i, that is, $H_a(k) = i$. Otherwise, let $S^n_{a,k} = \{0\}$. Clearly, $S^n_{a,k}$ is Σ^1_1 uniformly in n, a, k, and therefore there is a computable function f such that $S^n_{a,k}$ is the $f(n, a, k)$-th Σ^1_1 set $G_{f(n)}$. Note that if $\prec_x\restriction_n$ is well-founded, then the product $\prod_{\langle a,k\rangle} S^n_{a,k}$ consists of a unique jump hierarchy for $\prec_x\restriction_n$. In particular, $S^n_{a,k}$ is a singleton for any $a \prec_x n$ and $k \in \mathbb{N}$ whenever $\prec_x\restriction_n$ is well-founded.

Given $p_{n,a,k} \in \Sigma^1_1\text{-}\mathsf{UC}^{\mathsf{tot}}_2(f(n, a, k))$, define $H_n = \bigoplus_{\langle a,k\rangle} p_{n,a,k}$. Note that if n is contained in the well-founded part of $\prec_x$, then H_n must be a jump hierarchy for $\prec_x\restriction_n$. By using an arithmetical power, first ask if H_n is a jump hierarchy for $\prec_x\restriction_n$ for every n. If yes, $\bigoplus_n H_n$ is a jump hierarchy along the whole ordering $\prec_x$, which is, in particular, large. If no, next ask if there exists a $\prec_x$-least n such that H_n is not a jump hierarchy for $\prec_x\restriction_n$. If yes, choose such an n, and then obviously n is not contained in the well-founded part of $\prec_x$. Hence, $\prec_x\restriction_n$ is a large initial segment of $\prec_x$. Moreover, by minimality of n, $\bigoplus\{H'_j : j \prec_x n\}$ is the jump hierarchy for $\prec_x\restriction_n$. If there is no such n, let j_0 be the $<_{\mathbb{N}}$-least number such that H_{j_0} is not a jump hierarchy for $\prec_x\restriction_{j_0}$, and $j_{n+1} \prec_x j_n$ be the $<_{\mathbb{N}}$-least number such that $H_{j_{n+1}}$ is not a jump hierarchy for $\prec_x\restriction_{j_{n+1}}$. By using an arithmetical power, one can find such an infinite sequence $(j_n)_{n\in\omega}$, which is clearly decreasing with respect to $\prec_x$. ■

▶ **Lemma IV.67.** $\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aou}}_{\mathbb{N}} \leq^{a}_{\mathsf{W}} \widehat{\mathsf{ATR}_{2'}}$.

Proof. Let S be a computable instance of $\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aou}}_{\mathbb{N}}$. Let $\prec_n$ be a linear order on an initial segment L_n of $\mathbb{N}$ such that $n \in S$ iff $\prec_n$ is ill-founded. Let H_n be a solution to the instance $\prec_n$ of $\mathsf{ATR}_{2'}$. Ask if there is n such that H_n is an infinite decreasing sequence w.r.t. $\prec_n$. If so, one can arithmetically find such an n, which belongs to S. Otherwise, each H_n is a jump hierarchy along a large initial segment J_n of L_n. In an arithmetical way, one can obtain J_n. Then ask if $L_n \setminus J_n$ is non-empty, and has no $\prec_n$-minimal element. If the answer to this arithmetical question is yes, we have $n \in S$.

Thus, we assume that for any n either $L_n = J_n$ holds or $L_n \setminus J_n$ has a $\prec_n$-minimal element. In this case, if $n \in S$ then J_n is ill-founded. This is because if J_n is well-founded, then J_n is exactly the well-founded part of L_n since J_n is large, and thus $L_n \setminus J_n$ is non-empty and has no $\prec_n$-minimal element. Moreover, since J_n admits a jump hierarchy while it is ill-founded,

J_n is a pseudo-well-order; hence H_n computes all hyperarithmetical reals. Conversely, if $n \notin S$ then H_n is a jump hierarchy along the well-order $J_n = L_n$, which is hyperarithmetic.

Now, ask if the following $(H_n)_{n\in\mathbb{N}}$-arithmetical condition holds:

$$(\exists i)(\forall j)\ H_i \not<_T H_j. \tag{IV.2}$$

By our assumption that $S \neq \emptyset$, there is $j \in S$, so that H_j computes all hyperarithmetic reals. Therefore, if (IV.2) is true, for such an i, the hierarchy H_i cannot be hyperarithmetic; hence $i \in S$. Then one can arithmetically find such an i. If (IV.2) is false, for any i there is j such that $H_i <_T H_j$. This means that there are infinitely many i such that H_i is not hyperarithmetical, i.e., $i \in S$. However, by our assumption, if S is infinite, then $S = \mathbb{N}$. Hence, any i is solution to S.

Finally, one can uniformly relativize this argument to any instance S. ■

Proof of Theorem IV.65 By Corollary IV.27.1 and Lemmas IV.66 and IV.67. ■

The $\mathrm{ATR}_2^{\mathsf{po}}$ degree

As the principle of construction by induction does not need the order to be linear, we can also define a partial order version of ATR_2.

Definition IV.68 ($\mathrm{ATR}_2^{\mathsf{po}}$, **[2]**)

$\mathrm{ATR}_2^{\mathsf{po}}$ is the multivalued function, with input any partial order $\trianglelefteq$ on the integers, and as output either an infinite decreasing sequence for $\trianglelefteq$, or any H such that H is a Jump-Hierarchy on $\trianglelefteq$.

▶ **Proposition IV.69.** $\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aou}}_{\mathbb{N}} \leq_{\mathsf{W}} \mathsf{ATR}_2^{\mathsf{po}}$.

Proof. Let S be a computable instance of $\Sigma^1_1\text{-}\mathsf{C}^{\mathsf{aou}}_{\mathbb{N}}$. Let T_n be a computable tree such that $n \in S$ iff T_n is ill-founded. Define

$$T = 00 \sqcup_n T_n = \{\langle\rangle, \langle 0\rangle, \langle 00\rangle\} \cup \{\langle 00n\rangle\sigma : \sigma \in T_n\}.$$

Let $i^\frown H$ be a solution to the instance T of $\mathsf{ATR}_2^{\mathsf{po}}$. If $i = 1$, i.e., if H is an infinite decreasing sequence w.r.t. T, then this provides an infinite path p through T. Then, choose n such that $00n \prec p$, which implies T_n is ill-founded, and thus $n \in S$. Otherwise, $i = 0$, and thus H is a jump hierarchy for T. We define $H^*_n = H_{\langle 00n\rangle}$. Note that if $n \notin S$ then H^*_n is hyperarithmetic, and if $n \in S$ then H^*_n computes all hyperarithmetical reals by Corollary IV.91.1 that we prove in section IV.6.1. By the definition of a jump hierarchy, we have $(H^*_n)'' \leq_T H$. Thus, the following is an H-computable question:

$$(\exists i)(\forall j)\ H^*_i \not<_T H^*_j. \tag{IV.3}$$

As in the proof of Lemma IV.67, one can show that if (IV.3) is true for i then $i \in S$, and if (IV.3) is false then any i is a solution to S. As before, one can uniformly relativize this argument to any instance S. ■

The unbounded construction

There is the other way to use the Arithmetical Transfinite Recursion principle of the perfect tree theorem; where we don't know in advance the number of steps that we need to construct the object we need, but we know that it will happen at some point.

We already defined the MPH principle in the context of second order arithmetic. However, this pseudo-hierarchy was in some sense a trick to counter the fact that we cannot prove the Perfect Tree Theorem directly, as it makes use of ω_1: we need to use a "pseudo-hierarchy" method as in Theorem IV.56. A principle is derivated from this in Definition IV.57 and has been proven equivalent to ATR in Theorem IV.58.

In the Weihrauch context, there is no need for such tricks, as we are not focused in "provability". Therefore, we define two Weihrauch degrees corresponding to these techniques (the plain one of Theorem IV.55 and the "pseudo-hierarchy" trick of Theorem IV.56). They both corresponds to one specific usage of the arithmetical transfinite recursion principle.

Definition IV.70

$\mathrm{ATR}_{\mathrm{unb}}$ is the multivalued function, with:

— **input**: an arithmetic formula $\Phi(n, \trianglelefteq, H)$ and a Σ^1_1 formula $\Psi(\trianglelefteq, H)$ such that there is an output,

— **output**: An ordinal α and a hierarchy H for $\hat{\Phi} \equiv \Phi(n, \alpha, H)$ up to α such that $\Psi(\alpha, H)$.

Definition IV.71

MPH is the multivalued function, with input an arithmetic formula $\Phi(n, \trianglelefteq, H)$ and a Σ^1_1 formula $\Psi(\trianglelefteq, H)$, and with output:

1. Either a well order $\alpha \in \mathbf{WO}$ and a Hierarchy H for $\hat{\Phi} \equiv \Phi(n, \alpha, H)$ up to ordinal α, with $\Psi(\alpha, H)$.
2. Or a code for a linear order $\trianglelefteq$, together with an infinite decreasing sequence (x_n), and a hierarchy H for $\hat{\Phi}$ on $\trianglelefteq$, such that $\neg\Psi(\trianglelefteq, H)$.

We consider $\mathrm{ATR}_{\mathrm{unb}}$ as more important than MPH, as we see the latter more as an artefact of second order arithmetic than an actual way to use arithmetical transfinite recursion.

IV.5.2 Comparing the different versions of ATR

We compare the previously defined different degrees for the arithmetical transfinite recursion principle. They are linearly ordered, but not every strictness is known. We have the following:

Theorem IV.72

$$\mathrm{ATR} <_{\mathrm{W}} \mathrm{ATR}_{2'} \leq_{\mathrm{W}} \mathrm{ATR}_2 \leq_{\mathrm{W}} \mathrm{ATR}_2^{\mathsf{po}}$$

Proof. It is clear that $\mathrm{ATR} \leq_{\mathrm{W}} \mathrm{ATR}_{2'}$, by identity witnesses, as given a well-order, $\mathrm{ATR}_{2'}$ has no other choice than answering the Jump Hierarchy for the whole order. This strictness comes from the fact that ATR always has a hyperarithmetic solution, while there exists an instance of $\mathrm{ATR}_{2'}$ that has no hyperarithmetic solution: for any pseudo-well order, as defined in Definition IV.86, both an infinite descending sequence and a jump-hierarchy cannot

be hyperarithmetic by Corollary IV.91.1 and Theorem IV.96.

It is clear that $\mathrm{ATR}_{2'} \leq_{\mathrm{W}} \mathrm{ATR}_2$ by identity functions, as any solution for the latter is a solution for the former. Finally, again by identity witnesses, we have $\mathrm{ATR}_2 \leq_{\mathrm{W}} \mathrm{ATR}_2^{\mathsf{po}}$ as the latter is the former with a bigger domain. ■

The principles introduced in this thesis are strictly (far) above the previously defined principles:

Theorem IV.73

$\mathrm{ATR}_2^{\mathsf{po}} <_{\mathrm{W}} \mathrm{ATR}_{unb}$ and $\mathrm{ATR}_2 <_{\mathrm{W}} \mathrm{ATR}_{unb}$.

Proof. This follows from Theorem IV.78 and Theorem IV.77. ■

▶ **Question IV.74.** Do we have $\mathrm{ATR}_2 \equiv_{\mathsf{W}}^{a} \mathrm{ATR}_{2'} \equiv_{\mathsf{W}}^{a} \mathrm{ATR}_2^{\mathsf{po}}$? Or do we have $\mathrm{ATR}_{2'} <_{\mathrm{W}} \mathrm{ATR}_2 <_{\mathrm{W}} \mathrm{ATR}_2^{\mathsf{po}}$

▶ **Question IV.75.** Do we have $\mathrm{ATR}_{\mathrm{unb}} \equiv_{\mathrm{W}} \mathrm{MPH}$?

IV.5.3 Relation between choices and ATR.

The Choice principles are good references when studying new Weihrauch degree, as they are already well-studied and provide a good hierarchy.

Theorem IV.76

We have that $\mathrm{ATR} \equiv_{\mathrm{W}} \Sigma^1_1\text{-}\mathrm{UC}_{\omega^\omega}$.

Proof. Let us prove that $\mathrm{ATR} \leq_{\mathrm{W}} \Sigma^1_1\text{-}\mathrm{UC}_{\omega^\omega}$. Given a well-order $\trianglelefteq$, the set of jump-hierarchies on $\trianglelefteq$ is a Π^0_2 singleton, therefore ATR can be solved using $\Sigma^1_1\text{-}\mathrm{UC}_{\omega^\omega}$. Conversely, let Φ be a Σ^1_1 formula defining a singleton $\mathcal{A} = \{A\}$. The singleton is Δ^1_1, as $n \in A$ iff $\exists X \in \mathcal{A}$, $n \in X$ iff $\forall X \in \mathcal{A}$, $n \in X$. Then, there is a computable bound α such that $n \in A$ iff $f(n) \in \mathcal{O}_{\leq\alpha}$ for all n. But $\mathcal{O}_{\leq\alpha}$ can be defined by an α long induction. ■

Theorem IV.77 ([51])

We have $\mathrm{ATR}_2^{\mathsf{po}} <_{\mathrm{W}} \mathrm{C}_{\omega^\omega}$.

Proof. We first prove that $\mathrm{ATR}_2^{\mathsf{po}} <_{\mathrm{W}} \mathrm{C}_{\omega^\omega}$. Given a partial order $\trianglelefteq$, the set of jump-hierarchies on $\trianglelefteq$ is a Π^0_2 set, and the set of infinite decreasing sequences for $\trianglelefteq$ is a Π^0_1 set. Therefore $\mathrm{ATR}_2^{\mathsf{po}}$ can be solved using $\Sigma^1_1\text{-}\mathrm{C}_{\omega^\omega} \equiv_{\mathrm{W}} \mathrm{C}_{\omega^\omega}$.

Now, let us argue that $\mathrm{C}_{\omega^\omega} \not\leq_{\mathrm{W}} \mathrm{ATR}_2^{\mathsf{po}}$. Suppose otherwise via Turing functionnals Φ and Ψ. Then, we have that T is ill-founded if and only if

$$\Phi(T) \text{ is a partial order} \wedge \forall S \in \mathrm{ATR}_2^{\mathsf{po}}(\Phi(T)),\ \Psi(T,S) \in [T]$$

As $\mathrm{ATR}_2^{\mathsf{po}}(\Phi(T))$ is arithmetic, the above formula is arithmetic, which is a contradiction with the set of ill-founded trees not being $\mathbf{\Delta^1_1}$. ■

Therefore, all the principles ATR, $\mathrm{ATR}_{2'}$, ATR_2 and $\mathrm{ATR}_2^{\mathsf{po}}$ lay in the gap between Π^0_1-$\mathrm{C}_{\omega^\omega}$ and Π^0_2-$\mathrm{C}_{\omega^\omega}$. Between those two choice principles, we also have the other choice principles Σ^1_1-WKL and Σ^1_1-$\mathrm{AC}_{\mathbb{N}}$. While we already know that Σ^1_1-WKL is strictly in between, we will obtain the separation of independent and dependent choice as a corollary by proving that that ATR_2 is not below Σ^1_1-$\mathrm{AC}_{\mathbb{N}}$.

For the more powerful problems $\mathrm{ATR}_{\mathrm{unb}}$ and MPH, Choice on Baire space is not trivially sufficient to output a solution. The output of a solution needs to generate a sufficiently large ordinal, a Π^1_1 set. Those principles are above the closed choice in Baire space:

Theorem IV.78

$\mathrm{C}_{\omega^\omega} \leq_{\mathrm{W}} \mathrm{MPH}$ and $\mathrm{C}_{\omega^\omega} \leq_{\mathrm{W}} \mathrm{ATR}_{\mathrm{unb}}$.

Proof. Consider the construction which starts with a tree $T \subseteq \omega^{<\omega}$, remove leaves along the ordinals, and stop when there is no more leaves.

If we are given an ordinal α and a construction $(T_\beta)_{\beta\leq\alpha}$ where T_α has no leaves, then it is easy to compute a path in $T_\alpha \subseteq T$ as there is no dead-ends.

Suppose we are given a sequence $(T_{x_n})_{n\in\mathbb{N}}$ taken from $(T_\beta)_{\beta\leq\alpha}$ a construction on an ill-founded order α, with (x_n) being an infinite descending sequence. Then, one can compute an infinite path, as every leaf of T_{x_n} must have an extension in $T_{x_{n+1}}$. To compute the path, take the leftmost path of T_{x_0} until it reaches a dead-end. If that happens, continue by extending this left-most path in T_{x_1} until it reaches a dead-end, always taking the next tree when reaching a dead-end. ■

IV.5.4 Weihrauch degrees of theorems at the ATR_0 level

In section IV.5.1, we developed different Weihrauch degrees for the arithmetical transfinite recursion principle, arguing that they correspond to different usage of ATR_0 in proofs. We now see how the equivalence over RCA_0 of the theorems presented in section IV.4, transfers in the Weihrauch lattice.

We start with Σ^1_1-Sep. Due to the relative nature of Σ^1_1-definitions, as witnessed by Theorem II.18, we will not only consider Σ^1_1 separation but also Π^1_1 separation.

Definition IV.79

We define the following principles:

- Σ^1_1-Sep is the multivalued function with:
  - **Input**: Two Σ^1_1 descriptions of sets $A, B \subseteq \mathbb{N}$ with $A \cap B = \emptyset$,
  - **Output**: A set C such that $A \subseteq C$ and $B \cap C = \emptyset$.
- Π^1_1-Sep is the multivalued function with:
  - **Input**: Two Π^1_1 descriptions of sets $A, B \subseteq \mathbb{N}$ with $A \cap B = \emptyset$,
  - **Output**: A set C such that $A \subseteq C$ and $B \cap C = \emptyset$.

Note that this deals with subsets of $\mathbb{N}$. One could also define Σ^1_1-separation for subsets of $\mathbb{R}$, the output being a Borel code, however Π^1_1 separation by a Borel set does not hold.

Theorem IV.80 ([47])

We have $\mathrm{ATR} \equiv_{\mathrm{W}} \Sigma^1_1\text{-Sep} <_{\mathrm{W}} \Pi^1_1\text{-Sep} \equiv_{\mathrm{W}} \Sigma^1_1\text{-WKL}$.

This contrasts with their relation in reverse mathematics: Σ^1_1-separation is strictly stronger than Π^1_1-separation over RCA_0 [55]. The explanation is again that the interpretation of Σ^1_1 formulas are dependent on the model, as shown by Spector and Gandy, so that it is not surprising that they behave differently in the Weihrauch and the reverse math case.

The perfect tree theorem has many corresponding Weihrauch degrees. Similarly to ATR, it is for every instance, the disjunction of two Π^1_1 sentences. So there are at least three different Π^1_2 formulations of it: the straightforward formulation, and the two possible "negation of one case implies the other case".

Definition IV.81 ([47])

— PTT_1 is the multivalued function with:
- **Input**: A tree $T \subseteq \omega^{<\omega}$,
- **Output**: A perfect tree $T' \subseteq T$

— $\mathrm{PTT}_{\mathrm{list}}$ is the multivalued function with:
- **Input**: A tree $T \subseteq \omega^{<\omega}$,
- **Output**: An enumeration $(b_i)_{i\in\mathbb{N}}$ such that $[T] \subseteq \{b_i : i \in \mathbb{N}\}$.

— $\mathrm{PTT}_{\mathrm{det}}$ is the multivalued function with:
- **Input**: A tree $T \subseteq \omega^{<\omega}$,
- **Output**: Either a perfect tree $T' \subseteq T$, or an enumeration $(b_i)_{i\in\mathbb{N}}$ such that $[T] \subseteq \{b_i : i \in \mathbb{N}\}$.

In [47] Kihara, Marcone and Pauly proved that these three problems have very different degrees:

Theorem IV.82 ([47])

$\mathrm{ATR} \equiv_{\mathrm{W}} \mathrm{PTT}_{\mathrm{list}} <_{\mathrm{W}} \mathrm{PTT}_1 <_{\mathrm{W}} \mathrm{PTT}_{\mathrm{det}}$.

However, all of them are below the unbounded version of the arithmetical transfinite recursion principle:

Theorem IV.83

$\mathrm{PTT}_{\mathrm{det}} \leq_{\mathrm{W}} \mathrm{ATR}_{\mathrm{unb}}$.

Proof. By the exact same proof as Theorem IV.55. ∎

The last theorem that is equivalent to ATR_0 over RCA_0 is the comparability of well-orders. Kihara, Marcone and Pauly have shown in [47] that its Weihrauch degree is equivalent to ATR, and Goh showed that it is a robust one in [51].

Definition IV.84 ([47, 51])

CWO is the multivalued function with:
- **Input**: Two well-orders $\trianglelefteq$ and $\preccurlyeq$,
- **Output**: Either an embedding from $\trianglelefteq$ to an initial segment of $\preccurlyeq$, or an embedding from $\preccurlyeq$ to an initial segment of $\trianglelefteq$,

The multivalued function WCWO is the versions of CWO where the embedding is not required to be into an initial segment.

Theorem IV.85 ([47, 51])

We have CWO $\equiv_{\mathrm{W}}$ ATR $\equiv_{\mathrm{W}}$ WCWO.

It remains to see how the introduction of new principles might benefit the study of the Weihrauch degrees.

IV.6 Separating Σ^1_1 axioms choice and dependent choice

The goal of this section is to answer Question IV.25. Recall the informal presentation of this question: Suppose we have a black box that can find a path in any higher co-c.e. homogeneous tree, where homogeneous means that every nodes at a given level share the same extensions. Can we use this black box to solve the same problem where the trees are not required to be homogeneous?

We will solve this problem by showing that ATR_2 is not Weihrauch below Σ^1_1-AC_ω. As it is Weihrauch below Σ^1_1-DC_ω, we can conclude that the two axioms of choice are not Weihrauch equivalent. The crucial property of homogeneous sets we will use is that if H is homogeneous and $A \leq_s H$ via Φ, then if $\Phi^\sigma = \tau$ for some $\sigma \prec X \in H$, then there exists a Turing functional Ψ such that $\forall X \in H$, we have $\tau \prec \Psi^X \in A$. Indeed, choose Ψ^Y to be $\Phi^{X \cap [0,|\sigma|] \cup Y \cap [|\sigma|+1,\infty[}$. In other words, we can force a finite part of the computed elements of A.

In the context of being Weihrauch above ATR_2, any solution of an instance of ATR_2 specifies in its finite part which type of answer it is: a hierarchy or an infinite decreasing sequence. So by the above, if an instance is Medvedev below a homogeneous set, then every element of the homogeneous set uniformly computes a solution in a consistent manner: computed solutions are either all hierarchies, or all infinite descending sequences.

We can show that using this previous fact would allow us to define a Δ^1_1 set containing all the well-orders, together with some specific ill-orders, supposing ATR_2 below Σ^1_1-AC_ω. This does not directly consists of a contradiction as we only know that WO is not Δ^1_1, but a study of Σ^1_1 sets containing the recursive well-orders will allow us to conclude. This is what we do in the following subsection IV.6.1.

IV.6.1 Supportive orders and pseudo well orders

In this part, we present the phenomenon used by the method of pseudo-hierarchies: the existence of ill-founded orders that still allows some kind of induction along them. We will not consider arbitrary construction by induction, but only iteration of the jump, so we recall the definition of jump hierarchies:

▶ **Definition** (Jump-Hierarchy)**.** We say that H is a Jump-Hierarchy on $\trianglelefteq$ if H is a hierarchy for $\Phi(\langle a, b\rangle, X) \equiv b \in X'_a \equiv \varphi_b^{X_a}(b)\downarrow$. In other word, if H is a Jump-Hierarchy for $\trianglelefteq$, then

$$H_a = \bigoplus_{b \lhd a} H'_b.$$

The predicate "H is a Jump-Hierarchy for $\trianglelefteq$" is written $\mathsf{JH}(H, \trianglelefteq)$ and is Π^0_2.

We are now able to define the properties on the orders we will be interested in.

Definition IV.86 (Supportive orders and Pseudo-well orders)

Let $\lhd$ be a linear order on the integers.

1. We say that $\lhd$ is a Φ-*supportive order* if it supports a hierarchy for Φ, that is there exists H a hierarchy for Φ on $\lhd$. We write Φ-Supp for the set of indices of recursive Φ-supportive orders. When talking about Jump-hierarchies, we simply omit .
2. We say that $\lhd$ is a *pseudo-well order* if there is no infinite hyperarithmetic decreasing sequence. We write PWO for the set of indices of recursive pseudo well orders.

Note that, compared to the property of being well-ordered, the property of being supportive and pseudo-well-ordered is a property of the order and not the order-type.

These sets have already been well-studied, as in [42, 32, 62]. Gathering their work in this document, we will prove the following theorem, which appeared in [32]:

Theorem IV.87 ([32])

$\mathsf{WO} \subsetneq \mathsf{Supp} \subsetneq \mathsf{PWO}$.

