



HAL
open science

Flux combinatoires

Giti Omidvar

► **To cite this version:**

Giti Omidvar. Flux combinatoires. Informatique. Institut Polytechnique de Paris, 2023. Français.
NNT : 2023IPPAX152 . tel-04952717

HAL Id: tel-04952717

<https://theses.hal.science/tel-04952717v1>

Submitted on 17 Feb 2025

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.



INSTITUT
POLYTECHNIQUE
DE PARIS

NNT : 2023IPPAX152

Thèse de doctorat



Combinatorial Flows

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à École polytechnique

École doctorale n° 626 Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 11 Decembre 2023, par

MME GITI OMI DVAR

Composition du Jury :

Damiano Mazza Professeur, Université Sorbonne Paris Nord (LIPN)	President
Christian Retoré Professeur, Université de Montpellier (Equipe TEXTE, LIRMM)	Rapporteur
Mernoosh Sadrzadeh Professeur, University College London (Department of Computer Science)	Rapporteur
Willem Heijltjes Professeur, University of Bath (Department of Computer Science)	Examineur
Paolo Pistone Assistant Professeur, University of Bologna (Department of Computer Science)	Examineur
Lutz Straßburger Directeur de Recherche, Inria Saclay (PARTOUT)	Directeur de thèse

*To Morgoth and Manwë
May you find kindness in all that you meet*

Acknowledgements

Writing a thesis manuscript often appears as an impossible task, but it's indeed achievable, thanks in large part to the invaluable support received throughout this journey. I want to express my deep gratitude to the incredible people who have been instrumental in my success.

First and foremost, I owe an enormous amount of thanks to Lutz for his unwavering support from day one of meeting me, during my master's internship, and throughout my PhD which is more than three years. Your patience and guidance, both in scientific matters and personal growth, have been truly remarkable. Our meetings were always enlightening and enjoyable, and your words never failed to provide that extra push I needed.

While I had one official supervisor, I cannot overlook the invaluable guidance and lengthy discussions I received from other researchers in my circle. Willem is one of these special people, whom I've always looked up to in science. Thank you, Willem, for the numerous trips to Bath and the countless hours of insightful discussions, whether in front of a whiteboard in your office or on a table in a cozy cafe. In academia, having a mentor or a guiding figure is essential, and during my journey, Sonia has embodied the ideal academic mentor and an academic "older sister". Thank you, Sonia, for always being there for me, even when we were separated by physical distance. Your attendance at conferences significantly enhanced my experiences, making them extremely enriching. Additionally, your achievements as a successful woman in academia and research, particularly in a field such as proof theory where female representation is often scarce, have set an inspiring example for me. In our line of work, moments of solitude can be all too common, but Matteo has consistently made me feel less alone. Regardless of how nonsensical my ideas looked, you always lent a listening ear, found sense in them, and went as far as discussing them for several hours. A special thanks to my office mate and friend, Marianela, for sharing numerous moments during these years, especially the times we were frustrated about COVID and bureaucratic procedures in France. Initially, I thought our paths would not cross for long due to your initial earlier departure, but I consider myself fortunate that you stayed longer. Getting to know you has been an honor, and your presence made every day of the past year feel less lonely, especially as we embarked on the journey of completing our theses together. I also would like to extend my gratitude to Anupam, Marianna, Alessio, Torie, Dale, and Iris for their invaluable guidance and assistance along the way.

Beyond the realms of academia and professional life, my incredible partner, friends, and family have been the pillars of strength that drove me forward every day. To Igor, I consider myself extremely lucky to have found you. Your steady and untiring support has been my anchor during countless moments when I questioned my ability to continue. Especially during these past three months, you have been so patient with me and all my complaints about how hard it is to find a job and write a thesis at the same time. You never failed to stay positive and help me get through these moments. Moreover, traveling for conferences, workshops, and schools is an inseparable part of a researcher's life, it would have been impossible to attend most of these events if not for you. To Mary and Fred, I can never thank you enough for your friendship. You opened your home to me for several weeks when I had nowhere else to go, and our board game nights and discussions have left

me with several core memories. Morgan and Öykü, you are my inspiration as a couple. Knowing both of you has been a privilege, and your individual journeys inspire me to grow and embrace creativity every day. To Nicolas and Will, our friendship blossomed during the thematic month in CIRM. While we didn't become flatmates, we became great friends. Spending weekends with you, as well as Morgan, discussing category theory (which still remains a challenge for me to grasp) has been delightful. I wouldn't trade these moments for anything, and who knows, maybe one day I will learn enough category theory from listening to our conversations, or maybe Morgan will teach me. To my sister Mahtab, I want you to know that I miss you dearly. Although we have been thousands of kilometers apart for more than four years, you have always been a significant part of my life. From our lengthy book discussions to the moments when we simply chatted nonsense during video calls, I am proud to have you in my life. Finally, to my beloved Mom and Dad, your support during my education and life journey has been the foundation of my achievements. Without your dedication and support, I would not have made it to France or accomplished all that I have.

This thesis would have been an impossible endeavor without the support of my academic team members (PARTOUT), Inria personnel, and the Ecole Doctoral IP Paris.

To each and every one of you, I extend my heartfelt thanks, and I will forever be grateful for this unforgettable opportunity.

Contents

Acknowledgements	v
Table of Contents	vii
List of Figures	viii
Introduction (English)	1
Introduction (Français)	7
I Preliminaries	15
1 Classical Logic	17
1.1 Syntax	17
1.2 Semantics	19
1.2.1 Truth Tables	19
1.2.2 Semantics and Interpretation	19
2 Proof Systems	21
2.1 Natural Deduction	21
2.1.1 Rules	22
2.2 Sequent Calculus	23
2.3 Deep Inference	26
2.3.1 Open Deduction	26
2.4 Normal Proofs	29
3 Atomic Flows	31
4 Graphs	35
II Multiplicative and Additive Flows	41
5 From Derivations to Flows	43
6 Multiplicative Flows	49
7 Additive Flows	53
7.1 Skew Fibrations	53
7.2 a^\downarrow -flows and a^\uparrow -flows	55

7.3	A-flows	57
III	Combinatorial Flows and Their Normalization	71
8	Combinatorial Flows	73
8.1	Combinatorial Proofs with Cuts	75
8.2	Combinatorial Flows and Other Proof Systems	77
8.2.1	Sequent Calculus	77
8.2.2	Natural Deduction	78
9	Normalization	89
9.1	Purification	94
9.2	Cut Normalization	99
10	Conclusion and Future Work	111
	Bibliography	113
A	Graph Theory Proofs	119
B	Decomposition and Composition of Skew Fibrations	121

List of Figures

2.1	Natural deduction rules for classical propositional logic	22
2.2	An example of a natural deduction proof	23
2.3	Sequent Calculus SC inference rules	25
2.4	Inference rules of system SKS	27
2.5	An open deduction derivation with premise $(a \vee a) \wedge a \wedge \bar{a}$ and conclusion $a \vee f$	28
3.1	Generators for atomic flows	32
4.1	MIX R&B-proof-net connectors	38
4.2	Construction of R&B-trees	39
5.1	Inference rules of system SKS and their translation into flowboxes	45
5.2	Inference rules of system SKS and their translation into colored flowboxes	46
5.3	A deep inference derivation from $(a \vee a) \wedge (a \wedge \bar{a})$ to $a \vee f$ and its translation to a preflow and an atomic flow	47
6.1	Multiplicative inference rules of system SKS and their translation into m-flows (for the full system translation, see Figure 5.2)	51
7.1	Coalescence Rules	59
7.2	Rules of SKS_a and their translations to a-flows	63
7.3	Correctness of the a-flow $\phi = \langle (A \wedge B) \vee (C \wedge D), (A \vee C) \wedge (B \vee D), \mathbb{B}_\phi \rangle$ for a medial rule	65
8.1	An open deduction derivation with hypothesis $(a \vee a) \wedge (a \wedge \bar{a})$ and conclusion $a \vee f$, and its translation to a preflow with colors	74
8.2	A simplification of the combinatorial flow in Figure 8.1 and its purification	75
8.3	Translating combinatorial proofs with cuts into combinatorial flows.	76
8.4	Translation of \wedge logical rules of sequent calculus to combinatorial flows	79
8.5	Translation of \vee logical rules of sequent calculus to combinatorial flows	80
8.6	Translation of \neg logical rules and cut rule of sequent calculus to combinatorial flows	81
8.7	Translation of structural rules of sequent calculus to combinatorial flows	82
8.8	Translation of conjunctive and disjunctive rules of natural deduction to combinatorial flows	85
8.9	Translation of negation and implication rules of natural deduction to combinatorial flows	86
8.10	Translation of the rules ex and fE of natural deduction to combinatorial flows	87
8.11	Example of applying “Currying” to a combinatorial flow.	88
9.1	Normalization steps to apply on a combinatorial flow	92

9.2	Translation of a three-element normal combinatorial flow to a combinatorial proof	92
9.3	Horizontal compositions of normal combinatorial flows	94
9.4	A disjunctive composition of two normal combinatorial flows and its normalization	95
9.5	Purification Normalization Example	99
9.6	purification is not confluent	107
9.7	Purification cases for conjunction where the hypothesis of φ is unit-free formula A	108
9.8	Purification cases for conjunction where the hypothesis of φ is t	109
9.9	Purification cases for conjunction where the hypothesis of φ is f	110
B.1	Vertical composition of skew fibrations is not a skew fibration	121

Introduction (English)

Reasoning is a non-detachable part of mankind. Many claim that humans are superior to any other species known to us because of our analysis and reasoning abilities. We use reasoning in almost all the actions we take. However, some humans exercise reasoning rigorously on a daily basis or as a profession, namely mathematicians, philosophers, etc. More precisely, mathematicians explore abstract mathematical concepts and theories via heavy usage of different styles of proofs and reasoning. Philosophers dissect complex problems, arguments, and ideas meticulously while examining them from multiple perspectives to arrive at a reasoned conclusion. Proof theory aims at studying this common skill among all humans by employing logical and mathematical concepts and tools. In this thesis, we direct our attention toward the challenge of gaining a deeper understanding of equivalence among proofs.

Proof Theory

The core elements of mathematics are theorems and proofs. Often, a specific theorem can be proven via taking different approaches, sparking the desire to study the similarities and distinctions among them. Mathematicians aim to understand the commonalities and differences between proofs in general, exploring the possibility of transforming them into one another. Sometimes different theorems have common patterns and we want to study the context where we can apply such particular form of reasoning.

Despite the rich landscape of mathematical proofs, mathematicians traditionally communicate them using natural language, albeit with the inclusion of mathematical symbols. This inherent need to delve deeper into the nature of proofs, analyze their properties, and formalize them in a purely mathematical language has given rise to the field of proof theory. This discipline seeks to consider proofs as mathematical objects worthy of study, a concept initially proposed by Frege in 1879 [21]. Hilbert furthered these ideas by proposing a deductive system for formalizing reasoning [44].

Proof theory has solidified its position as one of the foundational pillars of mathematical logic. It introduces various formalisms to view proofs as mathematical entities, allowing for the exploration of their properties through formal mathematical methods. This has practical applications, such as the development of computer programs capable of automated theorem proving and proof verification. Additionally, we can extract algorithms from proofs, especially when we prove the existence of an object constructively. Moreover, we can extract counterexamples in cases of invalid theorems, a process known as proof mining [58]. Reverse mathematics [23], i.e. the study of axioms necessary for the proof of specific theorems, and proof complexity, i.e. comparing different properties and methods of proofs and more specifically the size of proof, are two more abstract consequences of proof theory.

Hilbert's Program

In the early 1900s, attempts were made by mathematicians such as Frege (in his *Grundgesetze der Arithmetik* [22]), and Bertrand Russell and Alfred North Whitehead (in *Principia Mathematica* [89]) to clarify the foundations of mathematics. However, these attempts failed and they all were found to suffer from paradoxes and inconsistencies, such as Russell's paradox shown in [76]. Consequently, Hilbert's program was proposed in the early 1920s as the solution to the foundational crisis of mathematics. Hilbert believed that an axiomatic approach is essential for the proper development of any kind of scientific subject. When offering an axiomatic approach, the theory is constructed without relying on intuition, while enabling a thorough examination of the logical connections among fundamental ideas and axioms. Hilbert emphasized that an axiomatic treatment should primarily focus on developing the theory independently from the intuition of the axioms and, most importantly, ensuring their coherence and absence of contradictions (i.e. consistency).

Gödel's incompleteness theorems, which he published in 1931 [29], dealt a significant blow to Hilbert's ambitious program. In his first theorem, Gödel demonstrated that any consistent system equipped with a computable set of axioms capable of expressing arithmetic could never achieve completeness. This means that within such a system, it's possible to formulate statements that are true but cannot be derived using the formal rules of the system itself. In his second theorem, Gödel went further to show that such a system couldn't even prove its own consistency, let alone the consistency of anything more powerful. This contradicted Hilbert's belief that a finitistic system could establish its own consistency and, by extension, prove the consistency of all other mathematical propositions.

Despite the setbacks caused by Gödel's theorems, modern research in mathematical logic, including fields like proof theory and reverse mathematics, can be seen as natural progressions from Hilbert's original program. Although the program had to be adjusted to adapt to the new realities exposed by Gödel, much of its essence and objectives remain a driving force in modern mathematical inquiry. An example of such progression is the development in the study of proof theory toward the 24th problem of Hilbert.

Towards the 24th Problem of Hilbert

In 1900 David Hilbert gave a lecture in Paris [43] in which he presented his now-famous 23 problems. Later on a 24th problem [86] was discovered written in his notebook [42], translated below:¹

“As 24th problem in my Paris lecture, I wanted to ask the question: Find criteria of simplicity or rather prove the greatest simplicity of given proofs. More generally develop a theory of proof methods in mathematics. Under given conditions there can be only one simplest proof. And if one has two proofs for a given theorem, then one must not rest before one has reduced one to the other or discovered which different premises (and auxiliary means) have been used in the proofs: When one has two routes then one must not just go these routes or find new routes, but the whole area lying between these two routes must be investigated...”

After more than a century from Hilbert's time, we find that proof theory has evolved into a well-established mathematical discipline. However, Hilbert's original challenge, as outlined

¹Translated by Lutz Straßburger [84].

in his notebook, remains an unsolved puzzle. The fundamental question of what makes a proof "simple" continues to baffle us. We lack clear criteria to determine the simplicity of a proof, and we struggle to compare proofs effectively. Surprisingly, even the simple establishment of a notion for proof identity has turned out to be extremely hard. This sets proof theory apart from many other mathematical fields where identifying objects of interest is more straightforward. For example, in group theory, two groups are considered the same if they are isomorphic, and in topology, two spaces are equivalent if they are homomorphic. This situation poses a significant challenge for proof theory as a mathematical discipline. A first vital step toward resolving Hilbert's 24th problem is to develop a satisfactory notion of proof identity.

In contemporary discussions about proof identity, there are mainly two strategies employed. The first approach involves identifying two proofs as equivalent if they can be transformed into each other using specific proof transformations. This can be achieved through techniques like proof normalization [68] or rule permutations [57]. The second approach focuses on defining canonical representations for proofs. Some well-known examples include λ -terms [16], proof nets [27], and combinatorial proofs [47].

A recurring concept in the second approach is the notion of tracking the occurrences of formulas or atoms within proofs. This idea has a history that dates back to the work of Kelly and Maclane [55], who used *coherence graphs* to determine morphism identity in a category. Later, Christian Retoré introduced *proof-nets* as bicolored graph representations [74] of tracing atoms and subformulas in a proof of Linear logic. This idea continued to evolve in the form of *string diagrams* [78]. All these three concepts have proven effective in the context of linear logic, where there are no contraction or weakening operations. For instance, proof nets serve as a canonical representation for multiplicative linear logic (MLL) [27].

However, in classical settings, where we add contraction and weakening to MLL, the idea of tracking formulas in a derivation took a different turn. Buss introduced *logical flow graphs* in [12], which was later studied by Carbone in [13, 14] to explore the relationship between cuts, contractions, and cycles in these flow graphs. It is important to note that these cycles do not exist in linear setting. To gain a more precise understanding of these cycles, Guglielmi and Gundersen developed *atomic flows* [32]. Atomic flows refine the concept of logical flow graphs by specifically tracing atoms and completely detaching the flow from the derivation. This approach brings atomic flows visually closer to coherence graphs and string diagrams. Moreover, atomic flows can serve as invariants for proofs and they can play a key role in the introduction of normalization procedures for deep inference derivations. However, atomic flows lose too much information from the proof. It is not possible to form a canonical proof representation of atomic flows. Moreover, there is no polynomial correctness criterion.² Therefore they do not form a proof system in the sense of Cook and Reckhow [17], i.e. the study of proof systems from the perspective of computational complexity.

Although this problem has already been solved by *combinatorial proofs* [47], it comes with an extremely high a price. Combinatorial proofs present a total separation between the linear part and the resource management part of the proof, where the linear part and its correctness criterion is due to Retoré [74]. However, performing this multiplicative/additive separation is as expensive as normalization, and therefore leads to a size explosion.

In this thesis, we present the idea of combinatorial flows, originally inspired by L. Straßburger's paper [81]³, which can be observed as a merging of atomic flows and combinatorial proofs.

²Das has shown in [20] that no such criterion is possible, under the assumption that integer factoring is hard for $P/poly$.

³The term combinatorial flows in this paper is used for composition over combinatorial proofs. However, in this thesis we use the same definitions as in [66] to show that in combinatorial flows we allow more flexibility in the compositions to the extent that we compose the linear parts of the proof freely with the resource management parts.

Normalization

Mathematicians often construct a proof for a theorem by establishing intermediate statements called lemmas. These lemmas serve as pivotal steps in the deductive process, akin to the Modus Ponens rule in a deductive system in Hilbert style. To prove a theorem, mathematicians typically employ Modus Ponens by first proving a relevant lemma and then deducing the theorem from it. Lemmas can involve clever tricks, such as using topology to solve numerical equations. However, in some contexts, shifting to a different theory or language than the theorem poses challenges, leading to a preference for lemma-free proofs, even if they are less elegant and more repetitive.

A lemma-free proof, often referred to as an *analytic proof*, is one that refrains from invoking additional lemmas and remains entirely within the language and theory of the theorem itself. This notion was introduced by Bolzano in 1817 [8] where he presented a “Purely analytic proof” of the intermediate value theorem. Such proofs rely solely on concepts introduced in the theorem, as opposed to introducing external mathematical methods. Gentzen later adopted and developed this concept in his thesis [25, 26], where he introduced the sequent calculus and demonstrated that any proof within this purely logical system can be transformed into an analytic normal form (Hauptsatz Theorem), the *cut* rule which is a reformulation of Modus Ponens is redundant in a proof by sequent calculus. This achievement led to the subformula property, ensuring that every formula appearing in the premises of an inference rule is a subformula of its conclusion, making the proof internally consistent.

In mathematics and computer science, we often use the term “normalization” to refer to a procedure of rewriting steps to produce a canonical form of an object, normalization of algebraic varieties and database normalization are examples of such procedures. Hauptsatz theorem is an instance of normalization, often referred to as cut-elimination. Furthermore, structural proof theory seeks to extend Gentzen’s results to other mathematical theories often expressed as axioms. However, the addition of any new axioms or rules to a sequent calculus can disrupt the cut-elimination process, a problem referred to as “lack of modularity” by Girard in [28]. More specifically, cut-elimination on sequent calculus heavily relies on syntactic arguments and has tedious case analysis.

One might ask whether the source of “lack of modularity” is in the heart of cut-elimination itself. Atomic flows, introduced by A. Guglielmi and T. Gundersen in [32], give a negative answer to such questions. They capture the skeleton of a proof while ignoring the syntactical content via using a “graphical” notation. Atomic flows are only an example of such objects. In fact, Girard’s proof nets [19, 27], which was originally proposed for linear logic [27], and Buss’ logical flow graphs [12], proposed for classical logic, employ similar notions to atomic flows. In this thesis, we will show that combinatorial flows have an independent normalization procedure from the syntactical design of logical rules.

Outline

This thesis consists of three main parts: preliminaries, multiplicative and additive flows, and combinatorial flows and their normalization. Part one is dedicated to recalling literature on mathematical logic and graph theory while setting the foundation for part two and three where we present our contributions.

- **Part I**, preliminaries, consists of chapters 1 to 4 and serves as the foundational backdrop for the subsequent exploration of proof theory and combinatorial flows. We embark on a journey through the literature of mathematical logic and graph theory, acquainting the reader with essential concepts and terminology.

-
- In **Chapter 1**, we introduce Classical logic, its syntax, and semantics. Every logical system is composed of two fundamental components: a formal language and a proof system.
 - In **Chapter 2**, we will show the latter by introducing sequent calculus, natural deduction, and deep inference formalisms. Moreover, we discuss normal proofs and the importance of normalization in proof theory and computer science.
 - **Chapter 3**, is dedicated to the presentation of atomic flows, introduced by A. Guglielmi and T. Gundersen in [32]. Atomic flows offer a graphical representation of proofs and formulas, allowing for a visual understanding of the logical structure underlying mathematical reasoning. We delve into the theoretical underpinnings of atomic flows.
 - In **Chapter 4**, we delve into the basics of graphical representations of proofs and formulas. This section is a recalling of the literature and work carried on by Christian Retore in [73, 72, 74]. We explore various graphical techniques for visualizing logical reasoning, providing insights into the relationship between graphical representations and formal proof systems. This chapter serves as a bridge between traditional formalisms and the emerging field of combinatorial proofs and combinatorial flows.
 - **Part II**, multiplicative and additive flows, consists of chapters 5 to 7 where we first define how to translate derivation to basic components of combinatorial flows. As we mentioned, the remaining two parts of this thesis (i.e this part and the next one) are dedicated to our contributions where we introduce *combinatorial flows*, originally inspired by Lutz Straßburger’s paper [81]. Combinatorial flows can be observed as a combination of atomic flows’ and combinatorial proofs’ desired characteristics. The first version of our combinatorial flows is presented as a joint work with Lutz Straßburger in WoLLIC 2022 [66] where we have introduced combinatorial flows as the free composition of multiplicative flows and additive flows (i.e resource management parts of the proof referred to as a^\uparrow -flows and a^\downarrow -flows). Below we outline the contents of chapter 5 to chapter 7:
 - In **Chapter 5**, we define the building blocks of combinatorial flows defined as flow-boxes as well as different possibilities of composing them. We show how we translate a deep inference derivation to such flows. Moreover, we dive more deeply into the purpose of our later separation between additive and multiplicative fragments of the logic.
 - **Chapter 6** and **Chapter 7** are dedicated to defining smaller elements of Combinatorial flows which are multiplicative flows and additive flows. To present the correctness criterion, we recall RB-cographs from Chapter 4 and we define the correctness for the linear parts (multiplicative flows) and the resource management parts (additive flows), respectively.
 - Additionally, in **Chapter 7**, upon further investigations on the properties of the additive fragment (or resource management part) of classical logic, we have been able to show that there exists no immediate need for a separation between weakening and contraction versus co-weakening and co-contraction which results in change of presentation of combinatorial flows in comparison with our paper [66]. In section 7.3, we show that in fact additive flows, i.e. a^\downarrow -flows and a^\uparrow -flows from [66], can be composed to form *a-flows* which are based on additive nets from [38] to accommodate the usage of units in our setting of classical logic. Moreover, we show that a-flows can be decomposed to a^\downarrow -flows and a^\uparrow -flows in a specific order.
 - **Part III**, shows combinatorial flows and their normalization separated into two following chapters:

-
- In **Chapter 8** we define combinatorial flows as correct preflows based on our paper [66]. Moreover, we show how to extend the translation, given in this paper, between combinatorial flows and Hughes’ combinatorial proof with cuts [49]. We adapt the soundness and completeness of combinatorial flows, which is proven via their close correspondence with deep inference system SKS, to the new definition of combinatorial flows. As an additional result in this thesis, we show the translation of natural deduction and sequent calculus proof to combinatorial flows in section 8.2 to better establish the correspondence between combinatorial flows and other proof systems.
 - **Chapter 9** investigates the normalization of combinatorial flows, which consists of purification, horizontal composition normalization, and vertical composition normalization. We introduce purification, in section 9.1, as a terminating procedure on formulas inside combinatorial flows that normalizes them into unit-free formulas or units, also in [66].

Introduction (Français)

La capacité de raisonner fait partie intégrante de l'humanité. Beaucoup prétendent que les humains sont supérieurs à toute autre espèce que nous connaissons en raison de nos capacités d'analyse et de raisonnement. Nous utilisons le raisonnement dans presque toutes les actions que nous entreprenons. Cependant, il existe des humains qui exercent un raisonnement rigoureux au quotidien ou à titre professionnel, à savoir les mathématiciens, les philosophes, etc. Plus précisément, les mathématiciens explorent des concepts et des théories mathématiques abstraites en faisant un usage intensif de différents styles de preuves et de raisonnements. Les philosophes disséquent méticuleusement des problèmes, des arguments et des idées complexes tout en les examinant sous plusieurs perspectives pour arriver à une conclusion raisonnée. La théorie de la démonstration vise à étudier cette compétence commune à tous les humains en employant des concepts et des outils logiques et mathématiques. Dans cette thèse, nous portons notre attention vers le défi d'acquérir une compréhension plus profonde de l'équivalence entre les preuves.

Théorie de la démonstration

Les éléments fondamentaux des mathématiques sont les théorèmes et les preuves, aussi appelées démonstrations. Souvent, un théorème spécifique peut être prouvé en adoptant différentes approches, suscitant le désir d'étudier les similitudes et les distinctions entre eux. Les mathématiciens visent à comprendre les points communs et les différences entre les preuves en général, en explorant la possibilité de transformer certaines en d'autres. Parfois, différents théorèmes ont des modèles communs et nous souhaitons étudier le contexte dans lequel nous pouvons appliquer cette forme particulière de raisonnement.

Malgré le riche paysage de preuves mathématiques, les mathématiciens les communiquent traditionnellement en utilisant le langage naturel, mais en incluant des symboles mathématiques. Ce besoin inhérent d'approfondir la nature des preuves, d'analyser leurs propriétés et de les formaliser dans un langage purement mathématique a donné naissance au domaine de la théorie de la démonstration. Cette discipline cherche à considérer les preuves comme des objets mathématiques dignes d'étude, un concept initialement proposé par Frege en 1879 [21]. Hilbert a approfondi ces idées en proposant un système déductif pour formaliser le raisonnement [44].

Depuis, la théorie de la démonstration a consolidé sa position comme l'un des piliers fondamentaux de la logique mathématique. Elle introduit divers formalismes pour considérer les preuves comme des entités mathématiques, permettant l'exploration de leurs propriétés à travers des méthodes mathématiques formelles. Cela a des applications pratiques, telles que le développement de programmes informatiques capables de prouver automatiquement des théorèmes et de vérifier des preuves. De plus, nous pouvons extraire des algorithmes à partir de preuves, notamment lorsque nous prouvons l'existence d'un objet de manière constructive. Il est aussi possible d'obtenir des contre-exemples dans les cas de théorèmes invalides, un processus connu sous le nom de proof mining [58]. Les mathématiques à rebours [23], c'est-à-dire l'étude des axiomes nécessaires à la preuve de théorèmes spécifiques, et la complexité de la preuve, c'est-à-dire la comparaison de dif-

férentes propriétés et méthodes de preuves et plus spécifiquement de la taille d'une preuve, sont deux conséquences plus abstraites de théorie de la démonstration.

logique Classique

La logique classique trouve ses racines historiques dans les premières traditions philosophiques de la Grèce antique, notamment au sein des écoles stoïciennes et autres écoles philosophiques qui ont émergé dès le III^e siècle avant JC. Cependant, la version moderne de la logique classique et ses avancées ultérieures peuvent être largement attribuées aux recherches et aux contributions du mathématicien anglais du XIX^e siècle George Boole. De plus, la formalisation initiale du calcul propositionnel classique, qui sert de système axiomatique formel pour la logique classique, a été lancée par Gottlob Frege en 1879 [21].

Le raisonnement déductif, en général, se déroule dans le contexte d'un langage naturel enrichi de concepts et de symboles mathématiques. Ainsi, la question suivante se pose naturellement : est-il possible d'établir un langage formel capable de représenter rigoureusement un tel raisonnement déductif? Dans ce contexte, un langage formel est défini comme un ensemble structuré de chaînes utilisant un alphabet fixe et caractérisé par des règles de syntaxe et de grammaire précises. Isolé, un langage formel manque de signification inhérente ; ce n'est qu'à travers le prisme d'un système déductif et de sa sémantique que le sens est attribué aux formules qu'il contient. *Logique classique* est un exemple d'un tel langage formel, principalement conçu pour formaliser la pratique du raisonnement mathématique, qui possède un riche héritage historique. Dans cette thèse, nous nous concentrerons principalement sur l'aspect *propositionnel* de la logique classique, y compris les déclarations qui s'évaluent comme vraies ou fausses, communément appelées *propositions*. Ainsi, le terme *logique propositionnelle* caractérise de manière appropriée ce fragment de logique classique. Par exemple, une affirmation comme "la Terre est ronde" représente une proposition largement reconnue comme vraie, tandis qu'une phrase comme "Briana est drôle" ne correspond pas aux critères d'une proposition en raison de l'ambiguïté inhérente à la proposition. attribut "drôle". Même si l'on tente de lui attribuer une valeur de vérité, vraie ou fausse, la nature subjective de l'humour rend impossible une détermination définitive.

Dans les chapitres suivants, chaque fois que nous mentionnons *logique classique*, nous faisons référence au fragment propositionnel de la logique classique.

Programme de Hilbert

Au début des années 1900, des tentatives ont été faites par des mathématiciens tels que Frege (dans son *Grundgesetze der Arithmetik* [22]), ainsi que Bertrand Russell et Alfred North Whitehead (dans *Principia Mathematica* [89]) pour clarifier les fondements des mathématiques. Cependant, ces tentatives ont échoué et elles se sont toutes révélées souffrir de paradoxes et d'incohérences, comme le paradoxe de Russell montré dans [76]. Par conséquent, le programme de Hilbert a été proposé au début des années 1920 comme solution à la crise fondamentale des mathématiques. Hilbert croyait qu'une approche axiomatique est essentielle au bon développement de tout type de sujet scientifique. Lorsqu'elle propose une approche axiomatique, la théorie est construite sans s'appuyer sur l'intuition, tout en permettant un examen approfondi des liens logiques entre les idées et les axiomes fondamentaux. Hilbert a souligné qu'un traitement axiomatique devrait principalement se concentrer sur le développement de la théorie indépendamment de l'intuition des axiomes et, surtout, garantir leur cohérence et leur absence de contradictions (c'est-à-dire la cohérence).

Les théorèmes d'incomplétude de Gödel, qu'il publia en 1931 [29], portèrent un coup dur au programme ambitieux de Hilbert. Dans son premier théorème, Gödel a démontré que tout système cohérent équipé d'un ensemble calculable d'axiomes capables d'exprimer l'arithmétique

ne pourrait jamais atteindre l'complétude. Cela signifie qu'au sein d'un tel système, il est possible de formuler des déclarations qui sont vraies mais qui ne peuvent pas être dérivées à l'aide des règles formelles du système lui-même. Dans son deuxième théorème, Gödel est allé plus loin en montrant qu'un tel système ne pouvait même pas prouver sa propre cohérence, et encore moins celle de quelque chose de plus puissant. Cela contredisait la conviction de Hilbert selon laquelle un système finitiste pouvait établir sa propre cohérence et, par extension, prouver la cohérence de toutes les autres propositions mathématiques.

Malgré les revers causés par les théorèmes de Gödel, la recherche moderne en logique mathématique, y compris des domaines comme la théorie de la démonstration et les mathématiques à rebours, peut être considérée comme une progression naturelle du programme original de Hilbert. Bien que le programme ait dû être ajusté pour s'adapter aux nouvelles réalités exposées par Gödel, une grande partie de son essence et de ses objectifs restent une force motrice dans la recherche mathématique moderne. Un exemple d'une telle progression est le développement de l'étude de la théorie de la démonstration vers le 24^e problème de Hilbert.

Vers le 24^{ème} problème de Hilbert

En 1900, David Hilbert donna une conférence à Paris [43] dans laquelle il présenta ses désormais célèbres 23 problèmes. Plus tard, un 24^{ème} problème [86] a été découvert écrit dans son cahier [42], traduit ci-dessous :⁴

“Comme 24^{ème} problème de mon cours de Paris, je voulais poser la question : Trouver des critères de simplicité ou plutôt prouver la plus grande simplicité de données preuves. Plus généralement développer une théorie de la démonstration méthodes en mathématiques. Dans des conditions données il ne peut y avoir qu'une seule preuve la plus simple. Et si on a deux preuves pour un théorème donné, puis une il ne faut pas se reposer avant d'avoir été réduit à l'état autre ou découvert quels différents locaux (et moyens auxiliaires) ont été utilisés dans les preuves : Quand on a deux itinéraires alors il ne faut pas il suffit de suivre ces itinéraires ou de trouver de nouveaux itinéraires, mais le toute la zone située entre ces deux routes doit faire l'objet d'une enquête. . .”

Plus d'un siècle après Hilbert, nous constatons que la théorie de la démonstration est devenue une discipline mathématique bien établie. Cependant, le défi initial de Hilbert, tel qu'il est exposé dans son carnet, reste une énigme non résolue. La question fondamentale de savoir ce qui rend une preuve « simple » continue de nous dérouter. Nous manquons de critères clairs pour déterminer la simplicité d'une preuve et nous avons du mal à comparer efficacement les preuves. Étonnamment, même la simple mise en place d'une notion de preuve d'identité s'est révélée extrêmement difficile. Cela distingue la théorie de la démonstration de nombreux autres domaines mathématiques dans lesquels l'identification des objets d'intérêt est plus simple. Par exemple, en théorie des groupes, deux groupes sont considérés comme identiques s'ils sont isomorphes, et en topologie, deux espaces sont équivalents s'ils sont homomorphes. Cette situation pose un défi important à la théorie de la démonstration en tant que discipline mathématique. Une première étape essentielle vers la résolution du 24^{ème} problème de Hilbert est de développer une notion satisfaisante de preuve d'identité.

Dans les discussions contemporaines sur la preuve d'identité, deux stratégies sont principalement utilisées. La première approche consiste à identifier deux preuves comme équivalentes si

⁴Traduit par Lutz Straßburger [84] de l'allemand vers l'anglais et par l'auteur de l'anglais vers le français.

elles peuvent être transformées l'une en l'autre à l'aide de transformations de preuve spécifiques. Ceci peut être réalisé grâce à des techniques telles que la normalisation des preuves [68] ou les permutations de règles [57]. La deuxième approche se concentre sur la définition de représentations canoniques pour les preuves. Quelques exemples bien connus incluent les λ -terms [16], les réseaux de preuves [27] et les preuves combinatoires [47].

Un concept récurrent dans la seconde approche est la notion de suivi des occurrences de formules ou d'atomes dans les preuves. Cette idée a une histoire qui remonte aux travaux de Kelly et Maclane [55], qui ont utilisé des *graphes de cohérence* pour déterminer l'identité du morphisme dans une catégorie. Plus tard, Christian Retoré a introduit les *proof-nets* sous forme de représentations graphiques bicolores [74] de traçage d'atomes et de sous-formules dans une preuve de logique linéaire. Cette idée a continué à évoluer sous la forme de *diagrammes de chaînes* [78]. Ces trois concepts se sont révélés efficaces dans le contexte d'une logique linéaire, où il n'y a pas d'opérations de contraction ou d'affaiblissement. Par exemple, les réseaux de preuves servent de représentation canonique pour la logique linéaire multiplicative (MLL) [27].

Cependant, dans les contextes classiques, où l'on ajoute une contraction et un affaiblissement à MLL, l'idée de suivre les formules dans une dérivation a pris une tournure différente. Buss a introduit les *logical flow graphs* dans [12], qui ont ensuite été étudiés par Carbone dans [13, 14] pour explorer la relation entre les coupures, les contractions et les cycles dans ces graphiques de flux. Il est important de noter que ces cycles n'existent pas en milieu linéaire. Pour mieux comprendre ces cycles, Guglielmi et Gundersen ont développé *atomic flow* [32]. Les flux atomiques affinent le concept de graphiques de flux logiques en traçant spécifiquement les atomes et en détachant complètement le flux de la dérivation. Cette approche rapproche visuellement les flux atomiques des graphes de cohérence et des diagrammes de cordes. De plus, les flux atomiques peuvent servir d'invariants pour les preuves et peuvent jouer un rôle clé dans l'introduction de procédures de normalisation pour les dérivations d'inférence profonde. Cependant, les flux atomiques perdent trop d'informations dans la preuve. Il n'est pas possible de former une représentation canonique des flux atomiques. De plus, il n'existe pas de critère d'exactitude polynomiale.⁵ Par conséquent ils ne forment pas un système de preuve au sens de Cook et Reckhow [17], c'est-à-dire l'étude des systèmes de preuve du point de vue de la complexité informatique.

Bien que ce problème ait déjà été résolu par des *preuves combinatoires* [47], il a un prix extrêmement élevé. Les preuves combinatoires présentent une séparation totale entre la partie linéaire et la partie gestion des ressources de la preuve, où la partie linéaire et son critère de correction sont dus à Retoré [74]. Cependant, réaliser cette séparation multiplicative/additive est aussi coûteuse que la normalisation, et conduit donc à une explosion de taille.

Dans cette thèse, nous présentons l'idée de flux combinatoires, inspirée à l'origine de l'article de L. Straßburger [81]⁶, qui peut être observé comme une fusion de flux atomiques et de preuves combinatoires.

Normalisation

Les mathématiciens construisent souvent une preuve d'un théorème en établissant des énoncés intermédiaires appelés lemmes. Ces lemmes servent d'étapes cruciales dans le processus déductif, semblables à la règle de Modus Ponens dans un système déductif de style Hilbert. Pour prouver un théorème, les mathématiciens emploient généralement le Modus Ponens en prouvant d'abord un

⁵Das a montré dans [20] qu'aucun critère de ce type n'est possible, sous l'hypothèse que la factorisation entière est difficile pour $P/poly$.