Proof. By Lemma IV.88, Theorem IV.96 and Theorem IV.97. ■

We start with the easy part:

▶ **Lemma IV.88.** *For every arithmetic Φ, we have $\mathsf{WO} \subsetneq \Phi$-$\mathsf{Supp}$.*

Proof. By ATR, it is clear that $\mathsf{WO} \subseteq \Phi$-$\mathsf{Supp}$. However, Φ-Supp is a Σ^1_1 set: $e \in \Phi$-Supp if and only if $\exists H$ such that $\mathsf{Hier}_\Phi(H, \leq_e)$, which is a Σ^1_1 statement. As by Theorem II.2, WO is Π^1_1 complete, it cannot be Σ^1_1, and $\mathsf{WO} \subsetneq \Phi$-$\mathsf{Supp}$. ■

Although the previous proof is very short, and does not rely on complex theorems, the result is quite intriguing and unexpected. Indeed, it says that there exists a supporting order which is not well-founded, and so there is an infinite sequence $(X_n)_{n\in\mathbb{N}}$ such that for every n, $X'_{n+1} \leq_T X_n$. However, this existence result is inherently non constructive! And proving it directly would seem complicated, in fact we will see more formally that constructing such a sequence must be sufficiently complicated.

We now devote our efforts to prove the second part of Theorem IV.87, $\mathsf{Supp} \subseteq \mathsf{PWO}$. In order to do this, we will follow the following strategy:

0. If $\lhd$ is a well-order, it is also a pseudo-well-order, so we can already suppose that $\lhd$ is ill-founded, but supports a hierarchy (such hierarchies are called *pseudo-hierarchies*).
1. First, we will show that any pseudo-hierarchy associated with an ill-founded supportive order computes every hyperarithmetic set.
2. Second, show that in a sequence such that $X'_{n+1} \leq_T X_n$ for every n, the indices of the reduction is highly non uniform, even in some X_n,
3. Lastly, suppose that $\lhd$ is ill-founded, supportive and with an infinite decreasing sequence. Then, find a contradiction with 1. and 2.

One consequence of step 1. is that a pseudo-hierarchy cannot be hyperarithmetic. Pseudo-hierarchies for the jump contains sequences that decrease of at least one jump at each step. These sequences will be central, so we name them:

Definition IV.89

We say that $(X_n)_{n\in\omega}$ is a Jump-decreasing sequence if it is such that for every n, we have $X'_{n+1} \leq_T X_n$.

Pseudo-hierarchies and Jump-decreasing sequences

Recall that we already supposed that $\lhd$ supports a pseudo-hierarchy. In this part we study the computational strength of both pseudo-hierarchies and Jump-decreasing sequences. First, we show that a pseudo hierarchy always contains a Jump-decreasing sequence, in the following sense:

▶ **Fact IV.90.** If H is a pseudo-hierarchy, then there exists $(x_n)_{n\in\mathbb{N}}$ such that $(H_{x_n})_{n\in\mathbb{N}}$ is a Jump-decreasing sequence.

Proof. As H is a pseudo-hierarchy, it is a Jump-hierarchy on some ill-founded order $\lhd$. Let x_n be an infinite decreasing sequence for $\lhd$. Then, clearly $H'_{x_n+1} \leq_T H_{x_n}$ by just taking the projection of H_{x_n} on x_{n+1}. But then, $(H_{x_n})_{n\in\mathbb{N}}$ is a Jump-decreasing sequence. ■

Now, to give a lower bound on the computational power of pseudo-hierarchies, it suffices to give a common lower bound on every element of a Jump-decreasing sequence. This is what we achieve in the following theorem: By this fact, in order to show that a pseudo hierarchy uniformly computes every hyperarithmetic set, we can use the following theorem:

Theorem IV.91 ([31])

If $(X_n)_{n\in\omega}$ is Jump-decreasing, then every X_n computes all hyperarithmetic sets uniformly in its double jump.

Proof. We show by induction on α that for any Jump-decreasing sequence $(X_n)_{n\in\mathbb{N}}$ and $\alpha < \omega_1^{\mathrm{CK}}$, we have that $X_0 \geq_T 0^{(\alpha)}$. Note that we do not require any kind of uniformity in this statement.

For the base case, it is obvious that $X_0 \geq_T 0^{(0)}$ for any X_0. Now, if the result is true for α, as $(X_{n+1})_{n\in\mathbb{N}}$ is also a Jump-decreasing sequence, X_1 computes $0^{(\alpha)}$. But as $X_0 \geq_T X'_1$, $X_0 \geq_T 0^{(\alpha+1)}$, concluding the successor case.

Now we need to deal with the limit case, when λ is limit. The problem is that the successor proof is not uniform: it depends on the indices used for the reductions from X'_1 to X_0, and going back in the induction from X'_{i+1} to X_i. As we will see later, these reductions cannot be recovered uniformly. However, the reductions from X_i to $0^{(\alpha)}$ can be recovered using a finite number of jump, as seen below.

By the induction hypotheses, for every e code for $\alpha < \lambda$, we have that $X_2 \geq_T H_e$ where H_e is the Jump Hierarchy on $\leq_e$. Let l be a code for λ, that is ϕ_l codes an order $\leq_{\phi_l}$ of order-type *lambda*, and g be such that $\phi_g(i)$ codes for the well-order below the i-th element enumerated in the domain of $\leq_{\phi_l}$. Therefore, for any i, $\varphi_g(i)$ is a code for an ordinal below λ, and there exists r such that $\varphi_r^{X_2} = H_{\varphi_g(i)}$. As stated in Definition I.89 deciding this equality is Π^0_2 in $\varphi_g(i)$ and $\varphi_r^{X_2}$, thus given i and the double jump of X_2 one can compute H_l. As $X_0 \geq_T X''_2$, we have $X_0 \geq_T 0^{(\lambda)}$, concluding the limit case.

Now we have he result except for the uniformity, that we can get exactly the same way we did for the limit case. ■

▶ **Corollary IV.91.1.** *If H is a pseudo hierarchy on a computable order with no lowest element, then it computes all the hyperarithmetic sets.*

Proof. By Fact IV.90 let $H = (H_{x_n})_{n\in\mathbb{N}}$ be a Jump-Decreasing sequence. Then by the previous theorem, $H \geq_T H_{x_0}$ computes every hyperarithmetic set. ■

▶ **Corollary IV.91.2.** *The set* HYP *of all hyperarithmetic sets is not a model of* ATR_0

Proof. By Lemma IV.88, there exists a recursive order with no hyperarithmetic descending sequence. This order looks like a well-order inside HYP. However, by Corollary IV.91.1 any Jump-Hierarchy on this order would not be hyperarithmetic, so HYP does not satisfy ATR_0 for this specific order. ■

In particular any hierarchy on a supportive order that is not a well-order cannot be hyperarithmetic (otherwise it would compute its own jump). Contrary to this, a hierarchy on a recursive well-order is hyperarithmetic by definition. Therefore, we have a characterization of well-orders that has the Spector-Gandy Theorem as consequence:

Theorem IV.92 (Spector [61], Gandy [35])

Every Π^1_1 predicate $P(x)$ is equivalent to a predicate of the form: $\exists Y \in \mathrm{HYP} : A(Y, x)$ for some arithmetic A.

Proof. By Theorem II.2, $P(x)$ is equivalent to "$\varphi(x)$ codes a well-founded order" for some φ. We claim that

$$P(x) \iff \exists H \in \mathrm{HYP},\ H \text{ is a hierarchy on } \varphi(x).$$

Indeed, if $P(x)$ holds, then $\varphi(x)$ codes a well founded order and the hyperarithmetic $H_{\phi(x)}$ is a jump-hierarchy for $\varphi(x)$. If $\neg P(x)$ then $\varphi(x)$ is not well-founded, it might be supportive but by Corollary IV.91.1 it has no hyperarithmetic hierarchy. ■

Uniformity in Jump-decreasing sequences

We saw that any pseudo-hierarchy contains all the elements of a Jump-decreasing sequence. However, this does not imply that it computes a Jump-decreasing sequence, as it might be complex and non-uniform to know which layer to take. In particular, the key to recover a Jump-decreasing sequence from a pseudo-hierarchy is an infinite decreasing sequence, our initial interest. Therefore, in this section we are interested in uniformity inside Jump-decreasing sequences.

We will show that in any Jump-decreasing sequence, there is no arithmetic uniform way to go from one element to the next one. This is expressed by the following definition:

Definition IV.93 (P-sequence)

If P is an arithmetic formula, a *P-sequence* is a sequence $(X_n)_{n\in\omega}$ such that for every n, X_{n+1} is the unique set X such that $P(X_n, X)$. In this case, we write $X = P(X_n)$.

Therefore, we are able to express the negative result about uniformity in Jump-decreasing sequences:

Theorem IV.94 ([62])

There is no Jump-decreasing P-sequence.

Before doing the proof, it is useful in order to understand it more easily to draw a parallel with an informal proof of Gödel's Theorem. Suppose that T is a theory, and T proves $\mathrm{Con}(T)$. Then, by Gödel's completeness theorem, T proves[3] that there exists a model of T. Fix any model $\mathcal{M}_0$ of T. As it is a model of T, it contains $\mathcal{M}_1$ also model of T. We can continue and construct models of T $\mathcal{M}_n$ for every $n \in \omega$.

Now, suppose the construction of $(\mathcal{M}_n)_n$ is such that we can express the predicate R such that R is true in $\mathcal{M}_n$ if and only if "$\forall i > 0$, R is false in $\mathcal{M}_{n+i}$". Then, we can easily derive a contradiction: If R is true in $\mathcal{M}_0$, it is false in $\mathcal{M}_i$ for $i > 0$. In particular, it is false in $\mathcal{M}_1$, and therefore it is true in some $\mathcal{M}_j$ for $j > 1$, a contradiction with R being true in $\mathcal{M}_0$. If R is false in $\mathcal{M}_0$, it is true in some $\mathcal{M}_i$ and we use the same argument.

Now, we do the proof in the settings of P-sequence.

Proof of Theorem IV.94 Let $P(X, Y)$ be any arithmetical formula. We say that X is a *beginning* if there exists a Jump-decreasing P-sequence $(X_n)_{n\in\mathbb{N}}$ with by $X_0 = X$. Our goal is to show that there is no beginning. The element X_n of such a sequence would corresponds roughly to a model $\mathcal{M}_n$ of the previous paragraph. The next Lemma is a step to express the formula R, by identifying e, k with the e-th Σ^0_{k+1} formula.

▶ **Lemma IV.95.** *There exists an arithmetical formula $R(A, e, k)$ such that if A is any beginning, then:*

$$R(A, e, k) \iff \exists y > k + 1 : e \notin (P^y(A))^{(k+1)} \tag{IV.4}$$

Proof. What is needed to be shown for this lemma is just that the right part of the equality can be put in an arithmetic form. In its current form, it is Σ^1_1, as when expanded in a lower level language it becomes $\exists y > k + 1\ \exists (Y_n)_{n\leq y}$ such that $Y_0 = A$, $Y_{n+1} = P(Y_n)$ and $e \notin Y_y^{(k+1)}$.

However, in the Lemma we are in the special case where A is a beginning, that is $(P^n(A))_{n\in\mathbb{N}}$ is defined and a Jump-decreasing sequence. In particular, for every n, $A \geq_T P^n(A)$. This allows us to quantify over the cone below A instead of all Ys: an arithmetic quantification. Under this new light, R is defined to be:

$$R(A, e, k) \overset{\text{def}}{\iff} \exists y > k + 1, \exists (i_n)_{n\leq y} : \begin{cases} \Phi^A_{i_0} = A \quad \wedge \\ \forall n < y,\ \Phi^A_{i_{n+1}} = P(\Phi^A_{i_n}) \quad \wedge \\ e \notin \left(\Phi^A_{i_y}\right)^{(k+1)} \end{cases}$$

Therefore, R is arithmetic and satisfy the equivalence of the Lemma. ■

For the moment, in our analogy with the argument above the proof, R says: the e-th Σ^0_k formula false in some model $\mathcal{M}_y$ for $y \geq k$. We will consider the case where R is given its own code. More formally, as R is arithmetic, let e_0 and k_0 be codes for R, that is such that $R(A, x, y) \Leftrightarrow \langle x, y\rangle \in W_{e_0}^{A^{(k_0)}}$. Replacing R in (IV.4) and fixing $k = k_0$, we get that for every beginning A:

$$\forall e, \quad \langle e, k_0\rangle \in W_{e_0}^{A^{(k_0)}} \iff \exists y > k + 1 : e \notin P^y(A)^{(k_0+1)} \tag{IV.5}$$

Let $\widetilde{e}_0$ be such that $n \in W^X_{\widetilde{e}_0}$ if and only if $\langle n, k_0\rangle \in W^X_{e_0}$. We now apply (IV.5) with $e = \widetilde{e}_0$,

3. We suppose T strong enough to prove the completeness theorem.

so for every beginning A:

$$\langle \widetilde{e}_0, k_0 \rangle \in W_{e_0}^{A^{(k_0)}} \iff \exists y > k+1 : \widetilde{e}_0 \notin P^y(A)^{(k_0+1)} \tag{IV.6}$$

$$\widetilde{e}_0 \in W_{\widetilde{e}_0}^{A^{(k_0)}} \iff \exists y > k+1 : \widetilde{e}_0 \notin P^y(A)^{(k_0+1)} \tag{IV.7}$$

$$\widetilde{e}_0 \in A^{(k_0+1)} \iff \exists y > k+1 : \widetilde{e}_0 \notin P^y(A)^{(k_0+1)} \tag{IV.8}$$

Coming back to our analogy, the last statement is the one that is true if and only if it is false in some bigger $\mathcal{M}_n$.

■

Proof of Theorem IV.87

We are now ready to combine all the previous results to prove Theorem IV.87.

Theorem IV.96 ([32])

If $e \in \mathsf{Supp}$, then $e \in \mathsf{PWO}$.

Proof. Let H be a Jump-hierarchy on e and let $(x_n)_{n\in\mathbb{N}}$ be an infinite decreasing sequence. Toward a contradiction, suppose that $(x_n)_{n\in\mathbb{N}}$ is hyperarithmetic.

As $(H_{x_n})_{n\in\mathbb{N}}$ is a Jump-decreasing sequence, by Theorem IV.91 for every n the element H''_{x_n} uniformly computes the whole hyperarithmetic sequence $(x_n)_{n\in\mathbb{N}}$, via some fixed Turing functional Φ. Therefore, $(H_{x_n})_{n\in\mathbb{N}}$ is a P-sequence where P is such that :

$$\begin{aligned} P(X,Y) \iff \quad & X'' \text{ compute the sequence } (x_n) \text{ via } \Phi, \text{ and} \\ & \text{if } n \text{ is such that } X \text{ is a jump hierarchy for the order } x_n, \\ & \text{then } Y' = \{a : \langle x_{n+1}, a\rangle \in X\} \end{aligned}$$

This is a contradiction with Theorem IV.94. ■

We also mention the following theorem from Harvey Friedman's thesis:

Theorem IV.97 ([33, Chapter 3 Theorem 2])

There exists a pseudo well order which supports no jump hierarchy.

IV.6.2 Complexity of Supp and PWO

In the previous section, we have shown $\mathsf{WO} \subseteq \mathsf{Supp} \subseteq \mathsf{PWO}$. We will now tackle their complexity, using a very general theorem proved by Harrington.

We will work in the slightly more general setting of partial order. We extend the sets to WF, $\mathsf{Supp}_{\mathsf{po}}$ and PWF corresponding to the same notions for partial orders, in the very straightforward way. Harrington's theorem is the following:

Theorem IV.98

If $A \subseteq \mathbb{N}$ is a Σ^1_1 set such that $\mathsf{WF} \subseteq A \subseteq \mathsf{PWF}$, then A is Σ^1_1-complete.

Proof. Let A be such a set. In order to prove that A is Σ^1_1-complete, we will reduce to A the most famous Σ^1_1-complete set: IF, the set if indexes of ill-founded trees.

For any e, write U_e for the tree of code e. Also, as A is Σ^1_1, for any n there exists a tree V_n such that V_n is ill-founded if and only if $n \in A$.

The reduction f will be such that

$$[U_{f(e)}] = \{g : U_e \to V_{f(e)} \text{ order preserving}\}.$$

In order to define this f, we need to use the recursion theorem, in the following way: Given e, let h be the procedure that given n, output a code for the tree $\{g \subseteq: U_e \to V_n |\ g$ is finite and order preserving$\}$. By the recursion theorem, let n_0 be a fixed-point of h, that is n_0 is a code for $\{g \subseteq: U_e \to V_{n_0} : g$ is finite and order preserving$\}$ and define $f(e) = n_0$. Then, f is a computable function by the effectivity of the recursion theorem.

Before proving that f is a reduction, we prove that for every e, $U_{f(e)}$ is ill-founded. If $f(e) \in A$, then $V_{f(e)}$ is ill-founded by definition of $(V_n)_{n\in\mathbb{N}}$. But then, for every tree T there exists an order preserving function from T to $V_{f(e)}$, and $[U_{f(e)}] \neq \emptyset$. Otherwise, if $f(e) \notin A$, as $\mathsf{WF} \subseteq A$, $f(e) \notin \mathsf{WF}$ and $U_{f(e)}$ is ill-founded. Therefore, there exists an embedding from U_e to $V_{f(e)}$, that is a total order preserving function.

Now we prove that f is a reduction.

— Suppose $e \in \mathsf{IF}$, that is U_e is ill-founded. As U_e embeds into $V_{f(e)}$, the latter must be ill-founded, which by definition implies $f(e) \in A$.

— Otherwise, $e \notin \mathsf{IF}$, that is U_e is well-founded. As we already know that there exists an embedding from U_e to $V_{f(e)}$, by Lemma IV.99 below there must exist a hyperarithmetic one. But then, $U_{f(e)}$ contains a hyperarithmetic infinite descending sequence, and therefore $f(e) \notin \mathsf{PWF}$. As $A \subseteq \mathsf{PWF}$, $f(e) \notin A$.

It only remains to prove the Lemma:

▶ **Lemma IV.99.** *If $T_0, T_1 \subseteq \omega^{<\omega}$ are computable trees such that T_0 is well founded and there exists an order-preserving function from T_0 to T_1, then there exists such a function that is hyperarithmetic.*

Proof. We begin with some notation. We write T_σ for $\{\tau \in T : \tau \succ \sigma\}$. We call *rank* of the tree T the function l such that $l(\sigma) = \infty$ if T_σ is ill-founded, and $l(\sigma) = \sup_{\sigma \prec \tau \in T}(l(\tau)+1)$ otherwise. By convention, ∞ is greater than any ordinal.

As T_0 is well founded and computable, the rank l_0 is defined by induction over a computable ordinal, and therefore is hyperarithmetic. Let α_0 be the computable ordinal associated to the root, $l_0(\epsilon)$.

Contrary to l_0, the rank l_1 of T_1 may not be hyperarithmetic as T_1 may not be well-founded. However, the function l_1^* corresponding to the rank constructed only up to stage α_0 is hyperarithmetic: where we have $l_1^*(\sigma) = l_1(\sigma)$ if $l_1(\sigma) \leq \alpha_0$, and $l_1^*(\sigma) = \infty$ otherwise.

We will prove that for every $\sigma \in T_0$ and $\tau \in T_1$, the following are equivalent:

1. there exists an embedding f from $T_{0,\sigma}$ to $T_{1,\tau}$ with $f(\sigma) = \tau$
2. there exists a hyperarithmetic embedding from $T_{0,\sigma}$ to $T_{1,\tau}$ with $f(\sigma) = \tau$
3. $l_0(\sigma) \leq l_1(\tau)$
4. $l_0(\sigma) \leq l_1^*(\tau)$

The equivalence between 3 and 4 if obvious by definition of l_1^*. The implication from 2 to 1 is obvious, we will now prove 1 implies 3 and then 3 implies 2.

We prove 1 implies 3 by induction on the rank of σ. Fix an embedding f, we have that $l_0(\sigma) = \sup_{\sigma \prec \rho \in T_0} l_0(\rho) \leq \sup_{\sigma \prec \rho \in T_0} l_1(f(\rho))$ by induction hypothesis, but then as f is an embedding $l_0(\sigma) \leq \sup_{\tau \prec \rho \in T_1} l_1(\rho) = l_1(\tau)$.

We prove 3 implies 2. We define by induction the embedding $f_{\sigma\tau}$ from $T_{0,\sigma}$ to $T_{1\tau}$. To define $f_{\sigma,\tau}$, for every $\rho \succ \sigma$ direct successor of σ in T_0, fix any $\mu(\rho) \succ \tau$ such that $l_1(\mu(\rho)) \geq l_0(\rho)$, which must exist. Then, define $f_{\sigma,\tau}$ to be $\{(\sigma,\tau)\} \cup \bigcup\{f_{\rho,\mu(\rho)} : \rho \text{ direct successor of } \sigma\}$. Then, $f_{\sigma,\tau}$ is an hyperarithmetic embedding.

Applying the equivalence with σ and τ being the empty sequence, we get the existence of a hyperarithmetic embedding. ■

■

▶ **Corollary IV.99.1.** *If $A \subseteq \mathbb{N}$ is a Σ^1_1 set such that $\mathsf{WO} \subseteq A \subseteq \mathsf{PWO}$, then A is Σ^1_1-complete.*

Proof. Let A be such a set. If e is an index for a tree in Baire space, we write $\mathrm{KB}(e)$ for the index of the order of the associated Kleene-Brouwer ordering. We let B be $\{e \in \mathbb{N} : \mathrm{KB}(e) \in A\}$. Then, we have $\mathsf{WF} \subseteq B \subseteq \mathsf{PWF}$: If T is well-founded of code e, then the Kleene-Brouwer ordering of T is a well-order and $\mathrm{KB}(e) \in \mathsf{WO} \subseteq A$. If there exists an infinite hyperarithmetic descending sequence in T, then this sequence is also infinite decreasing for $\mathrm{KB}(e)$ and $\mathrm{KB}(e) \notin A$.

Therefore, B is Σ^1_1-complete. As there exists a many-one reduction from B to A, A is also Σ^1_1-complete. ■

▶ **Corollary IV.99.2.** Supp *and* PWO *are Σ^1_1-complete.*

IV.6.3 ATR_2 versus Σ^1_1-AC_ω and Σ^1_1-DC_ω

We are finally able to answer Question IV.25, using the background of this section. This is a very interesting situation, where the introduction of a new degree, ATR_2, purely inspired from reverse mathematics, allowed to close an open question asked by Pauly and Brattka in [13], a paper that is purely on Weihrauch considerations.

Theorem IV.100

$\mathrm{ATR}_2 \not\leq_{\mathrm{W}} \Sigma^1_1\text{-}\mathrm{AC}_\omega$

Proof. Toward a contradiction, we suppose that $\mathrm{ATR}_2 \not\leq_{\mathrm{W}} \Sigma^1_1\text{-}\mathrm{AC}_\omega$. As the above principle is parallelizable, we then have $\widehat{\mathrm{ATR}}_2 \not\leq_{\mathrm{W}} \Sigma^1_1\text{-}\mathrm{AC}_\omega$. Fix Φ and Ψ witnesses of this reduction.

Let $I = \bigoplus_{e\in\mathsf{LO}} I_e$ be the product of all the computable instances of ATR_2. As A is an instance of $\widehat{\mathrm{ATR}}_2$, $\Phi(A)$ is the presentation of a product of Σ^1_1 sets $S = \prod_n B_n$ such that for every $x \in S$, $\Psi(x)$ is the product of solutions to every computable instance of ATR_2.

Given $e \in \mathsf{LO}$ and $x \in S$, we are interested in what kind of solutions for I_e are computed by x. We write $\mathrm{Answer}_x(e) \subseteq \{\mathbf{H};\mathbf{D}\}$ for the kind of solution to I_e that are computed by x, that is $\mathbf{H} \in \mathrm{Answer}_x(e)$ iff and only if $\exists f$ computable with $f(x)$ is a Jump-hierarchy for I_e, and similarly for $\mathbf{D}$ with an infinite descending sequence. As $(\Psi(x))_e$ is always a solution, $\mathrm{Answer}_x(e) \neq \emptyset$ for every $x \in S$.

We define $\mathrm{answer}_S(e) = \bigcap_{x\in S} \mathrm{answer}_x(e)$. If $e \in \mathsf{WO}$, then we have that $\mathrm{answer}_S(e) = \{\mathbf{H}\}$, as there is no infinite descending sequence in I_e. Contrary to this, $\mathrm{answer}_S(e) = \{\mathbf{D}\}$ for $e \notin \mathsf{Supp}$ as there is no associated hierarchy. However, by Theorem IV.87 there are some $e \in \mathsf{Supp} \setminus \mathsf{WO}$ that we need to take care of.

For those $e \in \mathsf{Supp} \setminus \mathsf{WO}$, we argue that $\mathrm{answer}_S(e) \neq \emptyset$. Indeed, let $x \in S$, $(\Psi(x))_e$ is a solution of type $i \in \{\mathbf{H};\mathbf{D}\}$. Let $\sigma \prec x$ finite such that $\Psi(\sigma)$ already determines the type of solution it start to compute for I_e. Then, for every $y \in S$, $\Psi(\sigma(y_{\restriction>|\sigma|}))$ is a solution to I as by homogeneity $\sigma(y_{\restriction>|\sigma|}) \in S$. But then $i \in \mathrm{answer}_y(e)$ for all $y \in S$, and $i \in \mathrm{answer}_S(e)$.