⁶Le terme flux combinatoires dans cet article désigne les compositions de preuves combinatoires. Ici, nous utilisons les mêmes définitions que dans [66] pour montrer que dans les flux combinatoires nous autorisons plus de flexibilité dans les compositions dans la mesure où nous composons librement les parties linéaires de la preuve avec les parties de gestion des ressources.

lemme pertinent, puis en en déduisant le théorème. Les lemmes peuvent impliquer des procédés astucieux, comme l'utilisation de la topologie pour résoudre des équations numériques. Cependant, dans certains contextes, passer à une théorie ou un langage différent de celui du théorème pose des défis, conduisant à une préférence pour les preuves sans lemme, même si elles sont moins élégantes et plus répétitives.

Une preuve sans lemme, souvent appelée *preuve analytique*, est une preuve qui s'abstient d'invoquer des lemmes supplémentaires et reste entièrement dans le langage et la théorie du théorème lui-même. Cette notion a été introduite par Bolzano en 1817 [8] où il a présenté une « Preuve purement analytique » du théorème des valeurs intermédiaires. De telles preuves reposent uniquement sur des concepts introduits dans le théorème, et non sur l'introduction de méthodes mathématiques externes. Gentzen a ensuite adopté et développé ce concept dans sa thèse [25, 26], où il a introduit le calcul séquentiel et démontré que toute preuve au sein de ce système purement logique peut être transformée en une norme analytique. (Théorème de Hauptsatz), la règle *cut* qui est une reformulation de Modus Ponens est redondante dans une preuve par calcul séquentiel. Cette réussite a conduit à la propriété de sous-formule, garantissant que chaque formule apparaissant dans les prémisses d'une règle d'inférence est une sous-formule de sa conclusion, rendant la preuve cohérente en interne.

En mathématiques et en informatique, nous utilisons souvent le terme « normalisation » pour désigner une procédure d'étapes de réécriture visant à produire une forme canonique d'un objet, la normalisation de variétés algébriques et la normalisation de bases de données sont des exemples de telles procédures. Le théorème de Hauptsatz est un exemple de normalisation, souvent appelé élimination de coupure. De plus, la théorie de la démonstration structurelle cherche à étendre les résultats de Gentzen à d'autres théories mathématiques souvent exprimées sous forme d'axiomes. Cependant, l'ajout de nouveaux axiomes ou règles à un calcul séquentiel peut perturber le processus d'élimination des coupures, un problème appelé « manque de modularité » par Girard dans [28]. Plus précisément, l'élimination des coupures sur le calcul séquentiel repose fortement sur des arguments syntaxiques et nécessite une analyse de cas fastidieuse.

On pourrait se demander si la source du “manque de modularité” se trouve au cœur même de l'élimination des coupures. Les flux atomiques, introduits par A. Guglielmi et T. Gundersen dans [32], donnent une réponse négative à de telles questions. Ils capturent le squelette d'une preuve tout en ignorant le contenu syntaxique via l'utilisation d'une notation “graphique”. Les flux atomiques ne sont qu'un exemple de tels objets. En fait, les réseaux de preuve de Girard [19, 27], qui ont été initialement proposés pour la logique linéaire [27], et les graphes de flux logiques de Buss [12], proposés pour la logique classique, emploient des notions similaires aux flux atomiques. Dans cette thèse, nous montrerons que les flux combinatoires ont une procédure de normalisation indépendante de la conception syntaxique des règles logiques.

Contour

La thèse se compose de trois parties principales : les préliminaires, les flux multiplicatifs et additifs, et les flux combinatoires et leur normalisation. La première partie est consacrée au rappel de la littérature sur la logique mathématique et la théorie des graphes tout en jetant les bases des deuxième et troisième parties où nous présentons nos contributions.

- **Partie I**, préliminaires, se compose des chapitres 1 à 4 où nous familiarisons le lecteur avec la littérature.
 - Dans le **Chapitre 1**, nous introduisons la logique classique, sa syntaxe et sa sémantique. Tout système logique est composé de deux composants fondamentaux : un langage formel et un système de preuve.

-
- Dans le **Chapitre 2**, nous montrerons ce dernier en introduisant le calcul séquentiel, la déduction naturelle et les formalismes d’inférence profonde. De plus, nous discutons des preuves normales et de l’importance de la normalisation dans la théorie de la démonstration et en informatique.
 - **Chapitre 3**, est dédié à la présentation des flux atomiques, introduits par A. Guglielmi et T. Gundersen dans [32]. Les flux atomiques offrent une représentation graphique des preuves et des formules, permettant une compréhension visuelle de la structure logique sous-jacente au raisonnement mathématique. Nous approfondissons les fondements théoriques des flux atomiques.
 - Dans le **Chapitre 4**, nous approfondissons les bases des représentations graphiques des preuves et des formules. Cette section est un rappel de la littérature et des travaux menés par Christian Retore dans [73, 72, 74]. Nous explorons diverses techniques graphiques pour visualiser le raisonnement logique, fournissant ainsi un aperçu de la relation entre les représentations graphiques et les systèmes de preuve formels. Ce chapitre sert de pont entre les formalismes traditionnels et le domaine émergent des preuves combinatoires et des flux combinatoires.
- **Partie II**, flux multiplicatifs et additifs, se compose des chapitres 5 à 7 où nous définissons d’abord comment traduire la dérivation en composants de base des flux combinatoires. Comme nous l’avons mentionné, les deux parties restantes de cette thèse (c’est-à-dire cette partie et la suivante) sont consacrées à nos contributions où nous introduisons des *flux combinatoires*, inspirés à l’origine de l’article de Lutz Straßburger [81]. Les flux combinatoires peuvent être observés comme une combinaison des caractéristiques souhaitées des flux atomiques et des preuves combinatoires. La première version de nos flux combinatoires est présentée comme un travail conjoint avec Lutz Straßburger dans WoLLIC 2022 [66] où nous avons introduit les flux combinatoires comme composition libre de flux multiplicatifs et de flux additifs. (c’est-à-dire les parties de gestion des ressources de la preuve appelées a^\uparrow -flows et a^\downarrow -flows). Ci-dessous, nous décrivons le contenu du chapitre 5 au chapitre 7 :
 - Dans le **Chapitre 5**, nous définissons les éléments constitutifs des flux combinatoires définis comme des flowbox ainsi que les différentes possibilités de les composer. Nous montrons comment nous traduisons une dérivation d’inférence profonde à de tels flux. De plus, nous approfondissons le but de notre séparation ultérieure entre les fragments additifs et multiplicatifs de la logique.
 - **Chapitre 6** et **Chapitre 7** sont dédiés à la définition d’éléments plus petits des flux combinatoires qui sont des flux multiplicatifs et des flux additifs. Pour présenter le critère de correction, nous rappelons les RB-cographe du chapitre 4 et nous définissons la correction pour les parties linéaires (flux multiplicatifs) et les parties de gestion des ressources (flux additifs), respectivement.
 - De plus, au **Chapitre 7**, après des recherches plus approfondies sur les propriétés du fragment additif (ou partie de gestion des ressources) de la logique classique, nous avons pu montrer qu’il n’existe pas de besoin immédiat d’une séparation entre l’affaiblissement et la contraction par rapport à co-affaiblissement et co-contraction qui se traduisent par un changement de présentation des flux combinatoires par rapport à notre article [66]. Dans la section 7.3, nous montrons qu’en fait les flux additifs, c’est-à-dire les a^\downarrow -flows et les a^\uparrow -flows de [66], peuvent être composés pour former des *a-flows* qui sont basés sur des réseaux additifs de [38] pour s’adapter à l’utilisation d’unités dans notre cadre de logique classique. De plus, nous montrons que les *a-flows* peuvent être décomposés en a^\downarrow -flows et a^\uparrow -flows dans un ordre spécifique.
-

-
- **Partie III**, montre les flots combinatoires et leur normalisation séparés en deux chapitres suivants :
 - Dans le **Chapitre 8**, nous définissons les flux combinatoires comme des préflux corrects basés sur notre article [66]. De plus, nous montrons comment étendre la traduction, donnée dans cet article, entre les flux combinatoires et la preuve combinatoire de Hughes avec cuts [49]. Nous adaptons la solidité et l'exhaustivité des flux combinatoires, qui sont prouvées par leur correspondance étroite avec le système d'inférence profonde SKS, à la nouvelle définition des flux combinatoires. Comme résultat supplémentaire dans cette thèse, nous montrons la traduction de la preuve par déduction naturelle et par calcul séquentiel en flux combinatoires dans la section 8.2 pour mieux établir la correspondance entre les flux combinatoires et d'autres systèmes de preuve.
 - **Chapitre 9** étudie la normalisation des flux combinatoires, qui consiste en une purification, une normalisation de la composition horizontale et une normalisation de la composition verticale. Nous introduisons la purification, dans la section 9.1, comme une procédure de terminaison sur les formules à l'intérieur de flux combinatoires qui les normalise en formules ou unités sans unité, également dans [66].

Part I

Preliminaries

CHAPTER 1

Classical Logic

Classical logic finds its historical roots in the early philosophical traditions of ancient Greece, notably within the Stoic and other philosophical schools that emerged as far back as the 3rd century B.C. However, the modern version of classical logic and its subsequent advancements can be largely attributed to the research and contributions of the 19th-century English mathematician George Boole. Furthermore, the initial formalization of classical propositional calculus, which serves as a formal axiomatic system for classical logic, was pioneered by Gottlob Frege in 1879 [21].

Deductive reasoning, in general, unfolds within the context of a natural language enriched by mathematical concepts and symbols. Thus, the following question arises naturally: Is it possible to establish a formal language that can rigorously represent such deductive reasoning? In this context, a formal language is defined as a structured collection of strings utilizing a fixed alphabet and characterized by precise syntax and grammar rules. In isolation, a formal language lacks inherent meaning; it is only through the lens of a deductive system and its semantics that meaning is attributed to the formulas within it. *Classical logic* is an example of such a formal language, primarily designed to formalize the practice of mathematical reasoning, which has a rich historical heritage. In this thesis we will focus mainly on the *propositional* aspect of classical logic, including statements that evaluate as either true or false, commonly referred to as *propositions*. Thus, the term *propositional logic* characterizes this fragment of classical logic appropriately. For instance, a statement like “the Earth is round” represents a proposition, widely acknowledged as true, while a sentence such as “Briana is funny” does not fit the criteria of a proposition due to the inherent ambiguity surrounding the attribute “funny”. Even if one attempts to ascribe it a truth value of either true or false, the subjective nature of humor renders an impossible definitive determination.

In following chapters, whenever we mention *classical logic*, we are referring to the propositional fragment of classical logic.

1.1 Syntax

We define the *formulas* of classical propositional logic, denoted by A, B, \dots to be generated from a countable set of propositional variables $\{a, b, \dots\}$ and their negations $\{\bar{a}, \bar{b}, \dots\}$ with the following grammar. The propositional variables are the building blocks of classical logic.

$$A, B := t \mid f \mid a \mid \bar{a} \mid A \vee B \mid A \wedge B$$

Negation can be also extended to all formulas via De Morgan laws:

$$\bar{\bar{t}} = t \quad \bar{\bar{f}} = f \quad \bar{\bar{a}} = a \quad \overline{A \wedge B} = \bar{A} \vee \bar{B} \quad \overline{A \vee B} = \bar{A} \wedge \bar{B}$$

We define the equivalence relation \equiv on formulas:

$$\begin{array}{llll} A \wedge B \equiv B \wedge A & (A \wedge B) \wedge C \equiv A \wedge (B \wedge C) & A \wedge \text{t} \equiv A & \text{t} \vee \text{t} \equiv \text{t} \\ A \vee B \equiv B \vee A & (A \vee B) \vee C \equiv A \vee (B \vee C) & A \vee \text{f} \equiv A & \text{f} \wedge \text{f} \equiv \text{f} \end{array} \quad (1.1)$$

A formula A is a **unit** if $A = \text{t}$ or $A = \text{f}$, it is **unit-free** if it does not contain any units, and it is **pure** if either $A \equiv \text{t}$ or $A \equiv \text{f}$ or $A \equiv A'$ for some unit-free formula A' . An **atom** is an element in $\mathcal{A} = \{a, b, \dots\} \cup \{\bar{a}, \bar{b}, \dots\}$, and they are the only variables in classical propositional logic. A **multi-set of formulas** $\Gamma = A_1, \dots, A_n$ is a finite non-empty multiset of formulas. By conjunction on Γ , denoted as $\bigwedge \Gamma$, we mean the conjunction of all formulas in Γ which we show as:

$$\bigwedge \Gamma := A_1 \wedge \dots \wedge A_n$$

The disjunction on Γ , denoted as $\bigvee \Gamma$, has the same pattern as the conjunction:

$$\bigvee \Gamma := A_1 \vee \dots \vee A_n$$

The negation of Γ , denoted as $\bar{\Gamma}$, means that we just negate all the formulas in Γ :

$$\bar{\Gamma} := \bar{A}_1, \dots, \bar{A}_n$$

Definition 1.1. The **formula tree** of a formula A , denoted as $T(A)$, is a binary-tree with vertices labeled as subformulas of A defined inductively as follows:

- the root of the tree is the formula A ;
- if $A = B \vee C$, then the left child of A is B and the right child is C ;
- if $A = B \wedge C$, then the left child is B and the right one is C ;
- if A is an atom or unit, then it does not have a child and this is a leaf of the tree.

Note that the atom occurrences of A are the set of leaves in $T(A)$ that are not labeled with units. We denote it as $\lfloor A \rfloor$.

Remark 1.2. A set of formulas $\Gamma = \{A_1, \dots, A_n\}$ is a multi-set of formulas as well and the connectives $\bigwedge \Gamma$, $\bigvee \Gamma$, and $\bar{\Gamma}$ are defined the same. This is an important note due to usage of set of formulas in Section 2.1 for hypothesis of natural deduction derivation.

Definition 1.3. We define the **rank** of formula A , denoted as $|A|$, as a value defined inductively below:

- if A is a unit: $|\text{t}| = |\text{f}| = 0$
- if $A = a$ or $A = \bar{a}$ (is an atom): $|a| = |\bar{a}| = 1$
- if $A = B \wedge C$: then $|A| = |B| + |C|$
- if $A = B \vee C$: then $|A| = |B| + |C|$

Note that we defined negation of a formula inductively. Hence, the rank of negation of a formula such as $|\bar{A}|$ is defined inductively as well. For example if $A = B \wedge C$ then:

$$|\bar{A}| = |\overline{B \wedge C}| = |\bar{B} \vee \bar{C}| = |\bar{B}| + |\bar{C}|$$

We also can compare formulas by their rank. A formula A is **smaller** than a formula B , if $|A| < |B|$. Similarly, we say A has **smaller or the same size as** B , if $|A| \leq |B|$. A formula A is minimal if for all formulas $B \equiv A$, we have $|A| \leq |B|$.

From the inductive definition of negation using De Morgan Laws and rank of a formula follows:

Proposition 1.4. For a formula A , we have $|\bar{\bar{A}}| = |A|$.

Proof. trivial by Definition 1.3. □

A	B	$A \wedge B$	$A \vee B$
t	t	t	t
t	f	f	t
f	t	f	t
f	f	f	f

Table 1.1: Truth table for conjunction and disjunction

1.2 Semantics

As previously mentioned, the formal language of classical propositional logic borrows its meaning from its semantics, and this semantic aspect is crucial for understanding how logical propositions are evaluated. The semantics of classical propositional logic, known as *two-valued semantics*, is built upon the concept of truth values, specifically *true* and *false*. Each formula in the language is assigned one of these two truth values based on a set of well-defined rules. To begin, we demonstrate the practical application of truth tables as a tool for determining the truth value of each formula. These tables offer an informal presentation of semantics, with a more precise definition available in 1.2.2. Truth tables provide a systematic way to determine the truth value of complex formulas by exhaustively considering all possible combinations of truth values for the atomic propositions involved. Starting with atomic propositions, truth tables guide us in evaluating compound propositions step by step, ensuring that we adhere to the logical connectives' rules, including conjunction, disjunction, and negation.

For example, in a truth table, the formula $A \wedge B$ is evaluated as true only when both A and B are true, otherwise, it is false. Similarly, the formula $A \vee B$ is true if either A or B is true. Negation, represented as \bar{A} , flips the truth value of A . Units, such as t (always true) and f (always false), serve as reference points for constant truth values.

In this way, the semantics of classical propositional logic, grounded in two-valued semantics and exemplified by truth tables, provides a systematic and rigorous approach to assessing the truth or falsity of logical propositions, forming the basis for sound deductive reasoning within this logical system.

1.2.1 Truth Tables

A truth table shows all the possible values for a formula based on the values given to its atoms. We show the truth table for conjunction and disjunction in Table 1.1. We know every formula can have two values which leads to having four possible ways of choosing values for two formulas. Hence, there is four lines in the truth table, each line showing one possible way of assigning values. The column $A \wedge B$ shows the truth value of conjunction in each of these cases. Similarly, the column $A \vee B$ show the truth value of disjunction. These values follow the definition of conjunction and disjunction.

1.2.2 Semantics and Interpretation

An *interpretation* of the language is a tuple $\mathcal{M} = \langle d, I \rangle$ where d is a subset of \mathcal{A} which is the domain of interpretation and $I: d \rightarrow \{t, f\}$ is the interpretation function on d . The interpretation function assigns true and false values to every formula in its domain. Below, we show how the interpretation function is expanded to cover the whole language. We use the notation $\mathcal{M} \models A$ to denote that the interpretation \mathcal{M} *satisfies* the formula A (the value assigned to A under the interpretation \mathcal{M} is true). If we use $\not\models$ it means that the formula is not being satisfied (the value of the formula under the interpretation is false).

- $\mathcal{M} \vDash t$
- $\mathcal{M} \not\vDash f$
- $\mathcal{M} \vDash a$, where $a \in d$, if $I(a) = t$
- $\mathcal{M} \not\vDash a$, where $a \in d$, if $I(a) = f$
- $\mathcal{M} \vDash A \wedge B$ if and only if $\mathcal{M} \vDash A$ and $\mathcal{M} \vDash B$
- $\mathcal{M} \vDash A \vee B$ if and only if either $\mathcal{M} \vDash A$ or $\mathcal{M} \vDash B$ (note that here it is possible that both cases are happening)

We define a formula A to be a *tautology* if and only if for every interpretation \mathcal{M} , we have $\mathcal{M} \vDash A$. For a multi-set of formulas Γ , we say Γ is *satisfied* with an interpretation \mathcal{M} (denoted as $\mathcal{M} \vDash \Gamma$) if and only if \mathcal{M} satisfies every formula $A \in \Gamma$. The formula A is a *semantical consequence* of a multi-set of formulas Γ (denoted as $\Gamma \vDash A$) if and only if for all interpretation \mathcal{M} for which $\mathcal{M} \vDash \Gamma$, we also have $\mathcal{M} \vDash A$.

CHAPTER 2

Proof Systems

A *logical system* comprises two fundamental components: a *formal language* and a set of *axioms* and *inference rules* for guiding logical reasoning, i.e. a *deductive system* or a *proof system*. In our exploration of classical propositional logic in the previous chapter, we have already introduced the formal language that serves as the framework for expressing logical propositions. In this chapter, we will discuss several possibilities for reasoning statements in the formal language, namely natural deduction, sequent calculus, and deep inference.

As soon as a proof system is introduced, we need to show that it is meaningful. For this purpose we use soundness and completeness. A proof system S is *sound* if any formula provable in S is also true in all structures of the semantics for the logic upon which the system is based. A system S is *complete* if all the tautologies in the logic are also provable in S . It is *strongly complete* if for every multi-set of formulas Γ as premises, any semantical consequence of Γ is also derivable in S from Γ .

2.1 Natural Deduction

The term *Natural Deduction* refers to a class of proof systems founded on inference rules that draw from traditional modes of reasoning with a rich historical heritage. The inception of formal natural deduction systems is credited to Gentzen [25, 26] and Jaśkowski [53], who independently developed these systems. Natural deduction emerged as an alternative to the Hilbert-style axiomatic systems. Gentzen's formulation, notably suited for investigating the structure of proofs, contrasted with Jaśkowski's version which is more suitable for practical proof-search applications. A key distinction in natural deduction, when compared to other axiomatic systems, is its incorporation of free assumptions or hypotheses. These assumptions, initially made without commitment, are subsequently discharged within the framework of the applied inference rules, granting flexibility in proof-search strategies. In essence, natural deduction facilitates the composition and decomposition of formulas during the proof process.

While it may be tempting to trace the historical lineage of natural deduction back to ancient Greece, particularly to Aristotle's syllogistic logic, such claims hold limited theoretical significance. The introduction of natural deduction marked a profound advancement in modern logic. For instance, natural deduction plays a pivotal role in computer science, thanks to the Curry-Howard isomorphism [45]. This isomorphism allows us to interpret natural deduction proofs as programming constructs, demonstrating the enduring relevance and versatility of this approach. An illustrative example of this is the rule of modus ponens, shown below, a fundamental concept to be explored in greater detail later, emphasizing natural deduction's enduring importance in modern logical and computational discourse.

$$\begin{array}{c}
 \frac{A \quad B}{A \wedge B} \wedge I \qquad \frac{A \wedge B}{A} \wedge E_l \qquad \frac{A \wedge B}{B} \wedge E_r \\
 \\
 \frac{A}{A \vee B} \vee I_l \qquad \frac{B}{A \vee B} \vee I_r \qquad \frac{\begin{array}{c} \Gamma, [A] \\ \mathcal{D}_A \\ C \end{array} \quad \begin{array}{c} \Delta, [B] \\ \mathcal{D}_B \\ C \end{array}}{A \vee B \quad C} \vee E \\
 \\
 \frac{\begin{array}{c} \Gamma, [A] \\ \mathcal{D} \\ f \\ A \end{array}}{A} \neg I \qquad \frac{A \quad \bar{A}}{f} \neg E \qquad \frac{\begin{array}{c} \Gamma, [\bar{A}] \\ \mathcal{D} \\ f \\ A \end{array}}{A} \text{ex} \\
 \\
 \frac{f}{A} \text{fE} \qquad \frac{A \quad A \rightarrow B}{B} \rightarrow E \qquad \frac{\begin{array}{c} \Gamma, [A] \\ \mathcal{D} \\ B \end{array}}{A \rightarrow B} \rightarrow I
 \end{array}$$

Figure 2.1: Natural deduction rules for classical propositional logic

$$\frac{A \quad A \rightarrow B}{B}$$

2.1.1 Rules

In this thesis, we primarily focus on Gentzen's version of natural deduction. For an in-depth comparison with Jaškowski's version, please refer to [37, 70].

Natural deduction comprises a set of inference rules that define the behavior of logical connectives. For each connective, there exists a corresponding elimination rule and an introduction rule, as depicted in Figure 2.1. Here, the letters A , B , and C represent logical formulas. To express that formula A can be proven using a set of formulas denoted as Γ as premises within the framework of natural deduction, we employ the notation $\Gamma \vdash_{\mathcal{NK}} A$. This notation signifies that there exists at least one valid proof within the system of natural deduction (i.e. \mathcal{NK}), employing the inference rules depicted in Figure 2.1. Historically, the notation \vdash is used in proof theory to show the existence of a proof from hypothesis (shown in the left hand side of \vdash) to conclusion (shown in the right hand side of \vdash). Moreover, we depict a natural deduction derivation \mathcal{D} for $\Gamma \vdash_{\mathcal{NK}} A$ as below:

$$\frac{\Gamma}{\mathcal{D}} \frac{}{A}$$

In natural deduction there exists several hypothesis and only one conclusion. The letters I and E mean introduction and elimination of the connectives in question which are conjunction,

$$\begin{array}{c}
 \frac{[p \vee q]^1 \quad \frac{[q]^4 \quad [q \rightarrow r]^2 \rightarrow E}{r} \rightarrow E \quad \frac{[p]^4 \quad [p \rightarrow r]^3 \rightarrow E}{r} \vee E_4}{r} \vee E_4 \\
 \frac{\frac{r}{(p \vee q) \rightarrow r} \rightarrow I_1}{(q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r)} \rightarrow I_2 \\
 \frac{(q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r)}{(p \rightarrow r) \rightarrow ((q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r))} \rightarrow I_3
 \end{array}$$

Figure 2.2: An example of a natural deduction proof

disjunction, negation, and implication. So far we have introduced the first three connectives but we have not mentioned implication or how a formula with implication is defined. This is due to the fact that the fragment of classical propositional logic that does not include implication is isomorphic to the classical propositional logic with implication and we can interpret implication as $A \rightarrow B := \overline{A} \vee B$. For simplicity and aesthetic reasons we will keep implication in the language whenever natural deduction is being discussed.¹ We also have an extra rule representing excluded middle law (or double negation elimination) which is specific to classical logic. Brackets [] around a premise show the discharging of it from the set of premise in the proof. One can observe discharging of premises in rules such as $\vee E$, $\neg I$, or ex . As an example, if a disjunction elimination is being used in the proof, we have a premise $A \vee B$, there exists a derivation \mathcal{D}_A with at least some instance of premise A and conclusion C , and there exists a derivation \mathcal{D}_B with at least some instance of premise B and conclusion C , then by discharging premises A and B from the derivation \mathcal{D}_A and \mathcal{D}_B , we can conclude C .

Example 2.1. The proof shown in Figure 2.2 is an example of a proof in Gentzen's format. It uses trees as representations of proof and is similar to sequent calculus proof trees.

The root of this proof tree shows the conclusion $((p \rightarrow r) \rightarrow ((q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r)))$ and the leaves are discharged hypothesis (p , and q , and $p \vee q$, and $p \rightarrow r$, and $q \rightarrow r$). The formulas p and q are discharged using the inference rule $\vee E$ and the rest by using $\rightarrow I$. The number on top of each bracket indicates is matched with the number next to the rule that is discharging that hypothesis.

Beyond the aesthetical aspect of introduction and elimination rules for every constant lies the attempt to realise a deeper intuition of logical constants philosophically. It was claimed that if a set of rules is intuitive for characterization of a constant, then it can express the meaning of it.

Theorem 2.2. *The conclusion of a natural deduction proof is always a semantical consequence of the premises. In other words, if $\Gamma \vdash_{\mathcal{NK}} A$, then $\Gamma \vDash A$.*

Theorem 2.3. *All semantical consequences in propositional logic are provable in natural deduction. That is, if $\Gamma \vDash A$, then $\Gamma \vdash_{\mathcal{NK}} A$.*

2.2 Sequent Calculus

Sequent calculi is the family of deductive systems using *sequents* as basic judgment and there exists many formulations of such systems. Historically, Gentzen [26] introduced natural deduction systems \mathcal{NK} and \mathcal{NJ} for classical and intuitionistic logic. However, it was not possible for him

¹In Section 8.2.2 of Chapter 8, we will show how to translate a natural deduction to combinatorial flows using the interpretation of implication from formulas in natural deduction to formulas in combinatorial flows.

to achieve a normal form for \mathcal{NK} .² To this purpose, he came up with classical sequent calculus \mathcal{LK} which uses sequents to keep track of the hypothesis formulas.

A *sequent* is shown as $\Gamma \Rightarrow \Delta$, where Γ and Δ are multi-set of formulas. The *precedent* of such sequent is Γ and its *succedent* is Δ (they correspond to hypothesis and conclusion of a proof). Gentzen's original formulation of sequent calculus showed multi-set of formulas as lists of formulas.³ Hence, Gentzen used rules to permit the exchange of order of formulas in a sequence. However, proofs look more simplified if we treat multi-set of formulas as finite multisets which means that they are lists with multiplicity and no order.

According to Gentzen [26], the denotational interpretation of a sequent in this setting is that the conjunction of precedents implies the disjunction of succedents which in the negation normal form would be treated as the formula $\bigvee \bar{\Gamma} \vee \bigvee \Delta$.

Unlike natural deduction there exists no law of excluded middle in sequent calculus. The difference between classical logic and intuitionistic logic is made by the fact that in \mathcal{LJ} the succedent is restricted to be at most one formula.

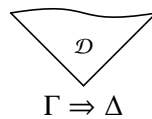
Moreover, in natural deduction, elimination rules get rid of formulas while in sequent calculus, non active (formulas that are not affected by the rule) formulas do not change or get removed. In other words, sequent calculus has *subformula property* which is defined as:

Definition 2.4. A proof system has *subformula property* if and only if all formulas in a derivation in such system are subformulas of the conclusion of that derivation.

Sequent calculus allows easier proof analysis than natural deduction. The subformula property allows easy induction over proof-steps. In sequent calculus every rule introduces a term on either side of a sequent and no elimination is used which means that the usage of each inference rule makes the sequent simpler (i.e. no new formulas are introduced). Moreover, sequent calculus is extremely symmetrical and for every connective there exists a right rule and a left rule.

In this thesis, we show a different variant of sequent calculus than the original \mathcal{LK} . The goal here is to point out the difference between the multiplicative and additive fragments of classical logic.

Definition 2.5. The sequent calculus proof \mathcal{SC} consists of the inference rules shown in Figure 2.3. These rules are used in the same proof tree format as natural deduction. A *derivation* \mathcal{D} in \mathcal{SC} is a proof tree using \mathcal{SC} inference rules with a sequent $\Gamma \Rightarrow \Delta$ in root and we denote it as $\mathcal{D}: \Gamma \vdash_{\mathcal{SC}} \Delta$, depicted below. A sequent $\Gamma \Rightarrow \Delta$ is *provable* in \mathcal{SC} , denoted as $\Gamma \vdash_{\mathcal{SC}} \Delta$, if there exists a derivation $\mathcal{D}: \Gamma \vdash_{\mathcal{SC}} \Delta$.



Note that we have shown two different instances of right rules and left rules for \vee and \wedge . These are due to the fact that we can have independent or shared context which are also called additive (for independent) and multiplicative (for shared) by the linear logic community. We will use the terms multiplicative and additive in the second part of the thesis due to the fact that this separation is between the two fragments is shown better with the combinatorial flows.

Theorem 2.6. *The root sequent of a sequent calculus proof is valid. In other words, if $\Gamma \vdash_{\mathcal{SC}} \Delta$, then $\Gamma \vDash \bigvee \Delta$.*

²In section 2.4 we go into more details about normal proofs.

³Kleene used the word "sequent" for the first time in [57] where he explains the origin of the term: "Gentzen says 'sequenz', which we translate as 'sequent', because we have already used 'sequence' for any succession of objects, where the German is 'Folge'." Nowadays, this is standard terminology.

Identity Rules:

$$\frac{}{\Gamma, A \Rightarrow \Delta, A} \text{ax} \qquad \frac{\Gamma \Rightarrow \Delta, A \quad \Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \text{cut}$$

Logical Rules:

$$\frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \overline{A}} \neg R \qquad \frac{\Gamma \Rightarrow \Delta, A}{\Gamma, \overline{A} \Rightarrow \Delta} \neg L$$

$$\frac{\Gamma \Rightarrow \Delta, A \quad \Sigma \Rightarrow \Pi, B}{\Gamma, \Sigma \Rightarrow \Delta, \Pi, A \wedge B} \wedge R_m \qquad \frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta} \wedge L_m$$

$$\frac{\Gamma \Rightarrow \Delta, A \quad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B} \wedge R_a \qquad \frac{\Gamma, A_i \Rightarrow \Delta}{\Gamma, A_1 \wedge A_2 \Rightarrow \Delta} \wedge L_a$$

$$\frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B} \vee R_m \qquad \frac{\Gamma, A \Rightarrow \Delta \quad \Sigma, B \Rightarrow \Pi}{\Gamma, \Sigma, A \vee B \Rightarrow \Delta, \Pi} \vee L_m$$

$$\frac{\Gamma \Rightarrow \Delta, A_i}{\Gamma \Rightarrow \Delta, A_1 \vee A_2} \vee R_a \qquad \frac{\Gamma, A \Rightarrow \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \vee B \Rightarrow \Delta} \vee L_a$$

Structural Rules:

$$\frac{\Gamma \Rightarrow \Delta, A, A}{\Gamma \Rightarrow \Delta, A} \text{cR} \qquad \frac{\Gamma, A, A \Rightarrow \Delta}{\Gamma, A \Rightarrow \Delta} \text{cL}$$

$$\frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A} \text{wR} \qquad \frac{\Gamma \Rightarrow \Delta}{\Gamma, A \Rightarrow \Delta} \text{wL}$$

Figure 2.3: Sequent Calculus *SC* inference rules

Theorem 2.7. *If a sequent $\Gamma \Rightarrow \Delta$ is valid in propositional logic, then it is provable in sequent calculus. That is, if $\Gamma \models \Delta$, then $\Gamma \vdash_{SC} \Delta$.*

The sequent calculus has many applications in computer science. However, its earliest application was in the proof theory of arithmetic, in Gentzen’s thesis [25, 26] and in a decisive way in [24] in the proof of the consistency of arithmetic. Moreover, Troelstra in [87] mentioned that Ketonen’s work [56] was an early analysis of cut-free proofs in sequent calculus.

2.3 Deep Inference

Both proof systems, natural deduction and sequent calculus, that we introduced so far share a common feature: they advance by manipulating the main connectives (i.e. the outermost connective on a formula). This leads us to the main subject of this section, the deep inference formalism, which deviates from the significance of the main connective. The calculus of structures [31] is the first such formalism that challenges the conventional emphasis on the main connective and permits the rewriting of formulas deep within any context.⁴ It derives its name from the fact that it blurs the distinction between sequents and formulas, offering a unique syntactic structure seen as an equivalence class of formulas, accommodating associativity, commutativity, and unit equivalences shown in (1.1). The inception of deep inference dates back to 1999-2003, starting with A. Guglielmi’s initial manuscript in 1999 [30], followed by the first refereed papers in 2001 [10, 34].

Another formalism designed and developed in deep inference is open deduction [33]. We will show open deduction formalism whenever we are addressing deep inference derivations.⁵ In essence, deep inference aims to explore the concepts of proof composition and normalization (see Section 2.4 for more details on normalization and normal proofs), including cut elimination [25, 26]. The roots of deep inference trace back to its early inspiration from linear logic [27]. Linear logic, among other various ideas, suggests that logic possesses a geometric character, and there exists a more insightful analysis of proofs concealed beneath their syntax. This notion can be given technical significance by identifying linearity within proofs. In the realm of computing, linearity can be construed as a means to manage quantity or resources (i.e. the distinction between multiplicative and additive connectives).

2.3.1 Open Deduction

As already mentioned, open deduction [33] is a deep inference formalism [34], allowing to write derivations in a way such that the same operations that are used to build formulas from atoms are also used to build derivations from inference rules.

Figure 2.4 shows the inference rules of system SKS [10], i.e. multiplicative rules (ai \downarrow axiom rule, ai \uparrow cut rule, s switch, and mix) and additive rules (aw \downarrow atomic weakening, aw \uparrow atomic co-weakening, ac \downarrow atomic contraction, ac \uparrow atomic co-contraction, and m medial). They have to be understood as rule schemas, where a can stand for an arbitrary atom and A, B, C, D for arbitrary formulas. We call an *inference system* any such set of inference rules. The rules can be composed

to *derivations*, which are defined inductively below, and which are denoted as $\mathcal{D} \parallel \mathcal{S}$ where A is the *premise* and B is the *conclusion* of the derivation \mathcal{D} , and \mathcal{S} is the set of inference rules used in \mathcal{D} .

1. A formula A is a derivation with premise A and conclusion A .

⁴The original motivation for the calculus of structures was to overcome some restrictions of the sequent calculus which could not express a certain logic with a self-dual non-commutative connective.

⁵A third approach, known as nested sequents [9], integrates deep inference principles into a conventional Gentzen formalism.

$$\begin{array}{ccc}
 \frac{t}{a \vee \bar{a}} \text{ ai}\downarrow & \frac{(A \vee B) \wedge C}{A \vee (B \wedge C)} \text{ s} & \frac{\bar{a} \wedge a}{f} \text{ ai}\uparrow \\
 \\
 \frac{a \vee a}{a} \text{ ac}\downarrow & \frac{(A \wedge C) \vee (B \wedge D)}{(A \vee B) \wedge (C \vee D)} \text{ m} & \frac{a}{a \wedge a} \text{ ac}\uparrow \\
 \\
 \frac{f}{a} \text{ aw}\downarrow & \frac{f}{t} \text{ mix} & \frac{a}{t} \text{ aw}\uparrow
 \end{array}$$

Figure 2.4: Inference rules of system SKS

2. Every inference rule $\frac{A}{B} \rho$ in \mathbf{S} , is a derivation with premise A and conclusion B .

3. If $\mathcal{D}_1 \parallel \mathbf{S}$ and $\mathcal{D}_2 \parallel \mathbf{S}$ are derivations, then the compositions $\mathcal{D}_1 \wedge \mathcal{D}_2$ and $\mathcal{D}_1 \vee \mathcal{D}_2$ are derivations
- $$\begin{array}{cc}
 A_1 & A_2 \\
 \mathcal{D}_1 \parallel \mathbf{S} & \mathcal{D}_2 \parallel \mathbf{S} \\
 B_1 & B_2
 \end{array}$$
- and denoted as $\mathcal{D}_1 \parallel \mathbf{S} \wedge \mathcal{D}_2 \parallel \mathbf{S}$ and $\mathcal{D}_1 \parallel \mathbf{S} \vee \mathcal{D}_2 \parallel \mathbf{S}$, respectively.
- $$\begin{array}{cccc}
 A_1 & A_2 & A_1 & A_2 \\
 \mathcal{D}_1 \parallel \mathbf{S} & \mathcal{D}_2 \parallel \mathbf{S} & \mathcal{D}_1 \parallel \mathbf{S} & \mathcal{D}_2 \parallel \mathbf{S} \\
 B_1 & B_2 & B_1 & B_2
 \end{array}$$

4. If \mathcal{D}_1 is a derivation with premise A_1 and conclusion B_1 , and \mathcal{D}_2 is a derivation with premise A_2 and conclusion B_2 , and $A_2 \equiv B_1$, we can compose \mathcal{D}_1 and \mathcal{D}_2 directly to $\mathcal{D}_1 \circ \mathcal{D}_2$ denoted as:

$$\begin{array}{ccc}
 \begin{array}{c} A_1 \\ \mathcal{D}_1 \parallel \mathbf{S} \\ B_1 \\ \equiv \\ A_2 \\ \mathcal{D}_2 \parallel \mathbf{S} \\ B_2 \end{array} & \text{or} & \begin{array}{c} A_1 \\ \mathcal{D}_1 \parallel \mathbf{S} \\ A_2 \\ \mathcal{D}_2 \parallel \mathbf{S} \\ B_2 \end{array} & \text{or} & \begin{array}{c} A_1 \\ \mathcal{D}_1 \parallel \mathbf{S} \\ B_1 \\ \mathcal{D}_2 \parallel \mathbf{S} \\ B_2 \end{array} & (2.1)
 \end{array}$$

Finally, we define $\mathcal{D} \parallel \mathbf{SKS}$ to be a *deep inference open deduction derivation* proving the formula $\bar{A} \vee B$. The *size* of a derivation \mathcal{D} , denoted as $|\mathcal{D}|$, is the number of deep inference rules used in \mathcal{D} .