Define the Σ^1_1 sets

$$\begin{aligned} B &= \{e : \mathbf{H} \notin \mathrm{answer}_S(e)\} \\ C &= \{e : \mathbf{D} \notin \mathrm{answer}_S(e)\}. \end{aligned}$$

As for every $e \in LO$, $\mathrm{answer}_S(x) \neq \emptyset$, $A \cap B = \emptyset$. We also have $\mathsf{WO} \subseteq C$ and $\mathsf{LO} \setminus \mathsf{Supp} \subseteq B$. As B and C are Σ^1_1, by Lusin's Separation Theorem for integer, corresponding to Theorem IV.53, there exists a Δ^1_1 set D with $\mathsf{WO} \subseteq C \subseteq D$ and $D \cap B = \emptyset$. But then by Theorem IV.87, $B \supseteq \mathsf{PWO}$ and so $D \cap \mathsf{PWO} = \emptyset$. By Corollary IV.99.1, D is Σ^1_1-complete, a contradiction. ■

We can now answer the question about axioms of choice and dependant choice.

▶ **Corollary IV.100.1.** $\Sigma^1_1\text{-}\mathrm{AC}_\omega \not\leq_{\mathrm{W}} \Sigma^1_1\text{-}\mathrm{DC}_\omega$

Proof. We have $\mathrm{ATR}_2 \not\leq_{\mathrm{W}} \Sigma^1_1\text{-}\mathrm{AC}_\omega$, but $\mathrm{ATR}_2 \leq_{\mathrm{W}} \Sigma^1_1\text{-}\mathrm{C}_{\omega^\omega} \equiv_{\mathrm{W}} \Sigma^1_1\text{-}\mathrm{DC}_\omega$, so the two principles cannot be equivalent. ■

Chapter V

Randomness and genericity within ITTMs

Gotlib, RàB T2. Le matou matheux

One of the striking observation of mathematical logic is that given a countable alphabet, the set of things we can define is countable, so in this language we can only speak of countably many reals even though there are many more of them. This can in some ways be given a formal statement, known as the descending Löwenheim-Skolem theorem. So, coming back to the reals, we *need* the existence of reals that we *cannot define*, making them rather difficult to deal with.

So if mathematics only deals about an infinitesimal part of the real numbers, what do the other largest part look like? Even if we cannot define some individual, can we speak about them as a whole? As they are not really definable, they should not have any discriminating property. In other words, we use the interesting small part of the reals, the major part being a mass of typical objects, all of them different but sharing the same features, making them indistinguishable.

There are several ways of studying these reals, the mass sharing the most common properties. Usually, we define a notion of largeness to decide which are the most common properties. Then, we get a hierarchy on typicality of reals, defined by the complexity of the simplest common property that is not verified. This includes two very different notions of typicality: randomness and genericity.

The study of randomness fills a gap left by probability theory. Consider the following sequences of bits: "00000000000000000000", "01010101010101010101", and "11001001110100111000". Probability theory says that if they are obtained by tossing a fair coin many times, all three outcomes are equally probable. However, our intuition strongly suggests that the first one was using a biased coin, the second was purely cheating and the last one might have been obtained by toss-

ing a coin [1]. Similarly, if we compare infinite sequences of bits, they all have probability 0 to be obtained by tossing a coin infinitely many times. Yet, we expect them to have many properties, such as having roughly as many 0s than they have 1s, to follow the law of large numbers, and in some sense, to be very disordered.

The study of the randomness of infinite sequences and the one of finite sequences both converged to the foundation of algorithmic randomness, providing a clear and satisfying answer to the above contradiction between the theory and the intuition. The resulting theory of algorithmic randomness measures the randomness content of a finite string as the size of the most efficient way of compressing the string, still being able to recover it in a computable fashion (or at least, using a reasonably powerful oracle). It measures the randomness of an infinite string by finding how typical it is with regard to the sets of measure 1. It turns out that many such notions are linked, and an infinite string is random if and only if its finite prefixes all have maximal randomness, up to a constant.

Algorithmic randomness, since the first randomness notion by Martin-Löf in 1966 [53], has known an impressive development. A very rich theory has emerged, as a complex and beautiful answer to the original philosophical question of what are random objects. Just like recursion theory had been extended to higher recursion theory, to α-recursion theory and to the theory of ITTMs, algorithmic randomness is meant to follow a similar development. This has been started with Higher randomness by Hjorth and Nies [45], Chong and Yu [23] [24] and Bienvenu, Greenberg and Monin [39] [7] [54]. Recently this was extended to ITTMs by Carl and Schlicht [19]. The goal of this chapter is first to pursue their work.

This chapter aims at continuing the study of randomness and genericity at the higher levels. In Section V.1 we recall some basic facts and definitions about typicality notions. In Section V.2, we develop a framework that can be used in general to study randomness and genericity within Gödel's constructible hierarchy. We use this framework to answer several open questions of Carl and Schlicht, and we also ask new ones.

Some formal details of Section V.2 may be a bit tedious to read, and there is no way around that. Any recursion theorist may have struggled in its early days to read all the technical details on the equivalence between various models of computations, and developed after that a very solid intuition of what is computable, without the necessity of coming back every time to the formal definitions. Thus, the reader who is not familiar with constructibility will certainly need to furnish an effort with some proofs of Section V.2, whereas the reader who is used to it will certainly have no problem admitting these theorems without reading the proofs. Despite the difficulties inherent to the material presented here, we tried as much as possible to never confuse rigor and formalism, by ensuring the former without getting trapped in the latter.

In Section V.3, even though we answer several questions of [19], we feel that this section's main achievement is not there, but more in a new question (Question V.30) that we ask on the separation of two randomness notions defined by Carl and Schlicht. It seems so clear at first that the two notions should be different, that the question was not asked so far. The reason is certainly that the analogues of these two notions in Higher randomness actually differ for simple reasons. We emphasize here that things are not so simple in the settings of ITTMs, and we show that the two notions are much closer than we think, even though we are not able to settle the question.

This question was the original motivation for the work presented in Section V.4: In order to argue that it is not absurd to think that these two randomness notions may actually coincide, we show that it is the case for their categorical analogues. Note that the versions of these analogues with Higher genericity are also known to differ for simple reasons, like it is the case

1. In fact, it was obtained using a pseudo-random generator. So, continuing this way would result in something strongly non-random in the sense of this chapter.

with randomness. In some sense, Theorem V.59 that shows equality of these notions, may actually be the most important of this chapter: it uses the new phenomenons that occur within some levels of the constructible hierarchy to show that two classes collapse in a very unexpected way. Despite that, we decided to leave this section at the end, so that the chapter follows the logic exposed so far, that we now sum up.

In Section V.1, we expose the definitions and basic results of randomness and genericity. In Section V.2 we develop a general framework to study randomness in any limit level of Gödel's constructibles, in Section V.3 we study randomness notions with respect to ITTMs, focusing first on the question we mentioned above and proving several results meant to delineate and sharpen the contour and outline of this question. In this same section we then answer several questions of [19], the most interesting theorem about that being maybe Theorem V.41. In Section V.4 we define and study, in the setting of ITTMs, the categorical analogues of the studied randomness notions. The section focuses on answering for categoricity the question that is still too hard in the randomness case.

V.1 Notions of typicality

Genericity and randomness are two different approaches to study typical objects, that is, objects having "all the typical properties" for some notion of typicality. For randomness, a property is typical if the class of reals sharing it is of measure 1, whereas for genericity, a property is typical if the class of reals sharing it is co-meager. In both cases, for any countable collection of typical properties, it is still a typical property to have all of them: the intersection of countably many measure-one sets is still a measure-one set, and the intersection of countably many co-meager sets is still a co-meager set. Depending on the countably many properties we consider, the reals that share all of them may be of great interest, in forcing constructions or to study various notions of degrees, from Turing to α-degrees.

Computability theory establishes many countable hierarchies of complexity, making these a good choice for the study of typical objects. Indeed, if one defines a random real as being a real with all the measure 1 properties among a countable class, it is natural to include in the class the less complex properties, before the more exotic ones (such as $2^\omega \setminus \{X\}$ where X is a set that looks like very random).

These hierarchies of complexity define hierarchies of randomness and genericity. A set is more random if it appears random with regards to all sets of a higher complexity. Many notions of randomness and genericity have been defined, ranging from rather simple sets to ITTM-semi-decidable sets. Following the previous chapters of this thesis, we will be more interested in the high-end of the spectra, possibly the strongest class of randomness defined so far.

In this section, we review the two notions of typicality: first, randomness which relies on measure theory, and genericity which relies on Baire categoricity. Measure and categoricity are two different ways to measure the largeness of a subset of the reals.

V.1.1 Measure Theory

A measure is a value of largeness given to subsets of reals. Measuring every subsets of the real in a consistent but non trivial way may not seem a difficult task, however there are so many of them ($\mathcal{P}(\mathbb{R})$ is of cardinality $2^{2^{\aleph_0}}$), perhaps even some are not constructible, that we restrict the measurable sets to a smaller class: a σ-algebra.

Definition V.1 (σ-algebra)

A *σ-algebra* $\mathcal{A}$ on X is a subset of $\mathcal{P}(X)$, closed by complementary and countable union, and containing X.

Of course, $\mathcal{P}(\mathbb{R})$ is a σ-algebra. However, as said before it is not so clear that there exists a well-behaving measure on so many complex sets. Therefore we will restrict ourselves to the smallest σ-algebra containing the open sets.

▶ **Example V.2.** The Borel sets form a σ-algebra. It is the smallest σ-algebra containing the open intervals.

Proof. Let us show that the Borel sets form a σ-algebra. First, the Borel sets is a collection closed by intersection. If $A_n \in \mathbf{\Pi^0_{\alpha_n}}$ for all $n \in \mathbb{N}$, as the α_n are countable we have that $\alpha = \sup \alpha_n$ is also countable. But then, $A = \bigcup_n A_n$ is $\mathbf{\Pi^0_{\alpha+1}}$, and the Borel sets are also closed by countable unions, therefore consist of a σ-algebra.

We now show the converse, that any σ-algebra containing the open sets must contain the Borel sets: Let $\mathcal{A}$ be a σ-algebra containing the open intervals. Suppose that there exists an α and a $\mathbf{\Pi^0_\alpha}$ set $A \notin \mathcal{A}$ with $\alpha < \omega_1$. Let α be the smallest such one, it cannot be 1 as $\mathcal{A}$ contains all $\mathbf{\Sigma^0_1}$ sets. Then A is the complement of a union of sets in $\mathcal{A}$, as every $\mathbf{\Pi^0_\beta}$ set for $\beta < \alpha$ is in $\mathcal{A}$. But then, the complement of A is in $\mathcal{A}$, so $A \in \mathcal{A}$. ■

Then, we measure the sets of a σ-algebra.

Definition V.3 (Measure)

If $\mathcal{A}$ is a σ-algebra on X, then a function $\mu : \mathcal{A} \to \mathbb{R} \cup \{\infty\}$ is a measure if we have the following:

1. $\mu(\emptyset) = 0$, and
2. If $(A_n)_{\in\mathbb{N}}$ is a family of disjoint sets in $\mathcal{A}$, then $\mu(\bigcup_n A_n) = \sum_n \mu(A_n)$.

We call the second requirement *σ-additivity*. If we also have $\mu(X) = 1$, then we say that μ is a *probability measure*.
Elements of the σ-algebra are called *measurable sets*. A measurable set of measure 0 is called a *null set*.

There exists some measures on the full σ-algebra on 2^ω, such as for instance, given some fixed $x \in 2^\omega$, the measure valued 1 on a set A if $x \in A$ and 0 otherwise. This measure is called the Dirac measure on x. However, if we want a more natural measure on the reals, we may have to restrict the σ-algebra first. However, given a σ-algebra and a measure on it, there is sometimes a cheap way to increase the number of measurable sets, by *completing* a measure.

Definition V.4

Let $\mathcal{A}$ be a σ-algebra on X and μ a measure on it. We say that $N \subseteq X$ is negligible if N is included in some measure 0 set. Then, the set $\mathcal{B} = \{A \cup N : A \in \mathcal{A} \text{ and } N \text{ is negligible}\}$ is a σ-algebra, and the measure defined by $\mu^*(A \cup N) = \mu(A)$, for some negligible N, is a measure on $\mathcal{B}$. It is called the *completed measure of* μ. If a measure equals its own completion, then the measure is said to be *complete*.

Definition V.5 (Lebesgue Measure)

There exists a unique complete measure μ on the Borel sets, such that $\mu([a,b]) = b - a$ where $a, b \in \mathbb{R}$. This measure is called *Lebesgue measure*.

Usually, Lebesgue measure is written λ. In this thesis, we avoid this notation as λ is already used to denote the supremum of writable ordinals.

Before continuing to algorithmic randomness, we need to show that for a probability measure, a countable intersection of measure 1 sets is also of measure 1. The complement of a measure 1 set is a measure 0 set. By σ-additivity, the countable union of null sets is null. Therefore, the countable intersection of co-null set is co-null.

V.1.2 Algorithmic Randomness

Once we have defined the largeness notion and showed that countably many "large" sets are "large", we can use computability hierarchies to define typical objects.

Definition V.6 ($\mathcal{C}$-randomness)

Let $\mathcal{C} \subseteq \mathcal{P}(2^\omega)$ be countable. A set $x \in 2^\omega$ is *$\mathcal{C}$-random* if it is in no null set of $\mathcal{C}$, that is $\forall A \in \mathcal{C},\ \mu(A) = 0 \implies x \notin A$.

We explain why this definition corresponds to our intuition of random sets, and yields robust notions of randomness. There are usually three paradigms to define randomness:

1. The first paradigm is the one used here: the difficulty to find some non randomness in the set. If a set is obtained by tossing a coin infinitely many times, then we expect the set to have no special properties.
2. The second paradigm is unpredictability. If a sequence of 0 and 1 is random, then given a finite initial segment, there should be no way to "guess" the next bit. This is usually formalized by bounding the expectation of a gambler, who tries to win money by betting the next bits of the sequence, steps after steps, but has limited computational power.
3. The third paradigm is the incompressibility of prefixes. If a sequence is random, then there must be no structure in the sequence allowing to compress the initial segment in a code, still being able to recover the initial segment using limited computational power.

We give the most famous choice of $\mathcal{C}$ in algorithmic randomness:

Definition V.7

A *Martin-Löf test* is a Π^0_2 set effectively of measure 0, that is a uniform intersection $\bigcap_{n\in\mathbb{N}} \mathcal{U}_n$ of open sets $\mathcal{U}_n$ with $\mu(\mathcal{U}_n) \leq 2^{-n}$.
We say that x is *Martin-Löf random* if X is in no Martin-Löf test.

V.1.3 Baire categoricity

We start by giving the analogue of co-null sets for Baire categoricity, that is the largeness notion.

Definition V.8

Let (X, τ) be a topological space. Then, a set $A \subseteq X$ is *co-meager* is it contains a countable intersection of dense open sets. A set is *meager* if it is the complement of a co-meager set.

The terminology of meager and co-meager sets suggests that we defined them reversed of the usual way. The reason is that we feel that "dense open" is more easily understood than using the notion of "nowhere density".

We now argue that co-meagerness is indeed a good candidate for being a notion of largeness. It can be seen as the notion of largeness induced by open dense sets, when closing the notion by countable intersection. Open and dense sets look like a good largeness notion: if a set A is open dense, it means that for every x, not only there is an approaching sequence to x in A, but there are intervals in A arbitrarily close to x.

Yet, it remains to prove that co-meagerness is not a trivial notion. This is so in what is usually called "Baire spaces", but the terminology clashes with the (also standard) notion of "the" Baire space, ω^ω. In this new notion, a Baire space is a topological space in which every intersection of countably many dense open sets is dense. The Baire category theorem states that every complete metric space, as well as every locally Hausdorff space, are Baire spaces. We only do the proof for *the* Baire space and the Cantor space.

Theorem V.9

For Baire and Cantor spaces, the intersection of countably many dense open sets is dense.

Proof. Let $(D_n)_{n\in\mathbb{N}}$ be a countable union of dense open sets, and σ be a finite string (of bits, or of integers). We define by induction a sequence $(\sigma_n)_{n\in\mathbb{N}}$, where $[\sigma_n] \subseteq D_n$, $\sigma_n \prec \sigma_{n+1}$ and the length is strictly increasing. We start with $\sigma_{-1} = \sigma$ the empty string, and if σ_{n-1} has been defined, we use the density of D_n to define $x_n \succ \sigma_{n-1}$ in D_n, followed by the use of the openness of D_n to find $\hat{\sigma}_n \prec x_n$ such that $[\hat{\sigma}_n] \subseteq D_n$. If $\hat{\sigma}_{n-1} \prec \sigma_{n-1}$ we define $\sigma_n = \sigma_{n-1}{}^\frown 0$, otherwise $\sigma_n = \hat{\sigma}_n$.

It is clear that $x \succ \sigma$ is in every D_n, as $\sigma_n \prec x$, concluding the theorem. ■

V.1.4 Genericity

Just as we defined random objects as those which have all co-null simple properties, we define generic objects as those which have all co-meager simple properties. Note that every open set can be "densified": if $P \subseteq \omega^\omega$ is an open set, then $\hat{P} = P \cup \bigcup\{[\sigma] : [\sigma] \cap P = \emptyset\}$ is a dense open set. This densified set corresponds to the fact that given any open property, and beginning σ of a constructed object, either we can force the final object to be in P, or we can force the final object to avoid P, by just choosing an extension $\tau \succ \sigma$ in $\hat{P} \setminus P = \{[\sigma] : [\sigma] \cap P = \emptyset\}$ (if the former is false, then we must already have $\sigma \in \hat{P} \setminus P$). After an extension τ is chosen, no matter how the construction continues above τ, we already know if the constructed object will belong to P or not.

So, we can see the generic objects as the one having all the properties that can be ensured in a finite way. For instance, having 1000 zeros one after the other can be ensured after any beginning of a string. So generic objects will have this property. Note that this is also a property of random objects, as never having 1000 zeros in a row in the binary expansion has probability 0.

Definition V.10

Let $\mathcal{D}$ be a countable family of dense open sets. Then, a sequence g is generic over $\mathcal{D}$ if for every $D \in \mathcal{D}$, $g \in D$.

So, let us give a property that generic objects have, but random objects don't. The set $P_n = \bigcup\{[\sigma^\frown 0^{\mathrm{lh}(\sigma)}] : \sigma \in 2^{<\omega} \wedge \mathrm{lh}(\sigma) > n\}$ is a dense open set, as if $\sigma \in 2^{<\omega}$ is any sufficiently long finite binary string, then $[\sigma^\frown 0^{\mathrm{lh}(\sigma)}] \subseteq P_n$. So if $g \in 2^\omega$ is generic, then there must be infinitely often a sequence of zeros as long as the sequence before the zeros, which happens with probability 0.

Genericity is closely related to forcing. In fact, Cohen used the previously defined notion of genericity, over the sets of a countable model of ZFC, to enlarge the model with a generic real in some way. He called this method forcing. By modifying the topology to another Baire space, we get other definitions of forcing, but in this thesis we will be only interested in the above defined Cohen genericity.

It is straightforward to use hierarchies from computability and logic to define hierarchies of genericity notions.

Definition V.11

Let $\alpha < \omega_1^{\mathrm{CK}}$. A sequence is *weakly-$\alpha$-generic* if it belongs to every Σ^0_α dense open sets. It is *α-generic* if it belongs to every densification of Σ^0_α dense open set.

The fact that we require the Σ^0_α to be open might seem a restrictive condition to the set of properties that can be ensured for generic elements. However, it is not such a restriction. In fact, a potentially not open property can be approximated in an open property, up to a meager set. Then, any "generic enough" element will be in the open approximation, but not in the meager set of errors, therefore the generic will have the initial property.

Definition V.12

A set A has the *Baire property* if there exists an open set U and a meager set M such that $A = U \Delta M$, where Δ is the symmetric difference: $A \Delta B$ is $(A \setminus B) \cup (B \setminus A)$.

Having the Baire property is being equal to an open set up to a meager error. Now, we need that a Σ^0_α set have the Baire property in a way that works well with α-genericity.

Theorem V.13 (Baire property theorem for lightface Borel sets)

For any Σ^0_α set $A \subseteq 2^\omega$, one can find uniformly in an index for A a Σ^0_α-open set U and uniformly in n a $\Pi^0_{<\alpha}$-closed set F_n such that $A = U \Delta B$ for some set B included in $\bigcup_n \delta F_n$, where δF_n is the boundary of F_n.
For any Π^0_α set $A \subseteq 2^\omega$, one can find uniformly in an index for A a Π^0_α-open set U and a uniformly in n a Π^0_α-closed set F_n such that $A = U \Delta B$ for some set B included in $\bigcup_n \delta F_n$, where δF_n is the boundary of F_n.

Therefore, for every Σ^0_α P, there is a Σ^0_α open set U equal to P for α-generic.

Theorem V.14

Let $\alpha < \omega_1^{\mathrm{CK}}$, let $\mathcal{G}_\alpha$ be the set of α-generic reals and $\mathcal{WG}_\alpha$ be the set of weakly-α-generic reals. Then, for every Σ^0_α set P we have:

1. There exists a Σ^0_α-open set $\mathcal{U}$ with $P \cap \mathcal{WG}_\alpha = \mathcal{U} \cap \mathcal{WG}_\alpha$.
2. There exists a Π^0_α-open set $\mathcal{U}$ with $P \cap \mathcal{G}_\alpha = \mathcal{U} \cap \mathcal{G}_\alpha$.

V.2 Forcing in the constructibles

Algorithmic randomness normally deals with Borel sets of positive measure. Working in the constructibles will make this task a little bit harder, and requires to go into usual naming and forcing in L.

We will however not formally define a forcing relation. Instead we go around the need of defining one, by directly dealing with Borel sets. The reason we do so is to stick with what is traditionally done with algorithmic randomness: the manipulation of Borel sets. We believe that for our purpose, it is a bit more clear to use Borel sets rather than a formal forcing relation.

V.2.1 Borel codes

In order to be able to speak about sets of reals in L_α, we need to code them into elements of L_α. We do that with the notion of ∞-Borel codes and Borel codes. In this chapter, due to technical reasons that will be made clear later, we need to be careful about the L-rank of our Borel codes. In particular, if $\{c_n\}_{n\in\omega}$ are Borel codes for $\Sigma^0_{\alpha+k}$ sets $\mathcal{B}_n$ such that each c_n has L-rank, say β, we need a code of $\bigcap_{n\in\omega}\mathcal{B}_n$ also to have L-rank β. In particular we cannot for instance define a code of $\bigcap_{n\in\omega}\mathcal{B}_n$ to be a set containing the codes c_n.

In what follows the coding trick is achieved with (3) and (4), by coding sequences of sequences of codes to be a partial function defined in $F \subseteq \omega$, using the usual bijection between ω and ω^2. This way the L-rank of a sequence of code stay at the same level.

Definition V.15 (∞-Borel codes and Borel codes)

We define, by induction, ∞-Borel codes together with their rank r, type $t = \Sigma_r$ or Π_r and interpretation ι:

1. The set $c = \langle 2, \{\sigma_i\}_{i<k}\rangle$, for any finite sequence $\{\sigma_i\}_{i<k}$ with each $\sigma_i \in 2^{<\omega}$, is an ∞-Borel code, with rank $r(c) = 0$, type $\Sigma_0 = \Pi_0 = \Delta_0$ and interpretation $\iota(c) = \bigcup_{i<k}[\sigma_i]$
2. Suppose that for some set I, there exists a function $f : i \in I \mapsto c_i$ such that c_i is an ∞-Borel code for every $i \in I$. Then $d_0 = \langle 0, f\rangle$ and $d_1 = \langle 1, f\rangle$ are ∞-Borel codes, with rank $r(d_0) = r(d_1) = \sup_{i\in I}(r(c_i) + 1)$, type respectively Σ^0_r and Π^0_r and interpretation $\iota(d_0) = \bigcup_{i\in I}\iota(c_i)$ and $\iota(d_1) = \bigcap_{i\in I}\iota(c_i)$.
3. Suppose for some set I, there is $k \in \omega$ and a function $i \in I \mapsto c_i$ where for every $i \in I$, the set $c_i = \langle 0, f_i : I^k \to L\rangle$ is an ∞-Borel code. Then we define $f : I^{k+1} \to L$ by $f(i, a_1 \ldots, a_k) = f_i(a_1 \ldots, a_k)$. The set $c = \langle 1, f : I^{k+1} \to L\rangle$ is an ∞-Borel code, with rank $r(c) = \sup_{i\in I}(r(c_i) + 1)$, type Π_r and interpretation $\iota(c) = \bigcap_{i\in I}\iota(c_i)$.
4. Suppose for some set I, there is $k \in \omega$ and a function $i \in I \mapsto c_i$ where for every $i \in I$, the set $c_i = \langle 1, f_i : I^k \to L\rangle$ is an ∞-Borel code. Then we define $f : I^{k+1} \to L$

by $f(i, a_1 \dots, a_k) = f_i(a_1 \dots, a_k)$. The set $c = \langle 0, f : I^{k+1} \to L\rangle$ is an ∞-Borel code, with rank $r(c) = \sup_{i\in I}(r(c_i) + 1)$, type Σ_r and interpretation $\iota(c) = \bigcup_{i\in I} \iota(c_i)$.

A Borel code is an ∞-Borel code where each set I involved equals ω. Note that a Borel code can be encoded by a real.

In order to lighten the notations, we will write $b = \bigvee_{i\in I} b_i$ if b is the ∞-Borel code of $\bigcup_{i\in I} \iota(b_i)$ and $b = \bigwedge_{i\in I} b_i$ if b is the ∞-Borel code of $\bigcap_{i\in I} \iota(b_i)$. Note that given a Borel code $b = \bigvee_{i\in I} b_i$ or $b = \bigwedge_{i\in I} b_i$, one can uniformly find I (using the domain of the function involved in the code), and find the code b_i uniformly in $i \in I$:

▶ **Proposition V.16.** *We have:*

1. *The function which on an ∞-Borel code $b = \bigvee_{i\in I} b_i$ and some $i \in I$, associates the ∞-Borel code b_i, is $\Delta_1^{L_\alpha}$-definable uniformly in α limit. The same holds for $b = \bigwedge_{i\in I} b_i$.*
2. *The function which on an ∞-Borel code $b \in L_\gamma$ associates the ∞-Borel code d of $2^\omega - \iota(b)$ with $d \in L_\gamma$ and $r(b) = r(d)$, is $\Delta_1^{L_\alpha}$-definable uniformly in α limit.*
3. *The function which on ∞-Borel codes $b_0, \dots, b_k \in L_\gamma$ associates the ∞-Borel code $d \in L_\gamma$ with $r(d) = \max_{i\le k}(r(b_i))$ and $\iota(d) = \bigcup_{i<k} \iota(b_i)$, is $\Delta_1^{L_\alpha}$-definable uniformly in α limit.*

Proof. (1) is rather obvious: A code $b = \bigvee_{i\in I} b_i$ is of the form $\langle 0, f : I^{k+1} \to L\rangle$ for some $k \ge 0$. If $k = 0$ then b_i is given by $f(i)$. If $k > 0$ then b_i is given by $\langle 1, f_i : I^k \to L\rangle$ where f_i is defined by $f_i(a_1 \dots, a_k) = f(i, a_1 \dots, a_k)$. This is easily uniformly definable in L_α for any α limit. The same holds for $b = \bigwedge_{i\in I} b_i$.

(2) goes by propagating the complement into the ∞-Borel code, and (3) by propagating the finite union in the ∞-Borel code. Both (2) and (3) are straightforward by induction on γ, using bounded rank replacement of Proposition II.28. ■

▶ **Proposition V.17.** *The set of ∞-Borel codes and of Borel codes of L_α, are $\Delta_1^{L_\alpha}$-definable uniformly in any α limit.*

Proof. We define by Δ_0-induction on the rank of sets of L_α, a total function $f : L_\alpha \to \{0, 1\}$. The function returns 1 iff its parameter is a Borel code. It is defined as follow:

$$\begin{aligned} f(c) &= 1 \text{ if } c \text{ is of the form } \langle 2, \{\sigma_i\}_{i<k}\rangle \text{ for a sequence of strings } \{\sigma_i\}_{i<k} \\ &= 1 \text{ if } c \text{ is of the form } \bigvee_{i\in\omega} c_i \text{ or } \bigwedge_{i\in\omega} c_i \\ &\quad\ \text{and if for every } i \in \omega \text{ we have that } f(c_i) = 1 \\ &= 0 \text{ otherwise} \end{aligned}$$

Note that we are in the conditions of Proposition II.28, with sets L_β in place of sets E_β. One easily see that (1) (2) and (3) of Proposition II.28 are verified, which implies that f is well-defined in L_α for α limit, using bounded rank replacement.

The proof is similar for ∞-Borel codes. ■

V.2.2 The naming system

We use the naming system presented by Cohen in [26]: a name for a set $a \in L_\alpha(x)$ is given by the successive construction steps that lead to the construction of a, starting from an oracle x that we do not know.

We define P_0 as the set of names for elements of $L_0(x)$, that is, for $\{x\} \cup \omega$. The integer 0 is a name for x and the integer $n+1$ is a name for $n \in \omega$.

Suppose now by induction that for an ordinal α, the set of names P_α for elements of $L_\alpha(x)$ has been defined. We define the set of names $P_{\alpha+1}$ for elements of $L_{\alpha+1}(x)$. Let $b \in L_{\alpha+1}(x)$ be such that $b = \{a \in L_\alpha(x) \ : \ L_\alpha(x) \models \Phi(a, p_1, \ldots, p_n)\}$, for $p_1, \ldots, p_n \in L_\alpha(x)$. A name for b is given by the following $\dot{b} = \langle P_\alpha, \ulcorner \Phi \urcorner, \dot{p_1}, \ldots, \dot{p_n} \rangle$, where $\dot{p}_1, \ldots, \dot{p}_n \in P_\alpha$ are names for $p_1, \ldots, p_n$.

Suppose now that the set of names P_β have been defined for $\beta < \alpha$. Then we define $P_\alpha = \bigcup_{\beta<\alpha} P_\beta$.

In general if $a \in L_\alpha(x)$, its corresponding name is written $\dot{a}$. Note that the naming system allows us to speak about elements of $L_\alpha(x)$ without any requirement on x.

▶ **Proposition V.18.** *The function $\beta \mapsto P_\beta$ is $\Delta_1^{L_\alpha}$-definable uniformly in α is limit.*

Proof. We only sketch the proof here. It is straightforward by Δ_0-induction on ordinals, using bounded rank replacement of Proposition II.28, where $E_{<\beta}$ is simply β. One should show that for any β, the set P_β belongs to $L_{\beta+k}$ for some $k \in \omega$. This ensures (3) of Proposition II.28, whereas (1) and (2) are obvious. ■

We shall now argue that given a name $p \in P_\alpha$ and given $x \in 2^\omega$, we can, uniformly in p and x, define the set of $L_\alpha(x)$ that is coded by the name. Such a set will be denoted by $p[x]$, and is defined by induction on the rank of p as follows:

— If $p = 0$ then $p[x] = x$. If $p = n \in \omega - \{0\}$ then $p[x] = n - 1$.

— Suppose $p[x]$ has been defined for every name $p \in P_\alpha$. We define

$$P_\alpha[x] = \{p[x] \ : \ p \in P_\alpha\}$$

Note that $P_\alpha[x]$ is intended to equal $L_\alpha(x)$. Let $p = \langle P_\alpha, \ulcorner \Phi \urcorner, \dot{p_1}, \ldots, \dot{p_n} \rangle$ be a name of $P_{\alpha+1}$. Then $p[x]$ is defined as:

$$p[x] = \{q[x] \ : \ q \in P_\alpha \ \text{ s.t. } P_\alpha[x] \models \Phi(q[x], \dot{p_1}[x], \ldots, \dot{p_n}[x])\}$$

It is clear by induction that for any ordinal α, for any $x \in 2^\omega$ and any $p \in L_\alpha(x)$, we have $\dot{p}[x] = p$.

Note that with the definition we gave, we do not have $P_\alpha \subseteq P_{\alpha+1}$. However for $\beta < \alpha$ and $p \in P_\beta$, one can uniformly obtain a name $q \in P_\alpha$ such that $p[x] = q[x]$.

▶ **Proposition V.19.** *The function which to ordinals $\gamma < \beta$ and names $\dot{p_\gamma} \in P_\gamma$ of elements $p \in L_\gamma(x)$, associates names $\dot{p_\beta} \in P_\beta$ for the same element p, is $\Delta_1^{L_\alpha}$ uniformly in α limit.*

Proof. This is again Δ_0-induction on ordinals, using bounded rank replacement of Proposition II.28. If $\beta = 1$, given $\dot{p_0} \in P_0$, a name for some $p \in L_0(x)$, we let $\dot{p_\beta} = \langle P_0, \ulcorner a \in z \urcorner, \dot{p_0} \rangle \in P_1$. Note that z is a free variable in the formula, meant to be replaced by $\dot{p_0}$. It is clear that $\dot{p_\beta} \in L_k$ for some $k < \omega$ and that $\dot{p_\beta}$ is also a name for p.

Let β and let f be the function of the theorem defined on any $\gamma' < \beta' \leq \beta$ and names of $P_{\gamma'}$. Let us show that we can extend f on any $\gamma < \beta + 1$ and any name of P_γ. Let $\gamma < \beta + 1$ and $\dot{p_\gamma} \in P_\gamma$ be a name for some $p \in L_\gamma(x)$. If $\gamma < \beta$, using f we can find $\dot{p_\beta} \in P_\beta$, a name for p. Thus we can work as in the case $\gamma = \beta$ and consider that we always have a name $\dot{p_\beta} \in P_\beta$. In particular we have that $\dot{p_\beta}$ equals $\langle P_\gamma, \ulcorner \Phi(a, \overline{z_i}) \urcorner, \overline{p_i} \rangle$ for some $\ulcorner \Phi(a, \overline{z_i}) \urcorner$, some $\overline{p_i} \in P_\gamma$, and some $\gamma < \beta$ (with $\gamma = \beta - 1$ if β is successor).

Using f one can find names $\overline{q_i} \in P_\beta$ corresponding to the names $\overline{p_i} \in P_\gamma$. Note that a

name for L_γ is given by $\langle P_\gamma, \ulcorner a = a \urcorner\rangle$. Let $\Psi(a, r, \overline{p_i})$ be the conjunction of the formula $a \in r$, together with the formula $\Phi(a, \overline{p_i})$ where every instance of $\exists x$ (resp. $\forall x$) is replaced by $\exists x \in r$ (resp. $\forall x \in r$). The name $\dot{p_{\beta+1}} \in P_{\beta+1}$ is then given by:

$$\langle P_\beta, \ulcorner \Psi(a, r, \overline{q_i})\urcorner, \langle P_\gamma, \ulcorner a = a \urcorner\rangle, \overline{q_i}\rangle$$

If is clear that $\dot{p_{\beta+1}} \in L_{\beta+k}$ for some k. Therefore we are in the conditions of Proposition II.28 and the function of the proposition is $\Delta_1^{L_\alpha}$ uniformly in α limit.

Also for the limit case the induction is clear as for β limit we have $P_\beta = \bigcup_{\gamma<\beta} P_\gamma$. ■

V.2.3 The canonical Borel sets

We develop here the notations and the main theorem to deal with the canonical sets with ∞-Borel codes, that will be used in this chapter. Let β be an ordinal. Let $p_1, \ldots, p_n \in P_\beta$. Let $\Phi(p_1, \ldots, p_n)$ be a formula. Then we write:

$$B_\Phi^\beta(p_1, \ldots, p_n) \text{ for the set } \{x \in 2^\omega \ : \ L_\beta(x) \models \Phi(p_1[x], \ldots, p_n[x])\}$$

The upcoming theorem makes sure that $B_\Phi^\beta(p_1, \ldots, p_n)$ truly has an ∞-Borel code, definable uniformly in $\ulcorner\Phi\urcorner$, β and $p_1, \ldots, p_n$.

We will sometimes write B_Φ or B_Φ^β when the ordinal β and/or parameters $p_1, \ldots, p_n$ are not specified. Also given an ∞-Borel set B_Φ for a Σ_n formula Φ, we say that B_Φ^β is a Σ_n^β ∞-Borel set. Note that a fixed formula Φ gives rise to many possible ∞-Borel sets depending on the model L_β that we consider.

The second part of the following theorem says that for α limit, if Φ is Δ_0, then an ∞-Borel code for $B_\Phi^\alpha(p_1, \ldots, p_n)$ belongs to L_α, and can be found uniformly. It follows that one can picture a Σ_n^α Borel set with similar intuitions one has with the usual Σ_n Borel sets used in the realm of computable objects and algorithmic randomness : The Σ_1^α Borel sets can be seen as increasing uniform unions of Δ_0^α Borel sets over the names of elements of L_α. Note that if α is limit we have $P_\alpha \subseteq L_\alpha$ and:

$$\begin{aligned} & \{x \in 2^\omega \ : \ L_\alpha(x) \models \exists z \ \Phi(z, p_1[x], \ldots, p_n[x])\} \\ = \ & \textstyle\bigcup_{\dot{z} \in P_\alpha} \{x \in 2^\omega \ : \ L_\alpha(x) \models \Phi(\dot{z}[x], p_1[x], \ldots, p_n[x])\} \end{aligned}$$

Similarly, Σ_2^α Borel sets are unions of intersections of Δ_0^α Borel sets. Indeed we have for α limit that:

$$\begin{aligned} & \{x \in 2^\omega \ : \ L_\alpha(x) \models \exists z_1 \ \forall z_2 \ \Phi(z_1, z_2, p_1[x], \ldots, p_n[x])\} \\ = \ & \textstyle\bigcup_{\dot{z_1} \in P_\alpha} \bigcap_{\dot{z_2} \in P_\alpha} \{x \in 2^\omega \ : \ L_\alpha(x) \models \Phi(\dot{z_1}[x], \dot{z_2}[x], p_1[x], \ldots, p_n[x])\} \end{aligned}$$

One easily sees how to continue for Σ_n^α Borel sets in general.

Theorem V.20

Let α be limit. Then, we have the following:

1. A function which on β, $\ulcorner\Phi(x_1, \ldots, x_n)\urcorner$ and $p_1, \ldots, p_n \in P_\beta$, associates an ∞-Borel code for $B_\Phi^\beta(p_1, \ldots, p_n)$ is $\Delta_1^{L_\alpha}$ uniformly in α.
2. A function which on Δ_0 formulas $\ulcorner\Phi(x_1, \ldots, x_n)\urcorner$ and $p_1, \ldots, p_n \in P_\alpha$, associates an ∞-Borel code for $B_\Phi^\alpha(p_1, \ldots, p_n)$ is $\Delta_1^{L_\alpha}$ uniformly in α.

Moreover, if α is countable and Φ is Δ_0, then the ∞-Borel code for $B_\Phi^\alpha(p_1, \ldots, p_n)$ shows that $B_\Phi^\alpha(p_1, \ldots, p_n)$ is a $\mathbf{\Sigma}_\alpha^0$ set.

Proof. (1) is proved by a Δ_0-induction, using bounded rank replacement of Proposition II.28, with the class of elements of the form $(\beta, \ulcorner\Phi(\overline{x})\urcorner, \overline{p})$ in place of E: an ordinal β, a formula with n free variables, and n parameters of P_β. The induction is done only on the ordinal β. For a set F of formulas (for instance the atomic formulas) let $\mathcal{H}_\beta(F)$ be the induction hypothesis:

$$(\mathcal{H}_\beta(F)) \quad \begin{array}{l}\text{The function } f \text{ which on } \beta\text{, formulas } \ulcorner\Phi(\overline{x})\urcorner \in F \text{ and } \overline{p} \in P_\beta \\ \text{associates an } \infty\text{-Borel code for } B^\beta_\Phi(\overline{p})\text{, belongs to } L_{\beta+k} \text{ for some } k\end{array}$$

Let F_0 be the set of atomic formulas and F_∞ be the set of all formulas. We will show $\mathcal{H}_0(F_0)$. Then we will show $\mathcal{H}_\beta(F_0)$ implies $\mathcal{H}_\beta(F_\infty)$, then we will show $\mathcal{H}_\beta(F_\infty)$ implies $\mathcal{H}_{\beta+1}(F_0)$. Finally we will show $\bigwedge_{\gamma<\beta}\mathcal{H}_\gamma(F_\infty) \to \mathcal{H}_\beta(F_0)$, together with (2) of the Theorem.

Let us begin with $\mathcal{H}_0(F_0)$, Let $p_1, p_2 \in P_0$. Consider $B_= = \{x \in 2^\omega \ : \ L_0(x) \models p_1[x] = p_2[x]\}$ and $B_\in = \{x \in 2^\omega \ : \ L_0(x) \models p_1[x] \in p_2[x]\}$. Recall that p_1, p_2 must be integers, with 0 coding for x and $n+1$ coding for n. Therefore we have $B_= = 2^\omega$ if $p_1 = p_2$ and $B_= = \emptyset$ otherwise. We also have $B_\in = 2^\omega$ if $p_1, p_2 > 0$ and $p_1 \in p_2$ or if $p_1 \neq 0$, $p_2 = 0$ and $p_1 - 1 \in x$. Otherwise we have $B_\in = \emptyset$. It is clear that the two possible Borel codes (2^ω or $\emptyset$) belongs to L_k for some $k \in \omega$ and that the computable function which assign the right Borel code depending on the atomic formulas and parameters, also belongs to L_k for some $k \in \omega$ (recall that we start with $L_0 = \omega$).

Now we prove $\mathcal{H}_\beta(F_0) \Rightarrow \mathcal{H}_\beta(F_\infty)$. We proceed in 5 stages, first showing $\mathcal{H}_\beta(F_0) \Rightarrow \mathcal{H}_\beta(F_1)$, for F_1 the set of atomic formulas and their negations, then showing $\mathcal{H}_\beta(F_1) \Rightarrow \mathcal{H}_\beta(F_2)$, for F_2 the set of finite disjunctions of formulas of F_1, then showing $\mathcal{H}_\beta(F_2) \Rightarrow \mathcal{H}_\beta(F_3)$, for F_3 the set of finite conjunctions of formulas of F_2, then showing $\mathcal{H}_\beta(F_3) \Rightarrow \mathcal{H}_\beta(F_4)$ for F_4 the set of all formulas of F_3 closed by finitely many quantifications, and finally showing $\mathcal{H}_\beta(F_4) \Rightarrow \mathcal{H}_\beta(F_\infty)$.

The step $\mathcal{H}_\beta(F_0)$ implies $\mathcal{H}_\beta(F_1)$ simply follows from (2) of Proposition V.16. The step $\mathcal{H}_\beta(F_1)$ implies $\mathcal{H}_\beta(F_2)$ then follows from (3) of Proposition V.16, whereas the step $\mathcal{H}_\beta(F_2)$ implies $\mathcal{H}_\beta(F_3)$ follows from both (2) and (3) of Proposition V.16. Let us now show the step $\mathcal{H}_\beta(F_3)$ implies $\mathcal{H}_\beta(F_4)$. Let $\overline{p} \in P_\beta$ and let $\Phi(\overline{a}) = \exists a_1\ \forall a_2\ \ldots \Psi(a_1, a_2, \ldots, \overline{a})$ be any formula of F_4 (that is in prenex normal form with its quantifier-free part in disjunctive normal form, in particular with Ψ in F_3). We then have:

$$\begin{array}{rcl} B^\beta_\Phi(\overline{p}) & = & \{x \in 2^\omega \ : \ L_\beta(x) \models \exists x_1\ \forall x_2\ \ldots \Psi(x_1, x_2, \ldots, \overline{p}[x])\} \\ & = & \bigcup_{q_1 \in P_\beta} \bigcap_{q_2 \in P_\beta} \ldots \{x \in 2^\omega \ : \ L_\beta(x) \models \Psi(q_1[x], q_2[x], \ldots, \overline{p}[x])\} \end{array}$$

Using (3) and (4) of Definition V.15, and assuming we have the function given by $\mathcal{H}_\beta(F_3)$, it is easy to build the Borel code for $B^\beta_\Phi(\overline{p})$ whose rank does not increase with the number of quantification, and furthermore, to uniformly do so. In particular we obtain $\mathcal{H}_\beta(F_4)$. In order to obtain $\mathcal{H}_\beta(F_\infty)$, one simply has to use the computable function which transforms any formula into a formula in prenex normal form with its quantifier-free part in disjunctive normal form.

We continue by assuming $\mathcal{H}_\beta(F_\infty)$ and proving $\mathcal{H}_{\beta+1}(F_0)$. We let $p_1, p_2 \in P_{\beta+1}$ with $p_1 = \langle P_\beta, \ulcorner\Phi_1\urcorner, a_1, \ldots, a_n\rangle$ and $p_2 = \langle P_\beta, \ulcorner\Phi_2\urcorner, b_1, \ldots, b_m\rangle$. For $q \in P_\beta$, let:

$$\begin{array}{rcl} B_{\Phi_1}(q) & = & \{x \in 2^\omega \ : \ L_\beta(x) \models \Phi_1(q[x], a_1[x], \ldots, a_n[x])\} \\ B_{\Phi_2}(q) & = & \{x \in 2^\omega \ : \ L_\beta(x) \models \Phi_2(q[x], b_1[x], \ldots, b_m[x])\} \end{array}$$

Note that $q[x] \in p_1[x]$ iff $x \in B_{\Phi_1}(q)$ and $q[x] \in p_2[x]$ iff $x \in B_{\Phi_2}(q)$. Also by induction hypothesis, the function which on $q \in P_\beta$ and on any formula Ψ associates the code of $B_\Psi(q)$

belongs to $L_{\beta+k}$ for some $k \in \omega$. We have:

$$L_{\beta+1}(x) \models p_1[x] \in p_2[x] \quad \text{iff} \quad \exists q \in P_\beta,\ x \in B_{\Phi_2}(q) \wedge L_{\beta+1}(x) \models p_1[x] = q[x]$$

$$L_{\beta+1}(x) \models p_1[x] = p_2[x] \quad \text{iff} \quad \forall q \in P_\beta \text{ we have } x \in B_{\Phi_1}(q) \leftrightarrow x \in B_{\Phi_2}(q)$$

Thus we have:

$$\begin{array}{rcl} B_{=}^{\beta+1}(p_1,p_2) & = & \{x \in 2^\omega \ : \ L_{\beta+1}(x) \models p_1[x] = p_2[x]\} \\ & = & \bigcap_{q \in P_\beta} [B_{\Phi_1}(q) \cap B_{\Phi_2}(q)] \cup [(2^\omega - B_{\Phi_1}(q)) \cap (2^\omega - B_{\Phi_2}(q))] \end{array}$$

It is clear that a code for $B_{=}^{\beta+1}(p_1,p_2)$ can be obtained uniformly and belongs to $L_{\beta+1+k}$ for some k which is independent from p_1, p_2. It follows that we have $\mathcal{H}_{\beta+1}(F_0)$ for equality. Also the set

$$\begin{array}{rcl} B_{\in}^{\beta+1}(p_1,p_2) & = & \{x \in 2^\omega \ : \ L_{\beta+1}(x) \models p_1[x] \in p_2[x]\} \\ & = & \bigcup_{q \in P_\beta} [B_{\Phi_2}(q) \cap \{x \in 2^\omega \ : \ L_{\beta+1}(x) \models p_1[x] = q[x]\}] \end{array}$$

Using Proposition V.19 one can uniformly transform $q \in P_\beta$ into a name that belongs to $P_{\beta+1}$ and thus perform the induction given by the $=$ case just above. We thus have $\mathcal{H}_{\beta+1}(F_\infty)$.

We now deal with the limit case, together with (2) of the theorem. We shall show $\bigwedge_{\gamma<\beta} \mathcal{H}_\gamma(F_\infty) \to \mathcal{H}_\beta(F_0)$. We will actually show more in order to also show (2): We show $\bigwedge_{\gamma<\beta} \mathcal{H}_\gamma(F_\infty) \to \mathcal{H}_\beta(F_{\Delta_0})$ where F_{Δ_0} is the set of Δ_0 formulas.

For a Δ_0 formula $\Phi(p_1,\dots,p_n)$, let γ be the smallest such that $p_1,\dots,p_n \in P_\gamma$. Note that γ is $\Delta_1^{L_\beta}$-definable uniformly in $p_1,\dots,p_n$. We have that the Borel $B_\Phi^\beta(p_1,\dots,p_n)$ also equals the Borel $B_\Phi^\gamma(p_1,\dots,p_n) = \{x \in 2^\omega \ : \ L_\gamma(x) \models \Phi(p_1[x],\dots,p_n[x])\}$. By induction hypothesis the function which on Φ and $p_1,\dots,p_n \in P_\gamma$ gives the Borel code of $B_\Phi^\gamma(p_1,\dots,p_n)$ belongs to $L_{\gamma+k}$ for some k. As this function can be recognized with a Δ_0 formula uniformly in γ, this gives us (2). Note that the union of all these functions belongs to $L_{\beta+k}$ for some k, which gives us $\mathcal{H}_\beta(F_0)$. This concludes the proof. ■

Note that for the study of ITTMs, we can always assume that we work with Borel codes and not ∞-Borel codes. Indeed by Corollary II.62.1, for every $\alpha \leq \Sigma$ limit the sets L_α is a model of "everything is countable". We can then uniformly transform any ∞-Borel codes into a Borel code, working in L_α for α limit, by searching inductively for the smallest (in the sense of $<_L$) bijection between elements of a Borel code, and ω.

V.3 Randomness within ITTMs

We now have all the framework to be able to speak about Borel sets in the hierarchy of constructibles, and will therefore be able to define the different randomness notions. We start with a lemma extending computable measure theory to levels of the constructible hierarchy. Recall that μ denotes the Lebesgue measure on 2^ω.

▶ **Lemma V.21.**

1. *The function $b \mapsto \mu(\iota(b))$, defined on ∞-Borel codes b, is $\Delta_1^{L_\alpha}$ uniformly in any α limit.*
2. *We have the following, where b range over Borel codes and q over rationals:*
   — *The function $b, q \mapsto u$ such that u is the Borel code of an open set with $\iota(b) \subseteq \iota(u)$ and $\mu(\iota(u) - \iota(b)) \leq q$*

— *The function $b, q \mapsto c$ such that c is the Borel code of a closed set with $\iota(c) \subseteq \iota(b)$ and $\mu(\iota(b) - \iota(c)) \leq q$*

are $\Delta_1^{L_\alpha}$ definable uniformly in any α limit.

Proof.