Example 2.8. Figure 2.5 shows an example of an SKS derivation.

Theorem 2.9 (Soundness and Completeness of system SKS). *The formula $\bar{A} \vee B$ is a tautology if and only if there exists an open deduction derivation $\mathcal{D} \parallel \mathbf{SKS}$.*

Notice that in SKS we have shown weakening and contraction in atomic form. Below, we show other forms of these rules, for formula A .

$$\frac{f}{A} \text{ w}\downarrow \quad \frac{A \vee A}{A} \text{ c}\downarrow \quad \frac{A}{A \wedge A} \text{ c}\uparrow \quad \frac{A}{t} \text{ w}\uparrow \quad (2.2)$$

Theorem 2.10. *The rules $w\downarrow$, $c\downarrow$, $c\uparrow$, and $w\uparrow$ are derivable using additive rules in SKS (i.e. $aw\downarrow$, $ac\downarrow$, m , $ac\uparrow$, and $aw\uparrow$).*

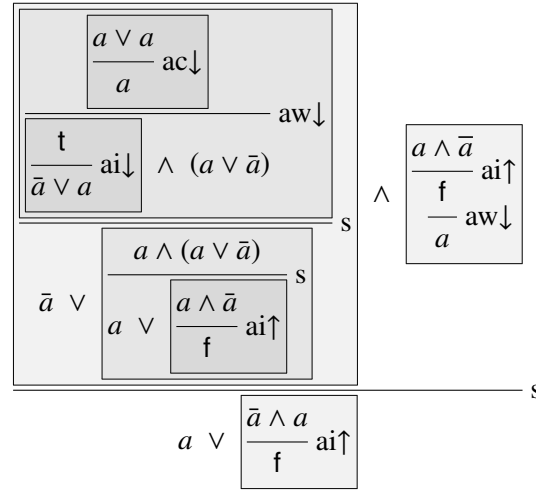


Figure 2.5: An open deduction derivation with premise $(a \vee a) \wedge a \wedge \bar{a}$ and conclusion $a \vee f$

Proof. We show the proof for $w\downarrow$ and $c\downarrow$ (for $w\uparrow$ and $c\uparrow$ is dual).

- $\frac{A \vee A}{A} c\downarrow$: We proceed by induction on A and show that it is derivable only using m and $ac\downarrow$. If $A = a$, then we use only $ac\downarrow$. If $A = B \wedge C$, then we have the following by induction hypothesis on B and C .

$$\frac{(B \wedge C) \vee (B \wedge C)}{\frac{\frac{B \vee B}{\|\{m, ac\downarrow\}} \wedge \frac{C \vee C}{\|\{m, ac\downarrow\}}}{B \wedge C} m} c\downarrow$$

If $A = B \vee C$, then we have the following by induction hypothesis on B and C .

$$\frac{(B \vee C) \vee (B \vee C)}{\frac{\frac{B \vee B}{\|\{m, ac\downarrow\}} \vee \frac{C \vee C}{\|\{m, ac\downarrow\}}}{B \vee C} \equiv} c\downarrow$$

- $\frac{f}{A} w\downarrow$: We proceed by induction on A . If $A = a$, then we use only $aw\downarrow$. If $A = B \wedge C$, then we have the following by induction hypothesis on B and C .

$$\frac{f}{\frac{\frac{f}{\|\{aw\downarrow\}} \wedge \frac{f}{\|\{aw\downarrow\}}}{B \wedge C} \equiv} w\downarrow$$

If $A = B \vee C$, then we have the following by induction hypothesis on B and C .

$$\frac{f}{\frac{\frac{f}{\|\{aw\downarrow\}} \vee \frac{f}{\|\{aw\downarrow\}}}{B \vee C} \equiv} w\downarrow$$

□

2.4 Normal Proofs

In the process of constructing a proof, we may utilize unnecessary inference rules to achieve our goal. Gentzen was interested in the idea of introducing a proof system which allows straightforward proofs, where no formulas are introduced during the proof that do not already exist in the conclusion. An example of such straightforward system is a proof system with the sub-formula property. The concept behind normal form of proofs is that by eliminating unnecessary formulas, we remove redundant proof steps—a process commonly referred to as normalization. These unnecessary formulas are also defined as *cut formulas*. In the context of natural deduction, the modus ponens rule (or implication elimination denoted as \rightarrow E) serves as an example of such an inference rule. In sequent calculus, the cut rule, depicted below, is such an example and the procedure of normalization is often referred to as *cut elimination*. Consequently, normalization is often termed as *cut normalization* or cut elimination by proof theorists.

$$\frac{\Gamma \Rightarrow \Delta, A \quad \Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \text{ cut}$$

Gentzen invented sequent calculus as an auxiliary proof system to prove normalization for natural deduction. Sequent calculus with multiple conclusions is particularly intended for Gentzen’s principal theorem (Hauptsatz, refer to [25, 26]).

It is worth emphasizing that achieving a normal form in proofs does not necessarily translate to simplicity or efficiency. In fact, non-normal proofs are often shorter and easier to comprehend for humans. Using the cut rule in a proof can be viewed the same as employing a lemma in a mathematical proof. This illustrates how significantly the length of a proof can be reduced when it is not constrained by the requirement of normalization. However, we pay the price of exponential growth on the size of the proof while using normalization. On the other hand, normal proofs are valuable since they provide conceptual simplicity, and a proof theoretical justification for deduction which leads to a fresh perspective on understanding the meaning of logics and proofs.

In subsequent developments, there was a growing concern about making proofs follow a specific pattern called *strong normalization*. In 1971, Prawitz [67] showed that any non-standard elements present in a proof can be systematically converted without any particular order being necessary. Notably, a cut can be addressed without specific order requirements. The result of this process is the assurance of a termination of normalization which leads to a unique, standardized proof as the final outcome. Prawitz’s theorem thus provided a robust framework for achieving strong normalization within proof systems, ensuring the reliability and consistency of formal logical reasoning.

During the same period as the development of strong normalization, an important concept known as the Curry-Howard correspondence emerged [45]. It essentially states that *there exists a correspondence between proof systems and models of computation*. In simpler terms, it suggests that these two families of formalisms can be viewed as equivalent and identical in their expressive power.

The development of two fundamental methodologies, namely decomposition (initially observed in [10, 35] within the context of classical logic) and splitting (credited to A. Guglielmi and first introduced in [31]), has significantly impacted the normalization in deep inference. The technique of cut-elimination via splitting has demonstrated its effectiveness across a wide spectrum of deep inference systems. For instance, we can refer to [83, 15] concerning cut-elimination for propositional linear logic. In Chapter 9, we delve deeper into more details of normal forms for combinatorial flows, offering a detailed examination of the normalization process.

CHAPTER 3

Atomic Flows

Atomic flows, initially introduced by A. Guglielmi and T. Gundersen in [32], serve as a means of representing the structural framework of a proof while disregarding its logical content. These structures, owing to their “graphical nature”, can be thought of as akin to Girard’s proof nets [19, 27] and Buss’ logical flow graphs [12]. Initially, proof nets were devised exclusively for linear logic, but various proposals have emerged to adapt them for classical logic, each with unique objectives and designs, as seen in works by Laurent [62], Robinson [75], and Lamarche and Straßburger [61]. Logical flow graphs, on the other hand, were defined specifically for classical logic, with their definition for linear logic being essentially identical, particularly in the case of the multiplicative fragment of linear logic (MLL). However, this equivalence no longer holds for classical logic, which is obtained by adding additional rules of contraction and weakening (i.e. resource management) to MLL. Atomic flows represent a development that combines the strengths of both approaches. Similar to proof nets, they simplify proof normalization by avoiding unnecessary intricacies arising from trivial rule permutations. At the same time, similar to logical flow graphs, they accurately capture the flow of information within a proof. Notably, they closely resemble a variant of proof nets discussed in [82]. Since atomic flows meticulously document every contraction and weakening applied to each atom occurrence, they can be employed to control proof size, making them relevant to the study of proof complexity (see [11]).

Atomic flows are also very similar to string diagrams for representing morphisms in monoidal categories (see [78] for a survey). However, in (classical) logic one usually finds two dual monoidal structures and not just one. Thus, atomic flows are, in spirit, more closely related to coherence graphs in monoidal closed categories [55]. Nonetheless, it should be stressed that atomic flows do not form a monoidal closed category. The following two flows are not the same, although, during the normalization process, we wish to reduce the atomic flow on the left (a cut connected to an identity) to the atomic flow on the right (a single edge).

$$\begin{array}{c} \text{---} \\ | \quad | \\ \text{---} \end{array} \quad \text{and} \quad | \tag{3.1}$$

In linear logic one can simply “pull the edges”, a process which is called *yanking*, and directly reduce the left atomic flow in (3.1) to the right one, whereas in classical logic this step might involve duplication of large parts of the proof. The main insight coming from atomic flows is that this duplication and the whole normalization process is independent from the logical content of the proof and independent from the design of the logical rules in use, as is discussed in [32].

Definition 3.1. A *type* is a finite list of atoms, denoted by p, q, r, \dots . We write $p|q$ to denote *list*

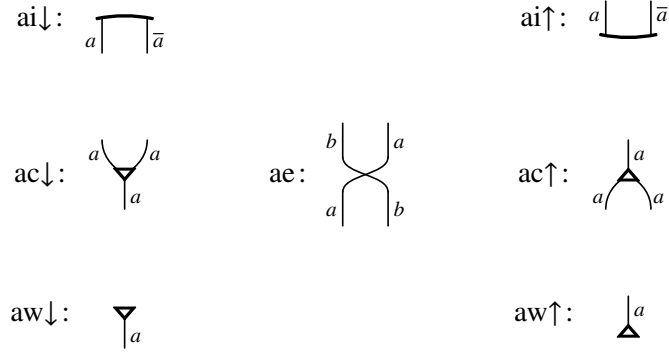
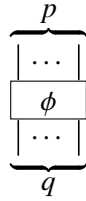


Figure 3.1: Generators for atomic flows

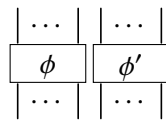
concatenation and we write $()$ to denote the *empty list*. An *atomic flow* $\phi: p \rightarrow q$ is a two-dimensional diagram [59], as shown below, where p is the *input* type and q is the *output* type. The number of edges corresponds to the lengths of the lists, and each edge is labelled by the corresponding list element.



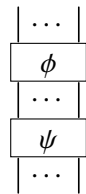
For each type q , we have the *identity atomic flow*, denoted as aid_q , shown below:



We can compose atomic flows horizontally so for two atomic flows $\phi: p \rightarrow q$ and $\phi': p' \rightarrow q'$, we can define their horizontal composition as $\phi|\phi': p|p' \rightarrow q|q'$, shown below:



We can also compose atomic flows vertically so for two atomic flows $\phi: p \rightarrow q$ and $\psi: q \rightarrow r$, we can define their vertical composition as $\phi \circ \psi: p \rightarrow r$, shown below:



The definition of identity atomic flows and compositions suggest the following equalities:

$$\phi \circ \text{aid} = \phi = \text{aid} \circ \phi \quad (\phi \circ \psi)|(\phi' \circ \psi') = (\phi|\phi') \circ (\psi|\psi') \quad (3.2)$$

Finally, Figure 3.1 shows the set of generators for atomic flows and below we explain the typing information shown for each generator:

-
- In $ai\downarrow$ (resp. $ai\uparrow$), the output edges (resp. the input edges) must have opposite atoms.
 - In $ac\downarrow$ (resp. $ac\uparrow$), all input and output edges must have same atoms.
 - In $aw\downarrow$ (resp. $aw\uparrow$), there is no restrictions.
 - In ae , the left input must have the same atom as the right output and the right input must have the same atom as the left output.

We will omit the typing of atomic flows when we show them in pictures and this information is irrelevant. The types are strictly preventing the occurrence of illegal atomic flows such as the one depicted below:



Theorem 3.2. *Let A and B be formulas, there exists an open deduction derivation $\mathcal{D} \parallel_{\text{SKS}}$ if and only if there exists an atomic flow $\phi: p \longrightarrow q$ where p and q are types achieved by listing the atom occurrences in A and B .*

Remark 3.3. We can not read back a proof from an arbitrary atomic flow and they loose too much information about the structure of the proof. That is, although atomic flows are sound and complete, they have no polynomial correctness criterion, A. Das in [20] has shown that no such criterion is possible under the assumption that factorization is hard for $P/poly$. Therefore, they do not form a proof system in the sense of Cook and Reckhow [17].

CHAPTER 4

Graphs

In this chapter we recall some definitions and theorems on graphs, e.g. cographs and RB-graphs. Graph theory, historically traced back to Euler's 1736 solution for "Seven Bridges of Königsberg" problem, is the study of mathematical structures used to model pairwise relations between objects. Graphs can be used to model many types of relations in various subjects in computer science, mathematics, social sciences, physics, and biology. Researchers have established that graph theory is an important mathematical tool in a wide variety of subjects and proof theory is not exempt from this. Cographs are an example of graphs which are being used in proof theory. They have a simple structural decomposition using disjoint union and complement operations which allows us to view them as a model for formulas. Additionally, we recall Christian Retore's handsome proof-nets [73, 74], because we use his correctness criterion in our definition of multiplicative flows in Chapter 6.

Definition 4.1. We say that a set X is \mathcal{A} -labelled if it is equipped with a *labelling function* $\ell_X: X \rightarrow \mathcal{A}$, mapping each element $x \in X$ to an atom.

As mentioned in Definition 1.1, for a formula A , we write $[A]$ to denote the set of leaves of the formula tree of A . This set is \mathcal{A} -labelled, with the labelling function $\ell_{[A]}: [A] \rightarrow \mathcal{A}$ mapping each leaf of the formula tree to the atom occurring in that position. Note that $A \equiv A'$ implies $[A] = [A']$.

Definition 4.2. An (*undirected*) graph $\mathcal{G} = \langle V_{\mathcal{G}}, R_{\mathcal{G}} \rangle$ is a set of vertices $V_{\mathcal{G}}$ accompanied with a binary edge relation $R_{\mathcal{G}} \subseteq V_{\mathcal{G}} \times V_{\mathcal{G}}$ which is irreflexive and symmetric. We omit the index \mathcal{G} in $V_{\mathcal{G}}$ and $R_{\mathcal{G}}$ when it is clear from the context. If $v, w \in V_{\mathcal{G}}$ are vertices in \mathcal{G} , we define $vR_{\mathcal{G}}w$ to be an edge in \mathcal{G} (i.e. $(v, w) \in R_{\mathcal{G}}$).

The two graphs $\mathcal{G} = \langle V_{\mathcal{G}}, R_{\mathcal{G}} \rangle$ and $\mathcal{H} = \langle V_{\mathcal{H}}, R_{\mathcal{H}} \rangle$ are *isomorphic*, denoted as $\mathcal{G} = \mathcal{H}$ if there exists a bijection $f: V_{\mathcal{G}} \rightarrow V_{\mathcal{H}}$ such that for any two vertices $v, w \in V_{\mathcal{G}}$, the edge $vR_{\mathcal{G}}w$ exists if and only the edge $f(v)R_{\mathcal{H}}f(w)$ exists.

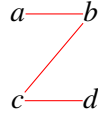
Definition 4.3. A graph $\mathcal{H} = \langle V_{\mathcal{H}}, R_{\mathcal{H}} \rangle$ is *subgraph* of \mathcal{G} if and only if $V_{\mathcal{H}} \subseteq V_{\mathcal{G}}$ and $R_{\mathcal{H}} \subseteq R_{\mathcal{G}}$. We define \mathcal{H} to be an *induced* subgraph when $R_{\mathcal{H}}$ is the maximal such set (i.e. it induces all the edges possible from $R_{\mathcal{G}}$).

In this thesis, we are only concerned with undirected graphs which have \mathcal{A} -labeled vertex set (i.e. the vertices are labeled by the set of atoms \mathcal{A}). Moreover, as we mentioned in the beginning of this chapter, we are recalling on Handsome Proof-nets [73, 74], which are bi-colored graph representations of proofs. So in the following definitions we aim at recalling on how to define a graph of a formula.

Definition 4.4. For two graphs \mathcal{G} and \mathcal{H} we define their *disjoint union* as $\mathcal{G} \uplus \mathcal{H} = \langle V_{\mathcal{G}} \uplus V_{\mathcal{H}}, R_{\mathcal{G}} \cup R_{\mathcal{H}} \rangle$ and their *join* as $\mathcal{G} \bowtie \mathcal{H} = \langle V_{\mathcal{G}} \uplus V_{\mathcal{H}}, R_{\mathcal{G}} \cup R_{\mathcal{H}} \cup \{(u, v), (v, u) \mid u \in V_{\mathcal{G}} \text{ and } v \in V_{\mathcal{H}}\} \rangle$. The

complement of \mathcal{G} is $\overline{\mathcal{G}} = \langle V_{\mathcal{G}}, \{(v, u) \mid (v, u) \notin R_{\mathcal{G}} \text{ and } v \neq u\} \rangle$. The labels are preserved for $\mathcal{G} \uplus \mathcal{H}$ and $\mathcal{G} \bowtie \mathcal{H}$ and negated in $\overline{\mathcal{G}}$.

A graph \mathcal{G} is a **cograph** if and only if it is constructed from single vertices using the operations \uplus and complement. A graph \mathcal{G} is a \mathcal{P}_4 if it is isomorphic to the graph shown below.



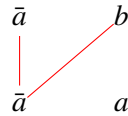
We define \mathcal{G} to be **\mathcal{P}_4 -free** if and only if there exists no induced subgraph of \mathcal{G} which is a \mathcal{P}_4 . Cographs have been discovered by several graph theorists in 70s, we can refer to H. A. Jung 1978 [54], D. Seinsche 1974 [77], and D. P. Sumner 1974 [85] as some of the early references on cographs.¹

Definition 4.5. An induced subgraph \mathcal{H} of graph \mathcal{G} is a **module** if and only if for all vertices $u \in V_{\mathcal{G}} \setminus V_{\mathcal{H}}$ the following statement holds:

$$\forall v, w \in V_{\mathcal{H}}. uR_{\mathcal{G}}v \Leftrightarrow uR_{\mathcal{H}}w \quad (4.1)$$

Definition 4.6. The **graph of** a formula A , denoted as $\mathcal{G}(A)$, is defined inductively via $\mathcal{G}(t) = \mathcal{G}(f) = \langle \emptyset, \emptyset \rangle$ (the empty graph), $\mathcal{G}(a) = \langle \{\bullet_a\}, \emptyset \rangle$ (a single vertex graph whose vertex is labeled by a), and $\mathcal{G}(B \vee C) = \mathcal{G}(B) \uplus \mathcal{G}(C)$, and $\mathcal{G}(B \wedge C) = \mathcal{G}(B) \bowtie \mathcal{G}(C)$.

Below we show an example of a graph of the formula $((\bar{a} \vee b) \wedge \bar{a}) \vee a$.



From the Definition 1.1 of formula tree and Definition 4.6 of graph of a formula, it follows that the multiset of labels in $V_{\mathcal{G}(A)}$ is equal to the multiset obtained by taking the labels of leaves in $[A]$. Moreover, using the previous definitions on cographs, \mathcal{P}_4 -free graphs, modules, we immediately arrive at some properties of cographs which are standard results which are well-known from literature and among graph-theorists:²

Proposition 4.7. Let \mathcal{G} be a cograph. Then there exists a graph \mathcal{H} which is isomorphic to \mathcal{G} and which can be constructed from single vertices using operations \uplus and \bowtie .

Proof. This property is referred to as *modular decomposition* and can be proved algorithmically, see [18, 36]. \square

Proposition 4.8. A graph \mathcal{G} is a cograph if and only if it is \mathcal{P}_4 -free.

Proof. See [64]. \square

Proposition 4.9. A graph \mathcal{G} is a cograph if and only if it is isomorphic to $\mathcal{G}(A)$ for some pure formula A .³ And for all pure formulas such as A , we have $\overline{\mathcal{G}(A)} = \mathcal{G}(A)$.

Proof. This is a direct consequence of Propositions 4.7 and 4.8. \square

¹On a side note, the mentioned representation of cographs (i.e. using disjoint union and complement operations on graphs) is used algorithmically to efficiently solve many problems such as finding the maximum clique.

²In order to maintain this thesis self-contained, we show proofs of these results in Appendix A.

³See Chapter 1 (p. 18) for the definition of pure formulas.

Proposition 4.10. *If A and B are pure formulas, then $A \equiv B$ implies that $\mathcal{G}(A) = \mathcal{G}(B)$.*

Proof. Let C_1, \dots, C_n be the sequence of formulas where $A = C_1$ and $B = C_n$ and $C_i \equiv C_{i+1}$ with the equivalence being one of the equivalence rules for classical propositional formulas (which means one can turn A to B using n equivalence relations). We proceed the proof by using induction on the pair n , which is the number of equivalence steps, and the formulas B . If $n = 1$, then A and B are the same formula so their corresponding graph isomorphic. If $n > 1$, then by induction hypothesis we have $\mathcal{G}(C_1) = \mathcal{G}(C_{n-1})$. Now for each the last equivalence step we have to look at all the equivalence relations possible. In the case that a unit is involved, it is obvious that the two graphs are isomorphic. The same holds for commutativity. If the step is concerning associativity, then we use induction hypothesis for the three smaller formulas to get isomorphic graphs for each of them. Consequently, it is easy to observe that $\mathcal{G}(C_{n-1}) = \mathcal{G}(C_n)$ which means $\mathcal{G}(C_1) = \mathcal{G}(C_n)$ which implies $\mathcal{G}(A) = \mathcal{G}(B)$. \square

Proposition 4.11. *If A and B are pure formulas which are not equivalent to units, then the isomorphism $\mathcal{G}(A) = \mathcal{G}(B)$, implies that $A \equiv B$.*

Proof. Let $\mathcal{G}(A)$ be isomorphic to $\mathcal{G}(B)$. By hypothesis, we know that $A \equiv C$ where C is a unit-free formula. So using Proposition 4.10, we have that $\mathcal{G}(A) = \mathcal{G}(C)$ which means that $\mathcal{G}(C) = \mathcal{G}(B)$. Then from the fact that there exists a modular decomposition of these graphs which is shown in [63] and C being unit free, we can deduce that B and C are equivalent as well. \square

Remark 4.12. It is important to note that there is no one to one correspondence between the equivalence of pure formulas and the isomorphism on graphs of such formulas. More specifically, the isomorphism on graphs of two pure formulas does not imply their equivalence and this is caused by the existence of units in the language and the definition of graph of a formula. An example is formulas $A = t \wedge a$ and $B = f \wedge a$ where $\mathcal{G}(A)$ and $\mathcal{G}(B)$ are single vertices labeled with the atom a thus they are isomorphic but A is not equivalent to B .

The following definition is mainly used to define correctness criterion in the second part of this thesis (Chapters 5 to 7). First, we use well-mated relations to define flowboxes and set the foundation for multiplicative and additive flows. Then, by using perfectly matched binary relations in Chapter 6, we define multiplicative flows.

Definition 4.13. A binary relation $\mathbb{B} \subseteq X \times X$ on an \mathcal{A} -labelled set X is *well-mated* if it is symmetric and we have that $x\mathbb{B}y$ implies $\ell_X(x) = \ell_X(y)$ (i.e. label of y is the negation of label of x).⁴ We say that \mathbb{B} is *perfectly matched* if it is well-mated and for every $x \in X$ there is exactly one $y \in X$ with $x\mathbb{B}y$.⁵

Remark 4.14. On a side note, we can notice some similarities between the definition of a well-mated binary relation and a graph *matching*, defined by W. T. Tutte in 1947 [88]. However, a matching cannot include edges of the graph that share vertices. We employ the well-mated relations to define flowboxes in Chapter 5.

Definition 4.15. Let \mathbb{R} and \mathbb{S} be binary relations on $X \uplus Y$ and $Y \uplus Z$ where $X \cap Z = \emptyset$. The composition of the two relations \mathbb{R} and \mathbb{S} (denoted as $\mathbb{R} \circ \mathbb{S}$) is a binary relation on $X \uplus Z$ where $(a, c) \in \mathbb{R} \circ \mathbb{S}$ if and only if there exists a sequence v_1, v_2, \dots, v_n of elements in $X \uplus Y \uplus Z$ such that

⁴Irreflexivity follows from this definition. Every symmetric irreflexive binary relation can be seen as an undirected graph, so a well-mated relation on an \mathcal{A} -labelled set is an undirected graph as well. The terminology is borrowed from Andrews' "matings" [5], the basic idea being connected dual atoms.

⁵We chose this terminology because of the resemblance to a perfect matching. It is justified because a relation \mathbb{B} is perfectly matched if and only if it is a perfect matching when seen as a graph in the sense of the previous footnote.

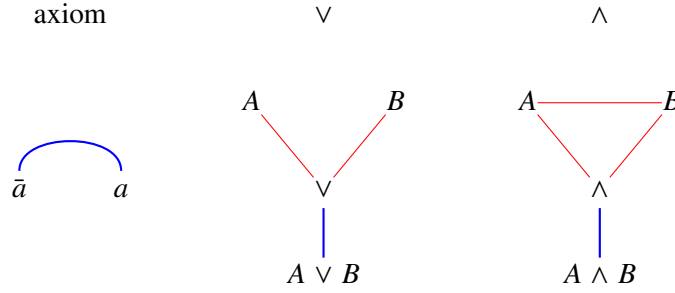


Figure 4.1: MIX R&B-proof-net connectors

$a = v_1, c = v_n$, and for every $i \in \{1, \dots, n-1\}$, we have $v_i \mathbb{R} v_{i+1}$ if i is odd and $v_i \mathbb{S} v_{i+1}$ if i is even. We define the *size* of a binary relation, denoted as $|\mathbb{R}|$, to be the number of elements in \mathbb{R} .

The following definitions are from Retoré's work on Handsome proof-nets which was firstly started by his ideas on series parallel orders during his PhD. We can refer to his 1996 paper on multiplicative proof-nets as RB-cographs [74] as the first publication on handsome proof-nets. In Chapter 6, we will show in detail the relation between multiplicative flows defined in this thesis and Handsome Proof-nets. More specifically, the correctness criterion, given in Chapter 6, for multiplicative flows is equivalent to Retoré's criterion on Handsome proof-nets introduced in his 1999 paper [73] which is an improvement on his prior work on the subject, later continued by his 2003 paper [72] which proves the same results using equivalence with Danos Reigner's criterion.

Definition 4.16. An **RB-graph** is a triple $\mathcal{G} = \langle V_{\mathcal{G}}, R_{\mathcal{G}}, \mathbb{B}_{\mathcal{G}} \rangle$ such that $\langle V_{\mathcal{G}}, R_{\mathcal{G}} \rangle$ is a labeled graph and $\mathbb{B}_{\mathcal{G}}$ is a perfectly matched binary relation on $V_{\mathcal{G}}$. An **RB-cograph** is an RB-graph \mathcal{G} where $\langle V_{\mathcal{G}}, R_{\mathcal{G}} \rangle$ is a cograph.

Definition 4.17. An **alternating elementary cycle** (\mathfrak{a} -cycle) in an RB-graph \mathcal{G} is a sequence of even number of vertices x_1, \dots, x_n (n is even) in $V_{\mathcal{G}}$, such that $x_i \neq x_j$ for all non equal indexes $i, j \in \{1, \dots, n\}$, and we have $x_1 R x_2 \mathbb{B} x_3 R x_4 \dots R x_n \mathbb{B} x_1$. A **chord** in an \mathfrak{a} -cycle c is an edge $x_i R x_j$ for $i, j \in \{1, \dots, n\}$ that does not participate in c . A **chordless \mathfrak{a} -cycle** is an \mathfrak{a} -cycle with no chords. An RB-graph \mathcal{G} is **\mathfrak{a} -acyclic** if and only if \mathcal{G} has no chordless \mathfrak{a} -cycle.

The following definition shows MIX R&B-proof-nets and we use the same notation as in [73]. We utilize this kind of proof-nets mainly to show soundness of our multiplicative flows, defined in Chapter 6, with respect to the multiplicative fragment of classical logic.

Definition 4.18. Let \mathcal{G} be an RB-cograph and let A be the formula where $\mathcal{G}(A) = \mathcal{G}$. We define a **R&B-tree** T_A to be a tree-shaped graph defined inductively using Figure 4.2 and the **connectors** shown in Figure 4.1. A **MIX R&B-proof-net** of \mathcal{G} , denoted as $P_{\mathcal{G}}$ is a R&B-tree with the addition of $\mathbb{B}_{\mathcal{G}}$ to the vertices labeled with atoms to complete the axiom connector.

We can observe that MIX R&B-proof-nets are RB-cographs by construction. We define a MIX R&B-proof-net to be **\mathfrak{a} -acyclic** if and only if it does not contain any alternating elementary cycle.

Theorem 4.19. Let A be a pure formula. For every \mathfrak{a} -acyclic RB-cograph \mathcal{G} with $\mathcal{G}(A) = \mathcal{G}$, the MIX R&B-proof-net $P_{\mathcal{G}}$ is \mathfrak{a} -acyclic.

Proof. See [73] for the proof of this theorem. □

Formula C	t or f	$a \in \mathcal{A}$	$A \vee B$	$A \wedge B$
R&B-tree T_C	\emptyset	a		

Figure 4.2: Construction of R&B-trees

Part II

Multiplicative and Additive Flows

CHAPTER 5

From Derivations to Flows

Definition 5.1. A *flowbox* is a triple $\phi = \langle A, B, \mathbb{B}_\phi \rangle$, where A and B are formulas and \mathbb{B}_ϕ is a well-mated relation on $[\bar{A}] \uplus [B]$. For flowboxes $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ and $\psi = \langle C, D, \mathbb{B}_\psi \rangle$, we define $\phi \wedge \psi = \langle A \wedge C, B \wedge D, \mathbb{B}_\phi \uplus \mathbb{B}_\psi \rangle$ and $\phi \vee \psi = \langle A \vee C, B \vee D, \mathbb{B}_\phi \uplus \mathbb{B}_\psi \rangle$, which are also flowboxes. If $\phi = \langle A, B_1, \mathbb{B}_\phi \rangle$ and $\psi = \langle B_2, C, \mathbb{B}_\psi \rangle$ are flowboxes with $B_1 \equiv B_2$, then $\phi \circ \psi = \langle A, C, \mathbb{B}_\phi \circ \mathbb{B}_\psi \rangle$ is also a flowbox.

Example 5.2. Here are some examples of flowboxes. We denote them by writing the formulas on top of each other and indicating the relation \mathbb{B} by edges between the atom occurrences.

The vertical composition of the first two flowboxes results in the third flowbox.

Definition 5.3. Two flowboxes $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ and $\psi = \langle C, D, \mathbb{B}_\psi \rangle$ are equal, denoted as $\phi = \psi$, if and only if $A \equiv C$, and $B \equiv D$, and $\mathbb{B}_\phi = \mathbb{B}_\psi$ (i.e. for every element in \mathbb{B}_ϕ there exists an equivalent of it in \mathbb{B}_ψ). We define the *size* of a flowbox $\phi = \langle A, B, \mathbb{B}_\phi \rangle$, denoted as $|\phi|$, to be $|\phi| = |A| + |B| + |\mathbb{B}_\phi|$.

Definition 5.4. For every formula A , we define an *identity* flow to be the tuple $\text{id}_A = \langle A, A, \mathbb{B}_{\text{id}} \rangle$ where every atom occurrence a in A is in relation with its corresponding occurrence in \bar{A} .¹

Moreover, we define *equivalence* flow for formulas A and B with $A \equiv B$ to be either the identity flow id_A or the identity flow id_B . Note that based on Definition 5.3, we have that id_A is equal to id_B in this case.

Proposition 5.5. Let A be a formula. The identity flow id_A is a flowbox.

Proof. Definition 5.4 shows that \mathbb{B}_{id} is well-mated and for every $x \mathbb{B}_{\text{id}} y$ we have that $\ell_{[A]}(x) = \ell_{[\bar{A}]}(y)$. Therefore id_A is a flowbox. \square

We can observe that the operation \circ on flowboxes is associative with id as unit.

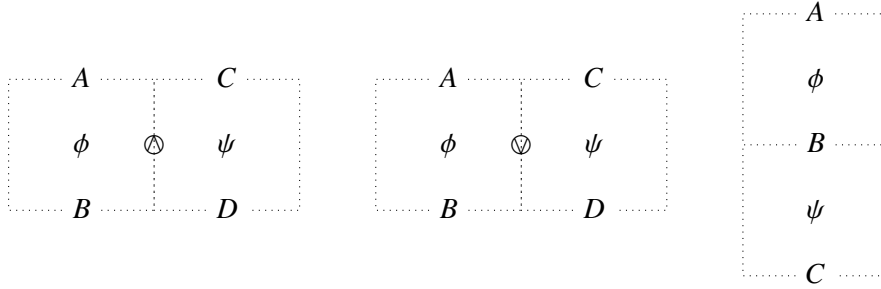
Remark 5.6. flowboxes are essentially the same as the B-nets where the well-mated relation \mathbb{B}_ϕ is a B-linking defined in Definition 2.1 of [61] and the operation \circ corresponds to the cut elimination of B-nets based on Section 5, Cut Elimination, of the same paper. As already observed in [61], the composition \circ forgets too much information about the proof and its complexity. In general, we cannot recover a derivation from a flowbox. To gain more control, we define a formal composition that keeps the structure of the proof.

¹Due to the fact that \mathbb{B}_{id} is a binary relation on $[\bar{A}] \uplus [A]$.

Definition 5.7. We define the set of *preflows* to be inductively constructed as follows:

1. A flowbox $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ is a preflow $\phi: A \vdash B$ with *premise* A and *conclusion* B .
2. If $\phi: A \vdash B$ and $\psi: C \vdash D$ are preflows, Then their *horizontal compositions* $\phi \otimes \psi: A \wedge C \vdash B \wedge D$ and $\phi \vee \psi: A \vee C \vdash B \vee D$ are preflows.
3. Let $\phi: A \vdash B_1$ and $\psi: B_2 \vdash C$ be preflows with $B_1 \equiv B_2$, then the *vertical composition* $\phi \circ \psi: A \vdash C$ is a preflow.

The *size* of a preflow ϕ , denoted as $|\phi|$, is the number of its flowboxes that are not identity or equivalence flows. We depict the horizontal and vertical composition of preflow as follows:



As suggested by this graphical notation, we consider preflows to be equivalent modulo associativity and the extra box equivalence shown below:

$$\begin{aligned}
 (\phi \otimes \psi) \otimes \xi &\equiv \phi \otimes (\psi \otimes \xi) & (\phi \vee \psi) \vee \xi &\equiv \phi \vee (\psi \vee \xi) \\
 (\phi \otimes \psi) \circ (\xi \otimes \pi) &\equiv (\phi \circ \xi) \otimes (\psi \circ \pi) & (\phi \vee \psi) \circ (\xi \vee \pi) &\equiv (\phi \circ \xi) \vee (\psi \circ \pi) \\
 \phi \circ (\psi \circ \xi) &\equiv (\phi \circ \psi) \circ \xi & &
 \end{aligned} \tag{5.2}$$

Moreover, preflows are equivalent modulo commutativity and identity flows acting as units, shown below:

$$\begin{aligned}
 \phi \otimes \psi &\equiv \psi \otimes \phi & \phi \vee \psi &\equiv \psi \vee \phi & \phi \circ \psi &\equiv \psi \circ \phi \\
 \text{id}_A \otimes \text{id}_B &\equiv \text{id}_{A \wedge B} & \text{id}_A \vee \text{id}_B &\equiv \text{id}_{A \vee B} & \text{id} \circ \phi &\equiv \phi
 \end{aligned} \tag{5.3}$$

We can now translate derivations into preflows by translating deep inference rules into flowboxes as indicated in Figure 5.1. In the case of translation from $\text{aw}\downarrow$ and $\text{aw}\uparrow$, we draw half lines for better readability.

Definition 5.8. The *translation* \mathbb{T} of a derivation \mathcal{D} in deep inference, denoted as $\mathbb{T}(\mathcal{D})$, is the preflow inductively obtained as follows. If \mathcal{D} is a formula A , then its translation is id_A . If \mathcal{D} is a rule instance, then the translation is shown in Figure 5.1. Finally, $\mathbb{T}(\mathcal{D}_1 \wedge \mathcal{D}_2) = \mathbb{T}(\mathcal{D}_1) \otimes \mathbb{T}(\mathcal{D}_2)$ and $\mathbb{T}(\mathcal{D}_1 \vee \mathcal{D}_2) = \mathbb{T}(\mathcal{D}_1) \vee \mathbb{T}(\mathcal{D}_2)$ and $\mathbb{T}(\mathcal{D}_1 \circ \mathcal{D}_2) = \mathbb{T}(\mathcal{D}_1) \circ \mathbb{T}(\mathcal{D}_2)$.

Definition 5.9. We define the *collapse* $\llbracket \phi \rrbracket$ of a preflow ϕ to be the flowbox inductively obtained via $\llbracket \phi \otimes \psi \rrbracket = \llbracket \phi \rrbracket \wedge \llbracket \psi \rrbracket$ and $\llbracket \phi \vee \psi \rrbracket = \llbracket \phi \rrbracket \vee \llbracket \psi \rrbracket$ and $\llbracket \phi \circ \psi \rrbracket = \llbracket \phi \rrbracket \circ \llbracket \psi \rrbracket$. In other words, the collapse “executes” the operations that are used to define preflows.