Both (1) and (2) are proved by a Δ_0-induction on ranks of Borel sets (their rank as elements of L). This uses the bounded rank replacement of Proposition II.28.

Proof of (1). For a Borel code b of rank 0, the measure is easily computable, as the measure of a clopen set. Let now $b = \bigvee_{n \in I} b_n$ and γ the smallest such that $b \in L_{\gamma+1}$. Note that each b_i belongs to L_γ. Let $P_f(I)$ be the set of finite subsets of I. We have:

$$\mu(\iota(b)) = \sup_{F \in P_f(I)} \left(\lambda \left(\bigcup_{b_i \in F} \iota(b_i) \right) \right) \tag{V.1}$$

Using (3) of Proposition V.16 we can obtain an ∞-Borel code d_F such that $\iota(d_F) = \bigcup_{b_i \in F} \iota(b_i)$ and such that $d_F \in L_\gamma$. It is also clear that the function which to I associates $P_f(I)$ is $\Delta_1^{L_\alpha}$-definable uniformly in α limit. The function can then be defined by the Δ_0-induction with bounded rank replacement of Proposition II.28.

To compute $\mu(\bigwedge_{n \in \omega} b_n)$, we can use (2) of Proposition V.16 to take the complement to 1 of the measure of $2^\omega - \iota(c)$.

Proof of (2). The function is also defined by Δ_0-induction over γ, using bounded rank replacement of Proposition II.28. In this first point, we could use ∞-Borel codes (and not Borel codes) but still compute the measure by considering all finite unions of codes of smaller complexity. In the second point, we do need to use Borel codes in order to associate a quantity 2^{-n} to each component of a Borel set.

For a Borel code b of rank 0, both the open and the clopen sets are given by b itself. Let now $b = \bigvee_{n \in \omega} b_n$ with γ^+ the smallest such that $b \in L_{\gamma^+}$. Note that each b_n belongs to L_γ. By induction, for each b_n we find codes u_n and c_n of respectively open and closed sets, such that $\mu(\iota(u_n) - \iota(b_n)) < 2^{-n} q$ and $\mu(\iota(b_n) - \iota(c_n)) \leq 2^{-n} q$. The code for the desired open set is then $\bigvee_{n \in \omega} u_n$. For the closed set, note that we have $\mu(\iota(\bigvee_n b_n) - \iota(\bigvee_n c_n)) \leq q$. It follows that there must be some m such that $\mu(\iota(\bigvee_n b_n) - \iota(\bigvee_{n<m} c_n)) \leq q$. The code for the closed set is then given by a code d equivalent to $\bigvee_{n<m} \iota(c_n)$, where we propagate the finite union using Proposition V.16. ■

V.3.1 Main definitions

We give the first definition, which in full generality extends algorithmic randomness to every level of the constructible hierarchy.

Definition V.22

Let α be a countable ordinal. A set x is *random over L_α* if x is in no null set with a Borel code in L_α.

This can be seen as an extension, to any level of the constructible hierarchy, of Δ_1^1-randomness, which corresponds to randomness over $L_{\omega_1^{ck}}$ in the above definition.

The most famous and studied randomness notion is undoubtedly Martin-Löf randomness

[53], whose counterpart for $L_{\omega_1^{ck}}$ was defined in [45]. We also extend the definition of Martin-Löf randomness to any level of the constructible hierarchy:

Definition V.23 (Carl, Schlicht [19])

An α-recursively enumerable open set $\mathcal{U}_n$ is an open set with a code Σ_1-definable in L_α (with parameters). A set x is *α-ML-random* if x is in no intersection $\bigcap_n \mathcal{U}_n$ where each set $\mathcal{U}_n$ is an α-recursively enumerable open set, uniformly in n, such that $\mu(\mathcal{U}_n) \leq 2^{-n}$.

We now turn to randomness notions which are specific to ITTMs. In order to do so, we first need the following definition:

Definition V.24 (Hamkins, Lewis [41])

A set $P \subseteq 2^\omega$ or $P \subseteq \omega$ is *ITTM-semi-decidable* if there is an ITTM M such that $x \in P \Leftrightarrow M(x)\downarrow$. A set $P \subseteq 2^\omega$ or $P \subseteq \omega$ is *ITTM-decidable* if it is both semi-decidable and co-semi-decidable, equivalently, there is an ITTM M such that $M(x)\downarrow= 1 \leftrightarrow x \in P$ and $M(x)\downarrow= 0 \leftrightarrow x \in P$.

If $x \subseteq \omega$, it is clear by admissibility of L_λ that x is ITTM-decidable iff $x \in L_\lambda$, and that x is ITTM-semi-decidable iff x is Σ_1-definable over L_λ.

Definition V.25 (Carl, Schlicht [19])

An ITTM-semi-decidable open set is an open set $\mathcal{U}$ with an ITTM-semi-decidable description $W \subseteq 2^{<\omega}$ such that we have $\bigcup_{\sigma \in W}[\sigma] = \mathcal{U}$. A set X is *ITTM-ML-random* if X is in no intersection $\bigcap_n \mathcal{U}_n$ where each set $\mathcal{U}_n$ is an ITTM-semi-decidable open set, uniformly in n, such that $\mu(\mathcal{U}_n) \leq 2^{-n}$.

Before we continue, we would like to make a small digression about the definition of ITTM-semi-decidable open sets. In the case of Turing machines, given an open set $\mathcal{U}$, it is equivalent to have a recursively enumerable set $W \subseteq 2^{<\omega}$ such that $\mathcal{U} = \bigcup_{\sigma \in W}[\sigma]$ and to have a functional Φ such that $\Phi(X)\downarrow \leftrightarrow X \in \mathcal{U}$.

In the case of computability over $L_{\omega_1^{ck}}$, the same holds: given an open set $\mathcal{U}$, it is equivalent to have a Π^1_1 set $W \subseteq 2^{<\omega}$ such that $\mathcal{U} = \bigcup_{\sigma \in W}[\sigma]$ and for the open set $\mathcal{U}$ to be Π^1_1 as a set of reals.

The corresponding fact with ITTMs does not hold:

▶ **Proposition V.26.** *Every ITTM semi-decidable open set $\mathcal{U} \subseteq 2^\omega$ is also ITTM semi-decidable as a set of reals, but there is an open set that is ITTM-decidable as a set of reals and which is not ITTM-semi-decidable as a set of strings.*

Proof. Suppose $\mathcal{U}$ has an ITTM semi-decidable code $W \subseteq 2^{<\omega}$. We can design another ITTM which on input x looks for some n such that $x_{\restriction n} \in W$. Whenever it find such an n it halts. It is clear that this other ITTM semi-decide $\mathcal{U}$ as a set of reals.

Let us now exhibit an open set $\mathcal{U}$ that is ITTM-decidable as a set of reals, but does not have an ITTM-semi-decidable code. Let c be given by the "Lost melody lemma" [41], that is $\{c\}$ is ITTM-decidable but c is not writable. Then, $A = 2^\omega - \{c\}$ is ITTM-decidable. However, no ITTM-semi-decidable set $W \subseteq 2^{<\omega}$ can be such that $\mathcal{U} = \bigcup_{\sigma \in W}[\sigma]$, as otherwise c would be writable by the following algorithm: if we know that $\sigma \prec c$, we compute a longer prefix of c by waiting for W to cover either $\sigma^\frown i$ (for $i = 0$ or 1), which will happen by compactness, and

then extend our prefix to $\sigma^\frown(1-i) \prec c$. ■

However, the two notions of ITTM-ML-randomness, where the test are ITTM-semidecidable as a description of an open set, or as a set, still the same.

Theorem V.27

A set X is ITTM-ML-random if and only if for every uniformly ITTM-semi-decidable sets $(W_n)_{n\in\mathbb{N}}$ with $\lambda([W_n]) \leq 2^{-n}$ we have $X \notin \bigcap_n [W_n]$.

Proof. The forward implication is clear, as the sets $[W_n]$ are open and uniformly semi-decidable. Now, let U_n be uniformly semi-decidable and open with $\lambda(U_n) \leq 2^{-n}$. As U_n is uniformly λ-recursive let ϕ be such that $x \in U_n$ if and only if $L_\lambda \models \exists a\ \phi(a,n,x)$. Fix p the projectum of λ into ω.

For every $\alpha < \lambda$, we define V_n^α such that $[V_n^\alpha] \supseteq \{x : L_\alpha[x] \models \exists a, \phi(a,n,x)\} = A_n^\alpha$ and $\lambda([V_n^\alpha]) \leq \lambda(A_n^\alpha) + 2^{-p(\alpha)}$. This is possible, by Lemma V.21 and the fact that A_n^α is has an ∞-Borel code.

Define the semidecidable set $V_n = \bigcup_{\alpha<\lambda} V_n^\alpha$. We do have $\lambda([V_n]) \leq \lambda(U_n) + 2^{-n}$, but only $V_n \supseteq U_n \cap \{x : \lambda^x = \lambda\}$. Yet, as U_n is open and most reals do not collapse λ, we can get around by defining:

$$W_n = \{\sigma : \lambda([\sigma]) = \lambda([V_n] \cap [\sigma])\}$$

It is clear that V_n and W_n have the same measure, and W_n is still semidecidable. Now, let us show that $W_n \supseteq U_n$: Let $x \in U_n$. Let $\sigma \prec x$ such that $[\sigma] \subseteq U_n$. Then as $\lambda(\{x : \lambda^x = \lambda\}) = 1$, we must have $\lambda([\sigma]) = \lambda([V_n] \cap [\sigma])$, so $\sigma \in W_n$ and $x \in [W_n]$. ■

We now turn to the most interesting randomness notions defined with ITTM.

Definition V.28 (Carl, Schlicht [19])

A real x is ITTM-random if it is in no semi-decidable null set of reals. A real x is ITTM-decidable random if it is in no decidable null set of reals.

ITTM-decidable randomness can be seen as a counterpart of Δ^1_1-randomness, and indeed Carl and Schlicht showed that ITTM-decidable randomness coincides with randomness over L_Σ.

The notion of ITTM-randomness is more interesting and is in many regards an equivalent for the notion of Π^1_1-randomness. We try to provide a better understanding of this notion.

V.3.2 ITTM-randomness

It is not immediately clear that every ITTM semi-decidable set is measurable. A semi-decidable set has the form $\{x \in 2^\omega : L_{\Sigma^x}(x) \models \Phi\}$ for some Σ_1-formula Φ. Such sets need not to be Borel, but we can separate them into a Borel part and a non-Borel part always included in a Borel set of measure 0. In particular any such set is included in the set $\{x \in 2^\omega : L_\Sigma(x) \models \Phi\} \cup \{x \in 2^\omega : \Sigma^x > \Sigma\}$. The fact that every ITTM semi-decidable set is measurable follows from the fact that the set $\{x \in 2^\omega : \Sigma^x > \Sigma\}$ is included in a Borel set of measure 0. This will be a consequence of Theorem V.31 together with Theorem II.55.

ITTM-randomness is by many aspects the ITTM counterpart of Π^1_1-randomness. For instance, there is a greatest Π^1_1 null set, and Carl and Schlicht showed that there is a greatest ITTM-semi-decidable null set. We also have that x is Π^1_1-random iff x is Δ^1_1-random and $\omega_1^x = \omega_1^{ck}$. Carl

and Schlicht proved the analogous statement for ITTM-randomness:

Theorem V.29 (Carl, Schlicht [19])

A real x is ITTM-random if and only if it is random over L_Σ and $\Sigma^x = \Sigma$.

There are of course differences between ITTM-randomness and Π^1_1-randomness. It is for instance straightforward to build a sequence that is Δ^1_1-random but not Π^1_1-random: to do so one can show that the set of Π^1_1-randoms is included in the set of Π^1_1-ML-randoms, which is included in the set of Δ^1_1-randoms. One can then build a sequence which is Δ^1_1-random but not Π^1_1-ML-random, with a construction similar to the one given in the proof of (1) implies (3) in Theorem V.41. The same thing is not possible with ITTM-randomness. We will see in particular that the Σ-ML-randoms are strictly included in the ITTM-randoms. Also, it is not clear that there are reals x which are randoms over L_Σ and such that $\Sigma^x > \Sigma$. We will for instance show later in Section V.4 that the equivalent notions for genericity collapse: a real x is ITTM-generic iff x is generic over L_Σ iff x is generic over L_Σ and $\Sigma^x = \Sigma$. The question for ITTM-randomness remains open:

▶ **Question V.30.** Does ITTM-randomness coincide with randomness over L_Σ ?

Although we are not able to answer the question here, we still can say meaningful things about ITTM-randomness. In [19] Carl and Schlicht proved the following:

Theorem V.31 (Carl, Schlicht [19])

Suppose that α is countable and admissible or a limit of admissibles ordinals. Then:

1. If $L_\beta \prec_1 L_\alpha$ and z is random over L_α, then $L_\beta(z) \prec_1 L_\alpha(z)$
2. If $L_\beta \prec_n L_\alpha$ and z is random over L_γ where $L_\gamma \models$"α is countable", then $L_\beta(z) \prec_n L_\alpha(z)$ for $n \geq 2$.

In order to understand better Σ-randomness, we introduce a stronger notion that will be enough to obtain (2) in the previous theorem.

Definition V.32

A weak α-ML test is given by a set $\bigcap_{q\in L_\alpha} \mathcal{B}_q$ such that the function which to q associates a Borel code of $\mathcal{B}_q$ is $\Delta^{L_\alpha}_1$ and such that $\mu(\bigcap_{q\in L_\alpha} \mathcal{B}_q) = 0$.
A real x is captured by a weak α-ML test if $x \in \bigcap_{q\in L_\alpha} \mathcal{B}_q$. Otherwise we say that x passes the test. A real x which passes all the weak α-ML tests is weakly α-ML-random.

▶ **Proposition V.33.** *Let α be admissible and $L_\alpha \models$"everything is countable". Then weak α-ML-randomness coincides with randomness over L_α.*

Proof. It is clear that weak α-ML-randomness implies randomness over L_α for any α. Suppose now α admissible and let $\bigcap_{q\in L_\alpha} \mathcal{B}_q$ be a weak α-ML test. Let $f : \omega \to L_\alpha$ be defined with $f(n)$ to be the smallest $r \in L_\alpha$, in the sense of $<_L$, such that $\mu(\bigcap_{q<_L r} \mathcal{B}_q) < 2^{-n}$. By admissibility of α there exists $\beta < \alpha$ such that $\forall n\ f(n) \in L_\beta$. We then have $\mu(\bigcap_{q\in L_\beta} \mathcal{B}_q) = 0$ and $\bigcap_{q\in L_\alpha} \mathcal{B}_q \subseteq \bigcap_{q\in L_\beta} \mathcal{B}_q$. As $\bigcap_{q\in L_\beta} \mathcal{B}_q$ has a Borel code in L_α we have that every element in $\bigcap_{q\in L_\alpha} \mathcal{B}_q$ belongs to a null set of L_α. ■

▶ **Proposition V.34.** *Weak Σ-ML randomness is strictly stronger than randomness over L_Σ.*

Proof. It is clear that weak Σ-ML randomness is stronger than randomness over L_Σ. Let us build $x \in 2^\omega$ that is random over L_Σ but not weakly Σ-ML random.

From Corollary II.63.1, let $b : \omega \to L_\Sigma$ be a bijection which is Σ_1-definable in L_Σ with parameter ζ. One can then simply diagonalize against every measure 1 set with a Borel code in L_Σ. We define σ_0 to be the empty string and $\mathcal{F}_0$ to be 2^ω. Suppose for some n and every $i \leq n$ we have defined a string σ_i and a closed set $\mathcal{F}_i$ uniformly in i such that $\mu(\sigma_i \cap \mathcal{F}_i) > 0$, such that $\sigma_i \preceq \sigma_{i+1}$, such that $\mathcal{F}_{i+1} \subseteq \mathcal{F}_i$ and such that if $b(i)$ is the Borel code of a co-null set, then $\mathcal{F}_i \subseteq \mathcal{B}_i$. Let us define $\mathcal{F}_{n+1}$ and σ_{n+1}. If $b(n+1)$ is the Borel code of a set of measure less than 1, we define $\sigma_{n+1} = \sigma_n$ and $\mathcal{F}_{n+1} = \mathcal{F}_n$. If $b(n+1)$ is the Borel code of a set $\mathcal{B}$ of measure 1, we uniformly find a closed set $\mathcal{F} \subseteq \mathcal{B}$ with a Borel code in L_Σ and with a measure sufficiently close to 1, so that we have $\mu(\sigma_n \cap \mathcal{F}_n \cap \mathcal{F}) > 0$, using Lemma V.21. We define $\mathcal{F}_{n+1} = \mathcal{F}_n \cap \mathcal{F}$. We then define $\sigma_{n+1} = \sigma_n i$ for $i \in \{0, 1\}$ such that $\mu(\sigma_{n+1} \cap \mathcal{F}_{n+1}) > 0$.

Let $x = \sigma_0 \preceq \sigma_1 \preceq \sigma_2 \preceq \dots$. It is clear that x is random over L_Σ. The weak Σ-ML test $\bigcap_{q \in L_\Sigma} \mathcal{B}_q$ is as follow: for $q \in L_\Sigma$, let $n = b^{-1}(q)$. Then define $\mathcal{B}_q = \sigma_n$. It is clear that $x \in \bigcap_{q \in L_\Sigma} \mathcal{B}_q$ and that $\mu(\bigcap_{q \in L_\Sigma} \mathcal{B}_q) = 0$. ■

We then need two lemmas. The first is the same as (1) in [19], but we believe that this proof is a bit simpler.

▶ **Lemma V.35.** *Let $\beta < \alpha$ with α countable and limit and $L_\alpha \models$"everything is countable" and $L_\beta \prec_1 L_\alpha$. Let z be random over α. Then we have $L_\beta(z) \prec_1 L_\alpha(z)$.*

Proof. Let $\Phi(p, q)$ be a Δ_0 formula with $p \in P_\beta$. Suppose z is random over L_α such that $L_\alpha(z) \models \exists q\ \Phi(p[z], q)$. Consider the set $\mathcal{B}^\alpha_\Phi = \{x \ : \ L_\alpha(x) \models \exists q\ \Phi(p[x], q)\}$. We have $\mathcal{B}^\alpha_\Phi = \bigcup_{\dot{q} \in P_\alpha} \mathcal{A}_{\dot{q}}$ where:

$$\mathcal{A}_{\dot{q}} = \{x \ : \ L_\alpha(x) \models \Phi(p[x], \dot{q}[x])\}$$

Let $\mu(\mathcal{B}^\alpha_\Phi) = m$. Note that as z is random over L_α we must have $m > 0$. For every ε with $0 < \varepsilon < m$ we have $L_\alpha \models \exists \dot{r}\ \mu(\bigcup_{\dot{q} <_L \dot{r}} \mathcal{A}_{\dot{q}}) > \varepsilon$. As $L_\beta \prec_1 L_\alpha$ we then have $L_\beta \models \exists \dot{r}\ \mu(\bigcup_{\dot{q} <_L \dot{r}} \mathcal{A}_{\dot{q}}) > \varepsilon$. As this is true for every ε we then have $\mu(\bigcup_{\dot{q} \in P_\beta} \mathcal{A}_{\dot{q}}) = m$.

Suppose for a contradiction that $z \notin \bigcup_{\dot{q} \in P_\beta} \mathcal{A}_{\dot{q}}$. There exists $\dot{r} \in P_\alpha$ such that $z \in \mathcal{A}_{\dot{r}}$. Note that we have $\bigcup_{\dot{q} \in P_\beta} \mathcal{A}_{\dot{q}} \subseteq \bigcup_{\dot{q} \in P_\alpha} \mathcal{A}_{\dot{q}}$ and $\mu(\bigcup_{\dot{q} \in P_\beta} \mathcal{A}_{\dot{q}}) = \mu(\bigcup_{\dot{q} \in P_\alpha} \mathcal{A}_{\dot{q}})$. Therefore we have $\mu(\mathcal{A}_{\dot{r}} - \bigcup_{\dot{q} \in P_\beta} \mathcal{A}_{\dot{q}}) = 0$. It follows that z belongs to a set of measure 0 with a Borel code in L_α, which is a contradiction. Therefore we have $z \in \bigcup_{\dot{q} \in P_\beta} \mathcal{A}_{\dot{q}}$ which implies $L_\beta(z) \models \exists q\ \Phi(p[z], q)$. ■

For the following lemma, we write $=^*, \subseteq^*$ for equality and inclusion, up to a set of measure 0.

▶ **Lemma V.36.** *Let $\beta < \alpha$ with α countable and limit, such that $L_\beta \prec_2 L_\alpha$. Let $\mathcal{B}^\alpha_\Phi = \bigcup_{\dot{q}_1 < P_\alpha} \bigcap_{\dot{q}_2 < P_\alpha} \mathcal{A}_{\dot{q}_1, \dot{q}_2}$ be a Σ^α_2 set with parameters in L_β. Then we have $\mathcal{B}^\alpha_\Phi =^* \mathcal{B}^\beta_\Phi$.*

Proof. By Lemma V.35 and Proposition II.34 we have that if z if random over L_α, then $z \in \bigcup_{\dot{q}_1 < P_\beta} \bigcap_{\dot{q}_2 < P_\beta} \mathcal{A}_{\dot{q}_1, \dot{q}_2}$ implies that $z \in \bigcup_{\dot{q}_1 < P_\alpha} \bigcap_{\dot{q}_2 < P_\alpha} \mathcal{A}_{\dot{q}_1, \dot{q}_2}$. It follows that

$$\bigcup_{\dot{q}_1 < P_\beta} \bigcap_{\dot{q}_2 < P_\beta} \mathcal{A}_{\dot{q}_1, \dot{q}_2} \subseteq^* \bigcup_{\dot{q}_1 < P_\alpha} \bigcap_{\dot{q}_2 < P_\alpha} \mathcal{A}_{\dot{q}_1, \dot{q}_2}$$

In particular if $\mu(\bigcup_{\dot{q}_1 < P_\alpha} \bigcap_{\dot{q}_2 < P_\alpha} \mathcal{A}_{\dot{q}_1, \dot{q}_2}) = 0$ then we are done. Suppose then that we have

$\mu(\bigcup_{\dot{q}_1<P_\alpha}\bigcap_{\dot{q}_2<P_\alpha}\mathcal{A}_{\dot{q}_1,\dot{q}_2}) = m > 0$. For every ε with $0 < \varepsilon < m$ we have:

$$L_\alpha \models \exists\langle\dot{q}_{1,0},\ldots,\dot{q}_{1,k}\rangle\ \forall\dot{r}_2\ \mu\left(\bigcup_{0\le i\le k}\bigcap_{\dot{q}_2<_L\dot{r}_2}\mathcal{A}_{\dot{q}_{1,i},\dot{q}_2}\right) > \varepsilon$$

Using $L_\beta \prec_2 L_\alpha$ we deduce:

$$L_\beta \models \exists\langle\dot{q}_{1,0},\ldots,\dot{q}_{1,k}\rangle\ \forall\dot{r}_2\ \mu\left(\bigcup_{0\le i\le k}\bigcap_{\dot{q}_2<_L\dot{r}_2}\mathcal{A}_{\dot{q}_1,\dot{q}_2}\right) > \varepsilon$$

We deduce that:

$$\mu\left(\bigcup_{\dot{q}_1<P_\beta}\bigcap_{\dot{q}_2<P_\beta}\mathcal{A}_{\dot{q}_1,\dot{q}_2}\right) \ge \varepsilon$$

As this is true for every ε with $0 < \varepsilon < m$, we must have the inequality $\mu(\bigcup_{\dot{q}_1<P_\beta}\bigcap_{\dot{q}_2<P_\beta}\mathcal{A}_{\dot{q}_1,\dot{q}_2}) \ge m$.

Together with the fact that $\bigcup_{\dot{q}_1<P_\beta}\bigcap_{\dot{q}_2<P_\beta}\mathcal{A}_{\dot{q}_1,\dot{q}_2} \subseteq^* \bigcup_{\dot{q}_1<P_\alpha}\bigcap_{\dot{q}_2<P_\alpha}\mathcal{A}_{\dot{q}_1,\dot{q}_2}$, we have the proposition. ■

Theorem V.37

Let $\beta < \alpha$ with α limit, be such that $L_\beta \prec_2 L_\alpha$. Let z be weakly α-ML random. Then $L_\beta(x) \prec_2 L_\alpha(x)$.

Proof. Let $p \in P_\beta$. Let $\Phi(x_1, x_2, x_3)$ be a Δ_0 formula.

Suppose $L_\beta(z) \models \exists a\ \forall b\ \Phi(a, b, p[z])$. In particular there exists $\dot{a} \in P_\beta$ such that $L_\beta(z) \models \forall b\ \Phi(\dot{a}[z], b, p[z])$. From Lemma V.35, $L_\alpha(z) \models \forall b\ \Phi(\dot{a}[z], b, p[z])$. Thus we have $L_\alpha(z) \models \exists a\ \forall b\ \Phi(a, b, p[z])$.