Example 5.10. Figure 5.3 shows an example of a deep inference derivation alongside its translation to an atomic flow and a preflow. One can observe how similar preflows and atomic flows translated from a derivation look like since they both are following the same tracing of atom occurrences as the derivation. The collapse of the preflow shown in bottom of Figure 5.3 is the rightmost flowbox in (5.1).

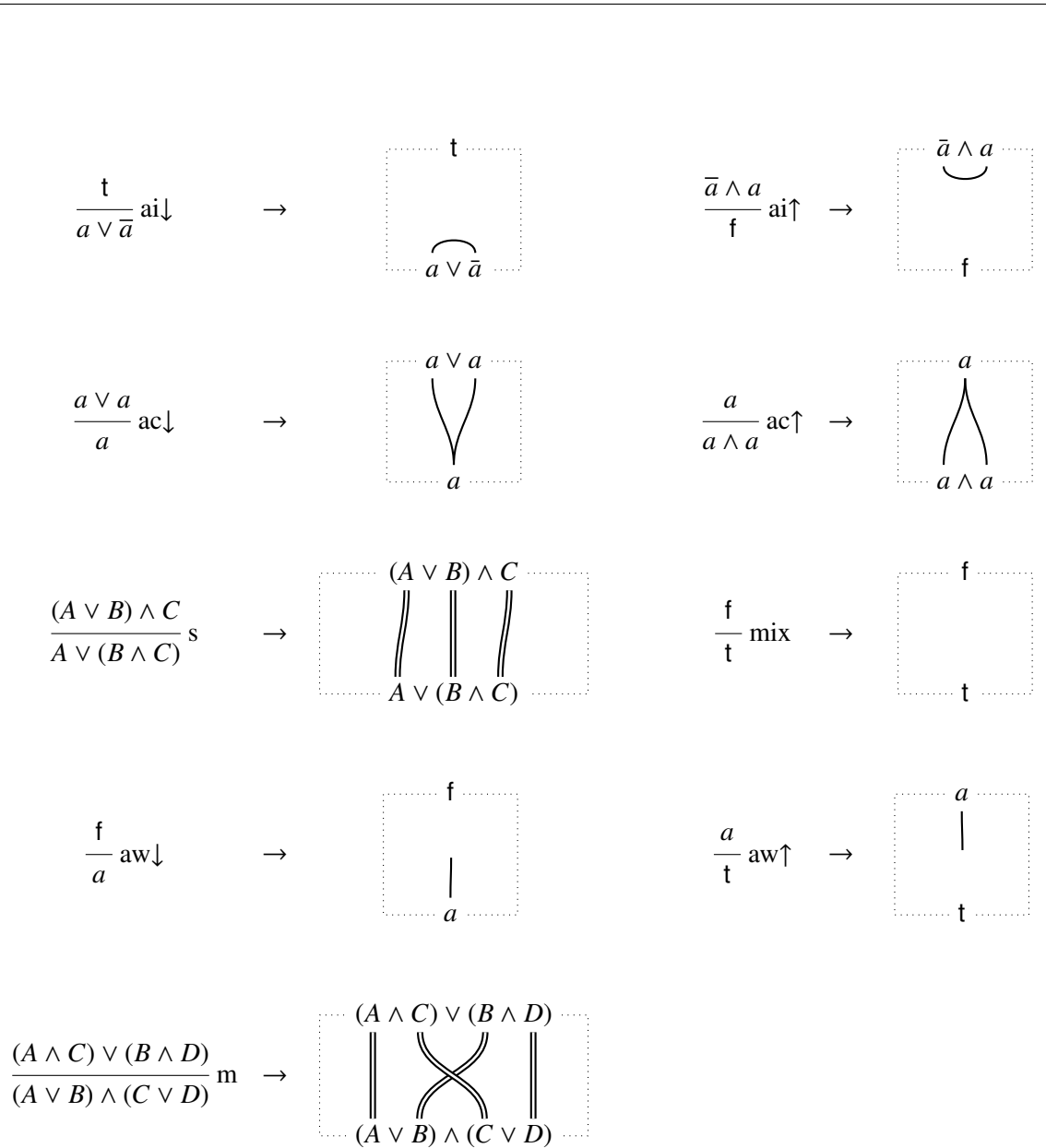


Figure 5.1: Inference rules of system SKS and their translation into flowboxes

Although preflows and atomic flows look the same, they differ in the fact that preflows keep as much information as the open deduction derivation. The collapse of a preflow forgets all such information and it does not keep the shape of the flow. Meanwhile, atomic flows can be obtained by forgetting all the formulas in a preflow and keeping the shape of the flow from hypothesis to conclusion. However, as we mentioned in Chapter 3 (p. 33), we cannot read back a proof from an atomic flow.

To better control the information that can be removed, we propose here a solution that assigns colors (blue and purple) to flowboxes and only allows to collapse them if they have the same color. The idea is that the blue flowboxes encode linear (or multiplicative) behavior, and therefore blue wires can be yanked. The purple flowboxes encode the resource management (or the additive) behavior of the proof, and therefore purple wires cannot be yanked. Figure 5.2 shows the translation of deep inference rules to flowboxes with such colors. Note that we use the same translation function from Figure 5.1 and just show the colors to differentiate between multiplicative and additive flowboxes. To indicate the color in a flowbox with empty \mathbb{B} (because of weakening), we

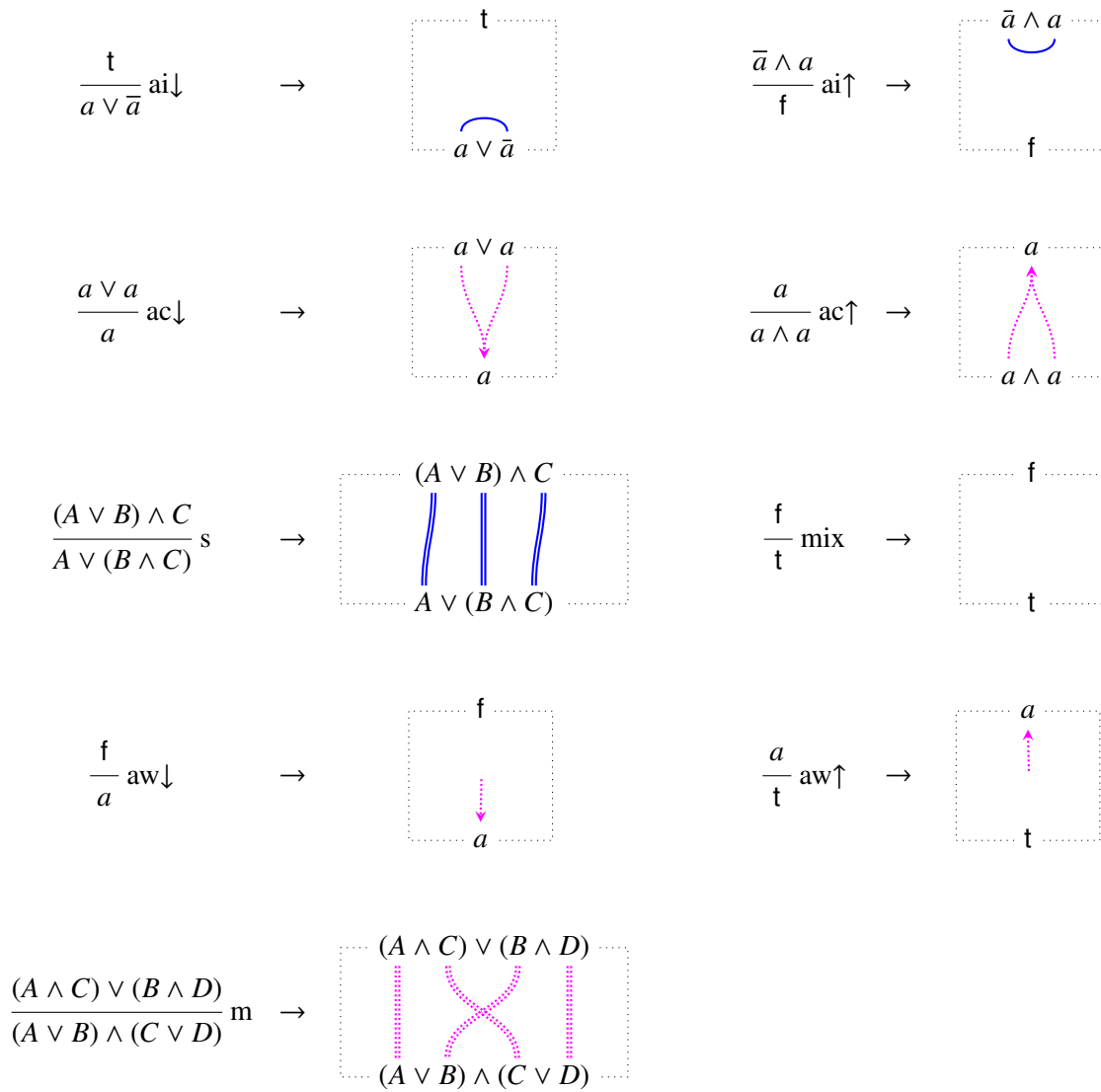


Figure 5.2: Inference rules of system SKS and their translation into colored flowboxes

sometimes draw half arrows to the atoms, as done in Figure 5.1 for the translation of inference rules $aw\downarrow$ and $aw\uparrow$.

Furthermore, both kinds of flowboxes obey different correctness criteria explained in Chapters 6, for blue, and 7, for purple. To define these correctness criterion, we heavily rely on RB-cographs from Chapter 4 (p. 38).

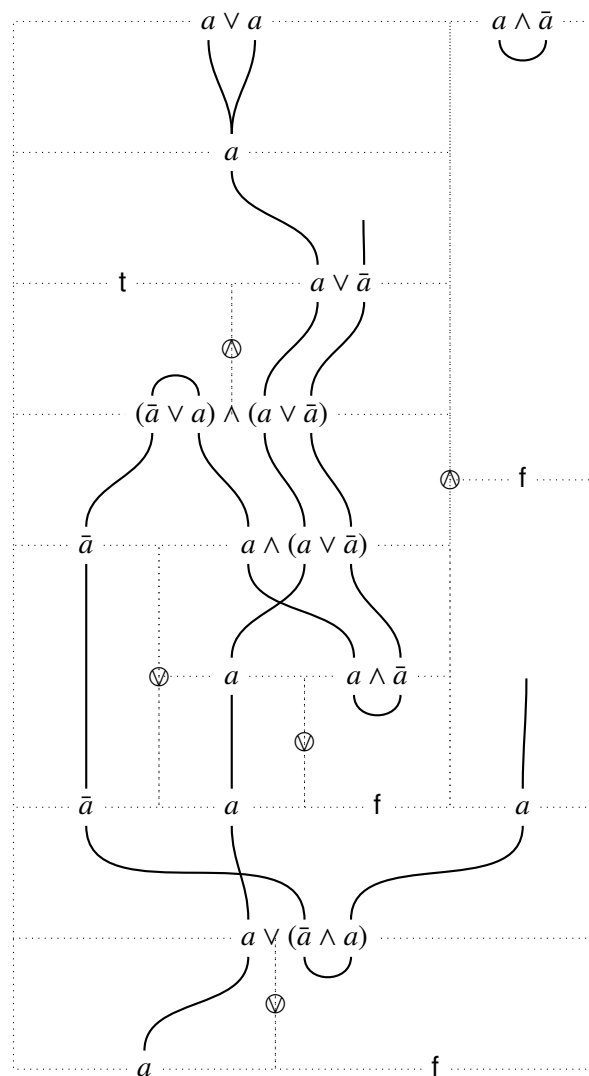
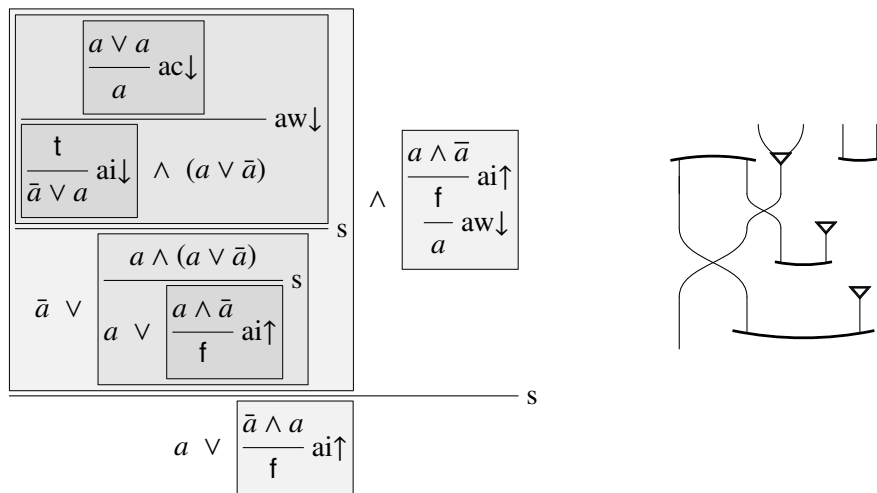


Figure 5.3: A deep inference derivation from $(a \vee a) \wedge (a \wedge \bar{a})$ to $a \vee f$ and its translation to a preflow and an atomic flow

CHAPTER 6

Multiplicative Flows

As mentioned in the previous chapter, the basic idea of the correctness criterion for combinatorial flows is the distinction between multiplicative and additive behavior of flowboxes. We begin in this chapter with the multiplicative part, which is based on Retoré's work on handsome proof nets [72].

Let $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ be a flowbox, where \mathbb{B}_ϕ is perfectly matched, defined in Definition 4.13 (p. 37). Then we can associate to ϕ the underlying RB-cograph $\mathcal{G}(\phi) = \langle V_\phi, R_\phi, \mathbb{B}_\phi \rangle$, where $\langle V_\phi, R_\phi \rangle$ is a graph $\mathcal{G}(\bar{A} \vee B)$. Observe that $V_\phi = V_{\mathcal{G}(\bar{A} \vee B)} = \lfloor \bar{A} \vee B \rfloor = \lfloor \bar{A} \rfloor \uplus \lfloor B \rfloor$, allowing us to use \mathbb{B}_ϕ in the graph. See Chapter 4 (p. 35 and 38) for the definitions of RB-cographs and graphs of formulas.

Definition 6.1. A flowbox $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ is *pure* if A and B are pure. It is *extra pure* $\bar{A} \vee B$ is also pure. A *multiplicative flow* (or *m-flow*) is an extra pure flowbox $\phi = \langle A, B, \mathbb{B}_\phi \rangle$, where \mathbb{B}_ϕ is perfectly matched and $\mathcal{G}(\phi)$ is \ae -acyclic¹, and we do not have both $A \equiv \text{t}$ and $B \equiv \text{f}$.²

Remark 6.2. Note that for each flowbox ϕ with \mathbb{B}_ϕ being perfectly matched, the graph $\mathcal{G}(\phi)$ is uniquely determined. However, the converse is not true. In particular, if A and B are unit-free formulas and $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ and $\phi' = \langle \text{t}, \bar{A} \vee B, \mathbb{B}_\phi \rangle$, then $\mathcal{G}(\phi) = \mathcal{G}(\phi')$.

Proposition 6.3. *Let A and B be unit-free formulas. Then $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ is an m-flow if and only if $\phi' = \langle \text{t}, \bar{A} \vee B, \mathbb{B}_\phi \rangle$ is an m-flow if and only if $\phi'' = \langle A \wedge \bar{B}, \text{f}, \mathbb{B}_\phi \rangle$ is an m-flow.*

Proof. Let A and B be unit-free formulas, then $\bar{A} \vee B$ and $A \wedge \bar{B}$ are unit-free formulas. We also have that $\bar{A} \vee B \equiv \text{f} \vee \bar{A} \vee B$. This means that using Proposition 4.11, we have $\mathcal{G}(\phi) = \mathcal{G}(\bar{A} \vee B) = \mathcal{G}(\text{f} \vee \bar{A} \vee B) = \mathcal{G}(\phi')$ and $\mathcal{G}(\phi) = \mathcal{G}(\bar{A} \vee B) = \mathcal{G}(\text{f} \vee (A \wedge \bar{B})) = \mathcal{G}(\phi'')$. Consequently either side of the proposition can be derived from the other because the binary relation is the same, hence the RB-cographs are equal and therefore one is \ae -acyclic if the other is. \square

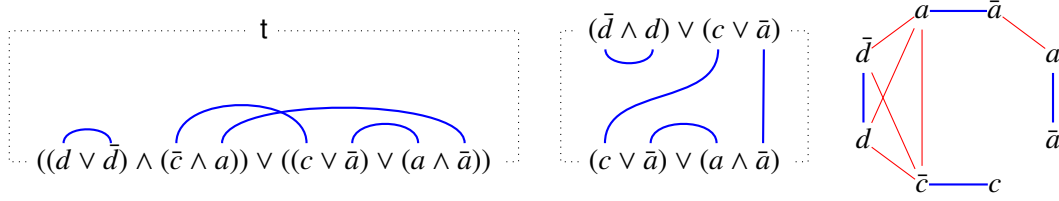
Proposition 6.4. *Let A , B , and C be pure formulas with $B \not\equiv \text{t}$. Then $\phi = \langle A, B \vee C, \mathbb{B}_\phi \rangle$ is an m-flow iff $\phi' = \langle A \wedge \bar{B}, C, \mathbb{B}_\phi \rangle$ is an m-flow.*

Proof. Similar to the previous Proposition using the graph isomorphisms $\mathcal{G}(\phi) = \mathcal{G}(\bar{A} \vee B \vee C) = \mathcal{G}(\phi')$. \square

¹We can observe that this correctness criterion is equivalent to C. Retoré's Handsome Proof Nets [74, 71], i.e every alternating elementary cycle in an RB-cograph contains at least a chord.

²Observe that we allow $A \equiv \text{t}$ or $B \equiv \text{f}$, but not both at the same time.

Example 6.5. Below on the left are two flowboxes that are m-flows and that have the same RB-cograph which is shown on the right below.



When we compose two flowboxes ϕ and ψ with the operations \wedge, \vee, \circ defined in Definition 5.1, then it is clear that the result is perfectly matched if ϕ and ψ are. From the work of Retoré [72] it follows that also the property of being \bar{x} -acyclic is preserved, shown in theorem's below. However, the result does not need to be pure even if ϕ and ψ are (see Example 6.11 below). In particular, if we have m-flows $\phi = \langle t, B, \mathbb{B}_\phi \rangle$ and $\psi = \langle C, D, \mathbb{B}_\psi \rangle$, where C is unit-free, then $\phi \vee \psi$ is not an m-flow because $t \vee C$ is not pure.

Lemma 6.6. *Let $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ and $\psi = \langle C, D, \mathbb{B}_\psi \rangle$ be m-flows with $A, B, C,$ and D as unit-free formulas, then $\phi \wedge \psi$ and $\phi \vee \psi$ defined in Definition 5.1 are m-flows.*

Proof. We show the proof that $\phi \wedge \psi$ is an m-flow and the proof for $\phi \vee \psi$ is dual. The formulas $A \wedge C$ and $B \wedge D$ are unit-free since $A, B, C,$ and D are unit-free formulas. Moreover, the relation binary relation $\mathbb{B}_{\phi \wedge \psi}$ is the disjoint union $\mathbb{B}_\phi \uplus \mathbb{B}_\psi$, based on Definition 5.1 is perfectly matched. It only remains to show that the underlying graph of $\phi \wedge \psi$ (i.e. $\mathcal{G}(\phi \wedge \psi)$) is \bar{x} -acyclic. Let c be a alternating elementary cycle in $\mathcal{G}(\phi \wedge \psi)$. If the vertices in c are all from $\mathcal{G}(\phi)$, there exists a chord for c because ϕ is an m-flow and therefore its underlying graph is \bar{x} -acyclic, resp. for $\mathcal{G}(\psi)$. Now let us assume that some vertices of c are in $\mathcal{G}(\phi)$ and the rest are from $\mathcal{G}(\psi)$. From the fact that the binary relation of $\phi \wedge \psi$ is the disjoint union of two binary relations \mathbb{B}_ϕ and \mathbb{B}_ψ , it follows that c has two different vertices from B and two different vertices from D . Hence, from the fact that there exists a conjunction between vertices of B and D , it follows that c has two chords. Therefore, the RB-cograph $\mathcal{G}(\phi \wedge \psi)$ is \bar{x} -acyclic and $\phi \wedge \psi$ is an m-flow. \square

Theorem 6.7. *Let $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ and $\psi = \langle C, D, \mathbb{B}_\psi \rangle$ be m-flows with $A, B, C,$ and D as pure formulas, then $\phi \wedge \psi$ is an m-flow if $A \wedge C$ and $B \wedge D$ are pure formulas (resp. $\phi \vee \psi$ is an m-flow if $A \vee C$ and $B \vee D$ are pure formulas).*

Proof. Lemma 6.6 is a special case of this theorem where $A \wedge C$ and $B \wedge D$ are equivalent to unit-free formulas. So we only need to show the proof for the case that at least one of these formulas is equivalent to a unit:

- If $A \wedge C \equiv f$, then we have $A \equiv f$ and $C \equiv f$. From Definition 6.1, we have that $\bar{A} \vee B$ and $\bar{C} \vee D$ should be pure formulas as well. Therefore, both B and D can not be equivalent to unit-free formulas which means $B \wedge D$ is also equivalent to a unit. So $\bar{A} \vee \bar{C} \vee (B \wedge D)$ is equivalent to a unit and therefore it is an extra pure formula which shows that $\phi \wedge \psi$ is a pure flowbox. Next by using the same proof as in Lemma 6.6, we can imply that $\phi \wedge \psi$ is an m-flow.
- If $A \wedge C \equiv t$, then we have $A \equiv t$ and $C \equiv t$. Then, the only constraint applied by Definition 6.1 is that B and D should not be equivalent to f . However, we already know that $B \wedge D$ is a pure formula which implies that $\bar{A} \vee \bar{C} \vee (B \wedge D)$ is pure. We can continue the proof same as the previous case.
- If $B \wedge C \equiv f$, then we have $B \equiv f$ and $D \equiv f$. So the proof is dual to the previous case.

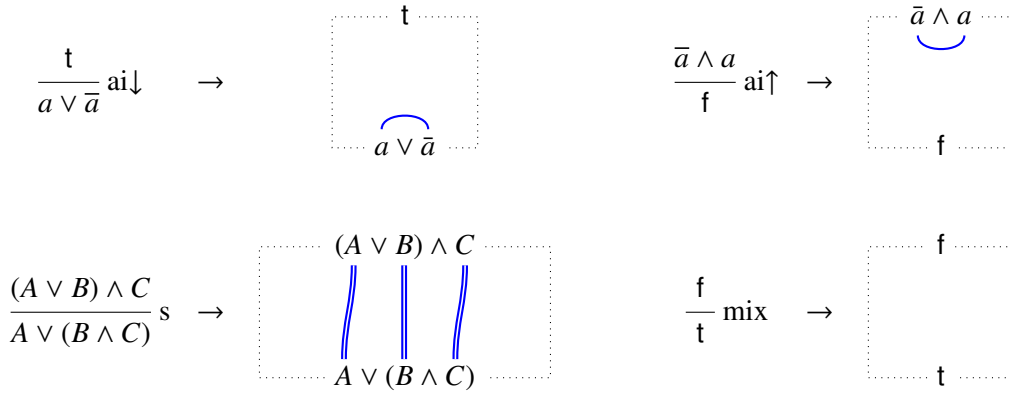


Figure 6.1: Multiplicative inference rules of system SKS and their translation into m-flows (for the full system translation, see Figure 5.2)

- If $B \wedge D \equiv t$, then we have $B \equiv t$ and $D \equiv t$. So the proof is dual to the case where $A \wedge C \equiv f$.

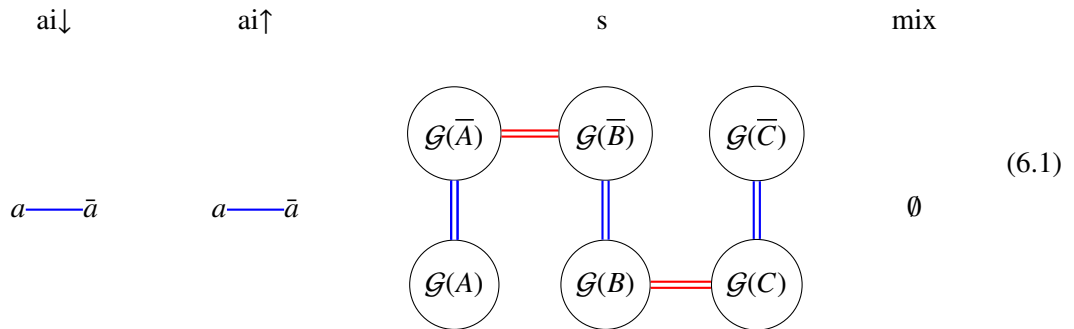
We prove that $\phi \vee \psi$ is an m-flow dually. \square

Theorem 6.8. Let $\phi = \langle A, B_1, \mathbb{B}_\phi \rangle$ and $\psi = \langle B_2, C, \mathbb{B}_\psi \rangle$ be m-flows with $A, B_1 \equiv B_2$, and C as pure formulas, then $\phi \circ \psi$ as defined in Definition 5.1 is an m-flow.

Proof. First we reduce both ϕ and ψ to MIX R&B-proof-nets using Theorem 4.19, shown in [72]. Now we can use standard cut elimination for linear logic proof nets to arrive at the result. Another more recent presentation of a proof for this theorem can be found in [48]. \square

Theorem 6.9. Let $\mathcal{D} \parallel \{ai\downarrow, ai\uparrow, s, mix\}$ be a derivation. If A and B are pure, then $\llbracket T(\mathcal{D}) \rrbracket$ is an m-flow.

Proof. First, we show that every flowbox occurring in $T(\mathcal{D})$ is an m-flow. Since \mathcal{D} only contains instances of $ai\downarrow$, $ai\uparrow$, s , and mix , it is enough to show that the translation of each rule to a flowbox, shown in Figure 6.1, is an m-flow. Below we show the underlying graph for all these flowboxes:



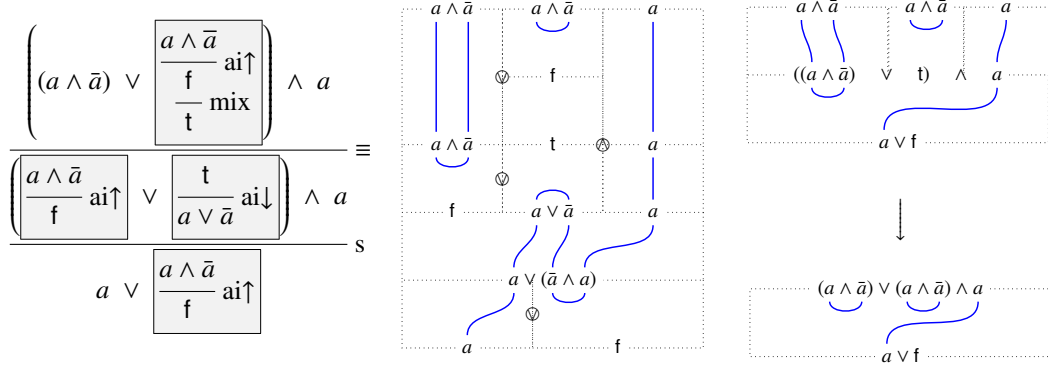
For the cases of $ai\downarrow$, $ai\uparrow$ and mix , it is trivial that the underlying graphs shown are RB-cographs (for mix , the notation \emptyset is standing for an empty graph). The underlying RB-cograph for the translation of s is shown schematically with graph of A , B , and C (and their negations) obtained inductively. As (6.1) suggests, an alternating elementary cycle can not exist in $\mathcal{G}(T(s))$. By construction shown in and Lemma 6.6 and Theorem 6.8 as well as Theorem 6.7, we have that $\mathbb{B}_{\llbracket T(\mathcal{D}) \rrbracket}$ is perfectly matched, and no alternating elementary cycle is introduced in $\mathcal{G}(\llbracket T(\mathcal{D}) \rrbracket)$ and that $\llbracket T(\mathcal{D}) \rrbracket$ is an

m-flow. Note that we also utilize the equalities (5.2) to prevent non-authorized compositions on m-flows at each step and this is possible due to the fact that A and B are pure formulas. \square

Theorem 6.10. *Let $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ be an m-flow. Then there is a derivation $\mathcal{D} \parallel_{\{ai\downarrow, ai\uparrow, s, mix\}}$ with $\llbracket T(\mathcal{D}) \rrbracket = \phi$.*

Proof. The simplest way to prove this is to by first reducing the underlying RB-cograph of ϕ to a MIX R&B-proof-net using Theorem 4.19. Next, we apply the sequentialization result from Retoré [72] to sequentialize the MIX R&B-proof-net to a sequent calculus derivation. By using the correspondence between sequent calculus and deep inference [34], we obtain the desired deep inference derivation. Moreover, a direct sequentialization from proof nets to deep inference derivations can be found in [84]. \square

Example 6.11. Below we show on the left a derivation, whose translation to preflows is shown in the middle. We can naively compose the lower half, but then would obtain flowboxes that are not pure, as shown on the upper right below. Nonetheless, the complete composition is pure and an m-flow (shown on the lower right).



This example also shows a case of yanking, see Chapter 3 (p. 31) for the explanation of yanking.

CHAPTER 7

Additive Flows

In the previous chapter we studied flows generated by multiplicative inference rules (i.e. $ai\downarrow$, $ai\uparrow$, s , and mix). In this chapter we investigate a different type of flows generated by additive inference rules (i.e. $ac\downarrow$, $aw\downarrow$, $aw\uparrow$, $ac\uparrow$, and m) mainly referred to as additive flows. To obtain any type of additive flows we consider flowboxes and explore the different possibilities of correctness criterion (namely skew fibrations [47] and additive nets [38]). Moreover, we show that checking correctness criterion is indeed of polynomial time and they are sound and correct with respect to the additive fragment of classical logic.

In Section 7.1, we present skew fibrations as well as their compositions. Next, using the definition below and properties of skew fibrations we are able to define a^\uparrow -flows and a^\downarrow -flows in Section 7.2. Moreover, we define a-flows and show that a^\uparrow -flows and a^\downarrow -flows can be composed freely to obtain an a-flow. Additionally, we show that a-flows can be decomposed into a^\uparrow -flows and a^\downarrow -flows. Note that this decomposition is not unique but the order is important here and not every order of decomposition is possible. We also show that decomposing the composition of a^\downarrow -flows and a^\uparrow -flows will not result in the same flows.

7.1 Skew Fibrations

Definition 7.1. A flowbox $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ is *function-like* if $x\mathbb{B}_\phi y$ implies that either $x \in [A]$ and $y \in [B]$ or $x \in [B]$ and $y \in [A]$, and for every $x \in [A]$, there is a unique y with $x\mathbb{B}_\phi y$. Then \mathbb{B}_ϕ defines a function $f_\phi^\downarrow: [A] \rightarrow [B]$, and we write ϕ as $\langle A, B, f_\phi^\downarrow \rangle$.

Similarly, we say that ϕ is *cofunction-like* if $x\mathbb{B}_\phi y$ implies that either $x \in [A]$ and $y \in [B]$ or $x \in [B]$ and $y \in [A]$, and for every $y \in [B]$ there is exactly one $x \in [A]$ with $x\mathbb{B}_\phi y$. In this case we interpret the function $[B] \rightarrow [A]$ defined by \mathbb{B}_ϕ as function $f_\phi^\uparrow: [B] \rightarrow [A]$, and we write ϕ as $\langle A, B, f_\phi^\uparrow \rangle$.

For specifying the desired properties of those functions, let us recall the notion of *skew fibration*.

Definition 7.2. Let \mathcal{G} and \mathcal{H} be graphs. A *graph homomorphism* $f: \mathcal{G} \rightarrow \mathcal{H}$ is a mapping $f: V_{\mathcal{G}} \rightarrow V_{\mathcal{H}}$ such that for every $v, w \in V_{\mathcal{G}}$, if $vR_{\mathcal{G}}w$ then $f(v)R_{\mathcal{H}}f(w)$, and for every $v \in V_{\mathcal{G}}$ we have $\ell_{\mathcal{G}}(v) = \ell_{\mathcal{H}}(f(v))$. A graph homomorphism $f: \mathcal{G} \rightarrow \mathcal{H}$ is *full* if for every vertex $v, w \in V_{\mathcal{G}}$ the existence of edge $f(v)R_{\mathcal{H}}f(w)$ implies $vR_{\mathcal{G}}w$. It is *injective* (denoted as $f: \mathcal{G} \rightarrow \mathcal{H}$) if for all vertices $v, w \in V_{\mathcal{G}}$, the equality $f(v) = f(w)$ implies that $v = w$. It is *surjective* (denoted as $f: \mathcal{G} \twoheadrightarrow \mathcal{H}$) if for all vertices $w \in V_{\mathcal{H}}$, there exists at least one vertex $v \in V_{\mathcal{G}}$ such that $f(v) = w$. It is *bijective* (denoted as $f: \mathcal{G} \xrightarrow{\sim} \mathcal{H}$) if is both injective and surjective.

Definition 7.3. A *skew fibration* is a graph homomorphism $f: \mathcal{G} \hookrightarrow \mathcal{H}$ where \mathcal{H} is non-empty, and for every $v \in V_{\mathcal{G}}$ and $w \in V_{\mathcal{H}}$ with $f(v)R_{\mathcal{H}}w$, there exists a vertex $z \in \mathcal{G}$ with $vR_{\mathcal{G}}z$ and $(f(z), w) \notin R_{\mathcal{H}}$.

For graph homomorphisms $f_1: \mathcal{G}_1 \rightarrow \mathcal{H}_1$ and $f_2: \mathcal{G}_2 \rightarrow \mathcal{H}_2$ we can define their horizontal compositions $f_1 + f_2: \mathcal{G}_1 \uplus \mathcal{G}_2 \rightarrow \mathcal{H}_1 \uplus \mathcal{H}_2$ and $f_1 \times f_2: \mathcal{G}_1 \bowtie \mathcal{G}_2 \rightarrow \mathcal{H}_1 \bowtie \mathcal{H}_2$, acting componentwise on the vertex set $V_{\mathcal{G}_1} \uplus V_{\mathcal{G}_2}$.

Lemma 7.4. If $f: \mathcal{G} \hookrightarrow \mathcal{H}$ and $g: \mathcal{I} \hookrightarrow \mathcal{J}$ are skew fibrations, then so is $f + g: \mathcal{G} \uplus \mathcal{H} \hookrightarrow \mathcal{I} \uplus \mathcal{J}$. Furthermore, if either $\mathcal{G}_1 \neq \emptyset \neq \mathcal{G}_2$ or $\mathcal{G}_1 = \emptyset = \mathcal{G}_2$ then $f \times g: \mathcal{G} \bowtie \mathcal{H} \hookrightarrow \mathcal{I} \bowtie \mathcal{J}$ is also a skew fibration.

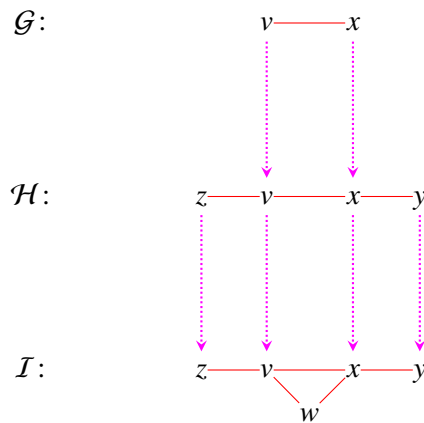
Proof. From Definitions 7.2 and 7.3, it immediately follows that $f + g$ is a skew fibration. If $\mathcal{G}_1 = \emptyset = \mathcal{G}_2$, then both f and g are empty functions and $f \times g$ is also an empty function. Hence, it is a skew fibration. If $\mathcal{G}_1 \neq \emptyset \neq \mathcal{G}_2$, then it follows from Definition 7.3 that $f \times g$ is a skew fibration. \square

Lemma 7.5. Let $f: \mathcal{G} \hookrightarrow \mathcal{H}_1 \uplus \mathcal{H}_2$ be a skew fibration. Then $f = f_1 + f_2$ for skew fibrations $f_1: \mathcal{G}_1 \hookrightarrow \mathcal{H}_1$ and $f_2: \mathcal{G}_2 \hookrightarrow \mathcal{H}_2$ with $\mathcal{G} = \mathcal{G}_1 \uplus \mathcal{G}_2$. Let $f: \mathcal{G} \hookrightarrow \mathcal{H}_1 \bowtie \mathcal{H}_2$ be a skew fibration. Then $f = f_1 \times f_2$ for skew fibrations $f_1: \mathcal{G}_1 \hookrightarrow \mathcal{H}_1$ and $f_2: \mathcal{G}_2 \hookrightarrow \mathcal{H}_2$ with $\mathcal{G} = \mathcal{G}_1 \bowtie \mathcal{G}_2$.

Proof. This follows immediately from the Definition 7.3 because a skew fibration is a graph homomorphism. \square

The vertical composition of homomorphisms $f: \mathcal{G} \rightarrow \mathcal{H}$ and $g: \mathcal{H} \rightarrow \mathcal{I}$ is defined as function composition $f \circ g: \mathcal{G} \rightarrow \mathcal{I}$ where every vertex $v \in V_{\mathcal{G}}$ is mapped to the vertex $g(f(v)) \in \mathcal{I}$. As a result, this composition is a graph homomorphism. However, we can not extend this result to skew fibrations in the general case.

Example 7.6. Below we show an example of the vertical composition of two skew fibrations $f: \mathcal{G} \hookrightarrow \mathcal{H}$ and $g: \mathcal{H} \hookrightarrow \mathcal{I}$. However, the composed homomorphism does not satisfy the skew fibration property. Note that, for simplicity of the presentation, we have labeled vertices in \mathcal{H} and \mathcal{I} with the label of their inverse image in the domain of skew fibration, if it exists and otherwise with a new label. For example, instead of writing $f(v)$ in \mathcal{H} we have written v and y is a vertex that has no inverse image in \mathcal{G} so we have used a new vertex label.



Let $v \in V_{\mathcal{G}}$ and $w \in V_{\mathcal{I}}$ and $vR_{\mathcal{I}}w$. We can observe that $f \circ g$ is not a skew fibration since there exists no vertex $u \in V_{\mathcal{G}}$ such that we have both $vR_{\mathcal{G}}u$ and $(u, w) \notin R_{\mathcal{I}}$. In fact, the only other vertex in \mathcal{G} is x and we have $xR_{\mathcal{I}}w$.