Suppose $L_\alpha(z) \models \exists a\ \forall b\ \Phi(a, b, p)$. Then, let $\mathcal{B}^\alpha_\Phi = \{x \in 2^\omega\ :\ L_\alpha(x) \models \exists a\ \forall b\ \Phi(a, b, p[x])\}$. We have $\mathcal{B}^\alpha_\Phi$ is the Σ^α_2 set given by $\bigcup_{\dot{q}_1<P_\alpha}\bigcap_{\dot{q}_2<P_\alpha}\mathcal{A}_{\dot{q}_1,\dot{q}_2}$, where

$$\mathcal{A}_{\dot{q}_1,\dot{q}_2} = \{x \in 2^\omega\ :\ L_\alpha(x) \models \Phi(\dot{q}_1[x], \dot{q}_2[x], p[x])\}$$

From Lemma V.36 we have that $\mathcal{B}^\alpha_\Phi =^* \mathcal{B}^\beta_\Phi$. Let $\dot{r}$ be such that $z \in \bigcap_{\dot{q}_2\in P_\alpha}\mathcal{A}_{\dot{r},\dot{q}_2}$. Then we have

$$\lambda\left(\bigcap_{\dot{q}_2\in P_\alpha}\left(\mathcal{A}_{\dot{r},\dot{q}_2} - \mathcal{B}^\beta_\Phi\right)\right) = 0$$

It follows that $\bigcap_{\dot{q}_2\in P_\alpha}(\mathcal{A}_{\dot{r},\dot{q}_2} - \mathcal{B}^\beta_\Phi)$ is a weak α-ML test. As z is not weakly α-ML random it does not belong to the test and then it must belong to $\mathcal{B}^\beta_\Phi$. Thus $z \in \bigcup_{\dot{q}_1\in P_\beta}\bigcap_{\dot{q}_2\in P_\beta}\mathcal{A}_{\dot{q}_1,\dot{q}_2}$. It follows that $L_\beta(z) \models \exists a\ \forall b\ \Phi(a, b, p[z])$. ■

▶ **Corollary V.37.1.** *Let $\beta < \alpha$ such that $L_\alpha \models$"everything is countable" and $L_\beta \prec_2 L_\alpha$. Suppose α is admissible. Let z be random over L_α. Then $L_\beta(z) \prec_2 L_\alpha(z)$.*

Proof. If α is admissible we have that weak α-ML-randomness coincides with randomness over L_α by Proposition V.33. Thus if z is random over L_α we must have $L_\beta(z) \prec_2 L_\alpha(z)$.

■

▶ **Corollary V.37.2.** *Let z be weakly Σ-ML random. Then z is ITTM-random.*

Proof. We have $L_\zeta \prec_2 L_\Sigma$. We also have that z is ITTM-random iff z is random over L_Σ and $\Sigma^z = \Sigma$. If z is weakly Σ-ML random we have $L_\zeta(z) \prec_2 L_\Sigma(z)$. In particular (ζ^z, Σ^z) is the lexicographically smallest pair of ordinal such that $L_{\zeta^z}(z) \prec_2 L_{\Sigma^z}(z)$, which implies $\Sigma^z = \Sigma$ and $\zeta^z = \zeta$. Also if z is weakly Σ-ML random, then it is random over L_Σ. It follows that z is ITTM-random. ■

We now give a more combinatorial equivalent characterization the notion of ITTM-randomness: a characterization in terms of being captured by sets of measure 0 having a specific complexity. For the following proposition, by Δ^Σ_3 set, we mean a set which is also Δ^ζ_3, that is, a set $\mathcal{B}^\Sigma_1 = \mathcal{B}^\Sigma_2$ with $\mathcal{B}^\Sigma_1$ which is Σ^Σ_3 and $\mathcal{B}^\Sigma_2$ which is Π^Σ_3, such that for the versions $\mathcal{B}^\zeta_1$ and $\mathcal{B}^\zeta_2$ we also have $\mathcal{B}^\zeta_1 = \mathcal{B}^\zeta_2$.

Theorem V.38

The following are equivalent:

1. z is ITTM-random.
2. z belongs to no Δ^Σ_3 set of measure 0, with parameters in L_ζ.

Proof. Let us show (2) implies (1). Suppose that z is not ITTM-random. If it is not random over L_Σ then clearly (2) is false with the Σ^Σ_1 set of measure 0 which is the union of all the Borel sets of L_Σ of measure 0. Otherwise z is random over L_Σ and there is a parameter $p \in L_\zeta(z)$ with a Δ_0 formula $\Phi(x_1, x_2, x_3)$ such that $L_\Sigma(z) \models \exists a\ \forall b\ \Phi(a, b, p)$ but $L_\zeta(z) \models \forall a\ \exists b\ \neg\Phi(a, b, p)$. Let $\Psi(p) \equiv \exists a\ \forall b\ \Phi(a, b, p)$.

We have $z \in \mathcal{B}^\Sigma_\Psi \cap \mathcal{B}^\zeta_{\neg\Psi}$. Also from Lemma V.36 we have $\mathcal{B}^\Sigma_\Psi =^* \mathcal{B}^\zeta_\Psi$ and thus $\mu(\mathcal{B}^\Sigma_\Psi \cap \mathcal{B}^\zeta_{\neg\Psi}) = 0$. We now have to transform the Π^ζ_2 set $\mathcal{B}^\zeta_{\neg\Psi}$ into a Π^Σ_2 set $\mathcal{B}^\Sigma_\phi$ such that $\mathcal{B}^\Sigma_\phi =^* \mathcal{B}^\zeta_{\neg\Psi}$ and $\mathcal{B}^\Sigma_\phi$ still contains z. Let $\beta < \zeta$ be such that $p \in L_\beta(z)$.

We define $\mathcal{B}^\Sigma_\phi$ to be $\bigcap_{\beta \le \alpha \prec_1 \Sigma} \mathcal{B}^\alpha_{\neg\Psi}$. Formally, the corresponding formula ϕ is given by $\phi(\beta, p) \equiv \forall \alpha \ge \beta\, [L_\alpha$ is not Σ_1 stable or $L_\alpha(x) \models \neg\Psi(p)]$.

Using Proposition II.35 it is clear that $\mathcal{B}^\Sigma_\phi$ is Π^Σ_2. We shall now show that as long as z is random over L_ζ we have $z \in \mathcal{B}^\Sigma_\phi$ iff $z \in \mathcal{B}^\zeta_{\neg\Psi}$. As $L_\zeta \prec_1 L_\Sigma$ we have $\mathcal{B}^\Sigma_\phi \subseteq \mathcal{B}^\zeta_{\neg\Psi}$. Let us show that if z is random over L_ζ and $z \in \mathcal{B}^\zeta_{\neg\Psi}$, then $z \in \mathcal{B}^\Sigma_\phi$.

To do so let us first show that for every α with $\zeta < \alpha < \Sigma$ we have that $\neg L_\alpha \prec_1 L_\Sigma$. Fix such an ordinal α. By Theorem II.58, every accidentally writable ordinal becomes writable with parameter ζ. In particular $\{\alpha\}$ is Σ_1-definable in L_Σ with some Σ_1 formula $\Phi(\zeta, \alpha)$ (intuitively the program that writes α and halts). It follows that $L_\Sigma \models \exists\alpha\ \Phi(\zeta, \alpha)$ but $\neg L_\alpha \models \exists\alpha\ \Phi(\zeta, \alpha)$. Thus we do not have $L_\alpha \prec_1 L_\Sigma$.

Suppose now z is random over L_ζ and $z \in \mathcal{B}^\zeta_{\neg\Psi}$. Let $\alpha \ge \beta$ be such that $L_\alpha \prec_1 L_\Sigma$. Then we must have $\alpha \le \zeta$. Also if $L_\Sigma \models \Phi(p)$ for some Σ_1 formula Φ with parameter $p \in L_\alpha$, we must have $L_\alpha \models \Phi(p)$ and then $L_\zeta \models \Phi(p)$. Therefore $L_\alpha \prec_1 L_\zeta$. Now as z is random over L_ζ and $z \in \mathcal{B}^\zeta_{\neg\Psi}$, we must have by Lemma V.35 and Proposition II.34 that $z \in \mathcal{B}^\alpha_{\neg\Psi}$. It follows that $z \in \bigcap_{\beta \le \alpha \prec_1 \Sigma} \mathcal{B}^\alpha_{\neg\Psi}$.

We then have that $z \in \mathcal{B}^\Sigma_\Psi \cap \mathcal{B}^\Sigma_\phi$, with $\mu(\mathcal{B}^\Sigma_\Psi \cap \mathcal{B}^\Sigma_\phi) = 0$, and with $\mathcal{B}^\Sigma_\Psi \cap \mathcal{B}^\Sigma_\phi$ a Δ^Σ_3 set with parameters in L_ζ. Note that $\mathcal{B}^\zeta_\Psi \cap \mathcal{B}^\zeta_\phi$ is also a Δ^ζ_3 set.

Let us show (1) implies (2). Suppose now that there is a Π^Σ_3 set $\mathcal{B}^\Sigma_\Phi$ and a Σ^Σ_3 set $\mathcal{B}^\Sigma_\Psi$, with

parameters in L_ζ, such that $z \in \mathcal{B}^\Sigma_\Phi = \mathcal{B}^\Sigma_\Psi$ and $\mu(\mathcal{B}^\Sigma_\Phi) = \mu(\mathcal{B}^\Sigma_\Psi) = 0$, with also $\mathcal{B}^\zeta_\Phi = \mathcal{B}^\zeta_\Psi$.

If $z \notin \mathcal{B}^\zeta_\Phi$ then $L_\Sigma(z) \models \Phi$ and $\neg L_\zeta(z) \models \Phi$ for the Π_3-formula Φ. By Proposition II.34 we then have $\neg L_\zeta(x) \prec_2 L_\Sigma(x)$ and thus z is not ITTM-random.

Otherwise $z \in \mathcal{B}^\zeta_\Phi$ and thus $z \in \mathcal{B}^\zeta_\Psi$. We also have that $\mu(\mathcal{B}^\Sigma_\Psi) = 0$. Also $\mathcal{B}^\Sigma_\Psi = \bigcup_{\dot{q}_1 \in P_\Sigma} \bigcap_{\dot{q}_2 \in P_\Sigma} \bigcup_{\dot{q}_3 \in P_\Sigma} \mathcal{A}_{\dot{q}_1,\dot{q}_2,\dot{q}_3}$. For any name $\dot{q}_1 \in P_\zeta$ we have that $\mu(\bigcap_{\dot{q}_2 \in P_\Sigma} \bigcup_{\dot{q}_3 \in P_\Sigma} \mathcal{A}_{\dot{q}_1,\dot{q}_2,\dot{q}_3}) = 0$ and from Lemma V.36 we then must have that $\mu(\bigcap_{\dot{q}_2 \in P_\zeta} \bigcup_{\dot{q}_3 \in P_\zeta} \mathcal{A}_{\dot{q}_1,\dot{q}_2,\dot{q}_3}) = 0$. In particular there is $\dot{q}_1 \in P_\zeta$ such that $z \in \bigcap_{\dot{q}_2 \in P_\zeta} \bigcup_{\dot{q}_3 \in P_\zeta} \mathcal{A}_{\dot{q}_1,\dot{q}_2,\dot{q}_3}$. It follows that z is not random over L_Σ and thus not ITTM-random. ■

So ITTM-randomness is equivalent to Δ^Σ_3-randomness for sets with parameters which are at most eventually writable, but not accidentally writable. We shall now see that it is actually very close to randomness over L_Σ, which can be shown to be equivalent to a similar test notion:

Theorem V.39

The following are equivalent:

1. z is random over L_Σ.
2. z is in no Σ^Σ_2 set of measure 0, with parameters in L_ζ.
3. z is in no Π^Σ_2 set of measure 0, with parameters in L_ζ.

Proof. It is clear that both (2) and (3) imply (1), using the Σ^Σ_1 set of measure 0 which is the union of all the Borel sets of L_Σ of measure 0.

Let us show (1) implies (2). Let $\mathcal{B}^\Sigma_\Phi$ be a Σ^Σ_2 set equal to $\bigcup_{\dot{q}_1 \in P_\alpha} \bigcap_{\dot{q}_2 \in P_\alpha} \mathcal{A}_{\dot{q}_1,\dot{q}_2}$ with $\mu(\mathcal{B}^\Sigma_\Phi) = 0$. The following argument is a combination of the Σ_2-stability of L_ζ in L_Σ, together with the admissibility of L_ζ.

By Lemma V.36 we have $\mu(\mathcal{B}^\zeta_\Phi) = 0$. Then, $\forall \dot{q}_1 \in P_\zeta\ \mu(\bigcap_{\dot{q}_2 \in P_\zeta} \mathcal{A}_{\dot{q}_1,\dot{q}_2}) = 0$. Fix $\dot{q}_1 \in P_\zeta$. By admissibility of ζ, there must exists $\dot{r} \in P_\zeta$ such that $\mu(\bigcap_{\dot{q}_2 <_L \dot{r}} \mathcal{A}_{\dot{q}_1,\dot{q}_2}) = 0$. It follows that $L_\zeta \models \forall \dot{q}_1\ \exists \dot{r}\ \mu(\bigcap_{\dot{q}_2 <_L \dot{r}} \mathcal{A}_{\dot{q}_1,\dot{q}_2}) = 0$. As $L_\zeta \prec_2 L_\Sigma$ we also have $L_\Sigma \models \forall \dot{q}_1\ \exists \dot{r}\ \mu(\bigcap_{\dot{q}_2 <_L \dot{r}} \mathcal{A}_{\dot{q}_1,\dot{q}_2}) = 0$. In particular every real in $\mathcal{B}^\Sigma_\Phi$ is in a set of measure 0 with a Borel code in L_Σ.

Let us show (1) implies (3). Let $\mathcal{B}^\Sigma_\Phi$ be the Π^Σ_2 set of measure 0. By Lemma V.36 we must have $\mathcal{B}^\zeta_\Phi =^* \mathcal{B}^\Sigma_\Phi$. Let $z \in \mathcal{B}^\Sigma_\Phi$. Suppose $z \notin \mathcal{B}^\zeta_\Phi$. Then we have $L_\Sigma(z) \models \Phi$ and $\neg L_\zeta(z) \models \Phi$ for a Π_2 formula Φ with parameters in L_ζ. By Lemma V.35 together with Proposition II.34 we then have that z is not random over L_Σ. Suppose now $z \in \mathcal{B}^\zeta_\Phi$. Then z is in a set of measure 0 with a Borel code in L_Σ which implies that z is not random over L_Σ. ■

V.3.3 Martin-Löf randomness in the constructibles

It was shown in [19] that randomness over L_λ is the counterpart of Δ^1_1-randomness for ITTMs, and λ-ML-randomness the counterpart of Π^1_1-ML-randomness. Carl and Schlicht asked if as in the hyperarithmetic case these two notions really differ. We give a general answer to this question by characterizing the ordinals α for which the two notions are different.

Separation of randomness over L_α and α-ML-randomness

We first give the easy relation between randomness over L_α and α-ML-randomness:

▶ **Proposition V.40.** *Let α be limit. Then α-ML-randomness is stronger than randomness over L_α*

Proof. Let $\mathcal{B}$ be a Borel set with code in L_α. By Lemma V.21, we define an α-ML-test $\bigcap_n \mathcal{U}_n$ such that for all n, we have $\mathcal{B} \subseteq \mathcal{U}_n$, and $\mu(\mathcal{U}_n) \leq \mu(\mathcal{B}) + 2^{-n} = 2^{-n}$. Then $\mathcal{B} \subseteq \bigcap_n \mathcal{U}_n$, this proves the property. ■

The following theorem characterizes exactly when randomness over L_α and α-ML-randomness coincide, for α admissible or α limit and $L_\alpha \models$"everything is countable".

Theorem V.41

Let α be admissible or α limit such that $L_\alpha \models$"everything is countable". The following are equivalent:

1. α is projectible into ω.
2. There is a universal α-ML-test.
3. α-ML-randomness is strictly stronger than randomness over L_α.

Proof. Note first that if α is limit, non-admissible and $L_\alpha \models$"everything is countable", then by Theorem II.63 α is projectible into ω. Therefore for (3) implies (1) and (2) implies (1), we can suppose α admissible.

The proof that (3) implies (1) is done by contraposition and Theorem II.38: if α is not projectible into ω, then L_α satisfies Σ_1-comprehension for subsets of ω and then every α-ML-test is in L_α, which implies that randomness over L_α is stronger than α-ML-randomness. Together with Proposition V.40 we have that the two notions of randomness coincide.

To prove (2) implies (1), suppose we have (2) and α is not projectible into ω, in order to get a contradiction. Then by Theorem II.38, the universal α-ML-test $\bigcap_n \mathcal{U}_n$ would be in some L_β with $\beta < \alpha$. We have that $2^\omega - \mathcal{U}_0$ is a closed set whose leftmost path is definable in L_β and then belongs to $L_{\beta+1}$. As this leftmost path is definable in L_α, it is not random over L_α, which contradicts the universality of the test.

Let us now prove (1) implies (2). Assuming that α is projectible into ω, it is then possible to α-recursively assign an integer to all the parameters in L_α, we will use this to assign an integer to every α-ML-test. We have an enumeration $\{\Phi_m(x,k,p,\sigma)\}_{m\in\omega}$ of every Δ_0 formula with four free variables and without parameters. We see any such formula as defining a uniform intersection of α-recursively enumerable open sets when given a parameter p: for some m the formula Φ_m together with a parameter p defines an intersection of open sets $\bigcap_k \mathcal{U}_k$, each $\mathcal{U}_k$ being the union of all the cylinders $[\sigma]$ such that $L_\alpha \models \exists x\ \Phi_m(x,k,p,\sigma)$.

Let π be a Σ_1-definable injection of L_α into ω. Note that if α is admissible we use the projection together with the bijection between α and L_α. Otherwise we use the bijection given by Theorem II.63. Let p be a parameter and n an integer such that $\pi(p) = n$. If $\bigcap_k \mathcal{U}_k$ is defined in L_α by the Σ_1 formula $\Phi_m(x,k,p,\sigma)$ with parameter p, then $\bigcap_k \mathcal{U}_k$ is also defined by the following parameter-free Σ_1 formula $\Psi_{n,m}(k,\sigma) \equiv \exists p\ \exists x\ \pi(p) = n \wedge \Phi_m(x,k,p,\sigma)$. Consequently, every uniform intersection of α-recursively enumerable open set $\bigcap_k \mathcal{U}_k$ is defined by a formula in the enumeration $\{\Psi_{m,n}(k,\sigma)\}_{\langle m,n\rangle\in\omega}$.

Now for integers m,n, the formula $\Psi_{m,n}(k,\sigma)$ might not define an α-ML-test, due to the measure requirement. For any n,m let $\widetilde{\psi}_{m,n}(z,k,\sigma)$ be a Δ_0 formula such that $L_\alpha \models$

$\exists z\ \widetilde{\psi}_{m,n}(z,k,\sigma)$ iff $L_\alpha \models \Psi_{n,m}(k,\sigma)$. We define the computable function g which to n,m associates the code $g(n,m)$ of the Δ_0 formula $\phi(z,k,\sigma)$

$$\phi(z,k,\sigma) \equiv \widetilde{\psi}_{m,n}(z,k,\sigma) \wedge \lambda\left(\bigcup\{[\tau] : \exists z' \leq_L z\ \widetilde{\psi}_{n,m}(z',k,\tau)\}\right) \leq 2^{-k}$$

The formula $\exists z\ \phi(z,k,\sigma)$ always defines a Martin-Löf test. Furthermore, if $\Psi_{n,m}(k,\sigma)$ defines an α-ML-test, then the formula $\exists z\ \phi(z,k,\sigma)$ defines the same test. It follows that $\{g(n,m)\}_{n,m\in\omega}$ is an enumeration of codes for α-ML-tests that contains all the α-ML-tests. This can then be used to define a universal α-ML test as in the lower settings: given an enumeration $\{\bigcap_k \mathcal{U}^n_k\}_{n\in\omega}$ of all the Martin-Löf tests, we define $\mathcal{V}_m = \bigcup_i \mathcal{U}^i_{i+m+1}$. We clearly have $\bigcup_n \bigcap_k \mathcal{U}^n_k \subseteq \bigcap_m \mathcal{V}_m$, and as $\mu(\mathcal{U}^i_{i+m+1}) \leq 2^{-m-i-1}$ we have $\mu(\mathcal{V}_m) \leq 2^{-m}$ which implies that $\bigcap_m \mathcal{V}_m$ is a Martin-Löf test. Thus (1) implies (2).

To prove (1) implies (3), we will build an α-ML test $\mathcal{U}$ capturing a real x which is random over L_α. Let π be a Σ_1-definable injection of L_α into ω. We proceed by stages where the stages are ordinals $s < \alpha$. The stages will approximate a set x random over L_α in a Δ^0_2 way, together with an α-ML test that capture x. To do so, for every integer n and every stage s, we will define a closed set $\mathcal{F}^n_s$ and a string σ^n_s with $|\sigma^n_s| = 2n$ and $\sigma^n_s \prec \sigma^{n+1}_s$, such that:

$$\lambda\left(\bigcap_{i\leq n} \mathcal{F}^i_s \cap [\sigma^n_s]\right) > 0 \tag{R^s_n}$$

$$\begin{array}{l}\text{If } \pi(a) = n \text{ for } a \in L_s \text{ such that } a \text{ is the code}\\ \text{of a Borel set } \mathcal{B}_a \text{ of measure 1, then } \mathcal{F}^n_s \subseteq \mathcal{B}_a\end{array} \tag{S^s_n}$$

Also the definition of σ^n_s and $\mathcal{F}^n_s$ will be independent from the definition of σ^n_t and $\mathcal{F}^n_t$ for $t <_L s$. At stage s, we define $\mathcal{F}^0_s$ to be 2^ω and σ^0_s be the empty string. It is clear that R^s_0 and S^s_0 are satisfied. Suppose $\mathcal{F}^i_s$ and σ^i_s have been defined for every $i \leq n$ such that R^s_i and S^s_i are satisfied. Let us define $\mathcal{F}^{n+1}_s$ and σ^{n+1}_s. If $\pi(a) = n+1$ for some $a \in L_s$ such that a is the Borel code of a set $\mathcal{B}_a$ of measure 1, then let $\bigcup_m \mathcal{S}_m \subseteq \mathcal{B}_a$ be a conull union of closed sets with a Borel code in L_α. Note that by Lemma V.21 we can obtain such a union uniformly. Let then k be the smallest such that:

$$\lambda\left(\bigcup_{i\leq k} \mathcal{S}_i \cap \bigcap_{i\leq n} \mathcal{F}^i_s \cap [\sigma^n_s]\right) > 0$$

Let $F^{n+1}_s = \bigcup_{i\leq k} \mathcal{S}_i$. Let σ^s_{n+1} be the first extension of σ^s_n by two bits such that R^s_{n+1} is satisfied.

Let x_s be the sequence $\sigma^s_1 \prec \sigma^s_2 \prec \sigma^s_3 \prec \dots$. Note that for each n, the sequences $\{\sigma^n_s\}_{s<\alpha}$ and $\{\mathcal{F}^n_s\}_{s<\alpha}$ change at most once per integer $i \leq n$ such that $\pi(a) = i$ for some Borel set $\mathcal{B}_a$ of measure 1 with $a \in L_s$. Thus these sequences change at most n times. In particular the whole process converges and the sequence x_s converges to some sequence x.

This can also be used to define the α-ML-test that contains x. We define $\mathcal{U}_n = \bigcup_{s<\alpha}[\sigma^n_s]$. This is an α-ML-test as there are at most n distinct versions of σ^n_s and for each of them we have $|\sigma^n_s| = 2n$. The measure of $\mathcal{U}_n$ is then bounded by $n \times 2^{-2n} \leq 2^{-n}$. This shows that x is not α-ML-random. Also by S^s_n we have that x is in every Borel set $\mathcal{B}_s$, so it is random over L_α. We then have (1) implies (3). ■

▶ **Corollary V.41.1.** *We have:*

— *λ-ML-randomness is strictly stronger than randomness over L_λ.*
— *ζ-ML-randomness is equal to randomness over L_ζ.*
— *Σ-ML-randomness is strictly stronger than randomness over L_Σ.*

Proof. By Corollary II.61, we have that λ is projectible over ω with no parameter, and ζ is not projectible into ω. By Corollary II.63.1, we have that Σ is projectible into ω with parameter ζ. ■

We shall now improve Corollary V.41.1 for Σ-ML-randomness, by showing that it is strictly stronger than weak Σ-ML randomness and thus than ITTM-randomness.

Theorem V.42

Σ-ML-randomness is strictly stronger than weak Σ-ML randomness and than ITTM-randomness.

Proof. We shall construct a real z such that for any Σ^Σ_1 set $\bigcup_{p\in L_\Sigma}\mathcal{B}_p$ of measure 1, we have $z \in \bigcup_{p\in L_\Sigma}\mathcal{B}_p$, together with a Σ-ML test $\bigcap_{n\in\omega}\mathcal{U}_n$ containing z, and with $\mu(\mathcal{U}_n) \leq 2^{-n}$. The proof is very similar to (1) implies (3) in Theorem V.41.

Let b be Σ_1-definable bijection of Corollary II.63.1 from ω to L_Σ. Using this bijection, let $\{\bigcup_{p\in L_\Sigma}\mathcal{B}_{n,p}\}_{n\in\omega}$ be an enumeration of all the union of Borel sets of L_Σ.

We will define a computation, stage by stage, of a set z, that will be approximated in a Δ^0_2 way, together with a Σ-ML test that will capture z. To do so, for every integer n and every stage s, we will define a closed set $\mathcal{F}^n_s$ and a string σ^n_s with $|\sigma^n_s| = 2n$ and $\sigma^n_s \prec \sigma^{n+1}_s$, such that for every n, s we have

$$\lambda\left(\bigcap_{i\leq n}\mathcal{F}^i_s \cap [\sigma^n_s]\right) > 0$$

and for every n, if $\mu(\bigcup_{p\in L_\Sigma}\mathcal{B}_{n,p}) = 1$ there exists t such that for all $s \geq t$ we have $\mathcal{F}^n_s \subseteq \bigcup_{p\in L_\Sigma}\mathcal{B}_{n,p}$. Note also that the definition of $\mathcal{F}^n_s$ and σ^n_s will not depend on $\mathcal{F}^m_t$ or σ^m_t for $m \in \omega$ and $t < s$.

At stage s, we define $\mathcal{F}^0_s$ to be 2^ω and σ^0_s to be the empty string. Suppose $\mathcal{F}^i_s$ and σ^i_s have been defined for every $i \leq n$. Let us define $\mathcal{F}^{n+1}_s$ and σ^{n+1}_s:

Suppose $\mu(\bigcup_{p\in L_s}\mathcal{B}_{n+1,p} \cap \bigcap_{i\leq n}\mathcal{F}^i_s \cap (\sigma^n_s)) > 0$. Then let us find some closed set $\mathcal{F}^{n+1}_s \subseteq \bigcup_{p\in L_s}\mathcal{B}_{n+1,p}$ such that $\mu(\bigcap_{i\leq n+1}\mathcal{F}^i_s \cap [\sigma^n_s]) > 0$. Let then σ^s_{n+1} be the first extension of σ^s_n by two bits such that $\mu(\bigcap_{i\leq n+1}\mathcal{F}^i_s \cap [\sigma^{n+1}_s]) > 0$.

Let z_s be the sequence $\sigma^s_1 \prec \sigma^s_2 \prec \sigma^s_3 \prec \dots$. Note that for each n, the sequences $\{\sigma^n_s\}_{s<\Sigma}$ and $\{\mathcal{F}^n_s\}_{s<\Sigma}$ change at most once per integer i smaller than n. Thus these sequences change at most n times. In particular the whole process converges and the sequence z_s converges to some sequence z.

This can then be used to define the α-ML-test that contains z. We define $\mathcal{U}_n = \bigcup_{s<\Sigma}[\sigma^n_s]$. This is a Σ-ML-test as there are at most n distinct versions of σ^n_s and for each of them we have $|\sigma^n_s| = 2n$. The measure of $\mathcal{U}_n$ is then bounded by $n \times 2^{-2n} \leq 2^{-n}$. This shows that z is not Σ-ML-random. It is also clear that z is in every set $\bigcup_{p\in L_\Sigma}\mathcal{B}_p$ such that $\mu(\bigcup_{p\in L_\Sigma}\mathcal{B}_p) = 1$. ■

Summary

The following picture summarizes the relations between all the randomness notions we have seen:

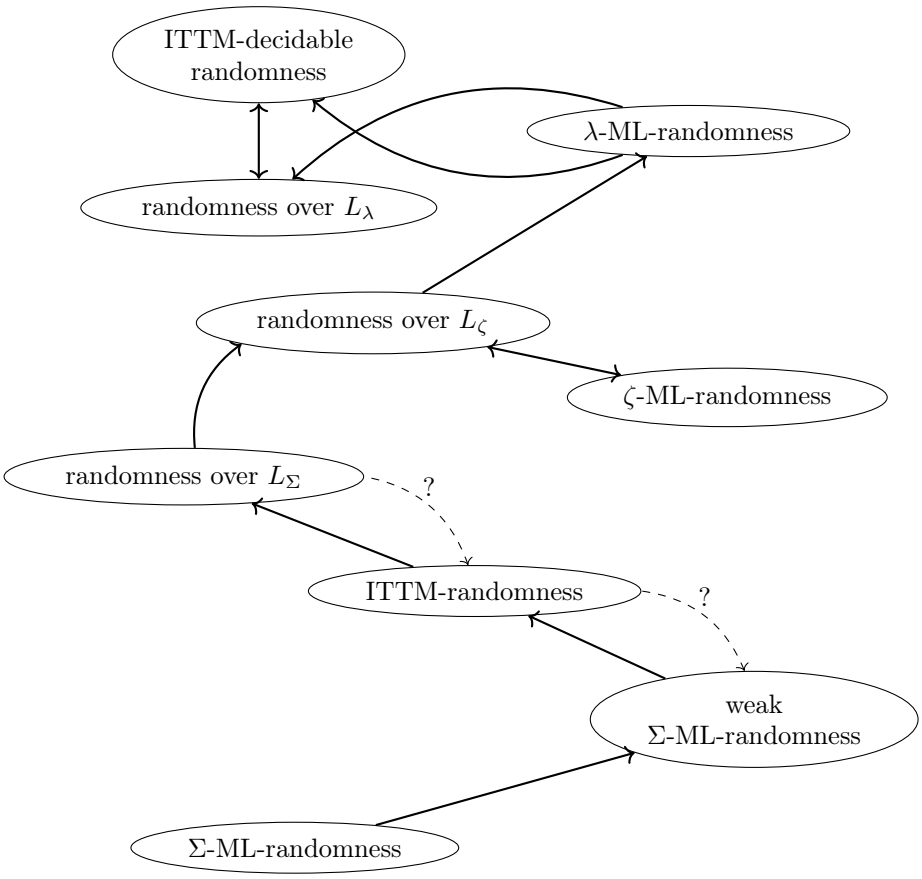


Figure V.2 – Higher randomness

We recall here the two remaining open questions:

▶ **Question V.43.** Is ITTM-randomness strictly stronger than randomness over L_Σ?

▶ **Question V.44.** Is weak Σ-ML randomness strictly stronger than ITTM-randomness?

Note that by Proposition V.34 a negative answer to one of the two questions would provide a positive answer to the other one.

Mutual λ-ML randoms computing common reals

When two sets are mutually random, we expect them to compute no common non-computable sets. However, depending on the randomness level we ask for, this is sometimes not the case. Carl and Schlicht asked in Question 5.5 from [19] if two mutually λ-ML-randoms could compute a common non-writable set. It is the case with Martin-Löf randomness, and sets which can be computed by two mutually Martin-Löf random must be K-trivials. We show that the same

happens with ITTMs: some non-writable sets can be ITTM-computed by two mutually λ-ML-randoms. We do not study here however the notion of K-triviality for ITTMs, even though we conjecture that most of the work done about K-trivials and about higher K-trivials (K-trivials defined over $L_{\omega_1^{ck}}$) lifts to the world of computability inside L_λ, using the fact that λ is projectible into ω.

First, we need to expand, in a straightforward way, some definitions from ML-randomness to the ITTM settings. In the following, we focus on ITTMs but the proofs also work for α such that there exists a universal α-ML-test, in other word by Theorem V.41 when α is projectible in ω and such that either α is admissible or both α is limit and $L_\alpha \models$"everything is countable".

Definition V.45

An ITTM-Solovay test is a sequence of uniformly ITTM-semi-decidable open sets $(\mathcal{S}_s)_{s<\lambda}$ such that $\Sigma_{s<\lambda}\mu(\mathcal{S}_s) < \infty$. We say that $Z \in 2^\omega$ passes the test if Z belongs to only finitely many $\mathcal{S}_s$.

▶ **Proposition V.46.** *Let $z \in 2^\omega$. The following are equivalent:*

1. *z passes every ITTM-Solovay tests.*
2. *z is λ-ML-random.*

The proof of this characterization of λ-ML test via ITTM-Solovay tests is exactly the same as the one from the lower case, that can be found in [29]. Our witness for answering the question will be the even and odd parts of a specific λ-ML-random, an approximable one.

Definition V.47 (Chaitin's Ω for ITTMs)

Let $\bigcap_n \mathcal{U}_n$ be a universal λ-ML-test. We define Ω as being the leftmost path of $2^\omega - \mathcal{U}_0$. In particular Ω is λ-ML-random and has a left-c.e. approximation in L_λ.

In [19] Carl and Schlicht discuss the van Lambalgen theorem for λ-ML randomness. It holds using the fact that λ is projectible into ω. The proof is the same as the one for ω_1^{ck}-ML randomness (called Π^1_1-ML randomness in the literature) and works for any α limit such that α is projectible into ω. In particular for $\Omega = \Omega_1 \oplus \Omega_2$ we have that Ω_1 and Ω_2 are mutually λ-ML random.

Theorem V.48

There exists a non ITTM-writable set A which is ITTM-writable from both Ω_0 and Ω_1, the two halves of Chaitin's Ω for ITTMs.

Proof. Let us first show the following version of the Hirschfeldt and Miller theorem for ITTMs (see for example [57, Theorem 5.3.15]): let $\bigcap_n \mathcal{U}_n$ be a uniform intersection of λ-recursively enumerable open sets, with $\mu(\bigcap_n \mathcal{U}_n) = 0$. Then there exists a non-writable set A such that A is x-writable in every λ-ML random $x \in \bigcap_n \mathcal{U}_n$. The set A will be a λ-recursively enumerable simple set, that is, it will be co-infinite and intersect any infinite λ-recursively enumerable set of integers. Let $\bigcap_n \mathcal{U}_n$ be a uniform intersection of λ-recursively enumerable open sets of measure 0. Note that we can suppose without loss of generality that $\mathcal{U}_{n+1} \subseteq \mathcal{U}_n$. Let $\{W_e\}_{e\in\omega}$ be an enumeration of the λ-recursively enumerable sets.

The enumeration of A is defined by stages. At ordinal stage $s = \omega \times \alpha + \langle n, e\rangle$, if we have:

1. $n > 2e$,
2. $A[< s] \cap W_e[s] = \emptyset$,

3. $n \in W_e[\alpha]$
4. $\mu(\mathcal{U}_n[\alpha]) \leq 2^{-e}$

Then we add n to A at stage s.

First, let's show that A is simple. It is obviously co-infinite, as $|A \cap [0, 2n]| \leq n$ by requirements (1) and (2). Let e be such that W_e is infinite, and towards a contradiction, suppose that $W_e \cap A = \emptyset$. Then, let $m > 2e$ such that $\mu(\mathcal{U}_m) < 2^{-e}$, together with $n \geq m$ and α such that $n \in W_e[\alpha]$. Note that we have $\mu(\mathcal{U}_n) \leq \mu(\mathcal{U}_m) < 2^{-e}$. At stage $s = \omega \times \alpha + \langle n, e\rangle$ if $A[s] \cap W_e[s] = \emptyset$ then (1) (2) (3) and (4) will be met and $n \in W_e$ will be added to A at stage s.

Now, let's show that A is x-writable from every λ-ML-random element of $\bigcap_n \mathcal{U}_n$. We build the following ITTM-Solovay test: each time we add n into A at stage $s = \omega \times \alpha + \langle n, e\rangle$, we put $\mathcal{U}_n[\alpha]$ in the Solovay test. Note that by (4) we have $\mu(\mathcal{U}_n[\alpha]) < 2^{-e}$, in particular the measure requirement of the Solovay test is satisfied. Now if $x \in \bigcap_n \mathcal{U}_n$ is λ-ML-random it belongs to only finitely many such sets $\mathcal{U}_n[\alpha]$. In particular, there exists k such that for every $m \geq k$, if $m \in A$, then $m \in A[s]$ for $s = \omega \times (\alpha + 1)$ where α is the smallest such that $x \in \mathcal{U}_m[\alpha]$. We can then use x to write A.

Finally, it remains only to prove that Ω_0 and Ω_1 are both in a common uniform intersection $\bigcap_n \mathcal{U}_n$ of λ-open sets, with $\mu(\bigcap_n \mathcal{U}_n) = 0$. Each set $\mathcal{U}_n$ is given by

$$\mathcal{U}_n = \bigcup_{\alpha<\lambda} [\Omega_0[\alpha] \upharpoonright n] \cup \bigcup_{\alpha<\lambda} [\Omega_1[\alpha] \upharpoonright n]$$

It is clear that each set $\mathcal{U}_n$ is a λ-recursively enumerable open set which contains both Ω_0 and Ω_1. Let $S_0 = \{\Omega_0[\alpha] : \alpha < \lambda\} \cup \{\Omega_0\}$ and $S_1 = \{\Omega_1[\alpha] : \alpha < \lambda\} \cup \{\Omega_1\}$. To show that $\bigcap_n \mathcal{U}_n$ has measure 0, we use the following argument from [7, Proposition 5.1]: if $x \in \bigcap_n \mathcal{U}_n$, then x is at a distance of 0 from the set $S_0 \cup S_1$. Also it is clear that both S_0 and S_1 are closed sets, and thus that $S_0 \cup S_1$ is a closed set (in particular because for every i the sequences $\{\Omega_0(i)[\alpha]\}_{\alpha<\lambda}$ and $\{\Omega_1(i)[\alpha]\}_{\alpha<\lambda}$ change only finitely often). As x is at a distance 0 from a closed set, it is a member of the closed set. As the closed set is countable it has measure 0. It follows that $\mu(\bigcap_n \mathcal{U}_n) = 0$. ■

V.4 Genericity within ITTMs

Just like we define as random the sequences which are in every measure 1 set, among countably many sets, we define as *generic* the sequences which are in every co-meager set, among countably many sets. Both notions are obtained by considering a notion of largeness (measure 1 sets for randomness and co-meager sets for genericity), together with a countable class of large sets. For this reason both notions present many similar properties, and of course also many differences, as they are somehow opposite notions: whereas the random sets have no atypical property, the generic sets have them all.

The notion of genericity was designed by Cohen, as a canonical forcing notion. He considered as generic, the sets that belongs to no meager set, with a Borel code, in a countable model of ZFC. Various weakenings of this notion have then been considered in the literature. This has been done in computability by Jockush and Kurtz [49] [50], in higher computability by Greenberg and Monin [39], and for ITTMs by Carl and Schlicht [18]. In the later paper, the authors mostly focus on sets that are computable from every oracle in a large set, for various notions of largeness, including co-meagerness. We focus here on various genericity notions, defined from ITTM. We define in particular the categorical analogue of ITTM-randomness, and we show that it is equivalent to ITTM-genericity over L_Σ, whereas the equivalent question remains open for

the randomness case.

V.4.1 Genericity over the constructibles

Again, we do not work with the forcing relation, traditionally defined to deal with generic sets, but we instead directly deal with Borel sets. The following proposition is the ITTM version of the Baire property for lightface Borel sets, Theorem V.13. It is the core tool of forcing with co-meager set:

Theorem V.49 (Baire property theorem for ITTMs)

There is a function $b \mapsto (o, m)$, which to any ∞-Borel code b, associates the ∞-Borel code o of an open set, and the ∞-Borel code m of a union of closed meager sets, such that for any $x \notin \iota(m)$ we have $x \in \iota(b)$ iff $x \in \iota(o)$. Moreover this function is uniformly $\Delta_1^{L_\alpha}$ for α limit.

Proof. The function is defined by Δ_0 induction on the rank of sets of L_α with the bounded rank replacement of Proposition II.28. If b is the ∞-code of an open set then $o = b$ and m is the ∞-code of the empty set. If b is the ∞-code of a closed set then o is the interior of b and m is the boundary of b. We leave to the reader the proof that the function which to an ∞-Borel code of a closed set associates the ∞-Borel code of its interior and boundary, is uniformly $\Delta_1^{L_\alpha}$ for α limit.

Consider now an ∞-Borel code $b = \bigvee_{i\in I} c_i$. Note that the rank of each c_i in L_α is smaller than the rank of b. By induction we uniformly find ∞-Borel codes o_i and m_i such that for any i and any $x \notin \iota(m_i)$ we have $x \in \iota(b_i)$ iff $x \in \iota(o_i)$. We have that o is given by a code of $\bigcup_{i\in I} \iota(o_i)$ and m is given by a code of $\bigcup_{i\in I} \iota(m_i)$. It is clear that for any $x \notin \iota(m)$ we have $x \in \iota(b)$ iff $x \in \iota(o)$.

Consider now an ∞-Borel code $b = \bigwedge_{i\in I} c_i$. Note that the rank of each c_i in L_α is smaller than the rank of b. By induction we uniformly find ∞-Borel codes o_i and m_i such that for any n and any $x \notin \iota(m_i)$ we have $x \in \iota(b_i)$ iff $x \in \iota(o_i)$. We have that o is given by a code of the open set generated by all the strings σ such that each open set $\iota(o_i)$ is dense in $[\sigma]$. For each such string σ we find $m_{\sigma,i}$, the ∞-Borel code of the closed set of empty interior $[\sigma] - \iota(o_i)$. Let m_s be a code of the meager set given by the union of each such $\iota(m_{\sigma,i})$. The meager set $\iota(m_s) \cup \bigcup_{i\in I} \iota(m_i)$ ensures that if $x \in \iota(o)$, then $x \in \iota(\bigwedge_{i\in I} c_i)$. We now need to ensure that if $x \in \iota(\bigwedge_{i\in I} c_i)$ then $x \in \iota(o)$. For that we add the following meager set: for each o_i we consider an ∞-Borel code u_i of $2^\omega - \iota(o_i)$. We then let m_t be the boundary of the closure of $\bigcup_i \iota(u_i)$. A code m of our full meager set is then given by a code of $\iota(m_t) \cup \iota(m_s) \cup \bigcup_{i\in\omega} \iota(m_i)$. Suppose now that for $x \notin \iota(m)$ we have $x \in \bigwedge_{i\in I} c_i$, and suppose that for no prefix $\sigma \prec x$ we have $[\sigma] \subseteq \iota(o)$. In particular for every prefix $\sigma \prec x$, there is an extension $\tau \succ \sigma$ and some i such that $[\tau] \subseteq \iota(u_i)$. Also because $x \notin \bigcup_{i\in I} \iota(m_i)$ we must have $x \in \iota(o_i)$ for every i and then $\tau \not\prec x$. It follows that x is in the boundary of the closure of $\bigcup_i \iota(u_i)$, which contradicts that $x \notin \iota(m)$. ■

We now use the previous proposition to define the forcing relation in L_α for α limit, as follows:

Definition V.50

Let α be limit. Let $\Phi(p)$ be a formula and $p \in L_\alpha$ a parameter. Let $\mathcal{B}^\alpha(p) = \{x : L_\alpha(x) \models \Phi(p)\}$. Let o and m be the Borel codes of Theorem V.49, such that for $x \notin \iota(m)$ we have $x \in \iota(u_n)$ iff $x \in \mathcal{B}^\alpha(p)$. Then we define $\sigma \Vdash_\alpha \Phi(\dot{p})$ if $[\sigma] \subseteq \iota(o)$.

It is clear that for $z \succ \sigma$ generic enough, that is, which does not belong to sufficiently many meager sets, we have $L_\alpha(z) \models \Phi(p)$ iff $\sigma \Vdash_\alpha \Phi(\dot{p})$.

▶ **Proposition V.51.** *Let α be countable and limit. Let $\Phi(p)$ be a formula with parameter $p \in L_\alpha$. For any σ, there exists $\tau \succeq \sigma$ such that $\tau \Vdash_\alpha \Phi(\dot{p})$ or $\tau \Vdash_\alpha \neg\Phi(\dot{p})$.*

Proof. Let o_1 be the open set which equals $\{x \in 2^\omega \ : \ L_\alpha(x) \models \Phi(\dot{p})\}$ and o_2 be the open set which equals $\{x \in 2^\omega \ : \ L_\alpha(x) \models \neg\Phi(\dot{p})\}$, both up to a union of closed meager sets of Borel code m. Suppose we have $[\sigma] \cap (\iota(o_1) \cup \iota(o_2)) = \emptyset$ for some σ. In particular there is $z \succ \sigma$ with $z \notin \iota(m)$ (because a countable union of meager closed set is nowhere dense, here we use that α is countable). Either $L_\alpha(z) \models \Phi(p)$ or $L_\alpha(z) \models \neg\Phi(p)$. In the first case we must have $z \in \iota(o_1)$ and in the second case we must have $z \in \iota(o_2)$, which contradicts $[\sigma] \cap (\iota(o_1) \cup \iota(o_2)) = \emptyset$. ■

We now see that the predicate $\sigma \Vdash_\alpha \Phi(\dot{p})$ for Δ_0 formulas with parameters $\dot{p}$ is uniformly Δ^α_1. We in fact need a bit more, in order to show that the forcing relation for more complex formulas is still not too complex, even when α is not admissible (see Corollary V.53.1):

▶ **Proposition V.52.** *The function which to a string σ and a Δ_0 formula $\Phi(\dot{p})$ returns 1 iff $\sigma \Vdash_\alpha \Phi(\dot{p})$ (and 0 otherwise) is $\Delta^{L_\alpha}_1$ uniformly in α limit, and more so, the function which on a Δ_0 formula $\Phi(\dot{p})$ returns the function $f : 2^{<\omega} \to \{0,1\}$ such that $f(\sigma) = 1$ iff $\sigma \Vdash_\alpha \Phi(\dot{p})$, is $\Delta^{L_\alpha}_1$ uniformly in α limit.*

Proof. By Theorem V.20 one can uniformly find the Borel code of $\mathcal{B}^\alpha(p) = \{x \ : \ L_\alpha(x) \models \Phi(p)\}$. Then by Theorem V.49 one can uniformly find the Borel code o of the open set such that $\mathcal{B}^\alpha(p)$ equals $\iota(o)$ up to a meager set, and let $f(\ulcorner \Phi(\dot{p}) \urcorner) = o$. The function f is simply given by $f(\sigma) = 1$ iff $[\sigma] \subseteq \iota(o)$. ■

In the previous proposition, note that the forcing relation is uniform in α: for $\alpha_1 < \alpha_2$ both limit, the same formula defines the forcing relation, interpreted as $\Vdash_{\alpha_1}$ when working in L_{α_1} and interpreted as $\Vdash_{\alpha_2}$ when working in L_{α_2}.

▶ **Proposition V.53.** *Let α be limit. Let $\Phi(a,p)$ be some formula with parameter $p \in L_\alpha$. We have:*

$$\begin{array}{lll} \sigma & \Vdash_\alpha & \exists a \ \Phi(a,\dot{p}) \text{ iff } \exists \dot{a} \ \sigma \Vdash_\alpha \Phi(\dot{a},\dot{p}) \\ \sigma & \Vdash_\alpha & \forall a \ \Phi(a,\dot{p}) \text{ iff } \forall \dot{a} \ \forall \tau \succ \sigma \ \exists \rho \succeq \tau \ \rho \Vdash_\alpha \Phi(\dot{a},\dot{p}) \end{array}$$

Proof. This follows from the construction of the Borel code o of Theorem V.49 together with the definition of the forcing relation: for each $a \in L_\alpha$, let $\mathcal{A}_{\dot{a}} = \{x \in 2^\omega \ : \ L_\alpha \models \Phi(\dot{a},\dot{p})\}$ and let $o_{\dot{a}}$ be Borel codes of open sets such that $\iota(o_{\dot{a}})$ equals $\mathcal{A}_{\dot{a}}$ up to a union of closed meager set.

Then we have that the Borel code of the open set o of Theorem V.49 corresponding to $\bigcup_{\dot{a} \in P_\alpha} \mathcal{A}_{\dot{a}}$ is given by $\bigcup_{\dot{a} \in P_\alpha} \iota(o_i)$. This gives us exactly $\sigma \Vdash_\alpha \exists a \ \Phi(a,\dot{p})$ iff $\exists \dot{a} \ \sigma \Vdash_\alpha \Phi(\dot{a},\dot{p})$.

Now the Borel code of the open set o of Theorem V.49 corresponding to the set $\bigcap_{\dot{a} \in P_\alpha} \mathcal{A}_{\dot{a}}$ is given by Borel code of the open set generated by all the strings σ such that each $\iota(o_{\dot{a}})$ is dense in $[\sigma]$. This gives us exactly $\sigma \Vdash_\alpha \forall a \ \Phi(a,\dot{p})$ iff $\forall \dot{a} \ \forall \tau \succ \sigma \ \exists \rho \succeq \tau \ \rho \Vdash_\alpha \Phi(\dot{a},\dot{p})$. ■

▶ **Corollary V.53.1.** *Let α be limit and $n \geq 1$. The function which to a string σ and a Σ_n formula $\Phi(\dot{p})$ returns 1 iff $\sigma \Vdash_\alpha \Phi(\dot{p})$ (and 0 otherwise) is $\Sigma^{L_\alpha}_n$ uniformly in α.*

Proof. By induction on the complexity of formula, starting with the function f of Proposition V.52. For the induction, note the the quantifiers $\forall \tau \succ \sigma$ and $\exists \tau \succ \sigma$ are bounded,

and that for the Π case, we have to use each time the function $f : 2^{<\omega} \to \{0,1\}$ given by Proposition V.52. ■

V.4.2 Main definitions

We now formally define the notions of genericity that will be used.

Definition V.54

If α is an ordinal, a sequence z is *generic over* L_α if z is in every dense open set $\mathcal{U}$ with a Borel code in L_α.

This previous definition applied to ITTM give that z is generic over λ (resp. generic over ζ, resp. generic over Σ) if z is in every dense open set with a writable Borel code (resp. an eventually writable Borel code, resp. an accidentally writable Borel code). These notions are somehow analogues of Δ^1_1-genericity, in the sense that Δ^1_1-genericity corresponds to genericity over $L_{\omega_1^{ck}}$ as defined above.