Upon closer inspection we notice that if the two graphs \mathcal{H} and \mathcal{I} are cographs, then $f \circ g$ is a skew fibration.

Lemma 7.7. *If $f: \mathcal{G} \hookrightarrow \mathcal{H}$ and $g: \mathcal{H} \hookrightarrow \mathcal{I}$ are skew fibrations and \mathcal{H} , and \mathcal{I} are cographs, then the composition $g \circ f: \mathcal{G} \hookrightarrow \mathcal{I}$ is a skew fibration.*

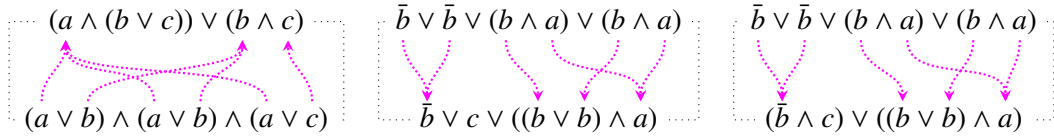
Proof. See [49] or [79] for the proof using the relation between skew fibration and additive rules in deep inference as well as the decomposition theorem on skew fibration. It is important to note that they include the weaker version of the theorem where we state that \mathcal{G} is also a cograph. However, by proving the theorem directly, proven in Theorem 3 of [65], we observe that this extra information is not necessary to the proof. We have shown the proof in Appendix B for easier access. \square

7.2 a^\downarrow -flows and a^\uparrow -flows

Definition 7.8. An a^\downarrow -flow is a function-like flowbox $\phi = \langle A, B, f_\phi^\downarrow \rangle$ where A and B are pure and $A \neq \top$ and f_ϕ^\downarrow is a skew fibration $f_\phi^\downarrow: \mathcal{G}(A) \rightarrow \mathcal{G}(B)$. Similarly, an a^\uparrow -flow is a cofunction-like flowbox $\phi = \langle C, D, f_\phi^\uparrow \rangle$ where C and D are pure and $D \neq \bot$ and f_ϕ^\uparrow is a skew fibration $f_\phi^\uparrow: \mathcal{G}(\bar{D}) \rightarrow \mathcal{G}(\bar{C})$.

An important observation is that checking correctness of a^\downarrow -flows and a^\uparrow -flows is polynomial time since one only needs to check that skew fibration property holds for each edge in the codomain of the graph homomorphism. Moreover, we will show the soundness and completeness for a^\downarrow -flows and a^\uparrow -flows with respect to the additive fragment of classical logic.

Example 7.9. When drawing flowboxes that are a^\downarrow -flows or a^\uparrow -flows, we use purple arrows to indicate the direction of the functions. Below are three examples. The first one is an a^\uparrow -flow. The second one is an a^\downarrow -flow. The third example is not a skew fibration because in the lower graph there is an edge between the \bar{b} and the c , violating the skew fibration property.



When composing flowboxes that are a^\downarrow -flows and a^\uparrow -flows, we are in a similar situation as for multiplicative flows in the previous section. In the general case, the horizontal composition of skew fibrations is not a skew fibration (see side condition in Lemma 7.4) so the side condition in the following theorem is important. The vertical composition follows from Lemma 7.7.

Theorem 7.10. *Let $\phi = \langle A, B, f_\phi^\downarrow \rangle$ and $\psi = \langle C, D, f_\psi^\downarrow \rangle$ be a^\downarrow -flows. If $A \wedge C$ and $B \wedge D$ are pure formulas, then $\phi \wedge \psi = \langle A \wedge C, B \wedge D, f_\phi^\downarrow \times f_\psi^\downarrow \rangle$ is an a^\downarrow -flow. If $A \vee C$ and $B \vee D$ are pure formulas, then $\phi \vee \psi = \langle A \vee C, B \vee D, f_\phi^\downarrow + f_\psi^\downarrow \rangle$ is an a^\downarrow -flow. The same statements hold for ϕ and ψ as a^\uparrow -flows.*

Proof. Let ϕ and ψ be a^\downarrow -flows. Then, by Definition 7.16, f_ϕ^\downarrow and f_ψ^\downarrow are skew fibrations. We conclude that $f_\phi^\downarrow \times f_\psi^\downarrow$ is a skew fibration using the fact that $A \wedge C$ and $B \wedge D$ are pure formulas and Lemma 7.4 therefore $\phi \wedge \psi$ is an a^\downarrow -flow. Similarly, using the fact that $A \vee C$ and $B \vee D$ are pure formulas accompanied by Lemma 7.5, we conclude that $f_\phi^\downarrow + f_\psi^\downarrow$ is a skew fibration, therefore $\phi \vee \psi$ is an a^\downarrow -flow. If ϕ and ψ are a^\uparrow -flows, we use the same arguments with more special care about skew fibration starting from the graph of conclusion and going to the graph of hypothesis in an a^\uparrow -flow. \square

Theorem 7.11. Let $\phi = \langle A, B, f_\phi^\downarrow \rangle$ and $\psi = \langle B, C, f_\psi^\downarrow \rangle$ be a^\downarrow -flows. Then, $\phi \circ \psi = \langle A, C, f_\psi^\downarrow \circ f_\phi^\downarrow \rangle$ is an a^\downarrow -flow and it is obtained by composing f_ϕ^\downarrow and f_ψ^\downarrow . Let $\phi = \langle A, B, f_\phi^\uparrow \rangle$ and $\psi = \langle B, C, f_\psi^\uparrow \rangle$ be a^\uparrow -flows. Then, $\phi \circ \psi = \langle A, C, f_\psi^\uparrow \circ f_\phi^\uparrow \rangle$ is an a^\uparrow -flow.

Proof. This follows directly from Lemma 7.7. \square

Using the two previous theorems, we have an analogous result as for m-flows shown in the theorems below:

Theorem 7.12. Let $\mathcal{D} \parallel_{\{aw\downarrow, ac\downarrow, m\}}$ be a derivation. If A and B are pure, then $\llbracket T(\mathcal{D}) \rrbracket$ is an a^\downarrow -flow.

Dually, if A and B are pure in $\mathcal{D} \parallel_{\{aw\uparrow, ac\uparrow, m\}}$ then $\llbracket T(\mathcal{D}) \rrbracket$ is an a^\uparrow -flow.

Proof. We prove this theorem by induction on the derivation \mathcal{D} . In the base case, a derivation is either one formula which means that $T(\mathcal{D})$ is an identity flow therefore it is an a^\downarrow -flow as well, or an instance of an additive inference rule (i.e. one of the three rules $aw\downarrow$, $ac\downarrow$, and m) for which $T(\mathcal{D})$ is shown in Figure 5.1 and these flowboxes are indeed a^\downarrow -flows. This proof is dual for a^\uparrow -flows and additive inference rules $aw\uparrow$, $ac\uparrow$, and m . Moreover, we should note that since A and B are pure formulas and down rules have a certain shape (specially the non existence of t in this rules), formulas inside the derivation are pure as well. Now, we proceed to the inductive cases.

If $\mathcal{D} = \mathcal{D}_1 \circ \mathcal{D}_2$ where \mathcal{D}_1 and \mathcal{D}_2 are derivations $\mathcal{D}_1 \parallel_{\{aw\downarrow, ac\downarrow, m\}}$ and $\mathcal{D}_2 \parallel_{\{aw\downarrow, ac\downarrow, m\}}$ with C as a

pure formula, then by induction hypothesis $\llbracket T(\mathcal{D}_1) \rrbracket$ and $\llbracket T(\mathcal{D}_2) \rrbracket$ are a^\downarrow -flows. Consequently, by Theorem 7.11, the flowbox $\llbracket T(\mathcal{D}_1 \circ \mathcal{D}_2) \rrbracket$ is an a^\downarrow -flow. If $\mathcal{D} = \mathcal{D}_1 \wedge \mathcal{D}_2$ is a derivation where \mathcal{D}_1

and \mathcal{D}_2 are derivations $\mathcal{D}_1 \parallel_{\{aw\downarrow, ac\downarrow, m\}}$ and $\mathcal{D}_2 \parallel_{\{aw\downarrow, ac\downarrow, m\}}$ with A_1, A_2, B_1 , and B_2 as pure formulas, then by induction hypothesis $\llbracket T(\mathcal{D}_1) \rrbracket$ and $\llbracket T(\mathcal{D}_2) \rrbracket$ are a^\downarrow -flows which then implies that $\llbracket T(\mathcal{D}) \rrbracket$ is an a^\downarrow -flow (using Theorem 7.10 and the fact that $A = A_1 \wedge A_2$ and $B = B_1 \wedge B_2$ are pure formulas). The same argument applies to $\mathcal{D} = \mathcal{D}_1 \vee \mathcal{D}_2$. The proof for derivations with up rules $\{aw\uparrow, ac\uparrow, m\}$ is dual. \square

As before, we also have the converse.

Theorem 7.13. Let $\phi = \langle A, B, f_\phi^\downarrow \rangle$ be an a^\downarrow -flow. Then there is a derivation $\mathcal{D} \parallel_{\{aw\downarrow, ac\downarrow, m\}}$ with $\llbracket T(\mathcal{D}) \rrbracket = \phi$. Dually, for every a^\uparrow -flow $\psi = \langle A, B, f_\psi^\uparrow \rangle$ we have $\mathcal{D} \parallel_{\{aw\uparrow, ac\uparrow, m\}}$ with $\llbracket T(\mathcal{D}) \rrbracket = \psi$.

Proof. This follows from one direction of Theorem 7.8 in [79] or Theorem 8.3 in [80]. We can show the proof for ϕ and for ψ it will be dual. By Definition 7.8, the function f_ϕ^\downarrow is a skew fibration $f_\phi^\downarrow : \mathcal{G}(A) \rightarrow \mathcal{G}(B)$. If f_ϕ^\downarrow is an empty function, it means that ϕ is a weakening so we can use a single $w\downarrow$ to prove B from A . Now using Theorem 2.10, we obtain the desired derivation. We continue for a non-empty function. Let us call a vertex in $V_{\mathcal{G}(A)}$ good if it is in the image of f_ϕ^\downarrow , and otherwise bad. For every bad vertex $v \in V_{\mathcal{G}(A)}$, by skew fibration property, there exists at least a good vertex that does not have an edge to v . Since there is at least one good vertex (f_ϕ^\downarrow is non-empty), we have for every bad vertex a a subformula $C \vee D$ in B such that (i) a is inside D , (ii) C contains a good vertex, and furthermore (iii) all vertices in D are bad. So we can apply $w\downarrow$ to delete D . Let B_0 be the formula obtained from B by repeating this process until no bad vertices remain. Then, for each atom a define n_a be the number of vertices in $V_{\mathcal{G}(A)}$ that f_ϕ^\downarrow maps

to a , and let B_1 be the formula obtained from B_0 by replacing each a by $a \vee \dots \vee a$ where there are n_a copies of a . Then there is a derivation from B_1 to B_0 the $\text{ac}\downarrow$ rule. We can define the skew fibration $f^\downarrow: \mathcal{G}(A) \rightarrow \mathcal{G}(B_1)$ which takes each vertex that f^\downarrow maps to a to one for the new copies of a . Now, we can observe that f^\downarrow is a bijective skew fibration. It follows from [79], Theorem 5.1, that there is a derivation from A to B_1 using only m rule. Alternatively, every skew fibration can be decomposed into a full injective, a full surjective, and a bijective skew fibrations which is then shown in [49] and [79] correspond to derivations with weakenings, contractins, and medials. Therefore, using the previous theorem, we can also observe that the translation of such derivations is in fact the starting flow. \square

7.3 A-flows

In this section we define a-flows, firstly introduced in [38] as additive nets, which are a second approach to show additive flows as we mentioned in the beginning of this chapter. Moreover, we show that unlike a^\downarrow -flows and a^\uparrow -flows, a-flows represent the whole additive fragment of classical logic. From which we can observe that the result of free composition of a^\downarrow -flows and a^\uparrow -flows is an a-flows.

We show the completeness and soundness for a-flows using additive deep inference rules $\text{SKS}_a = \{\text{aw}\downarrow, \text{ac}\downarrow, \text{m}, \text{ac}\uparrow, \text{aw}\uparrow\}$. We introduce two correctness criteria *coalescence* and *discreteness (pruning)* for a-flows. We will observe that these two correctness criterion are different in time complexity. Coalescence is a polynomial time criterion whereas pruning is exponential. However, we will show that satisfying each criterion implies the satisfaction of the other.

We continue this section by first building the foundation to the definition of an a-flow. Let A and B be propositional formulas. A **link** $C \bullet \bullet D$ on $\bar{A} \vee B$ is a tuple connecting a subformula C of A to a subformula D of B . An **atomic link** $a \bullet \bullet a$ is when a link is on atom a , i.e. $C = a = D$. A **linking**, denoted as $\mathbb{B}: A \text{ --- } B$ is a binary relation on the formula tree $T(\bar{A} \vee B)$ where each element of the relation is a link. As an example, the well-mated binary relation in a flowbox $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ is a linking if for every element $(a, b), (b, a) \in \mathbb{B}_\phi$, we have that a is an atom occurrence in A and b is an atom occurrence in B . An **axiom linking** is a linking with every link as an atomic link.

Definition 7.14. A **pruning** r ¹ for a pure formula A is function choosing one of the two conjuncts for each conjunction in the minimal pure formula $A' \equiv A$, i.e. for a subformula $B \wedge C$ of A' , we have that either $r(B \wedge C) = B$ or $r(B \wedge C) = C$. A subformula C of A is **retained** by r if C is equivalent to a minimal pure fomrula C' which is a subformula of the formula B_i (subformula of A') where $r(B_0 \wedge B_1) = B_i$ with $i = \{0, 1\}$. A subformula C is **discarded** if it is not retained by r . Dual to the pruning, a **co-pruning** for A chooses on disjunctions. A **pruning for formula sequent** \bar{A}, B is a pair $r = (r_A, r_B)$ where r_A is a co-pruning for A and r_B is a pruning for B . A link $C \bullet \bullet D$ in the linking $\mathbb{B}: A \text{ --- } B$ is retained by r if C is retained by r_A and D is retained by r_B .

Definition 7.15. A linking $\mathbb{B}: A \text{ --- } B$ is **discrete** if either of the following statements holds:

- Both formulas A and B are equivalent to some unit-free formula and every pruning for the formula sequent \bar{A}, B retains exactly one link $C \bullet \bullet D \in \mathbb{B}$.
- Either $A \equiv \text{f}$, or $B \equiv \text{t}$ (not both) and the linking is empty.

Definition 7.16. We define an **a-flow** $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ as a flowbox where A and B are pure formulas and the well-mated binary relation \mathbb{B}_ϕ is a discrete linking $\mathbb{B}_\phi: A \text{ --- } B$.

Discreteness is the first correctness criterion for a-flows. However, as we mentioned, we can observe that it takes exponential time to check whether a flowbox is discrete and this is due to the fact that pruning needs to check all the possible subformula in hypothesis and conclusion.

¹Pruning has first been defined by Dominic J. D. Hughes and Rob J. van Glabbeek in [51] as resolution

In the following definition we show the second correctness criterion called coalescence criterion which is a simple rewriting procedure with rules that imitate contraction and weakening in SKS and formula equivalences. We will show that coalescence is confluent on any flow. Moreover, we will show the relation between coalescence and discreteness.

Definition 7.17. *Coalescence* is a rewriting system on linkings and the steps are as follows (also shown in Figure 7.1). Without loss of generality, let $\mathbb{B} : X \multimap Y$ be a linking, then we can apply the following rewriting steps:

- Let A be a subformula of X and $B \vee C$ be a subformula of Y . If one of the links $A \multimap B$ or $A \multimap C$ exists, then we replace it with the link $A \multimap B \vee C$.
- Let A be a subformula of X and $B \wedge C$ be a subformula of Y . If both of the links $A \multimap B$ and $A \multimap C$ exist, then replace the two links with one link $A \multimap B \wedge C$.
- Let $A \vee B$ be a subformula of X and C be a subformula of Y . If both of the links $A \multimap C$ and $B \multimap C$ exist, then we replace both links with one link $A \vee B \multimap C$.
- Let $A \wedge B$ be a subformula of X and C be a subformula of Y . If one of the links $A \multimap C$ or $B \multimap C$ exists, then we replace it with the link $A \wedge B \multimap C$.
- Let $A \wedge t$ (resp. $t \wedge A$) be a subformula of X and B be a subformula of Y . If the link $A \multimap B$ exists, then we replace it with the link $A \wedge t \multimap B$ (resp. $t \wedge A \multimap B$).
- Let $A \vee f$ (resp. $f \vee A$) be a subformula of X and B be a subformula of Y . If the link $A \multimap B$ exists, then we replace it with the link $A \vee f \multimap B$ (resp. $f \vee A \multimap B$).
- Let A be a subformula of X and $B \wedge t$ (resp. $t \wedge B$) be a subformula of Y . If the link $A \multimap B$ exists, then we replace it with the link $A \multimap B \wedge t$ (resp. $A \multimap t \wedge B$).
- Let A be a subformula of X and $B \vee f$ (resp. $f \vee B$) be a subformula of Y . If the link $A \multimap B$ exists, then we replace it with the link $A \multimap B \vee f$ (resp. $A \multimap f \vee B$).

A linking $\mathbb{B} : A \multimap B$, with A and B equivalent to unit-free formulas, *weakly coalesces* if there is a sequence of coalescence steps starting with \mathbb{B} and ending in a single link $A \multimap B$. Moreover, in the case that either $A \equiv f$ or $B \equiv t$ (and not at the same time), we say that \mathbb{B} is empty and weakly coalesces (there is no link to apply coalescence steps on).

It *strongly coalesces* if any non-empty coalescence sequence terminates in a single link $A \multimap B$. It is also important to note that if a $\mathbb{B} : A \multimap B$ coalesces, the number of coalescence steps is of polynomial order with respect to the sum of the number of links and rank of A and B (i.e. $|A| + |B|$).

Lemma 7.18. *A linking $\mathbb{B} : A \multimap B$ that weakly coalesces, is discrete.*

Proof. If at least one of the formulas A and B is equivalent to a unit, then we know that the linking is empty therefore the second statement on Definition 7.15 is holding and \mathbb{B} is discrete.

Next, let us assume both A and B are equivalent to unit-free formulas. Let c_1, \dots, c_n be the sequence of coalescence steps that ends with a single link $A \multimap B$. We do induction on n which is the number of coalescence steps.

1. If $n = 0$, then no rewriting step is taken which means that there exists exactly one link and it is the link $A \multimap B$, therefore \mathbb{B} is discrete.
2. if $n = 1$, we show the theorem for two of the coalescence rewriting steps, $\rightsquigarrow_{c\uparrow}$ and $\rightsquigarrow_{w\uparrow}$. For the other two steps $\rightsquigarrow_{c\downarrow}$ and $\rightsquigarrow_{w\downarrow}$, the proof is dual. Also, the $\rightsquigarrow_{\equiv}$ steps aid us in using the equivalences on formulas (note that in discreteness we check the criterion on a

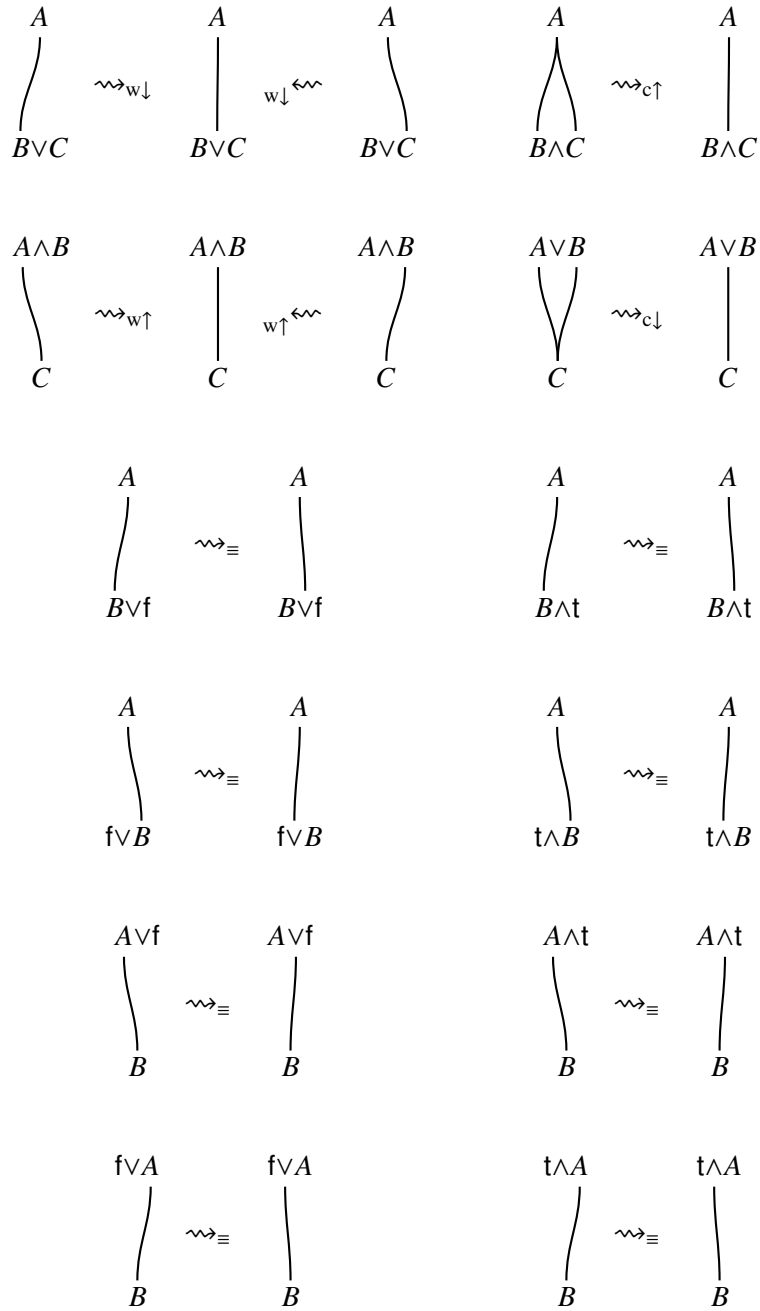
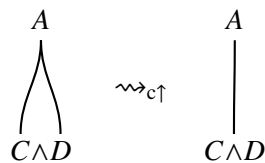


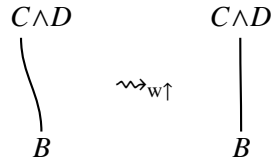
Figure 7.1: Coalescence Rules

unit-free equivalent formula). Let us assume that $c_1 = \rightsquigarrow_{c\uparrow}$ which is shown below (note that $B = C \wedge D$):



With this assumption, we can conclude that the linking is discrete since there exists only the two links $A \bullet \rightarrow C$ and $A \bullet \rightarrow D$ and any pruning r retains either C or D (and not both)

from the conclusion. Now, let us assume $c_1 = \rightsquigarrow_{w\uparrow}$ depicted below (with $A = C \wedge D$):



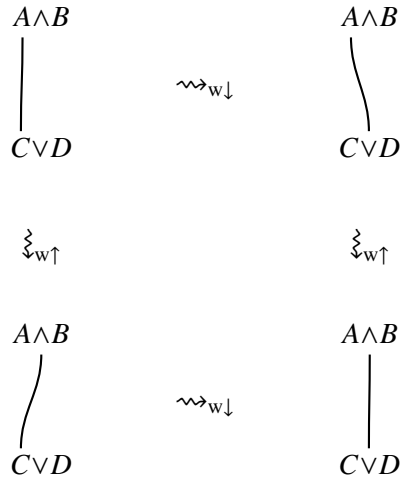
Any pruning will retain both C and D from the hypothesis which implies that the linking is discrete because only the link $C \bullet \bullet B$ is rewriting to the link $A \bullet \bullet B$ in the step c_1 .

3. If $n > 1$, using induction hypothesis on the $n - 1$ last steps as well as similar arguments to the previous case we can deduce that \mathbb{B} is discrete. □

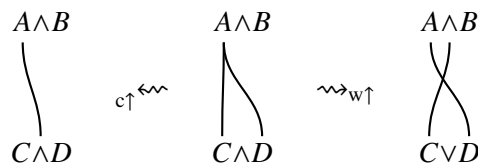
Lemma 7.19. *Coalescence is confluent on discrete linkings.*

Proof. We proceed the proof of this Lemma the same as in [40] since the coalesce steps $\rightsquigarrow_{\equiv}$ just help us to apply equivalence on formulas.

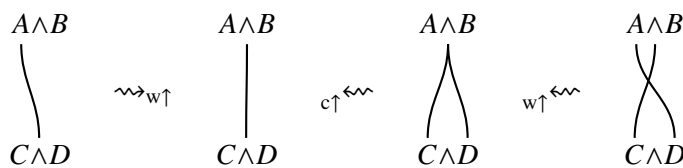
Note that every link can be replaced by at most two rewriting steps which means that a critical point in coalescence steps is when both rewriting steps can happen. We call these two possible rewriting steps a **critical pair**. For premise $A \wedge B$ and conclusion $C \vee D$, a critical pair is as follows and they both have the following immediate rewriting steps which result in the same link.



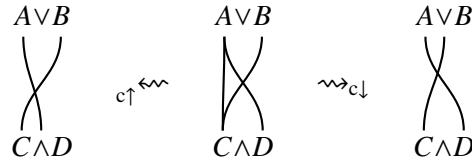
The links $A \bullet \bullet D$, and $B \bullet \bullet C$, and $B \bullet \bullet D$ have a similar proof for the same premise and conclusion. For premise $A \wedge B$ and conclusion $C \wedge D$ a critical pair is:



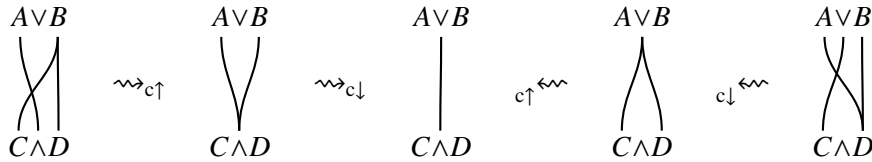
And both have the following immediate rewriting step which results in the same link.



The proof for a critical pair where the starting links are $B \bullet \bullet C$ and $B \bullet \bullet D$, is the same. The critical pairs with premise $A \vee B$ and conclusion $C \vee D$ can be proven using a dual argument. For the premise $A \vee B$ and conclusion $C \wedge D$, we have the following critical pair:



In order to continue coalescence steps in this critical pair, we need to show that some further coalescence steps will provide us with a link $B \bullet \bullet D$. Let us assume that the start configuration of coalescence is the linking $\mathbb{B} : X - Y$. Since coalescence implies discreteness, this linking is discrete, and any pruning r of \overline{X}, Y retaining \overline{B}, D must retain exactly one link. This link must lie within \overline{B}, D , since by changing r to choose A rather than B in X , or C rather than D in Y , one of $A \bullet \bullet C$, $A \bullet \bullet D$, or $B \bullet \bullet C$, is retained. Hence, by discreteness, no other link than the ones existing in $\overline{A \vee B}, C \wedge D$ may be retained by r . The links in \overline{B}, D form a discrete linking, which by Lemma 7.18 coalesces to a single link $B \bullet \bullet D$. Now using the present link $B \bullet \bullet D$, the critical pair have the following rewriting steps resulting in the same link.



The other cases where either of the links $A \bullet \bullet C$, or $A \bullet \bullet D$, or $B \bullet \bullet C$ is not present (while $B \bullet \bullet D$ is already present from the start) follow a similar proof. \square

Theorem 7.20. *Let $\mathbb{B} : A \vdash B$ be a linking. The following are equivalent:*

1. \mathbb{B} is discrete.
2. \mathbb{B} weakly coalesces.
3. \mathbb{B} strongly coalesces.

Proof.

- (1 \Rightarrow 3) Lemma 7.19 implies that any coalescence sequence that ends in one link, provides the link $A \bullet \bullet B$. Now we only have to show that there is exists a possible coalescence step to be applied unless we have only the link $A \bullet \bullet B$. By contradiction let us assume that after applying some coalescence steps, we arrive at the linking $\mathbb{B}' : A - B$ with no next possible coalescence step. This means that for every pattern of subformula that are the same as in $w\downarrow$ and $w\uparrow$, there exists no link, and for every pattern the same as $c\uparrow$ and $c\downarrow$, there exists less than two links. We can show immediately that we have a pruning that chooses the subformulas that have no link. Hence, this pruning is not retaining any links and \mathbb{B}' is not discrete. However, in proof of Theorem 7.18, we have shown that coalescence steps keep discreteness. We started from a discrete linking \mathbb{B} so we have contradiction and there exists a coalescence step to apply until we obtain the single link $A \bullet \bullet B$.
- (3 \Rightarrow 2) It follows trivially from the Definition 7.17.
- (2 \Rightarrow 1) This is shown in proof of Lemma 7.18. \square

Corollary 7.21. *The binary relation \mathbb{B}_ϕ in an a-flow $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ coalesces.*

Proof. Follows from Theorem 7.20 and Definition 7.16. \square

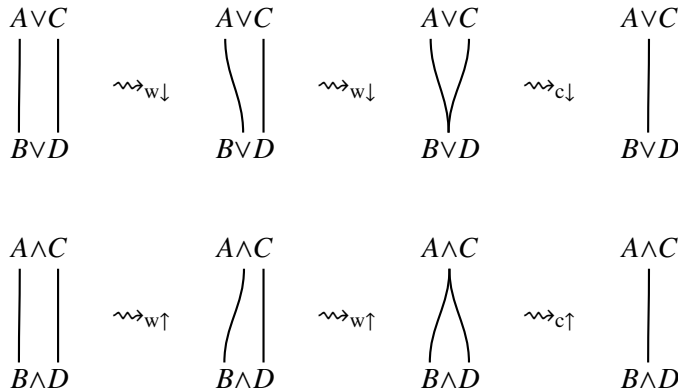
Corollary 7.22. *If a linking $\mathbb{B} : A \multimap B$ is well-mated and coalesces, then the flowbox $\phi = \langle A, B, \mathbb{B} \rangle$ is an a-flow.*

Proof. Follows from Theorem 7.20 and Definition 7.16. \square

The following theorems will investigate the compositionality of a-flows. We will show that a-flows have the same behaviors as other kinds of additive flows. Additionally, we explore the connection between a-flows, a^\downarrow -flows, and a^\uparrow -flows (i.e composing a^\downarrow -flows and a^\uparrow -flows freely results in a-flows). Furthermore, we will investigate the possibility of decomposing a-flows into a^\uparrow -flows and a^\downarrow -flows.

Theorem 7.23. *Let $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ and $\psi = \langle C, D, \mathbb{B}_\psi \rangle$ be a-flows. If $A \wedge C$ and $B \wedge D$ are pure formulas, then $\phi \wedge \psi = \langle A \wedge C, B \wedge D, \mathbb{B}_\phi \cup \mathbb{B}_\psi \rangle$ is an a-flow. If $A \vee C$ and $B \vee D$ are pure formulas, then $\phi \vee \psi = \langle A \vee C, B \vee D, \mathbb{B}_\phi \cup \mathbb{B}_\psi \rangle$ is an a-flow.*

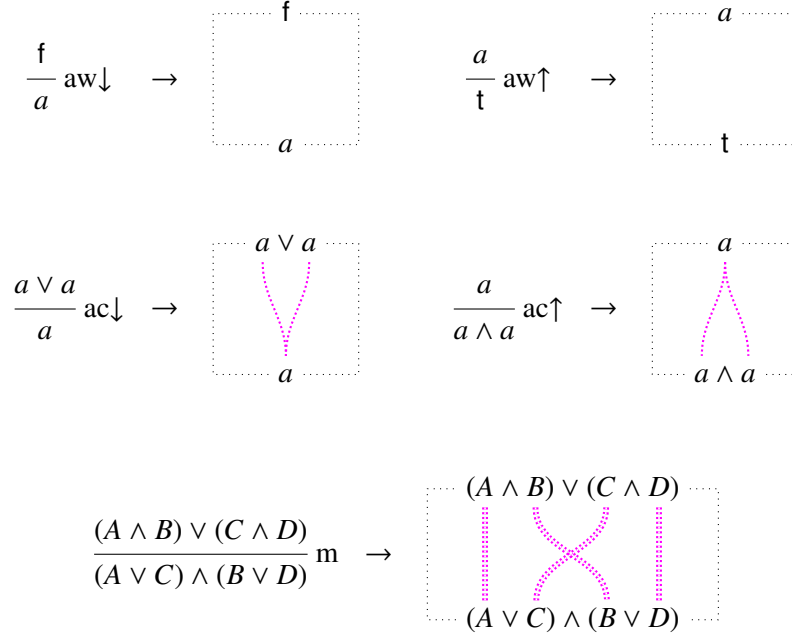
Proof. Using coalescence steps on ϕ and ψ and Corollary 7.21, we will have the links $A \bullet \bullet B$ and $C \bullet \bullet D$ in the case that all formulas are equivalent to unit-free formulas. Below we show that the linking consisting of these two links on both multi-sets of formulas $\overline{A \vee C}, B \vee D$ and $\overline{A \wedge C}, B \wedge D$ coalesces.



Hence, by Corollary 7.22 both $\phi \wedge \psi$ and $\phi \vee \psi$ are a-flows. If at least one of the formulas is equivalent to a unit, then its corresponding corresponding linking is empty. Hence, by hypothesis we know that $A \wedge C$ and $B \wedge D$ are pure formulas which implies that we can apply $\rightsquigarrow_{\equiv}$ steps to achieve goal link. Hence, $\phi \wedge \psi$ is an a-flow and the proof for $\phi \vee \psi$ is the same. \square

Theorem 7.24. *Let $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ and $\psi = \langle B, C, \mathbb{B}_\psi \rangle$ be a-flows. Then, $\phi \circ \psi = \langle A, C, \mathbb{B}_\phi \circ \mathbb{B}_\psi \rangle$ is an a-flow.*

Proof. For the case that either of the formulas $A, B,$ or C is equivalent to a unit, we have that either ϕ or ψ would have empty linking which then implies that $\phi \circ \psi$ has empty linking and therefore it is an a-flow. If $A, B,$ and C are all unit-free formulas, by way of contradiction we assume that $\phi \circ \psi$ is not an a-flow. So $\mathbb{B}_\phi \circ \mathbb{B}_\psi$ is not discrete which means that there exists a pruning $r = (r_A, r_C)$ and more than one link retained by it. Without loss of generality, let two of these links be $a_1 \bullet \bullet c_1$ and $a_2 \bullet \bullet c_2$. We now show that retaining these two links means that either ϕ or ψ is not an a-flow. Let b_1 and b_2 be atoms with links $a_1 \bullet \bullet b_1$ and $a_2 \bullet \bullet b_2$ in ϕ and links $b_1 \bullet \bullet c_1$ and $b_2 \bullet \bullet c_2$ in ψ . Now we consider prunings $r' = (r_A, r'_B)$ on formula sequent \overline{A}, B and $r'' = (r''_B, r_C)$ on formula sequent \overline{B}, C where either r'_B or r''_B retains both b_1 and b_2 (note that this is possible due to the fact


 Figure 7.2: Rules of SKS_a and their translations to a-flows

that r'_B is a pruning and r''_B is a co-pruning). Hence, the pruning that is retaining both b_1 and b_2 is retaining both links therefore either ϕ or ψ is not an a-flow which is in a contradictory with the hypothesis of the theorem. \square

Proposition 7.25. *Let $\phi = \langle A, B, \mathbb{B}_\phi \rangle$ be an a-flow. Then, the flowbox $\bar{\phi} = \langle \bar{B}, \bar{A}, \mathbb{B}_{\bar{\phi}} \rangle$ is an a-flow (if $C \bullet \bullet D$ is a link in \mathbb{B}_ϕ , then $\bar{D} \bullet \bullet \bar{C}$ is a link in $\mathbb{B}_{\bar{\phi}}$).*

Proof. It is straightforward to show that $\mathbb{B}_{\bar{\phi}}$ is discrete because of the dual nature of pruning and co-pruning defined in Definition 7.14. Hence, the flowbox $\bar{\phi}$ is an a-flow. \square

Lemma 7.26. *Let $\text{id}_A = \langle A, A, \mathbb{B}_{\text{id}} \rangle$ be an identity flowbox on a pure formula A , then id_A is an a-flow.*

Proof. If A is a unit, there exists no links so id_A is an a-flow. Let A be a unit-free formula, we prove the lemma by induction on A . For $A = a$ where a is an atom, there exists only one link $a \bullet \bullet a$ so id_A is an a-flow. If $A = B \wedge C$ or $A = B \vee C$, then by induction hypothesis we have that id_B and id_C are a-flows. We continue the proof by the following cases:

- If both B and C are equivalent to unit-free formulas, then the linkings of id_B and id_C coalesce to the links $B \bullet \bullet B$ and $C \bullet \bullet C$. By using these inductive coalescence steps continued by the steps shown below in (7.1), we will achieve the single link $A \bullet \bullet A$. Therefore the linking

\mathbb{B}_{id} coalesces and by Theorem 7.20 and Definition 7.16, the flowbox id_A is an a-flow.

$$\begin{array}{ccccccc}
 \begin{array}{c} B \wedge C \\ | \\ | \\ B \wedge C \end{array} & \rightsquigarrow_{w\uparrow} & \begin{array}{c} B \wedge C \\ | \quad | \\ B \wedge C \end{array} & \rightsquigarrow_{w\uparrow} & \begin{array}{c} B \wedge C \\ \diagup \quad \diagdown \\ B \wedge C \end{array} & \rightsquigarrow_{c\uparrow} & \begin{array}{c} B \wedge C \\ | \\ B \wedge C \end{array} \\
 & & & & & & (7.1) \\
 \begin{array}{c} B \vee C \\ | \\ | \\ B \vee C \end{array} & \rightsquigarrow_{w\downarrow} & \begin{array}{c} B \vee C \\ | \quad | \\ B \vee C \end{array} & \rightsquigarrow_{w\uparrow} & \begin{array}{c} B \vee C \\ \diagdown \quad \diagup \\ B \vee C \end{array} & \rightsquigarrow_{c\uparrow} & \begin{array}{c} B \vee C \\ | \\ B \vee C \end{array}
 \end{array}$$

- If both B and C are equivalent to units, then the linkings are empty and therefore the linking of id_A is also empty.
- If $B \equiv t$ and C is equivalent to a unit-free formula, then we can observe that only $A = B \wedge C$ is possible (otherwise A is not pure). So we have the following coalescence steps:

$$\begin{array}{ccc}
 \begin{array}{c} t \wedge C \\ | \\ t \wedge C \end{array} & \rightsquigarrow_{\equiv} & \begin{array}{c} t \wedge C \\ | \quad | \\ t \wedge C \end{array} & \rightsquigarrow_{\equiv} & \begin{array}{c} t \wedge C \\ | \\ t \wedge C \end{array}
 \end{array}$$

- For B equivalent to a unit-free formula and $C \equiv t$ we have the same proof as the previous case.
- If $B \equiv f$ and C is equivalent to a unit-free formula (resp. $C \equiv f$ and B equivalent to a unit-free formula), we continue the proof the same as two previous cases with the slight change that in this case $A = B \vee C$. \square

Theorem 7.27. *Let A and B be pure formulas. If $\mathcal{D} \parallel SKS_a$ is an additive deep inference proof for the premise A and conclusion B , then $\llbracket T(\mathcal{D}) \rrbracket$ is an a-flow.*

Proof. By Lemma 7.26, identity flows are a-flows so the translation of identity and equivalence rules is an a-flow. Now, We use induction on the size of the proof. If the proof is only one rule from SKS_a , then we use the translation of that rule shown in Figure 7.2. We should show that the linking shown in each translation coalesces. This is straightforward for the weakening and contraction rules due to the fact that they are instances of coalescence rules. Figure 7.3 demonstrates the steps for checking the correctness of the translation of the medial rule. By Theorem 7.20 and Definition 7.16, translation of each additive deep inference rule to a flowbox is an a-flow. Now using Theorem 7.24, we can deduce that the vertical composition of these translations, i.e $\llbracket T(\mathcal{D}) \rrbracket$, is an a-flow. \square

As we showed in the previous section, we can compose a^\downarrow -flows (resp. a^\uparrow -flows) but the composition of a^\downarrow -flows and a^\uparrow -flows freely was not investigated. In fact with the introduction of a-flows we show that this composition is possible. The following lemma puts the foundation for such claim.

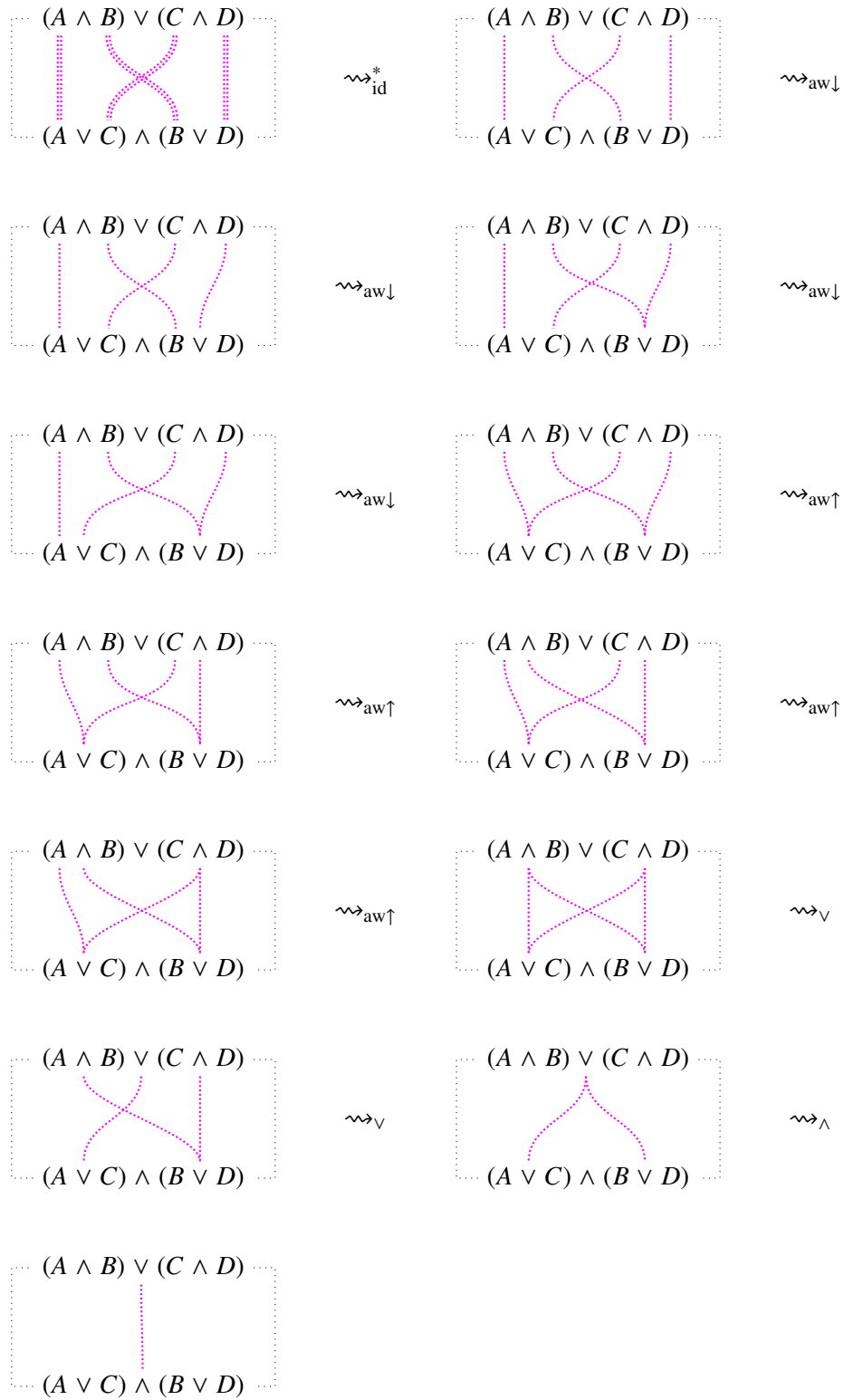


Figure 7.3: Correctness of the a-flow $\phi = \langle (A \wedge B) \vee (C \wedge D), (A \vee C) \wedge (B \vee D), \mathbb{B}_\phi \rangle$ for a medial rule

Lemma 7.28. An a^\downarrow -flow $\phi = \langle A, B, f_\phi^\downarrow \rangle$ is an a-flow. Similarly, an a^\uparrow -flow $\psi = \langle A, B, f_\psi^\uparrow \rangle$ is an a-flow.

Proof. By Theorem 7.13, there exists a derivation $\mathcal{D} \parallel_{\{ac\downarrow, aw\downarrow, m\}}$ such that $\llbracket T(\mathcal{D}) \rrbracket = \phi$ and by using soundness of a-flows shown in Theorem 7.27, $\llbracket T(\mathcal{D}) \rrbracket$ is an a-flow (Note that the translation of down additive inference rules are the same for a-flows and a^\downarrow -flows). Proof for ψ is similar. \square

Corollary 7.29. *The vertical and horizontal composition of several a^\downarrow -flows and a^\uparrow -flows is possible and the result is an a-flow.*

Proof. Follows from applying Theorems 7.23 and 7.24 to Lemma 7.28. \square

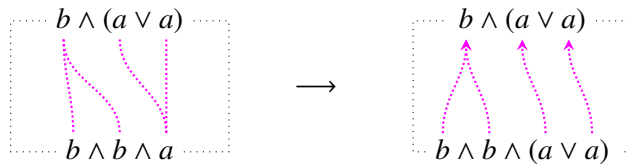
Furthermore, we show that a-flows can be decomposed to a^\downarrow -flows and a^\uparrow -flows which lays the foundation to prove the correctness of a-flows through decomposition. It is important to note that the decomposition theorem respects a certain order shown in [79]

Theorem 7.30. *Let $\phi = \langle A, C, \mathbb{B}_\phi \rangle$ be an a-flow. Then, we can decompose ϕ to an a^\uparrow -flow $\psi = \langle A, B, f_\psi^\uparrow \rangle$ and an a^\downarrow -flow $\chi = \langle B, C, f_\chi^\downarrow \rangle$ where $\phi = \psi \circ \chi$.*

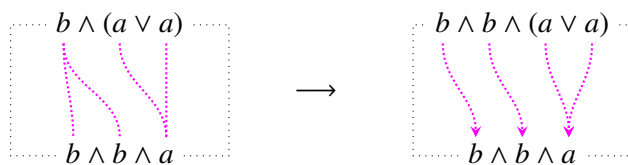
Proof. Let $\phi = \langle A, C, \mathbb{B}_\phi \rangle$ be an a-flow. With the following polynomial time algorithm on the formula C , we will create a pure formula B :

- put $B_0 := C$ and $i = 0$
- for each atom occurrence a in C :
 - let n be the number of links connected to a
 - increase i by 1
 - write the formula B_i as substitution of atom a in B_{i-1} with $\underbrace{a \vee \cdots \vee a}_{n \text{ times}}$, denoted as $B_i := B_{i-1}[a/\underbrace{a \vee \cdots \vee a}_{n \text{ times}}]$ (as an example if we have the formula $a \wedge (b \vee a)$ and we want to apply the substitution on the first occurrence of a a , we will have that $a \wedge (b \vee a)[a/a \vee a]$ is equal to the formula $(a \vee a) \wedge (b \vee a)$).

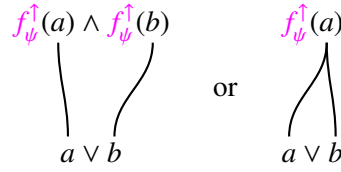
This algorithm terminates in polynomial time with respect to $|C|$ and we write $B = B_i$ where B_i is the last formula obtained by the algorithm. Now we set ψ to be the tuple $\langle A, B, f_\psi^\uparrow \rangle$ where $f_\psi^\uparrow(b) = a$ if the link $b \bullet \rightarrow a$ is in \mathbb{B}_ϕ . Note that in the case that an atom is substituted with copies of it in B , then each copy should only have one link. We show such an occurrence below where a is replaced by two copies and each copy has one link:



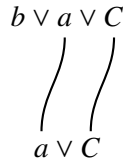
Next, we set χ to be the tuple $\langle B, C, f_\chi^\downarrow \rangle$ where $f_\chi^\downarrow(b) = c$ if either b is an atom occurrence in A and the link $b \bullet \rightarrow c$ is in \mathbb{B}_ϕ , or b is one of the instances of c after duplication during the algorithm. Below we show an example of χ with the same ϕ as the previous example:



Based on how we decomposed ϕ into ψ and χ , it is straightforward to show $\phi = \psi \circ \chi$ since we have kept all the links as they are and duplicated instances of atoms. It only remains to show that ψ is an a^\uparrow -flow and χ is an a^\downarrow -flow. The algorithm provides B in a way that every edge is preserved in f_χ^\downarrow and by the definition of χ we can observe that f_χ^\downarrow is a full homomorphism. Consequently, the flowbox χ is an a^\downarrow -flow. Next, we prove that ψ is an a^\uparrow -flow. To start, we can show that f_ψ^\uparrow is a function since every atom occurrence has only one link based on how we defined ψ and B . We also have to prove that f_ψ^\uparrow is a homomorphism. We continue by way of contradiction. Let a and b be atom occurrences in B where $aR_{\mathcal{G}(\bar{B})}b$ but $(f_\psi^\uparrow(a), f_\psi^\uparrow(b)) \notin R_{\mathcal{G}(\bar{B})}$ which means one of the following configuration happens (in the left case $f_\psi^\uparrow(a) = f_\psi^\uparrow(b)$):



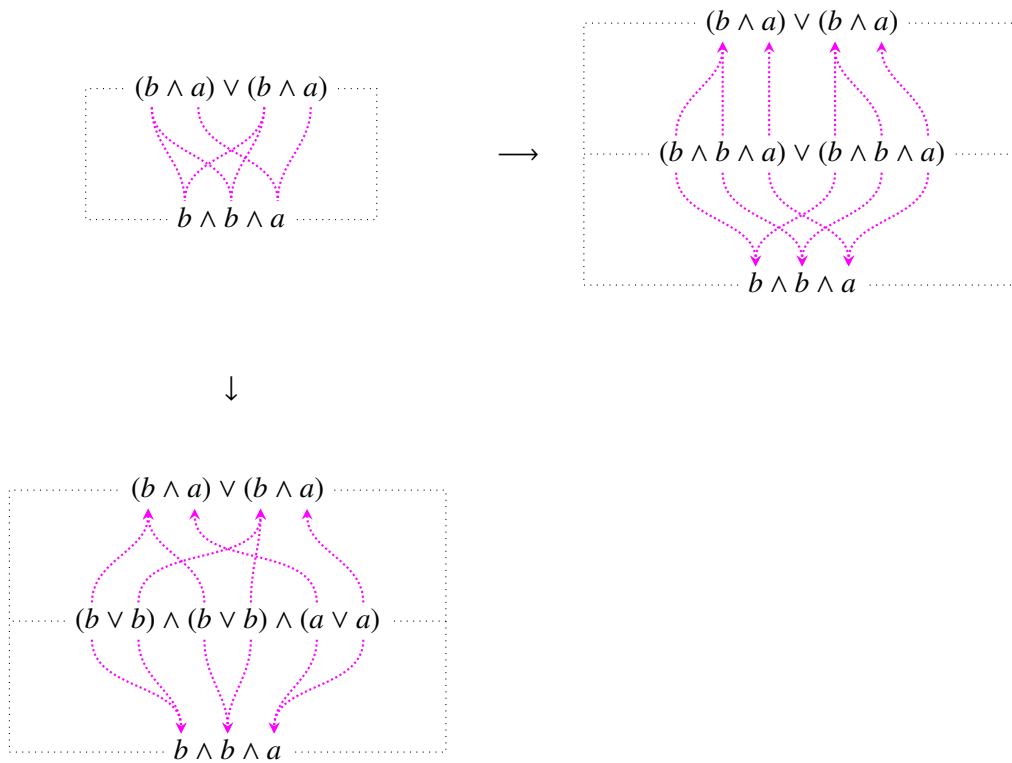
However, there is no coalescence steps applicable into any of the two configurations that could resolve these links which is in contradiction with the fact that ϕ is an a -flow using Theorem 7.20 and Definition 7.16. Hence, the function f_ψ^\uparrow is a homomorphism and it remains to show that it holds the skew fibration property. Let $v \in \mathcal{G}(\bar{B})$ and $w \in \mathcal{G}(\bar{A})$ be vertices with labels $l(v) = \bar{a}$ and $l(w) = \bar{b}$ where $f_\psi^\uparrow(v)R_{\mathcal{G}(\bar{A})}w$. By way of Contradiction let us assume for all vertices $z \in \mathcal{G}(\bar{B})$ such that $vR_{\mathcal{G}(\bar{B})}z$, there exists the edge $f_\psi^\uparrow(z)R_{\mathcal{G}(\bar{A})}w$ which means the following configuration exists in ϕ :



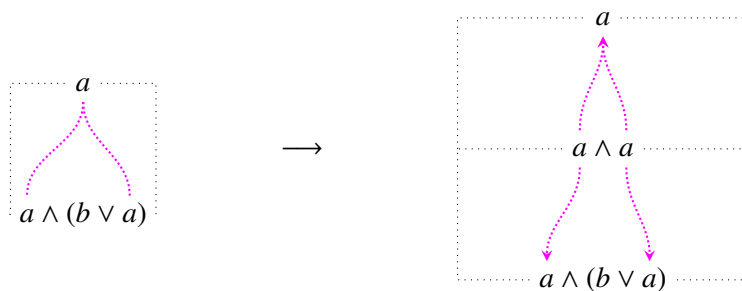
This configuration does not coalesce, the same as the previous two configurations it results in a contradiction from which we can conclude that f_ψ^\uparrow is a skew fibration. Therefore, the flowbox ψ is an a^\uparrow -flow. \square

Remark 7.31. Note that the decomposition is not unique and if the algorithm in the proof of the previous theorem is changed to a dual algorithm on the hypothesis formula, the middle formula will be different and the reason behind this is that medial rules can act as both a^\uparrow -flows and a^\downarrow -flows which results in several possibilities of decomposition. We show an example of such different

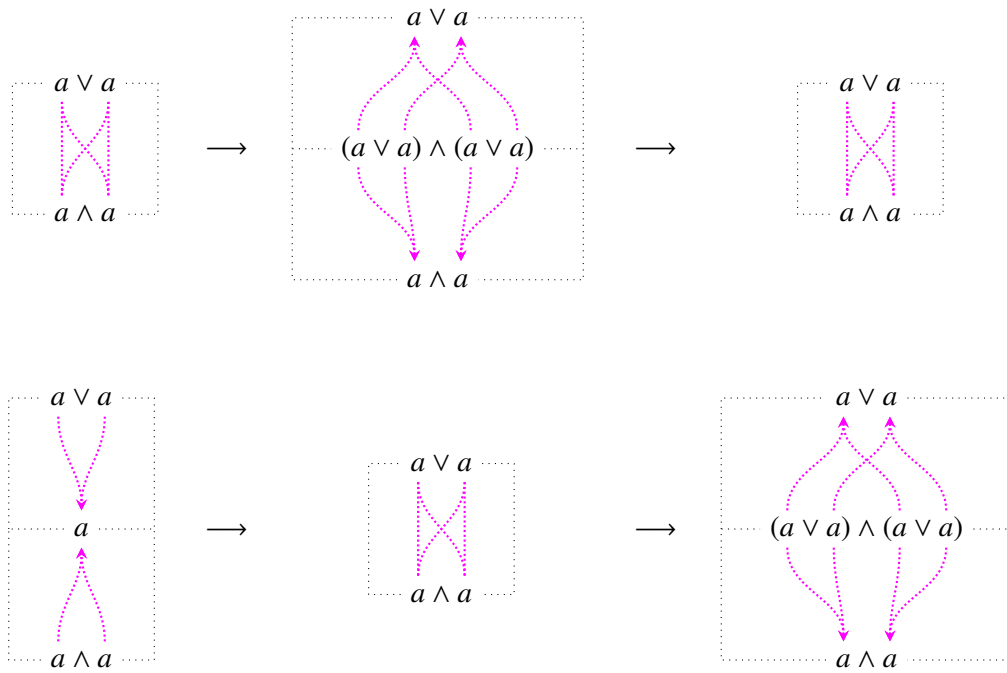
decompositions below:



Example 7.32. Below we show an example of an a -flow and its decomposition to an a^\uparrow -flow and a^\downarrow -flow.



Note that decomposition is done in a very specific order which also serves for the purpose of normalization and the normal form of flows, in Chapter 9, which are of size exponential with respect to the original flow. A result of such behavior is that composing an a^\downarrow -flows and an a^\uparrow -flows to obtain an a -flow and then decomposing this a -flow to an a^\uparrow -flow and an a^\downarrow -flow will not result in the same thing. However, decomposing an a -flow and then composing it back has the same result as the original a -flow. We show this with the example below:



Theorem 7.33. Let $\phi = \langle A, B, \mathbb{B}_{a\text{-flow}}\phi \rangle$ be an a -flow. Then, there exists a derivation $\mathcal{D} \parallel_{\text{SKS}_a}^A B$ where $\llbracket \mathbb{T}(\mathcal{D}) \rrbracket = \phi$.

Proof. Using theorem 7.30, we show that a -flows can be decomposed into a^\uparrow -flows and a^\downarrow -flows. Now, by Theorem 7.13 we have the derivation \mathcal{D} and also $\llbracket \mathbb{T}(\mathcal{D}) \rrbracket$ is in fact equal to ϕ (we use Corollary 7.29 to prove that that the composition of the translation of rules is indeed ϕ). \square

Part III

Combinatorial Flows and Their Normalization

CHAPTER 8

Combinatorial Flows

Definition 8.1. A *combinatorial flow* is a preflow $\phi: A \vdash B$ where every flowbox that occurs in ϕ is either a multiplicative flow or an additive flow (note that this includes a^\downarrow -flows, a^\uparrow -flows, and a -flows). We define the *size* of a combinatorial flow, denoted as $|\phi|$, as the number of flowboxes that are not identity or equivalence flows.

Lemma 8.2. Let $\phi: A \vdash B$ be a combinatorial flow. Then, there exists a combinatorial flow $\psi: \bar{B} \vdash \bar{A}$ and we can obtain it in linear time with respect to size of the combinatorial flow.

Proof. Straightforward using Propositions 6.3 or 7.25. \square

Theorem 8.3. Let \mathcal{D} be a derivation in SKS. Then $\mathsf{T}(\mathcal{D})$ is a combinatorial flow.

Proof. Using theorems 6.9, 7.12, and 7.27, we translate each block of multiplicative and additive rules to m -flows, a^\downarrow -flows, a^\uparrow -flows, and a -flows. Hence, based on Definition 8.1, the preflow $\mathsf{T}(\mathcal{D})$ is a combinatorial flow. \square

We can *simplify* combinatorial flows by “executing” the operations in Definition 5.7. More precisely, a *simplification* of a combinatorial flow $\phi: A \vdash B$ to a combinatorial flow $\psi: A \vdash B$ is the binary relation, denoted as $\phi \rightarrow_s \psi$, where ψ is obtained from ϕ by collapsing subflows in which all flowboxes have the same type (m -flow, a^\downarrow -flow, a^\uparrow -flow, or a -flows).

Note that the difference between the collapse (which is uniquely determined) and a simplification (which is not uniquely determined) is that in the former everything is simplified, whereas in the latter the property of being a combinatorial flow is preserved.

Proposition 8.4. Let $\phi: A \vdash B$ and $\psi: A \vdash B$ be combinatorial flows with $\phi \rightarrow_s \psi$. Then, size of ψ is smaller or equal to ψ (i.e. $|\psi| \leq |\phi|$).

Proof. Straightforward using Definitions 8.1 and 5.9. \square

Combinatorial flows are equivalent modulo the following equivalences:

$$\begin{aligned}
 (\phi \otimes \psi) \otimes (\xi \otimes \pi) &\equiv (\phi \otimes \xi) \otimes (\psi \otimes \pi) & (\phi \otimes \psi) \otimes \xi &\equiv \phi \otimes (\psi \otimes \xi) \\
 (\phi \otimes \psi) \otimes (\xi \otimes \pi) &\equiv (\phi \otimes \xi) \otimes (\psi \otimes \pi) & (\phi \otimes \psi) \otimes \xi &\equiv \phi \otimes (\psi \otimes \xi) \\
 \phi \otimes \psi &\equiv \psi \otimes \phi & (\phi \otimes \psi) \otimes \xi &\equiv \phi \otimes (\psi \otimes \xi) \\
 \phi \otimes \psi &\equiv \psi \otimes \phi & \text{id} \otimes \phi &\equiv \phi \equiv \phi \otimes \text{id} \\
 \phi \otimes \psi &\equiv \psi \otimes \phi & &
 \end{aligned} \tag{8.1}$$

Now, we can state the converse of Theorem 8.3.

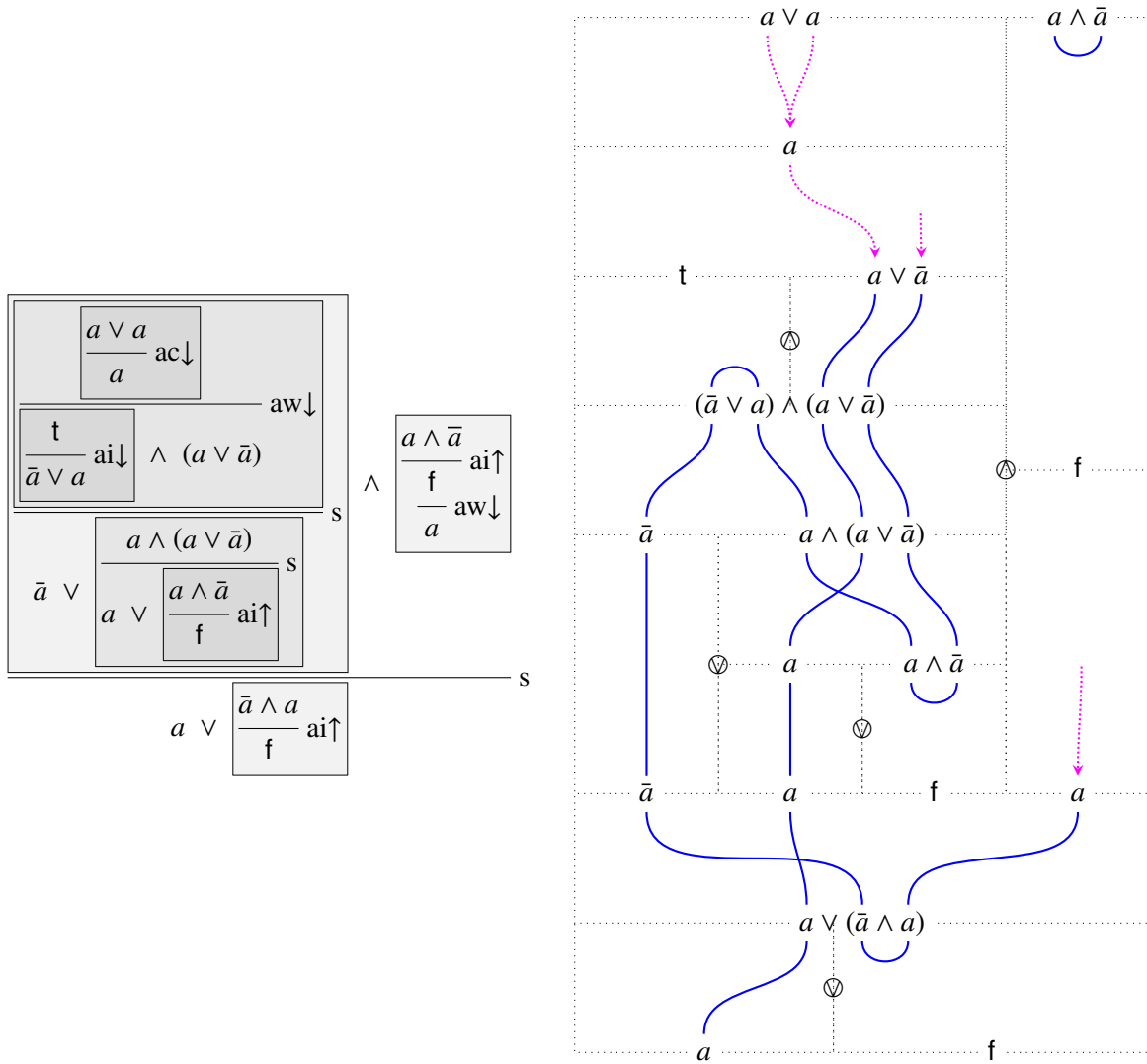


Figure 8.1: An open deduction derivation with hypothesis $(a \vee a) \wedge (a \wedge \bar{a})$ and conclusion $a \vee f$, and its translation to a preflow with colors

Theorem 8.5. Let $\phi: A \vdash B$ be a combinatorial flow. Then there is a derivation $\mathcal{D} \parallel_{\text{SKS}}$ such that $\phi \rightarrow_s \mathbb{T}(\mathcal{D})$.

Proof. This follows immediately from Theorems 6.10, 7.13, and 7.33 since every flowbox in ϕ is either a multiplicative flow (i.e. an m-flow), or an additive flow (i.e. an a^\downarrow -flow, or an a^\uparrow -flow, or an a-flow). \square

Example 8.6. Figure 8.1 shows an open deduction derivation \mathcal{D} on the left and we show its translation $\mathbb{T}(\mathcal{D})$ on the right. Now we can further simplify it to obtain the combinatorial flow shown in Figure 8.2 on the left.

Corollary 8.7. Combinatorial flows are sound and complete for classical logic.

Corollary 8.8. Combinatorial flows form a proof system in the sense of Cook and Reckhow [17].

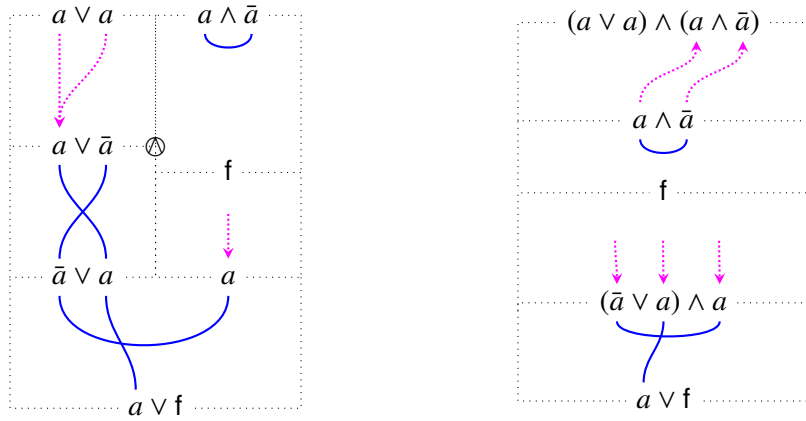


Figure 8.2: A simplification of the combinatorial flow in Figure 8.1 and its purification

Proof. It can be checked in polynomial time if a preflow is a combinatorial flow. \square

Example 8.9. Figure 8.11 (p. 88) shows a series of combinatorial flows, where we apply “Currying”: flipping some premises to the conclusion or vice versa. Because blue wires can be yanked, this is easily possible, and by Theorem 8.5 there is always a corresponding derivation. However, in the last combinatorial flow shown in the figure, there are two blue bends that are connected by a purple edge. Therefore, this blue wires can not be yanked:



8.1 Combinatorial Proofs with Cuts

Definition 8.10. A *combinatorial proof* [47] of a pure formula A is a skew fibration $f: \mathcal{H} \rightarrow \mathcal{G}(A)$ from an \mathfrak{a} -acyclic RB-cograph \mathcal{H} to the graph of A .¹

Translated to the setting of this thesis, a combinatorial proof is the composition $\phi \circledast \psi$ of an m -flow $\phi = \langle t, H, \mathbb{B}_\phi \rangle$ and an a^\downarrow -flow $\psi = \langle H, A, f_\psi^\downarrow \rangle$. In other words, combinatorial proofs are a special case of combinatorial flows that make a global separation between the multiplicative and the additive parts of a cut-free proof. In order to deal with cuts, the notion of combinatorial proof has been extended to sequents in [49].

Definition 8.11. A *combinatorial proof with cuts* of a sequent $\Gamma = A_1, \dots, A_n$ of unit-free formulas is a skew fibration $f: \mathcal{H} \rightarrow \mathcal{G}(A_1 \vee \dots \vee A_n \vee (\overline{C}_1 \wedge C_1) \vee \dots \vee (\overline{C}_k \wedge C_k))$ from an \mathfrak{a} -acyclic RB-cograph \mathcal{H} to the graph of $\Gamma, \overline{C}_1 \wedge C_1, \dots, \overline{C}_k \wedge C_k$, where C_1, \dots, C_k are arbitrary unit-free formulas and are called the *cut formulas* of the proof.

These cuts can be simulated by an m -flow in combinatorial flows. More precisely, assume we have a combinatorial proof with cuts for a sequent $\Gamma = \overline{A}, B$, as shown on the left in Figure 8.3. We have that $H = H_A \vee H_B$ and $\psi = \psi_A \circledast \psi_{BC}$ because of Lemma 7.5. We can translate this into a combinatorial flow $\phi: A \vdash B$, as shown on the right in Figure 8.3. There, the m -flow ϕ' exists by Proposition 6.3.

¹In [47], the units t and f are treated like atoms in the translation to graphs, so that the restriction to pure formulas was not needed. However, this would make composition difficult to define, and for this reason in [49] combinatorial proofs have been restricted to the unit-free setting.

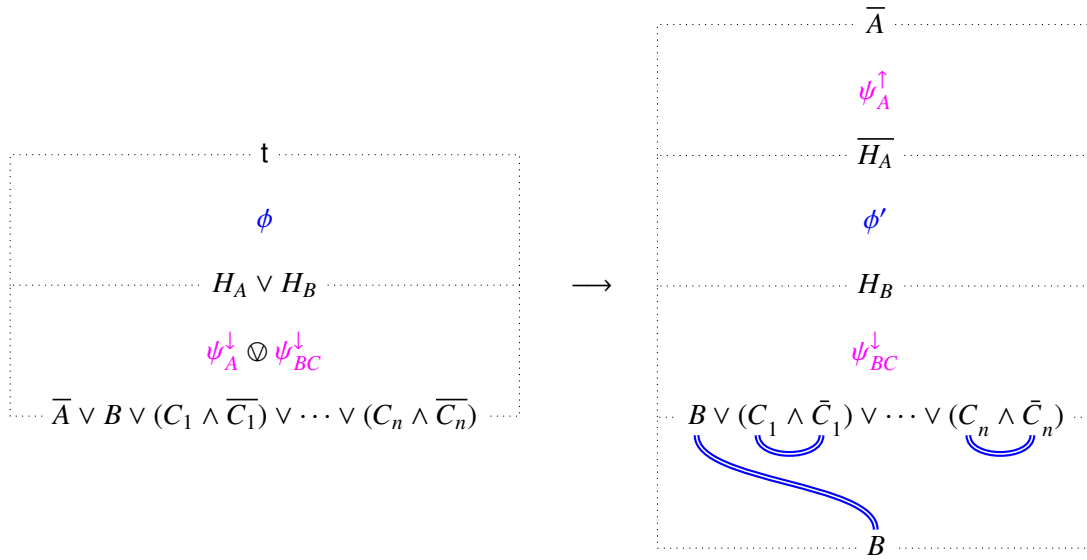
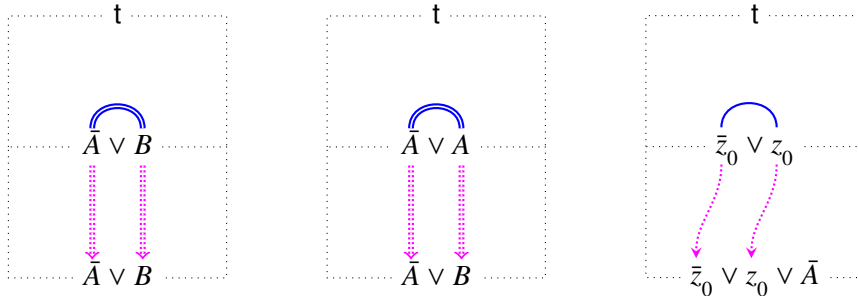


Figure 8.3: Translating combinatorial proofs with cuts into combinatorial flows.

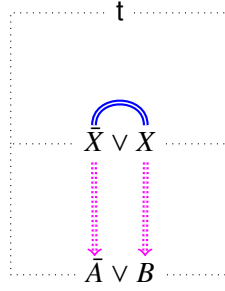
We are now going to show the converse, i.e., we will give a polynomial translation from combinatorial flows to combinatorial proofs with cuts. For this, we have to be a bit more careful with the units, as combinatorial proofs with cuts are only defined for unit-free formulas. We let z_0 be a fresh propositional variable, and we define the function $(\cdot)^0$ on pure formulas as follows: If $A \equiv B$ for some unit-free formula B , then $A^0 = B$. If $A \equiv \text{t}$, then $A^0 = \bar{z}_0 \vee z_0$. If $A \equiv \text{f}$, then $A^0 = \bar{z}_0 \wedge z_0$. Now assume we have a combinatorial flow $\phi: A \vdash B$. We can translate this inductively into a combinatorial proof with cuts of the sequent $\Gamma = \overline{A^0}, B^0$.

1. First, every m-flow, a^\downarrow -flow, and a^\uparrow -flow can be immediately translated into a combinatorial proof. For the cases $\langle A, B, \mathbb{B}_\phi \rangle$ and $\langle A, B, f_\psi^\downarrow \rangle$ and $\langle A, \text{t}, f_\psi^\uparrow \rangle$ these are shown below. The others are similar.



2. if $\phi = \langle A, B, \mathbb{B}_\phi \rangle$, first we decompose it to an a^\uparrow -flow $\psi = \langle A, X, f_\psi^\uparrow \rangle$ and an a^\downarrow -flow $\chi = \langle X, B, f_\chi^\downarrow \rangle$ where $\phi = \psi \circ \chi$, by using the algorithm shown in Theorem 7.30. Hence, we have

the following combinatorial proof:



3. If $\phi = \phi_1 \otimes \phi_2$ with $\phi_1: A_1 \vdash B_1$ and $\phi_2: A_2 \vdash B_2$ then we have by induction hypothesis combinatorial proofs with cuts of $\Gamma_1 = \overline{A_1^0}, B_1^0$ and $\Gamma_2 = \overline{A_2^0}, B_2^0$. By the construction in [49], we get one with conclusion $\Gamma = \overline{A_1^0}, \overline{A_2^0}, B_1^0 \wedge B_2^0$ which is equivalent to $(\overline{A_1} \vee \overline{A_2})^0, (B_1 \wedge B_2)^0$. The case for $\phi = \phi_1 \otimes \phi_2$ is similar.
4. If $\phi = \phi_1 \odot \phi_2$ with $\phi_1: A \vdash D_1$ and $\phi_2: D_2 \vdash B$ with $D_1 \equiv D_2$, we proceed similarly, with the difference that we add a new cut formula D_1^0 . Note that $\mathcal{G}(D_1^0) = \mathcal{G}(D_2^0)$ and $\overline{D_1^0} \wedge D_1^0$ is added to the conclusion sequent.

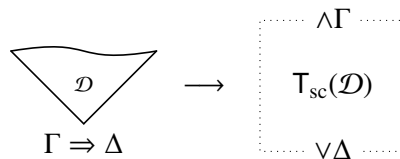
8.2 Combinatorial Flows and Other Proof Systems

In the beginning of this chapter we have shown combinatorial flows as well as their soundness and correctness using a translation function from deep inference derivations to combinatorial flows. In order to take closer steps to answering the problem of proof equivalence, we need to investigate the possibility of having a translation from other proof systems such as Sequent Calculus and Natural Deduction (due to [25, 26]), shown in Chapter 2, Sections 2.2, and 2.1. We go into more details for each of these two formalisms in the two following sections.