▶ **Proposition V.55.** *Let α be limit. Let $\Phi(\dot{p})$ be a Δ_0 formula. Let z be generic over L_α. Then $L_\alpha(z) \models \Phi(\dot{p}[z])$ iff $\exists \sigma \prec z\ \sigma \Vdash_\alpha \Phi(\dot{p})$.*

Proof. By Theorem V.20 one can uniformly find the Borel code of $\mathcal{B}^\alpha(p) = \{x : L_\alpha(x) \models \Phi(p)\}$. Then by Theorem V.49 one uniformly find the Borel code m of the union of meager closed sets such that for any $x \notin \iota(m)$ we have $x \in \mathcal{B}^\alpha(p)$ iff $\exists \sigma \prec x\ \sigma \Vdash_\alpha \Phi(\dot{p})$. As z is generic over L_α it does not belong to $\iota(m)$ and the result follows. ■

We now define the categorical analogues of ITTM-randomness and ITTM-decidable randomness. A first idea would be to define as ITTM-generic reals those which are in every ITTM-semi-decidable open sets (open sets generated by semi-decidable set of strings). However it is clear that such open sets cannot be enumerated beyond stage λ, and the notion we get is not so interesting (it is in fact equivalent to genericity over L_λ). Instead we need to use reals as oracle and the following definition seems to be the correct one:

Definition V.56

Let $z \in 2^\omega$. We say that z is:
- ITTM-generic if it is in no meager ITTM-semi-decidable set.
- coITTM-generic if it is no meager ITTM-co-semi-decidable set.
- ITTM-decidable generic if it is in no meager ITTM-decidable set.

The counterparts of these notions for Infinite Time Register Machines have already been studied in [17].

V.4.3 ITTM-genericity and ITTM-decidable genericity

In this section, we will fully characterize genericity over ITTM-decidable, semidecidable and cosemidecidable sets in terms of genericity over a level of the L-hierarchy. We will see in particular that ITTM-genericity coincides with genericity over L_Σ, whereas the analogue question remains open for randomness.

ITTM-genericity

We first see why ITTM-genericity is the categorical analogue of ITTM-randomness.

Theorem V.57

Let $\alpha < \beta$ limit with $L_\alpha \prec_1 L_\beta$. Suppose $z \in 2^\omega$ is generic over L_β. Then $L_\alpha(z) \prec_1 L_\beta(z)$.

Proof. Suppose $L_\beta(z) \models \exists q\ \Phi(q,p)$ for a Δ_0 formula Φ and $p \in L_\alpha$. Let q be such that $L_\beta(z) \models \Phi(q,p)$. As z is generic over L_β and as Φ is Δ_0, there must exist by Proposition V.55 a string $\sigma \prec z$ such that $\sigma \Vdash_\beta \Phi(\dot{q},\dot{p})$. In particular as $\exists \dot{q}\ \sigma \Vdash_\beta \Phi(\dot{q},\dot{p})$ we have $\sigma \Vdash_\beta \exists q\ \Phi(q,\dot{p})$. By Σ_1-stability of L_α in L_β we have $\sigma \Vdash_\alpha \exists q\ \Phi(q,\dot{p})$ and then we have $L_\alpha(z) \models \exists q\ \Phi(q,p)$. ■

Theorem V.58

Let $z \in 2^\omega$. Then the following are equivalent

1. z is ITTM-generic
2. z is generic over L_Σ and $\Sigma^z = \Sigma$.
3. z is generic over L_ζ and $\zeta^z = \zeta$.

Proof. We first prove (1) implies (2). Suppose z is ITTM-generic. Note first that the set $\mathcal{A} = \{x \in 2^\omega\ :\ \Sigma^x > \Sigma\}$ is ITTM-semi-decidable: given z, one simply has to look for two z-accidentally writable ordinals $\alpha < \beta$ such that $L_\alpha \prec_2 L_\beta$ and then halt. Such a machine halts exactly on oracles x such that $\Sigma^x > \Sigma$. Carl and Schlicht showed [18] that if x is generic over $L_{\Sigma+1}$, then $\Sigma^x = \Sigma$ (we will improve this result with Corollary V.59.1). Thus the set $\mathcal{A}$ is a meager semi-decidable set, which implies that $\Sigma^z = \Sigma$. We now have to show that z is generic over L_Σ. Suppose not for contradiction. We can then design the machine which given x on its input tape, look for all the accidentally writable Borel codes of unions of closed set of empty interior, and halt whenever it finds one such that x is in it. It is clear that such a machine semi-decides a meager set, and in particular halts on z, which contradicts that z is ITTM-generic.

Let us now show that (2) implies (1). Suppose z is generic over L_Σ and $\Sigma^z = \Sigma$. Let M be an ITTM that semi-decides a meager set M. Suppose for contradiction that $M(z)\downarrow$. As we have $\Sigma^z = \Sigma$ we must also have $\zeta^z = \zeta$, by Theorem II.58. By Theorem V.57 we have $L_\lambda(z) \prec_1 L_\zeta(z) = L_{\zeta^z}(z)$. As λ^z is the smallest ordinal α such that $L_\alpha(z) \prec_1 L_{\zeta^z}(z)$ and as $\lambda \leq \lambda^z$ we then have $\lambda = \lambda^z$. It follows that $M(z)\downarrow[\alpha]$ for some $\alpha < \lambda$. Thus the set $\mathcal{B} = \{x \in 2^\omega\ :\ L_\lambda(x) \models M(x)\downarrow[\alpha]\}$ is a Borel set with a code in L_λ. As M halts on a meager set, the set $\mathcal{B}$ must be meager. As $z \in \mathcal{B}$ it is not generic over L_λ, which is a contradiction.

It is clear that (2) implies (3). Let us now show (3) implies (2). Suppose z is generic over L_ζ and $\zeta^z = \zeta$. By Theorem II.58 we have that $\Sigma^z = \Sigma$. Suppose for contradiction that z is not generic over L_Σ. Then we can design the machine M that looks for the smallest accidentally writable ordinal α such that L_α contains the Borel code of a meager set containing z, and when it finds it, writes α and halts. As z is not generic over L_Σ the machine M with input z will write some accidentally writable ordinal α and halt. As z is generic over L_ζ it must be the case that $\alpha > \zeta$. It follows that $\lambda^z > \zeta$ and thus $\zeta^z > \zeta$, a contradiction. ■

▶ **Corollary V.58.1.** *There is a largest ITTM semi-decidable meager set.*

Proof. Such a set is given in the proof of (1) implies (2), in the previous theorem: let M be the ITTM which halt on x such that $\Sigma^x > \Sigma$, or on x such that x belongs to a meager set with an accidentally writable Borel code. It is clear that M semi-decides a meager set. Also this meager set contains all the elements x which are not generic over L_Σ, or such that $\Sigma^x > \Sigma$. ■

We now show our main theorem for this section, that is, genericity over L_Σ coincides with ITTM-genericity.

Theorem V.59

Let $\alpha < \beta$ with β limit, such that $L_\alpha \prec_2 L_\beta$. Let z be generic over L_β. Then $L_\alpha(z) \prec_2 L_\beta(z)$.

Proof. Let $\Phi(a, b, p)$ be a Δ_0 formula with parameter $p \in L_\alpha$. By Theorem V.57 and Proposition II.34 we have that if $L_\alpha(z) \models \exists a\ \forall b\ \Phi(a, b, p)$, then $L_\beta(z) \models \exists a\ \forall b\ \Phi(a, b, p)$. Suppose now that $L_\beta(z) \models \exists a\ \forall b\ \Phi(a, b, p)$. Let us show that $L_\alpha(z) \models \exists a\ \forall b\ \Phi(a, b, p)$. We shall prove that $\exists \sigma \prec z\ \sigma \Vdash_\beta \exists a\ \forall b\ \Phi(a, b, \dot{p})$. Note that this is not obvious because z is only generic over L_β and the equivalence of Proposition V.55 works only for Δ_0 formulas.

For any γ limit such that $p \in L_\gamma$, let us define

$$\begin{array}{rcl} A_1^\gamma & = & \{\sigma \in 2^{<\omega} \ : \ \sigma \Vdash_\gamma \exists a\ \forall b\ \Phi(a, b, \dot{p})\} \\ A_2^\gamma & = & \{\sigma \in 2^{<\omega} \ : \ \sigma \Vdash_\gamma \forall a\ \exists b\ \neg\Phi(a, b, \dot{p})\} \end{array}$$

Suppose for a contradiction that for no prefix $\sigma \prec z$ we have $\sigma \in A_1^\beta$. Suppose first that also for no prefix $\sigma \prec z$ we have $\sigma \in A_2^\beta$. By Proposition V.51 it must be the case that either A_1^β is dense along z, or that A_2^β is dense along z (without containing z). Also by the fact that $L_\alpha \prec_2 L_\beta$ and by Corollary V.53.1, we must have $A_1^\alpha = A_1^\beta$ and $A_2^\alpha = A_2^\beta$. By considering the boundary of the closure of the open set generated by whichever set among A_1^α of A_2^α is dense along z, we obtain a meager closed set containing z, with a Borel code in L_α, which contradicts that z is generic over L_β.

Thus if for no $\sigma \prec z$ we have $\sigma \notin A_1^\beta$, it must be the case that $\sigma \in A_2^\beta$ for some $\sigma \prec z$. Let us fix such a string σ. In particular we must have $\sigma \Vdash_\beta \forall a\ \exists b\ \neg\Phi(a, b, \dot{p})$. By the fact that $L_\alpha \prec_2 L_\beta$ and by Corollary V.53.1 we must have $\sigma \Vdash_\alpha \forall a\ \exists b\ \neg\Phi(a, b, \dot{p})$. By Proposition V.53 we have:

$$L_\alpha \models \forall \dot{a}\ \forall \tau \succ \sigma\ \exists \rho \succ \tau\ \exists \gamma\ \exists \dot{b} \in P_\gamma\ \rho \Vdash_\alpha \neg\Phi(\dot{a}, \dot{b}, \dot{p})$$

By Theorem II.36 we must have that L_α is admissible. Using admissibility of L_α we must have:

$$L_\alpha \models \forall \dot{a}\ \exists \gamma\ \forall \tau \succ \sigma\ \exists \rho \succ \tau\ \exists \dot{b} \in P_\gamma\ \rho \Vdash_\alpha \neg\Phi(\dot{a}, \dot{b}, \dot{p})$$

Now coming back to the definition of forcing we easily see that we have:

$$L_\alpha \models \forall \dot{a}\ \exists \gamma\ \forall \tau \succ \sigma\ \exists \rho \succ \tau\ \rho \Vdash_\alpha \exists b \in L_\gamma\ \neg\Phi(\dot{a}, b, \dot{p})$$

Which by the fact that $L_\alpha \prec_2 L_\beta$ gives us:

$$L_\beta \models \forall \dot{a}\ \exists \gamma\ \forall \tau \succ \sigma\ \exists \rho \succ \tau\ \rho \Vdash_\beta \exists b \in L_\gamma\ \neg\Phi(\dot{a}, b, \dot{p})$$

It follows that for every $\dot{a} \in P_\beta$, there exists $\gamma < \beta$ such that the open set generated by the strings ρ for which $\rho \Vdash_\beta \exists b \in L_\gamma\ \neg\Phi(\dot{a}, b, \dot{p})$, is dense in $[\sigma]$. Also this open set is clearly a set of L_β, and its complement in $[\sigma]$ is a meager closet set of L_β. It follows that we must have a

prefix $\rho \prec z$ such that $\rho \Vdash_\beta \exists b \in L_\gamma \neg\Phi(\dot{a}, b, \dot{p})$, which implies $L_\beta(z) \models \exists b \in L_\gamma \neg\Phi(a, b, p)$. As this is true for every $\dot{a} \in P_\beta$, we must have that $L_\beta(z) \models \forall a\ \exists b\ \neg\Phi(a, b, p)$, which contradicts that $L_\beta(z) \models \exists a\ \forall b\ \Phi(a, b, p)$.

Thus it must be in the first place that $\sigma \Vdash_\beta \exists a\ \forall b\ \Phi(a, b, \dot{p})$ for some prefix $\sigma \prec z$. Then we also must have $\sigma \Vdash_\alpha \exists a\ \forall b\ \Phi(a, b, \dot{p})$ which implies $L_\alpha(z) \models \exists a\ \forall b\ \Phi(a, b, \dot{p})$. This concludes the proof. ■

▶ **Corollary V.59.1.** *If z is generic over L_Σ then $\Sigma^z = \Sigma$. In particular the set*

$$\{z \in 2^\omega \ :\ \Sigma^z > \Sigma\}$$

is meager.

Proof. This is because $L_\zeta \prec_2 L_\Sigma$, and because Σ^z is the smallest ordinal such that $L_\alpha(z) \prec_2 L_{\Sigma^z}(z)$ for some α. By the previous theorem we must have $\Sigma^z = \Sigma$. ■

▶ **Corollary V.59.2.** *Let $z \in 2^\omega$. The following are equivalent:*

1. *z is generic over L_Σ.*
2. *z is ITTM-generic.*

Proof. The equivalence is given by the conjunction of Theorem V.59 and V.58. ■

ITTM-decidable genericity

Theorem V.60

Let $z \in 2^\omega$. The following are equivalent:

1. z is generic over L_λ,
2. z is ITTM-decidable generic,
3. z is co-ITTM generic.

Proof. The implications (3) $\Rightarrow$ (2) and (2) $\Rightarrow$ (1) are trivial. Thus, it remains only to prove (1) $\Rightarrow$ (3). Let z be a real generic over L_λ. Let M be a machine that halts on a co-meager set. By Corollary V.59.1 we have that the set $\{x \in 2^\omega \ :\ \Sigma^x > \Sigma\}$ is meager. Note also that if z is generic over L_Σ we have $L_\lambda(x) \prec_1 L_\zeta(x)$ together with $L_\zeta(x) \prec_2 L_\Sigma(x)$. Thus the set $\{x \in 2^\omega \ :\ \lambda^x > \lambda\}$ is actually also meager. It follows that the set $\{x \in 2^\omega \ :\ \exists\alpha < \lambda\ M(x) \downarrow [\alpha]\}$ is already co-meager.

In particular the set $\{\sigma \ :\ \sigma \Vdash_\lambda \exists\alpha\ M(x) \downarrow [\alpha]\}$ must be a dense set of strings. By admissibility of λ, there must exists $\beta < \lambda$ such that the set $\{\sigma \ :\ \sigma \Vdash_\lambda \exists\alpha < \beta\ M(x) \downarrow [\alpha]\}$ is already a dense set of strings.

It follows that $\{x \in 2^\omega \ :\ M(x) \downarrow [\beta]\}$ is co-meager in a dense open set and thus comeager. Furthermore its complement is a union of nowhere dense closed sets with Borel code in L_λ. In particular as z is generic over L_λ, it must be that $M(z) \downarrow [\beta]$. Thus z also is co-ITTM generic. ■

Gotlib, RàB T2. Le matou matheux

Bibliography

[1] J. L. Hirst A. R. Blass and S. G. Simpson. Logical analysis of some theorems of combinatorics and topological dynamics. *Logic and Combinatorics*, 65:125–156, 1987.

[2] Paul-Elliot Anglès d'Auriac and Takayuki Kihara. A comparison of various analytic choice principles. *arXiv e-prints*, page arXiv:1907.02769, Jul 2019.

[3] Jeremy Avigad and Richard Sommer. A model-theoretic approach to ordinal analysis. *The Bulletin of Symbolic Logic*, 3(1):17–52, 1997.

[4] Jeremy Avigad and Richard Sommer. The model-theoretic ordinal analysis of theories of predicative strength. *The Journal of Symbolic Logic*, 64(1):327–349, 1999.

[5] Jon Barwise. *Admissible sets and structures*, volume 7. Cambridge University Press, 2017.

[6] James E. Baumgartner. A short proof of Hindman's theorem. *J. Comb. Theory, Ser. A*, 17:384–386, 1974.

[7] Laurent Bienvenu, Noam Greenberg, and Benoit Monin. Continuous higher randomness. *Journal of Mathematical Logic*, 17(01):1750004, 2017.

[8] Laurent Bienvenu, Noam Greenberg, and Benoit Monin. Continuous higher randomness. *J. Math. Log.*, 17(1):1750004, 53, 2017.

[9] Laurent Bienvenu, Noam Greenberg, and Benoît Monin. Bad oracles in higher computability and randomness. Submitted.

[10] Vasco Brattka. A galois connection between turing jumps and limits. arXiv 1802.01355, 2018.

[11] Vasco Brattka and Guido Gherardi. Effective choice and boundedness principles in computable analysis. *Bulletin of Symbolic Logic*, 17:73 – 117, 2011. arXiv:0905.4685.

[12] Vasco Brattka and Guido Gherardi. Weihrauch degrees, omniscience principles and weak computability. *Journal of Symbolic Logic*, 76:143 – 176, 2011. arXiv:0905.4679.

[13] Vasco Brattka, Guido Gherardi, Rupert Hölzl, Hugo Nobrega, and Arno Pauly. Borel choice. in preparation.

[14] Vasco Brattka, Guido Gherardi, and Alberto Marcone. The Bolzano-Weierstrass Theorem is the jump of Weak König's Lemma. *Annals of Pure and Applied Logic*, 163(6):623–625, 2012. also arXiv:1101.0792.

[15] Vasco Brattka, Guido Gherardi, and Arno Pauly. Weihrauch complexity in computable analysis. arXiv 1707.03202, 2017.

[16] Vasco Brattka and Arno Pauly. On the algebraic structure of Weihrauch degrees. arXiv 1604.08348, 2016.

[17] Merlin Carl. Randomness and degree theory for infinite time register machines. *Computability*, 5(2):181–196, 2016.

[18] Merlin Carl and Philipp Schlicht. Infinite computations with random oracles. *Notre Dame Journal of Formal Logic*, 58(2):249–270, 06 2013.

[19] Merlin Carl and Philipp Schlicht. Randomness via infinite computation and effective descriptive set theory. *Journal of Symbolic Logic*, 83(2):766–789, 2018.

[20] Lorenzo Carlucci. "Weak yet strong" restrictions of Hindman's Finite Sums Theorem. *arXiv e-prints*, page arXiv:1610.07500, Oct 2016.

[21] Lorenzo Carlucci, Leszek Aleksander Kołodziejczyk, Francesco Lepore, and Konrad Zdanowski. New bounds on the strength of some restrictions of Hindman's Theorem. *arXiv e-prints*, page arXiv:1701.06095, Jan 2017.

[22] Douglas Cenzer and Peter G. Hinman. Density of the Medvedev lattice of Π^0_1 classes. *Arch. Math. Logic*, 42(6):583–600, 2003.

[23] Chi Tat Chong, André Nies, and Liang Yu. Lowness of higher randomness notions. *Israel J. Math.*, 166(1):39–60, 2008.

[24] Chi Tat Chong and Liang Yu. Randomness in the higher setting. *The Journal of Symbolic Logic*, 80(4):1131–1148, 2015.

[25] Chi Tat Chong and Liang Yu. *Recursion Theory: Computational Aspects of Definability*, volume 8. Walter de Gruyter GmbH & Co KG, 2015.

[26] Paul J Cohen. *Set theory and the continuum hypothesis*. Courier Corporation, 2008.

[27] W. W. Comfort. Ultrafilters: Some old and some new results. *Bull. Amer. Math. Soc.*, 83(4):417–455, 07 1977.

[28] Keith J Devlin. *Constructibility*, volume 6. Cambridge University Press, 2017.

[29] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Theory and Applications of Computability. Springer, 2010.

[30] Damir D. Dzhafarov, Carl G. Jockusch, Reed Solomon, and Linda Brown Westrick. *Effectiveness of Hindman's Theorem for Bounded Sums*, pages 134–142. Springer International Publishing, Cham, 2017.

[31] H. B. Enderton and Hilary Putnam. A note on the hyperarithmetical hierarchy. *J. Symbolic Logic*, 35(3):429–430, 09 1970.

[32] Harvey Friedman. Uniformly defined descending sequences of degrees. *J. Symbolic Logic*, 41(2):363–367, 06 1976.

[33] Harvey M. Friedman. *Subsystems of set theory and analysis*. PhD thesis, 1967.

[34] Sy-David Friedman and Philip D Welch. Two observations concerning infinite time turing machines. *BIWOC*, pages 44–47, 2007.

[35] R. O. Gandy. Proof of mostowski's conjecture. *Journal of Symbolic Logic*, 29(2):103–104, 1964.

[36] Gili Golan and Boaz Tsaban. Hindman's coloring theorem in arbitrary semigroups. *Journal of Algebra*, 395, 03 2013.

[37] R. L. Graham and B. L. Rothschild. Ramsey's theorem for n-parameter sets. *Transactions of the American Mathematical Society*, 159:257–292, 1971.

[38] R.L. Graham, B.L. Rothschild, and J.H. Spencer. *Ramsey Theory*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 1990.

[39] Noam Greenberg and Benoit Monin. Higher randomness and genericity. *Forum of Mathematics, Sigma*, 5:e31, 2017.

[40] P. Hájek and P. Pudlák. *Metamathematics of First-Order Arithmetic*. Perspectives in Logic. Cambridge University Press, 2017.

[41] Joel David Hamkins and Andy Lewis. Infinite time turing machines. *The Journal of Symbolic Logic*, 65(02):567–604, 2000.

[42] Joseph Harrison. Recursive pseudo-well-orderings. *Transactions of the American Mathematical Society*, 131(2):526–543, 1968.

[43] Neil Hindman. Finite sums from sequences within cells of a partition of N. *J. Comb. Theory, Ser. A*, 17(1):1–11, 1974.

[44] Neil Hindman and Dona Strauss. *Algebra in the Stone-Čech compactification*, volume 27 of *De Gruyter Expositions in Mathematics*. Walter de Gruyter & Co., Berlin, 1998. Theory and applications.

[45] Greg Hjorth and André Nies. Randomness via effective descriptive set theory. *Journal of the London Mathematical Society*, 75(2):495–508, 2007.

[46] Jussi Ketonen and Robert Solovay. Rapidly growing ramsey functions. *Annals of Mathematics*, 113(2):267–314, 1981.

[47] Takayuki Kihara, Albert Marcone, and Arno Pauly. Searching for an analogue of ATR_0 in the Weihrauch lattice, 2018. in preparation.

[48] Takayuki Kihara and Arno Pauly. Dividing by Zero – How Bad Is It, Really? In Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier, editors, *41st Int. Sym. on Mathematical Foundations of Computer Science (MFCS 2016)*, volume 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 58:1–58:14. Schloss Dagstuhl, 2016.

[49] Stuart Kurtz. Randomness and genericity in the degrees of unsolvability. *Dissertation Abstracts International Part B: Science and Engineering*, 42(9):1982, 1982.

[50] Stuart Kurtz. Notions of weak genericity. *The Journal of symbolic logic*, 48(03):764–770, 1983.

[51] Jun Le Goh. Some computability-theoretic reductions between principles around ATR_0. *arXiv e-prints*, page arXiv:1905.06868, May 2019.

[52] Nicolas Lusin. Sur les ensembles non mesurables b et l'emploi de la diagonale cantor. *CR Acad. Sci. Paris*, 181:95–96, 1925.

[53] Per Martin-Löf. The definition of random sequences. *Information and Control*, 9:602–619, 1966.

[54] Benoit Monin. Higher randomness and forcing with closed sets. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 25. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[55] Antonio Montalbán. On the pi11-separation principle. *Math. Log. Q.*, 54:563–578, 2008.

[56] Antonio Montalbán. Open questions in reverse mathematics. *The Bulletin of Symbolic Logic*, 17(3):431–454, 2011.

[57] André Nies. *Computability and Randomness*. Oxford Logic Guides. Oxford University Press, 2009.

[58] Ludovic Patey and Keita Yokoyama. The proof-theoretic strength of Ramsey's theorem for pairs and two colors. *arXiv e-prints*, page arXiv:1601.00050, Dec 2015.

[59] Gerald E. Sacks. *Higher recursion theory*. Perspectives in mathematical logic. Springer-Verlag, 1990.

[60] Stephen G. Simpson. *Subsystems of Second Order Arithmetic*. Perspectives in Logic. Cambridge University Press, 2 edition, 2009.

[61] C. Spector. Hyperarithmetical quantifiers. *Fundamenta Mathematicae*, 48(3):313–320, 1960.

[62] John Steel. Descending sequences of degrees. *The Journal of Symbolic Logic*, 40(1):59–61, 1975.

[63] Henry Towsner. Hindman's theorem: an ultrafilter argument in second order arithmetic. *J. Symb. Log.*, 76(1):353–360, 2011.

[64] Henry Towsner. A simple proof and some difficult examples for hindman's theorem. *Notre Dame J. Formal Logic*, 53(1):53–65, 2012.

[65] Philip D Welch. The length of infinite time turing machine computations. *Bulletin of the London Mathematical Society*, 32(2):129–136, 2000.

[66] Philip D Welch. Characteristics of discrete transfinite time turing machine models: Halting times, stabilization times, and normal form theorems. *Theoretical Computer Science*, 410:426–442, 02 2009.

Gotlib, RàB T2. Le matou matheux