8.2.1 Sequent Calculus

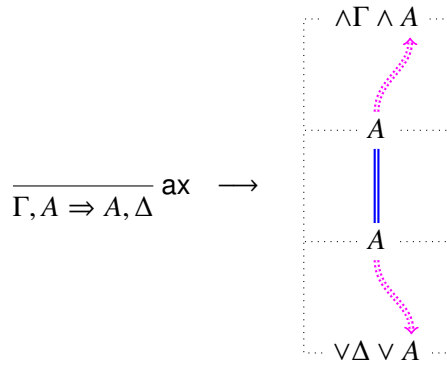
In this section, first we define a translation function from sequent calculus derivations to combinatorial flows. We also show that this translation keeps the flow in polynomial size with respect to the size of the derivation (i.e. the number of rules as well as the size of the formulas used in the proof).

Definition 8.12. The *translation* T_{sc} of a sequent calculus derivation $\mathcal{D}: \Gamma \vdash_{SC} \Delta$, denoted as $T_{sc}(\mathcal{D}): \wedge \Gamma \vdash \vee \Delta$, is the preflow inductively obtained as follows and we depict this procedure as shown below:



If \mathcal{D} is only an axiom rule proving the sequent $\Gamma, A \Rightarrow A, \Delta$, then $T_{sc}(\mathcal{D})$ returns the following

combinatorial flow:

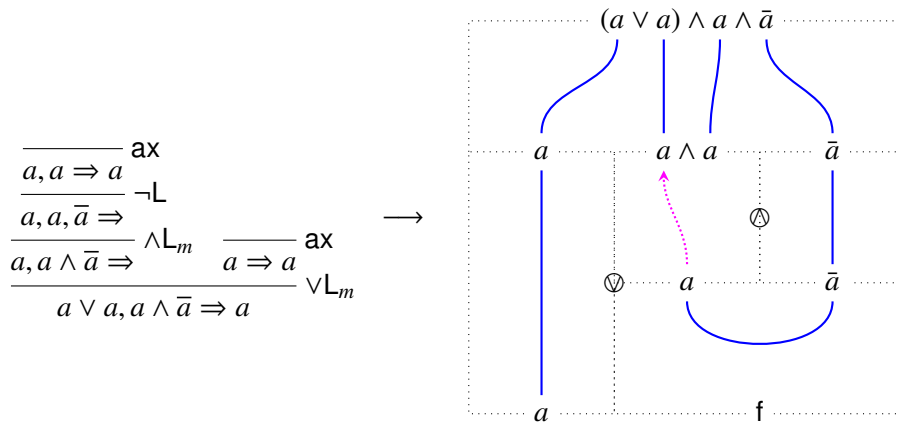


If \mathcal{D} is not just one axiom rule, Figures 8.4, 8.5, 8.6, and 8.7 show the inductive translation the preflow $T_{sc}(\mathcal{D})$.

Theorem 8.13. *If $\mathcal{D}: \Gamma \vdash_{SC} \Delta$ is a sequent calculus derivation for the sequent $\Gamma \Rightarrow \Delta$, then $T_{sc}(\mathcal{D}): \wedge \Gamma \vdash \vee \Delta$ is a combinatorial flow and the size of $T_{sc}(\mathcal{D})$ is of polynomial size with respect to the size of \mathcal{D} .*

Proof. The translation function is shown in Figures 8.6, 8.4, 8.5, and 8.7 and as shown in Definition 8.12, this function provides a combinatorial flow inductively from the hypothesis to conclusion. Moreover, we can observe that the size of the combinatorial flow (i.e. the number of flowboxes in the flow and size of their hypothesis and conclusion) is of polynomial size with respect to the number of rules and size of formulas in \mathcal{D} . \square

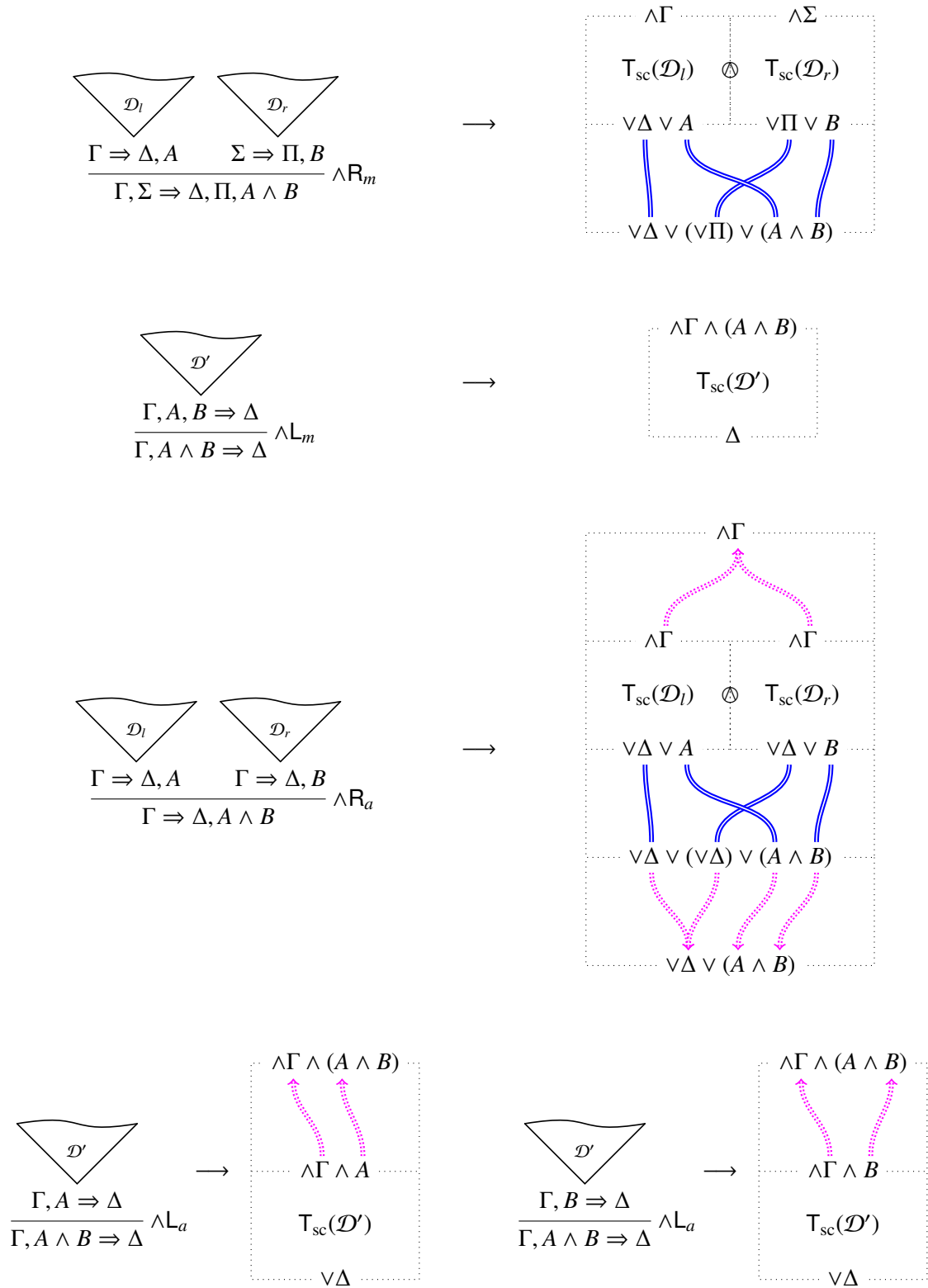
Example 8.14. Let \mathcal{D} be the sequent calculus derivation proving $a \wedge (b \vee c) \vdash_{SC} (a \wedge b) \vee c$ shown below left. Then, using the translation function shown in Definition 8.12, we can provide the translated combinatorial flow $T_{sc}(\mathcal{D})$ shown below right.

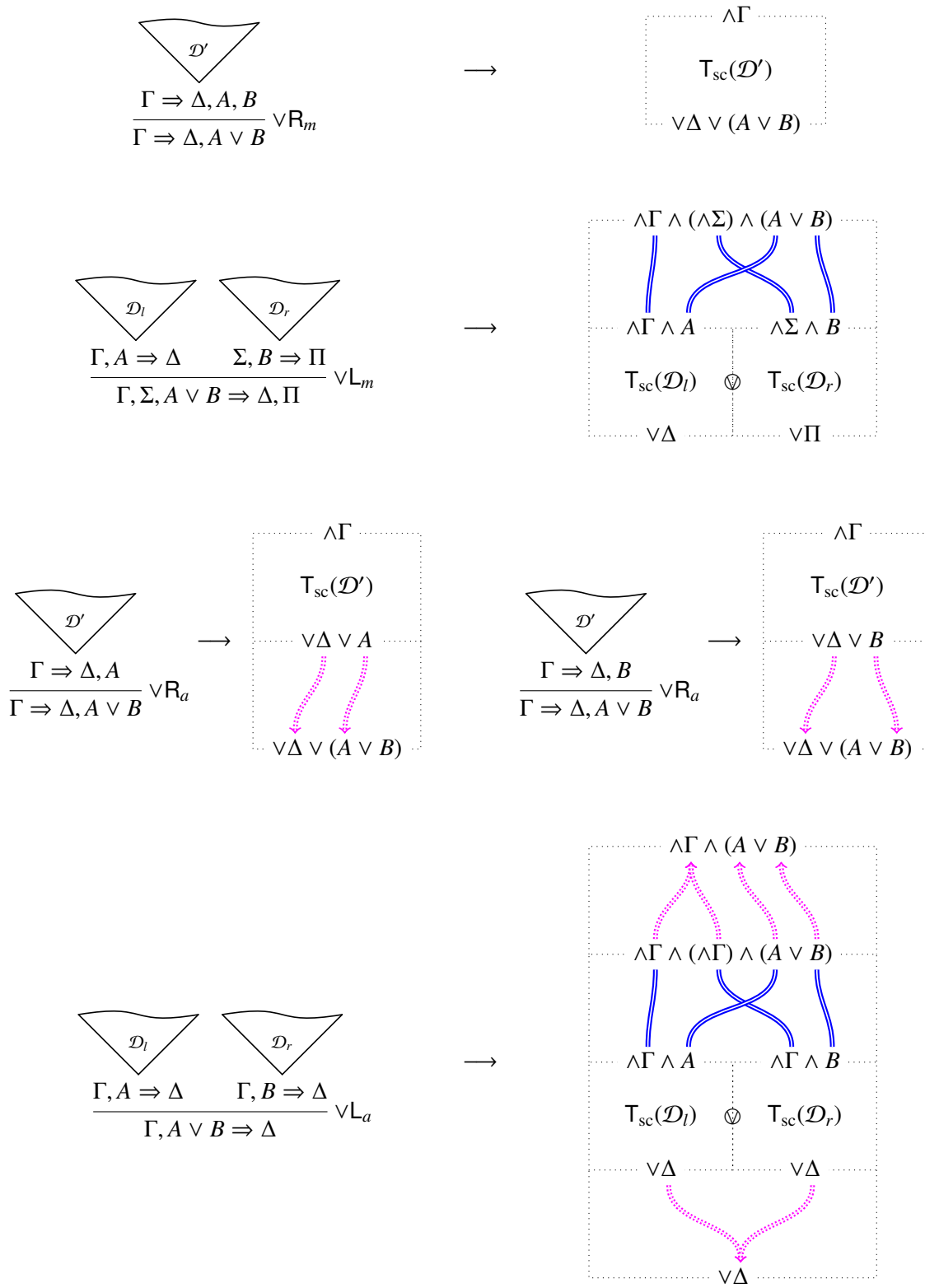


8.2.2 Natural Deduction

In this section we show the correspondence between Natural Deduction [25, 26, 53, 67] and Combinatorial flows.

As shown in Chapter 2, in natural deduction, a formula is deduced from a collection of premises by applying inference rules repeatedly. Figure 2.1 shows the inference rules for natural deduction for classical propositional logic. We utilize an extra connective (i.e. implication)


 Figure 8.4: Translation of \wedge logical rules of sequent calculus to combinatorial flows


 Figure 8.5: Translation of \vee logical rules of sequent calculus to combinatorial flows

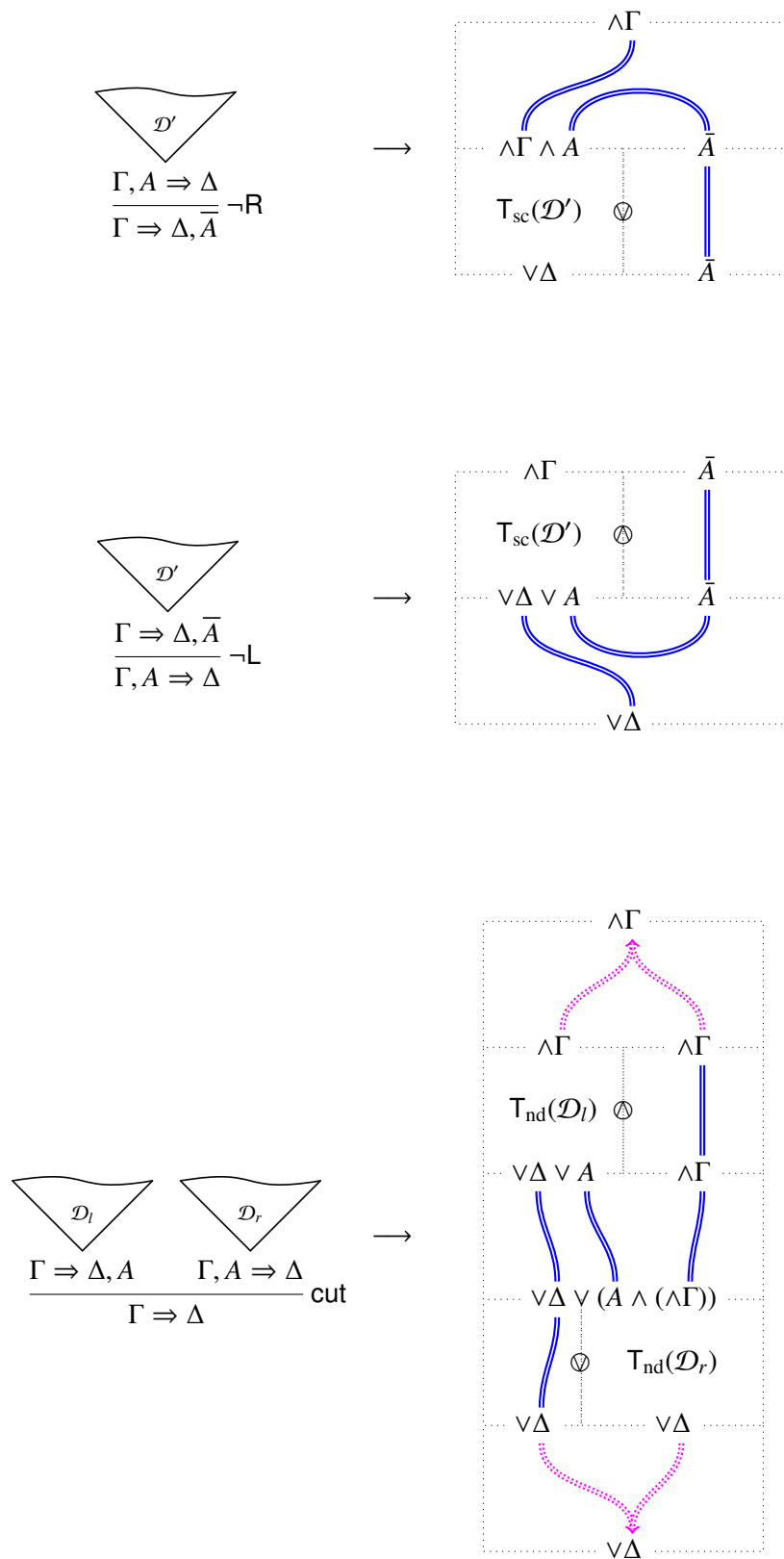


Figure 8.6: Translation of \neg logical rules and cut rule of sequent calculus to combinatorial flows

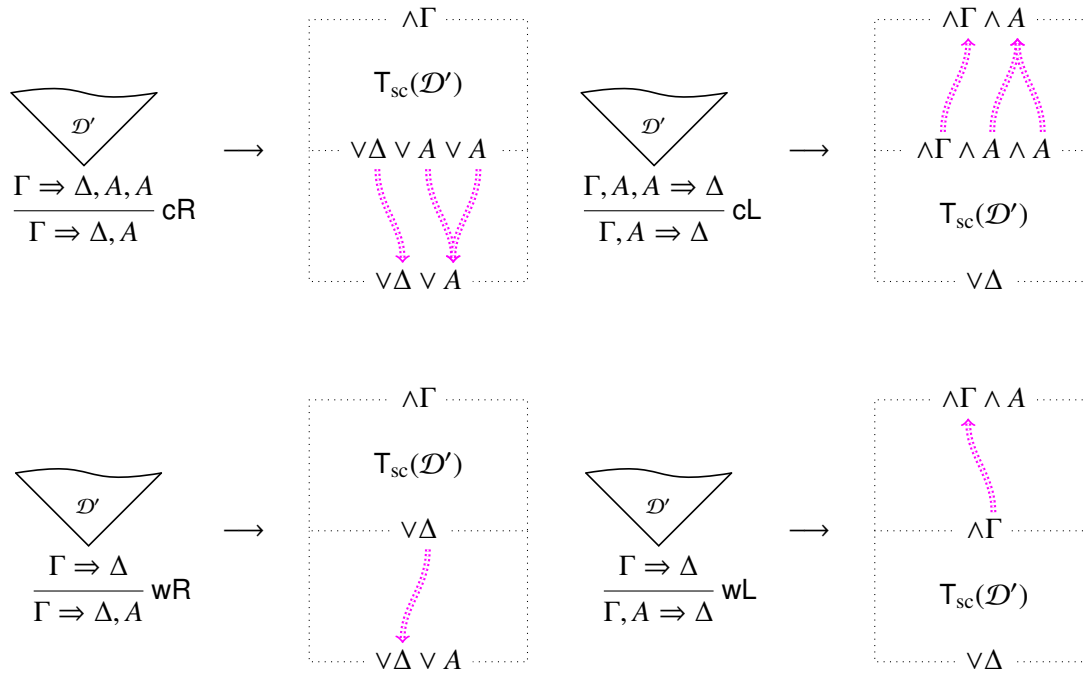
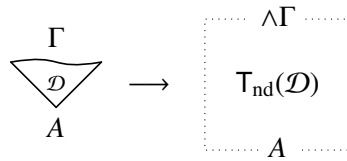


Figure 8.7: Translation of structural rules of sequent calculus to combinatorial flows

in the language when we consider natural deduction, shown in Section 2.1, and we will use the implication interpretation (i.e. $A \rightarrow B := \bar{A} \vee B$) to translate formulas in natural deduction proofs to formulas in the language of propositional classical logic.²

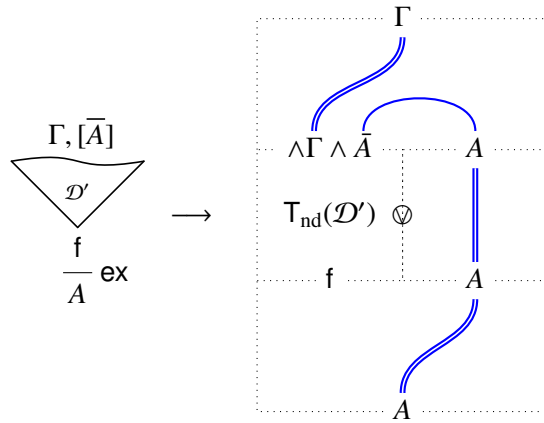
Definition 8.15. The *translation* T_{nd} of a natural deduction derivation $\mathcal{D}: \Gamma \vdash_{\mathcal{NK}} A$, denoted as $T_{\text{nd}}(\mathcal{D}): \wedge \Gamma \vdash A$, is the preflow inductively obtained as follows and we depict this procedure as shown below:



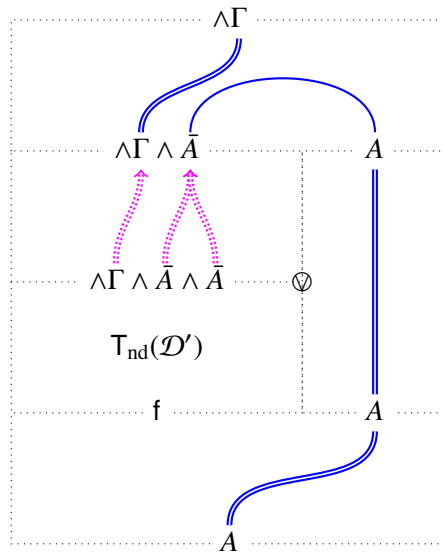
If \mathcal{D} is a formula A , then $T_{\text{nd}}(\mathcal{D})$ is the identity flowbox id_A , otherwise we show the inductive translation $T_{\text{nd}}(\mathcal{D})$ in Figures 8.8, 8.10, and 8.9.

It is important to note that in a translation of a rule such as *ex* shown below, we have assumed that the formula \bar{A} , which is discharged, occurs only once in the hypothesis of the derivation.

²Note that implication is generally present in the language. However, we base this thesis on the deep inference syntax which uses negation normal form of formulas and implication connective is not discussed if not needed, which is the case for classical propositional logic.



However, this is not always the case and we can assume that \bar{A} is presenting more occurrences. As an example below, we show the translation if there is two such occurrences. The solution is to add a co-contraction on top to get rid of multiple copies of \bar{A} .



Theorem 8.16. *If $\mathcal{D}: \Gamma \vdash_{\mathcal{NK}} A$ is a natural deduction derivation from set of hypothesis Γ to conclusion A , then $\mathbb{T}_{nd}(\mathcal{D}): \wedge \Gamma \vdash A$ is a combinatorial flow and the size of $\mathbb{T}_{nd}(\mathcal{D})$ is of polynomial size with respect to the size of \mathcal{D} .*

Proof. By Definition 8.15, we can observe that the translation of each rule returns a combinatorial flow inductively. Moreover, we can observe that the size of the translation is of polynomial size with respect to the size of \mathcal{D} . \square

Example 8.17. Let \mathcal{D} be a natural deduction derivation proving $a \wedge (b \vee c) \vdash_{\mathcal{NK}} (a \wedge b) \vee c$ shown below left. Then, using the translation function shown in Definition 8.15, we can provide the translated combinatorial flow $\mathbb{T}_{nd}(\mathcal{D})$ shown below right. Note that the sequent calculus derivation in Example 8.14 (78) has the same hypothesis and conclusion. However, the translation is not the same combinatorial flow as in this example. Therefore, one can conclude that these two proofs are

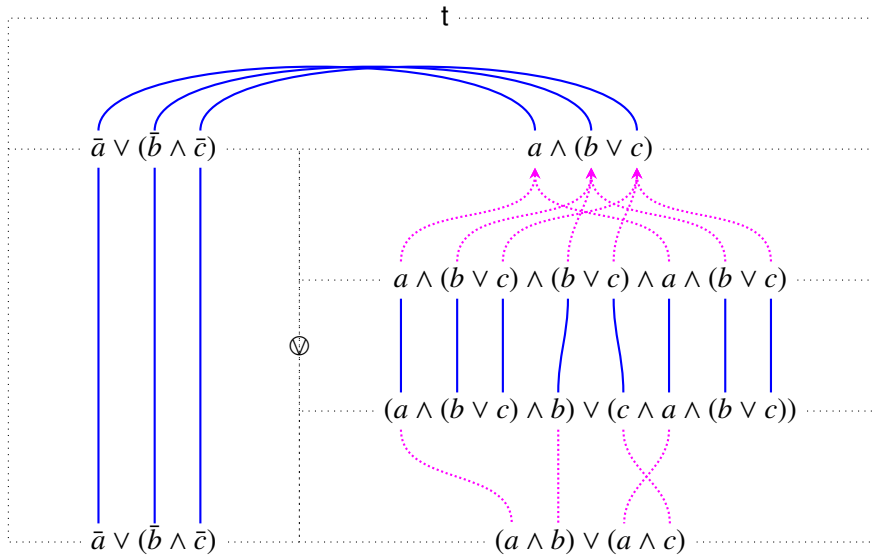
not equivalent.

$$\frac{\frac{(a \vee a) \wedge (a \wedge \bar{a})}{a \wedge \bar{a}} \wedge E_r}{a} \wedge E_l \rightarrow \begin{array}{|c|} \hline (a \vee a) \wedge a \wedge \bar{a} \\ \hline a \\ \hline \end{array}$$

Below, we show another example that utilizes discharge on several copies of one formula at the same time. Observe that the combinatorial flow shown is a simplified translation and there exists an a-flow which is the composition of several identity flows, a[↓]-flows, and a[↑]-flows.

$$\frac{\frac{[a \wedge (b \vee c)]^2}{b \vee c} \wedge E_r \quad \frac{\frac{[a \wedge (b \vee c)]^2}{a} \wedge E_l \quad [b]^1 \wedge I}{a \wedge b} \vee I_l \quad \frac{[a \wedge (b \vee c)]^2 \wedge E_l \quad [c]^1 \wedge I}{a \wedge c} \vee I_r}{(a \wedge b) \vee (a \wedge c)} \vee E_1}{(a \wedge (b \vee c)) \rightarrow ((a \wedge b) \vee (a \wedge c))} \rightarrow I_2$$

↓



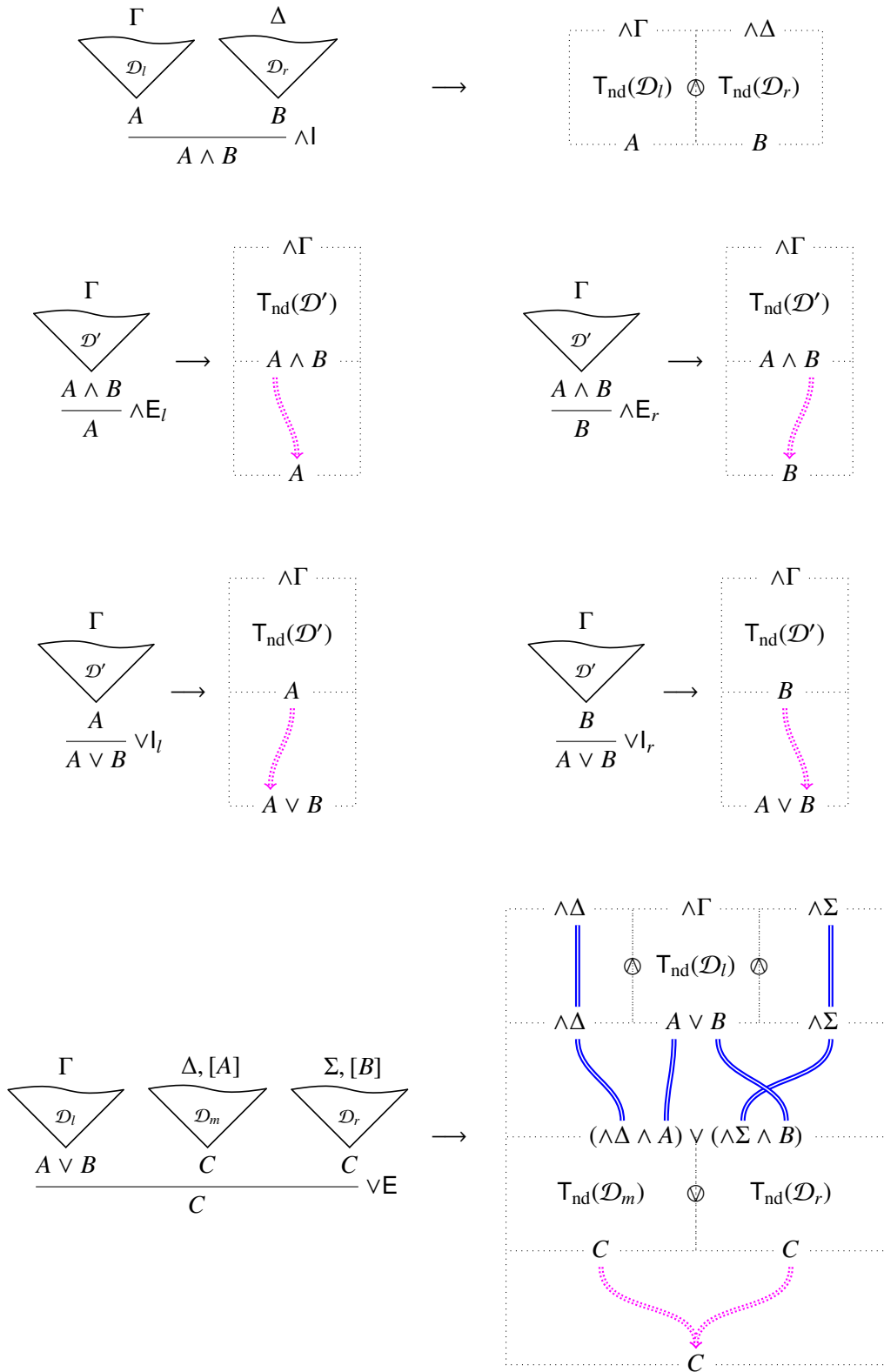


Figure 8.8: Translation of conjunctive and disjunctive rules of natural deduction to combinatorial flows

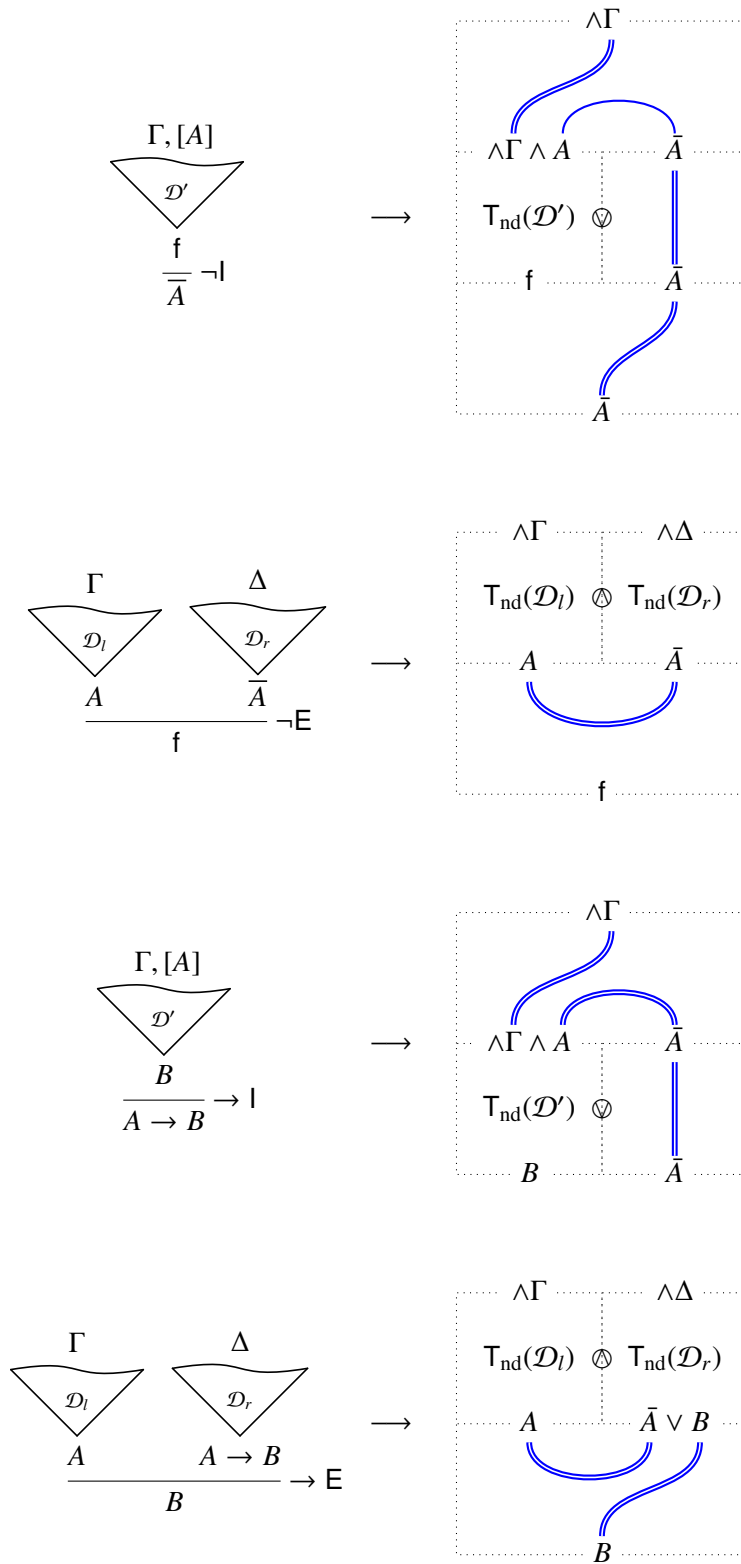


Figure 8.9: Translation of negation and implication rules of natural deduction to combinatorial flows

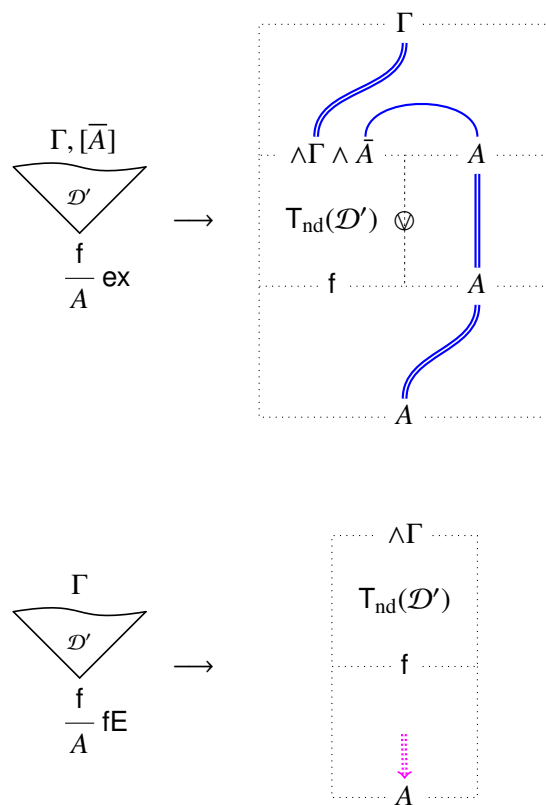


Figure 8.10: Translation of the rules *ex* and *fE* of natural deduction to combinatorial flows

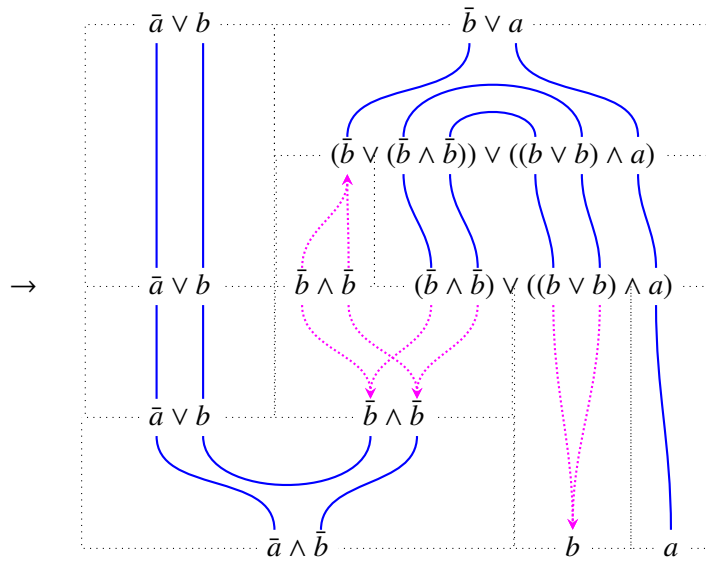
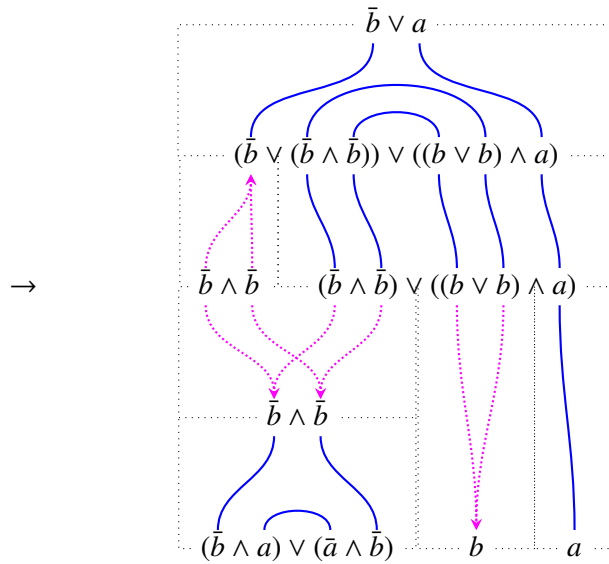
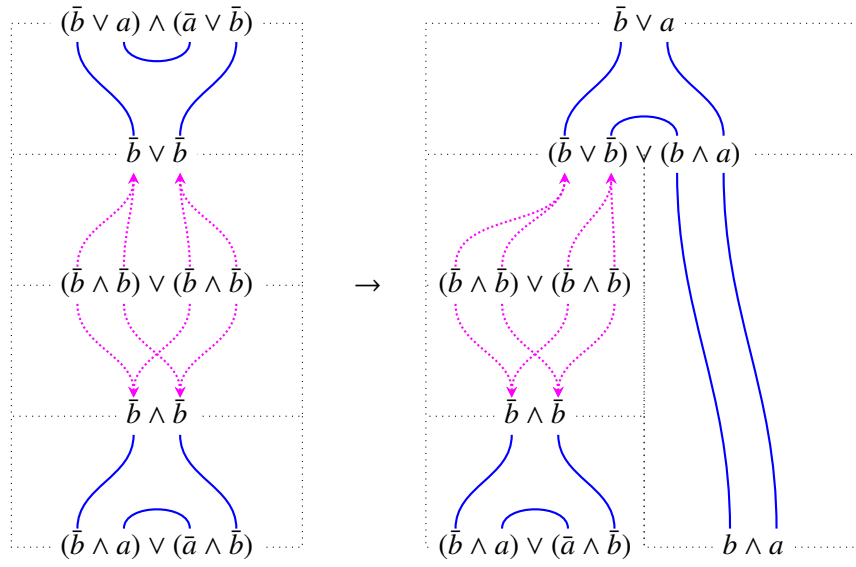


Figure 8.11: Example of applying “Currying” to a combinatorial flow.

CHAPTER 9

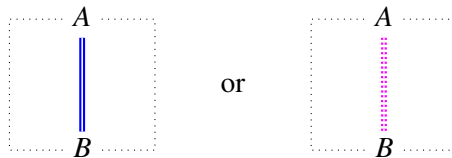
Normalization

In Chapter 2, we have introduced normal form of proofs especially for the proof systems deep inference, sequent calculus, and natural deduction as well as their applications in proof theory. As we mentioned before, normal proofs are viewed as a way of approaching the proof identity problem. Moreover, we gain more understanding of proofs and their structure by studying normalization.

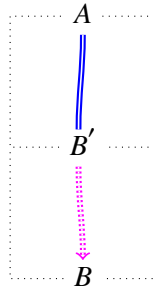
So far, we have shown how to normalize the compositions of m-flows and a-flows. However, as soon as these elements of combinatorial flows are composed freely, we need to take other approaches than naive composition. In fact, we have observed in Chapter 7 that it is not possible to compose a^\downarrow -flows (resp. a^\uparrow -flows) horizontally freely in the presence of units. The first step to doing normalization is to define the normal form of the proof:

Definition 9.1. A combinatorial flow $\phi: A \vdash B$ is *normal* if it has one of the shapes described below:

- It contains only one flowbox $\psi = \langle A, B, \mathbb{B}_\psi \rangle$ where ψ is either an m-flow or an a-flow.

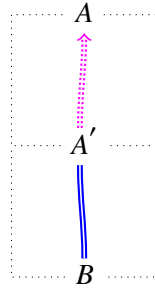


- It is the vertical composition of an m-flow $\psi = \langle A, B_1, \mathbb{B}_\psi \rangle$, and an a^\downarrow -flow $\chi = \langle B_1, B, f_\chi^\downarrow \rangle$ with $B_1 \equiv B' \equiv B_2$ (denoted as $\phi = \psi \odot \chi$).

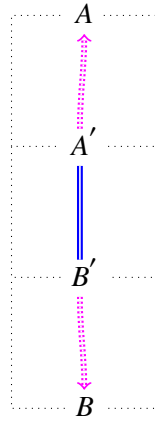


- It is the vertical composition of an a^\uparrow -flow $\psi = \langle A, A_1, f_\psi^\uparrow \rangle$, and an m-flow $\chi = \langle A_2, B, \mathbb{B}_\chi \rangle$

with $A_1 \equiv A' \equiv A_2$ (denoted as $\phi = \psi \odot \chi$).



- It is the vertical composition of an a^\uparrow -flow $\psi = \langle A, A_1, f_\psi^\uparrow \rangle$, an m-flow $\chi = \langle A_2, B_2, \mathbb{B}_\chi \rangle$, and an a^\downarrow -flow $\xi = \langle B_1, B, f_\xi^\downarrow \rangle$ with $A_1 \equiv A' \equiv A_2$ and $B_2 \equiv B' \equiv B_1$ (denoted as $\phi = \psi \odot \chi \odot \xi$).



The last shape in the previous definition is an important shape so we give the name **three-element normal form** to be able to refer to it later. A three-element normal form corresponds to combinatorial flows of [80] which have been the main inspiration for this thesis.

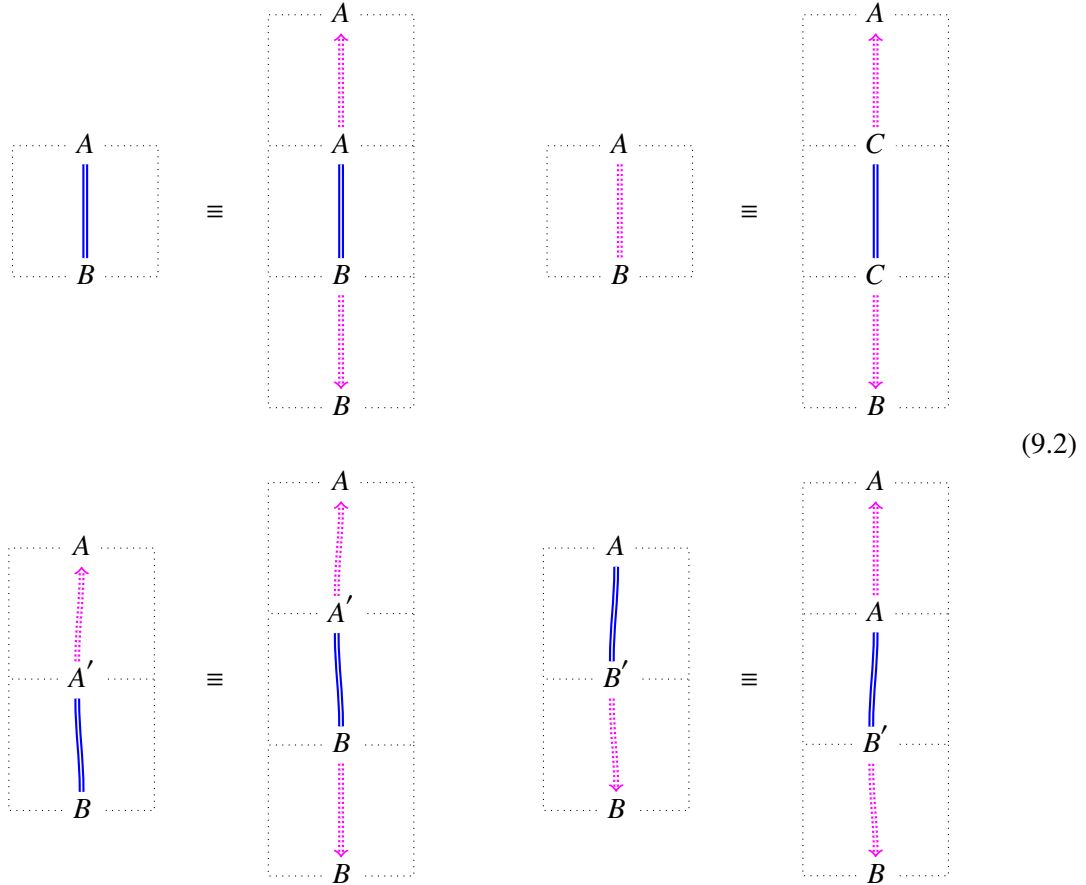
In Section 7.3 of Chapter 7, we have shown that a^\downarrow -flows and a^\uparrow -flows can be composed freely and this will result in a-flows (i.e. they are special cases of a-flows). Later in Chapter 8, we have shown that a-flows can be used as a base flowbox in the construction of combinatorial flows for more simplicity (the combinatorial flows take less space if they contain only a-flows and m-flows). We also have shown, in Theorem 7.30, that an a-flow can be decomposed into an a^\uparrow -flow and an a^\downarrow -flow and the resulting shape is shown below.¹ It can be observed that during the procedure of normalization shown in this chapter, we apply decomposition algorithm on every a-flow present.

(9.1)

¹Note that this decomposition is not unique, refer to Remark 7.31.

Proposition 9.2. *If a normal combinatorial flow $\phi: A \vdash B$ (where A and B are pure formulas and not units) is not a three-element normal form, there exists an equivalent three-element normal combinatorial flow with hypothesis A and conclusion B where the missing one or two elements are identity flows.*

Proof. For every normal form we show in (9.2) where the identity flows need to be added. Note that here identity flows are shown with flow-boxes that hypothesis and conclusion are the same but the links are colored (normally we show it with black) to refer to the element that they are representing. Note that in the a-flow case we are using Theorem 7.30 to decompose the flow first.



□

Knowing the normal form allows us to view normalization steps as successful vertical or horizontal composition of two normal combinatorial flows. Figure 9.1 shows the steps that we apply on a combinatorial flow $\phi: A \vdash B$ in order to obtain a normal combinatorial flow $\psi: A \vdash B$ that we call **normalized** version of ψ . In front of each step we have described the combinatorial flow that is the result of applying said step. We also show the notation used for each step.

Let $\phi: A \vdash B$ and $\psi: B \vdash C$ be combinatorial flows that are results of the Step 4 in the normalization steps, then we call B a **cut formula** for combinatorial flow $\phi \odot \psi$. In Section 8.1 of Chapter 8, we have shown the relation between combinatorial flows and combinatorial proofs and how to translate one to another. As mentioned before, combinatorial proofs do not have cuts or cut formulas by default. An auxiliary cut-conjunction is used for this purpose and the cut elimination procedure executes on such conjunctions. Moreover, upon closer inspection, one can observe that three-element normal combinatorial flows resemble combinatorial proofs and this has been detailed in [79] for the case that units are not present in the language. In Figure 9.2, we show how a

Start	Combinatorial flow ϕ	ϕ
Step 1 (Simplification)	Combinatorial flow ψ_0 which is obtained by applying simplification on ϕ	$\phi \rightarrow_s \psi_0$
Step 2 (Purification)	Purification of the simplification of ψ_0	$\psi_0 \rightsquigarrow_p^* \mathbf{p}(\psi_0)$
Step 3	Combinatorial flow ψ_1 which is a free composition of several three-element normal forms and it is equivalent to purification result	$\mathbf{p}(\psi_0) \equiv \psi_1$
Step 4 (Horizontal Composition)	Combinatorial flow ψ_2 which results after applying horizontal composition normalization on ψ_1 . Here ψ_2 is vertical composition of several three-element normal forms.	$\psi_1 \rightsquigarrow_{\otimes, \otimes}^* \psi_2$
Step 5 (Vertical Composition)	the normal combinatorial flow ψ obtained by applying vertical composition normalization on ψ_2	$\psi_2 \rightsquigarrow^* \psi$

Figure 9.1: Normalization steps to apply on a combinatorial flow

three-element combinatorial flow in the absence of units can be turned into a combinatorial proof. As we have shown in Proposition 9.2, any normal combinatorial flow is equivalent to a three-element normal combinatorial flow. In the presence of units, we add auxiliary atoms if needed in the form of $a \wedge \bar{a}$ to encode \mathbf{f} , and $a \vee \bar{a}$ to encode \mathbf{t} just as we did in Section 8.1. However, this procedure is not always needed. For example, if the \mathbf{m} -flow in a normal combinatorial flow has the hypothesis \mathbf{t} , then there is no need to encode this \mathbf{t} since by default a combinatorial proof is a proof of a tautology and there is no hypothesis. As a consequent of this, we can see that any combinatorial proof is already a normal combinatorial flow.

Lemma 9.3. *Let A, B, C be minimal pure formulas with $B \neq \mathbf{f}$. Then there is a one-to-one correspondence between normal combinatorial flows $\phi: A \vdash B \vee C$ and $\psi: A \wedge \bar{B} \vdash C$.*

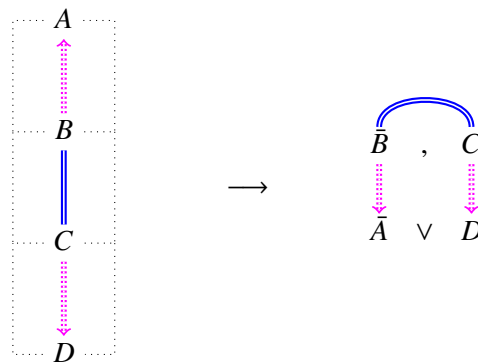


Figure 9.2: Translation of a three-element normal combinatorial flow to a combinatorial proof

Proof. This follows immediately from the Definition 9.1 of normal combinatorial flows and Proposition 6.4 and Lemma 7.5 (accompanied by Definition 7.8 on a^\downarrow -flows and a^\uparrow -flows). \square

The first step in the normalization steps is simplification shown in Chapter 8. We will show the second step (i.e. purification) in Section 9.1 and the last step (i.e. vertical composition) in Section 2.4. The third step is applying the Proposition 9.2 on the result of purification. In the remaining of this section, we will show the fourth step (i.e. horizontal composition normalization).

Theorem 9.4. *Let $\phi: A \vdash C$ and $\psi: B \vdash D$ be normal combinatorial flows, with $A, B, C,$ and D as unit-free formulas, which are not composed of any elements with units as hypothesis or conclusion. Then, there exist normal combinatorial flows $\chi: A \wedge B \vdash C \wedge D$ and $\omega: A \vee B \vdash C \vee D$ where χ is obtained by the conjunction normalization on the combinatorial flow $\phi \circledast \psi$ and ω is obtained by the disjunction normalization on the combinatorial flow $\phi \circledast \psi$ and the size of χ and ω is less or equal to the size of ϕ plus size of ψ (i.e. $|\chi| \leq |\phi| + |\psi|$ and $|\omega| \leq |\phi| + |\psi|$).*

Proof. We proceed with first using Proposition 9.2 to find three-element normal combinatorial flows $\phi' = \phi_1 \circledast \phi_2 \circledast \phi_3$ and $\psi' = \psi_1 \circledast \psi_2 \circledast \psi_3$ with $\phi' \equiv \phi$ and $\psi' \equiv \psi$ (note that this procedure is not affecting the size of the flows and we just add identity flows where needed) where ϕ_1 and ψ_1 are a^\uparrow -flows, and ϕ_2 and ψ_2 are m-flows, and ϕ_3 and ψ_3 are a^\downarrow -flows. Using Theorems 7.23, and 6.6, we have that $\chi = (\phi_1 \wedge \psi_1) \circledast (\phi_2 \wedge \psi_2) \circledast (\phi_3 \wedge \psi_3)$ and $\omega = (\phi_1 \vee \psi_1) \circledast (\phi_2 \vee \psi_2) \circledast (\phi_3 \vee \psi_3)$ are three-element normal combinatorial flows depicted in Figure 9.3 where in the middle we show ϕ' and ψ' and the arrows to right and left show the horizontal compositions $\phi' \circledast \psi'$ and $\phi' \circledast \psi'$. \square

Remark 9.5. In the presence of units the normalization on horizontal composition does not follow the same procedure as the previous theorem. In Section 9.1, we address this in more detail and we will show rewriting steps needed to be taken if a combinatorial flow contains units. However, it is important to note that in some cases we are still able to apply the same normalization on horizontal composition as the previous theorem.

Let $\phi: A \vdash C$ and $\psi: B \vdash D$ be normal combinatorial flows where at least one of the hypothesis A or B is the unit t and the remaining formulas are unit-free. Then, there exists a normal combinatorial flow $\chi: B \vdash C \wedge D$ obtained by applying conjunction normalization on the combinatorial flow $\phi \circledast \psi$. This case is possible because the flow with t as hypothesis does not contain a meaningful a^\uparrow -flow, its a^\downarrow -flow can not have a t as hypothesis, and the conjunctive composition of the two m-flows is possible, shown in Theorem 6.6. For the case that at least one of the conclusions C or D is the unit f and the remaining formulas are unit-free, the disjunction normalization on $\phi \circledast \psi$ is possible dually.

Example 9.6. Figure 9.4 shows an example of the algorithm shown in the proof of Theorem 9.4 where $\phi: a \vee a \vdash a \wedge a$ and $\psi: a \wedge \bar{a} \vdash a \wedge (\bar{a} \vee a)$, shown in the middle row, are normal combinatorial flows that are not three-element normal forms. We write their three-element normal combinatorial flow counterparts in the first step using Theorem 9.2, on bottom left. Next, we can perform the disjunction normalization using Theorem 9.4 to acquire $\phi \vee \psi$ by performing disjunctive composition \vee on a^\uparrow -flows, m-flows, and a^\downarrow -flows. Note that the end result of this example, therefore normalization, is not uniquely defined. As mentioned in the Decomposition Theorem 7.30, we made a choice on how to decompose a-flows which then affects the procedure of normalization and causes non-confluence. The other normalization path is shown on the top row.

Example 9.7. Below we depict two normal combinatorial flows where the procedure in Theorem 9.4 cannot be applied because of the existence of unit t in the conclusion of the flow in right. If we were to try the naive composition, the result would be a combinatorial flow with the conclusion $a \vee t$ which is not a pure formula.

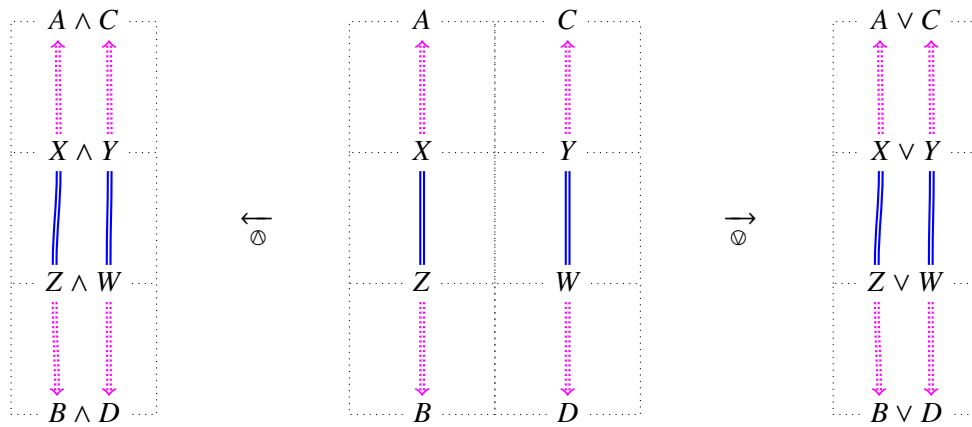
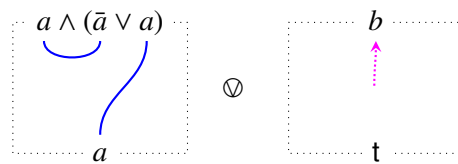


Figure 9.3: Horizontal compositions of normal combinatorial flows



In this example the left combinatorial flow is just an m -flow but this is not the main reason for inability of normalization on the composition. To prove this point, we can refer to Lemma 7.4 which shows that skew fibrations, therefore a^\downarrow -flows and a^\uparrow -flows, cannot be composed naively in the presence of a unit.

Observing the latest examples finally reveals the problem that arise by adding units to the language and the naive attempt of normalization. Therefore, in the following section we go into details on how to approach normalization in the presence of units.

9.1 Purification

We introduced the notion of *pure* in Chapter 1. Moreover, the notion of correctness criteria for additive and multiplicative flows can not exist if the formulas are not pure for the following reasons:

1. In the multiplicative case, we no longer have a canonical representation of an m -flow (see e.g. [7, 48, 60]) and checking equivalence is PSPACE-complete [39].
2. In the additive case, the horizontal composition can break the skew fibration property or coalescence or pruning, when units are involved. This is the reason for the side condition in Lemma 7.4.

As we showed in the examples in previous section, with the equivalence \equiv , some units occurring in formulas can be removed, but not all. And the presence of these units can block further simplification in combinatorial flows. We have $A \wedge f \not\equiv f$ and $A \vee t \not\equiv t$ because otherwise \equiv would change the number of atoms in a formula, and therefore break Propositions 4.9 and 4.11. However, the logical equivalences $A \wedge f \iff f$ and $A \vee t \iff t$ can be used in the setting of normalization. Using these equivalences, we can remove all units from a combinatorial flow, and we call this process

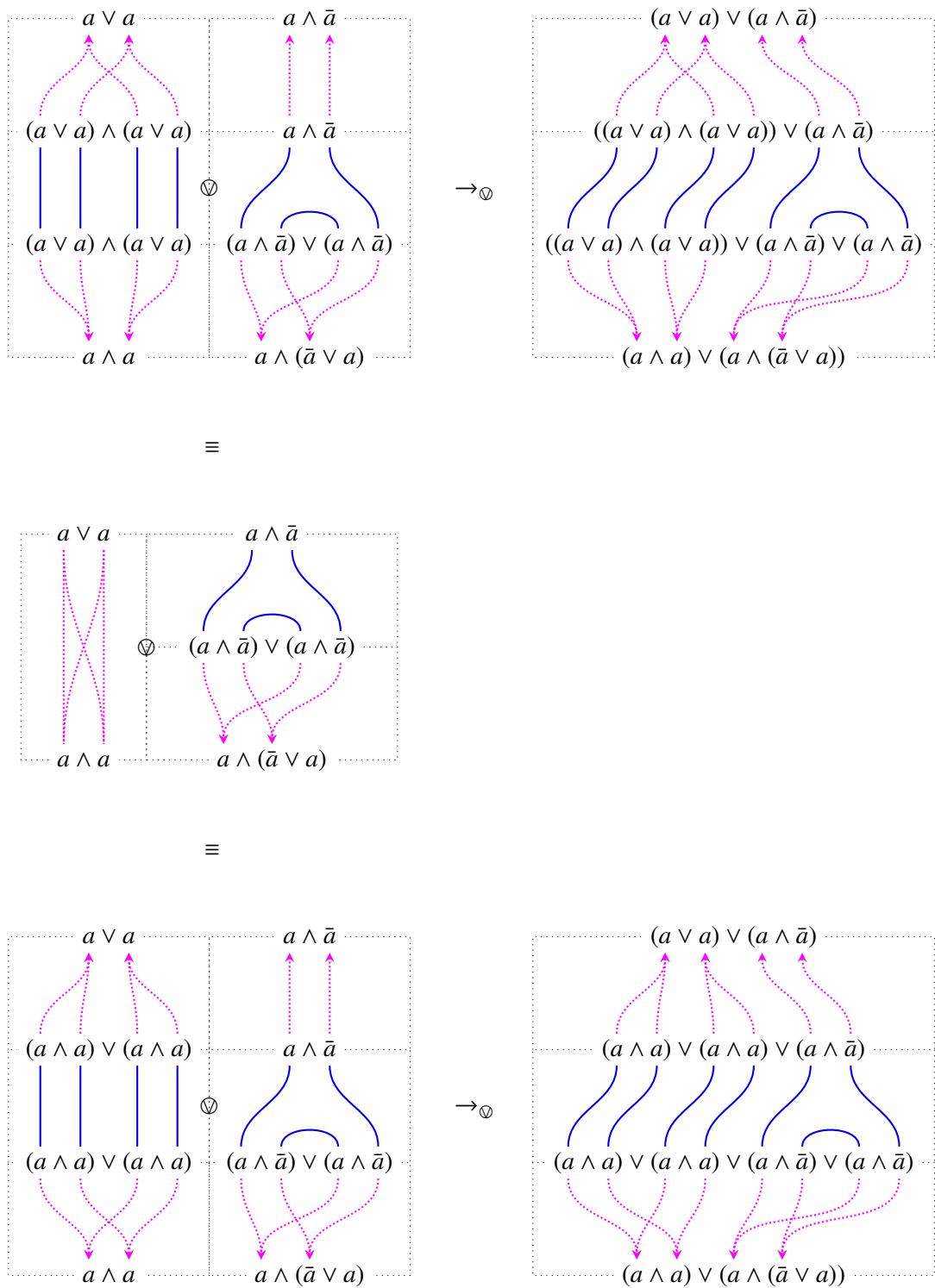
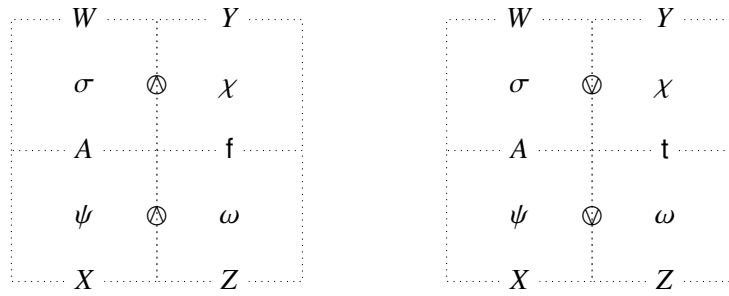
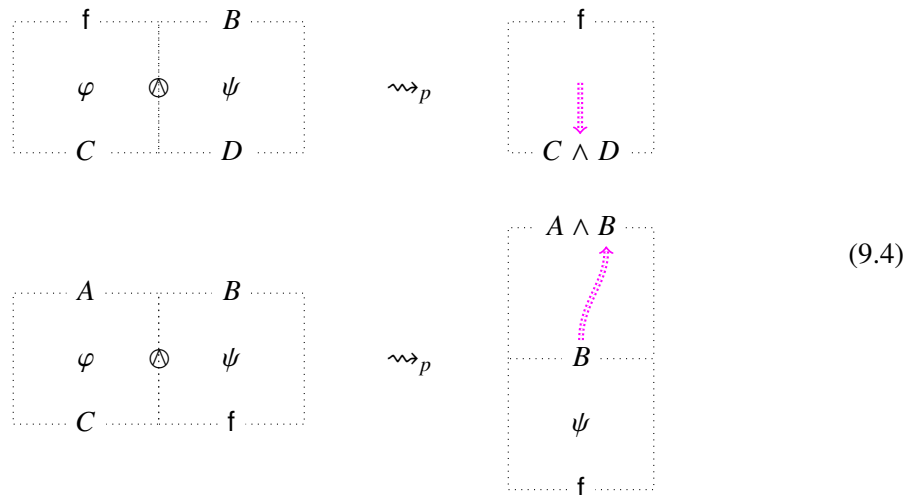


Figure 9.4: A disjunctive composition of two normal combinatorial flows and its normalization

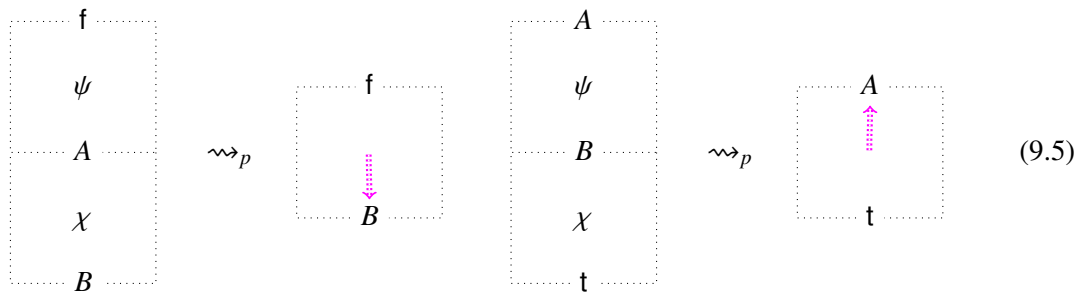
purification. The combination of purification with horizontal composition normalizations results in vertical composition of finite number of normal combinatorial flows.



After identifying the smallest subformula F that is not pure ($F = A \wedge f$ or $F = A \vee t$ for some pure formula A) of a non-pure slice, we can use rewriting rules on the horizontal composition of σ and χ , and horizontal composition of ψ and ω which are flowboxes containing F . We have listed all possible cases and designed a rewrite system that extends (9.3) to combinatorial flows. A list of these cases can be found in Figures 9.7, 9.8, and 9.9 for the horizontal composition using conjunction. Note that the cases for disjunction are dual to conjunction composition. Below are two representative cases (different cases and more details are presented in the Figure Explanation 9.1):



Moreover, let $\psi: f \vdash A$ be a flowbox which is identified inside ϕ and is vertically composed with another flowbox $\chi: A \vdash B$ where A and B are pure formulas, we can use the rewriting step shown below left and return a weakening flow (i.e. an a^\downarrow -flow flowbox that has the shape of a deep inference weakening rule aw^\downarrow).

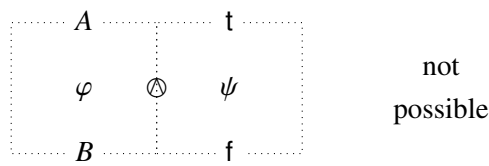


To the right, we show the dual case which is for t in conclusion which rewrites it to a single co-weakening. Each rewriting step reduces the number of flowboxes in ϕ that have a unit as premise

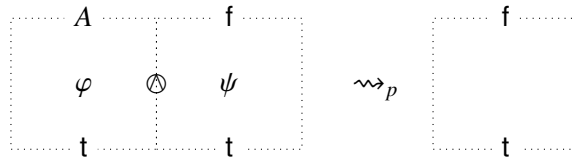
or conclusion, or reduces the size of the formula that containing the unit. Therefore the rewriting is terminating and the result combinatorial flow is pure. \square

Discussion: Here we are going to explain the purification rewriting steps in more details. Let φ and ψ be the flowboxes in question for rewriting. Then each hypothesis and conclusion of each of them have three different possibilities: it is a unit-free formula, or it is the unit f , or it is the unit t . Consequently, there exists $3^4 = 81$ cases of the appearance of $\varphi \otimes \psi$. All these cases and their rewriting step are depicted in Figures 9.7, 9.8, and 9.9. Upon closer inspection, one can notice that there exists only seven main patterns of rewriting steps. We show these patterns in more detail below:

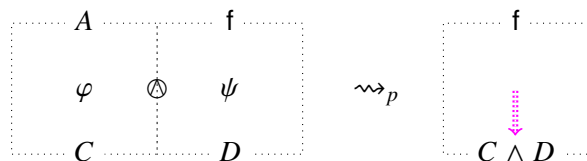
1. At least one of the flowboxes has hypothesis t and conclusion f : By Theorem 8.5, every combinatorial flow is sound. Hence, such combinatorial flow cannot exist:



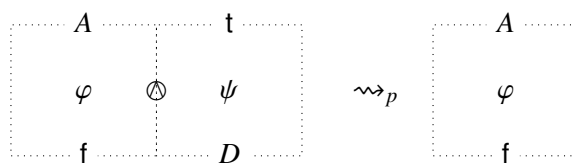
2. The purification on both hypothesis and conclusion return a unit resulting in an empty combinatorial flow.



3. The purification on only either hypothesis or conclusion returns a unit and this unit is allowed in skew fibration (i.e. t for conclusion, and f for hypothesis). Consequently, the result is a weakening or co-weakening. We delete everything and replace it with only one weakening or one co-weakening.



4. the purification on hypothesis and conclusions returns both the hypothesis and conclusion of one of the flowboxes. We will proceed by deleting the remaining flowbox.



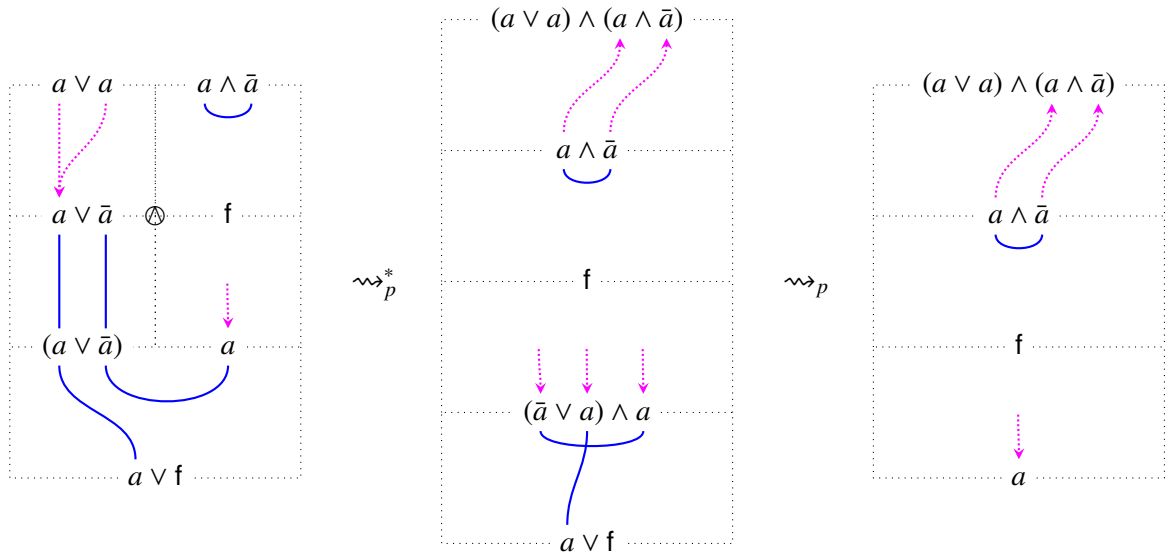


Figure 9.5: Purification Normalization Example

5. In this pattern, either only one of the conclusions is f , or only one of the hypothesis is t and the conclusion of the opposite proof is t . The example for this pattern is shown in (9.4) (on the bottom).
6. None of the above happens: we can keep both flowboxes and perform horizontal composition shown in Theorem 9.4. From another point of view, this pattern only happens if the purification on hypothesis and conclusions have no meaningful changes.

The purification procedure is not confluent and we have depicted an example of this in Figure 9.6.

Theorem 9.12. *Let $\phi: A \vdash B$ be a combinatorial flow. Then, there exists a finite set P of purification and horizontal normalization steps $\phi \rightsquigarrow_P \psi$ where $\psi = \psi_1 \odot \dots \odot \psi_n$ is a pure combinatorial flow with ψ_1, \dots, ψ_n normal combinatorial flows.*

Proof. First we perform the purification steps, Using Theorem 9.11, to obtain $\phi \rightsquigarrow_P^* \mathbf{p}(\phi)$. Next, we execute the horizontal normalization shown in Theorem 9.4 on $\mathbf{p}(\phi)$ to get $\mathbf{p}(\phi) \rightarrow_{\odot, \otimes}^* \psi$. This procedure terminates since in every step the number of horizontal composition in the flow reduce at least by one. Moreover, ψ is vertical composition of normal combinatorial flows because at the end there is no horizontal composition left and it is pure because the second step of normalization starts with a pure combinatorial flow and horizontal normalization steps never produce a non-pure slice in the combinatorial flow (in fact the number of units stay the same or decreases.) \square

The combinatorial flow on the right in Figure 8.2 is a purification of the one on the left. Moreover, in Figure 9.5 we show an example of steps taken in Theorem 9.12.

9.2 Cut Normalization

In the previous two sections, we defined cut formulas and introduced an algorithm for horizontal composition. As already mentioned, there are four normalization steps (i.e. simplification, purification, equivalences, and horizontal normalization) before arriving at vertical normalization. The

immediate result of the previous steps is a pure combinatorial flow that only has vertical compositions of smaller three-element normal combinatorial flows. In this section, we show the vertical normalization, referred to as cut normalization in sequent calculus, of a combinatorial flow which is an inductive process on all existing cut formulas in said flow.

Lemma 9.13. *Let $\phi: A \vdash B \wedge C$ be a normal combinatorial flow, then there exists combinatorial flows $\phi_l: A \vdash B$ and $\phi_r: A \vdash C$ which are obtained in polynomial time from ϕ and the size of each flowbox in both ϕ_l and ϕ_r are less than the size of each flowbox in ϕ .*

Proof. The existence of projections is not surprising with the knowledge that \wedge right introduction inference rule in sequent calculus is invertible. In other words, we should be able to recover proof of B and proof of C from proof of $B \wedge C$. We outline the algorithm of searching for empire of C (i.e. the largest subflow containing C as conclusion) which has been shown in more detail in [79] (Construction 6.5 Page 9).² First we write ϕ as a three-element normal form, using Proposition 9.2. Let us name the skew fibration function of the a^\downarrow -flow in ϕ as f_ϕ^\downarrow . By Lemma 7.5, we can write f_ϕ^\downarrow as the conjunction of two smaller skew fibration, with co-domains $\mathcal{G}(B)$ and $\mathcal{G}(C)$. Let \mathcal{G} be the underlying graph of the m-flow in ϕ . Let $f_\phi^{\downarrow^{-1}}(C)$ be the set of all vertices in \mathcal{G} that are mapped by f_ϕ^\downarrow to an atom occurrence of C . Let $\mathcal{E}_C \subseteq \mathcal{G}$ be the smallest set of vertices such that:

- $f_\phi^{\downarrow^{-1}}(C) \subseteq \mathcal{E}_C$
- If $v \in \mathcal{E}_C$ and $v \mathbb{B}_{\mathcal{G}} w$ and $w \notin C$, then $w \in \mathcal{E}_C$.
- If $V_1, V_2 \subseteq V_{\mathcal{G}}$ are modules of \mathcal{G} and $V_1 \cap V_2 = \emptyset$ and $V_1 \cup V_2$ is a module such that for all $v_1 \in V_1$ and $v_2 \in V_2$ we have that $v_1 \mathbb{R}_{\mathcal{G}} v_2$, then we also have that $V_2 \subseteq \mathcal{E}_C$.

Now let the m-flow part of ϕ_l be the flowbox with underlying graph which is the restriction of \mathcal{G} to the vertices $V_{\mathcal{G}} \setminus \mathcal{E}_C$. We also restrict the a^\downarrow -flow and a^\uparrow -flow to the new vertices.

Based on this construction, it is easy to show that the restriction of all flowboxes still follow the same correctness criterion. Therefore ϕ_l is a three-element normal combinatorial flow with hypothesis A and conclusion B . We can define ϕ_r similarly. \square

Dually, we can construct the right and left projections if there is a disjunctive formula in hypothesis of a normal combinatorial flow:

Lemma 9.14. *Let $\phi: A \vee B \vdash C$ be a normal combinatorial flow, then there exist two combinatorial flows $\phi_l: A \vdash C$ and $\phi_r: B \vdash C$ which are obtained in polynomial time from ϕ and both the size of each flowbox in both ϕ_l and ϕ_r are less than the size of each flowbox in ϕ .*

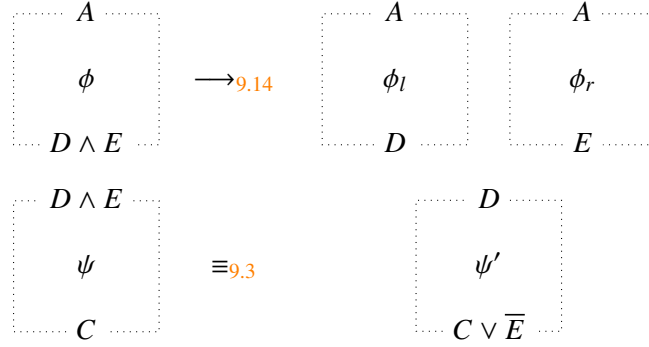
Proof. Dual to the proof of previous lemma. \square

Lemma 9.15. *Let $\phi: A \vdash B$ and $\psi: B \vdash C$ be three-element normal form combinatorial flows, then there is a normal combinatorial flow $\chi: A \vdash C$ (and we denote this as $\phi \rightsquigarrow^* \psi$).*

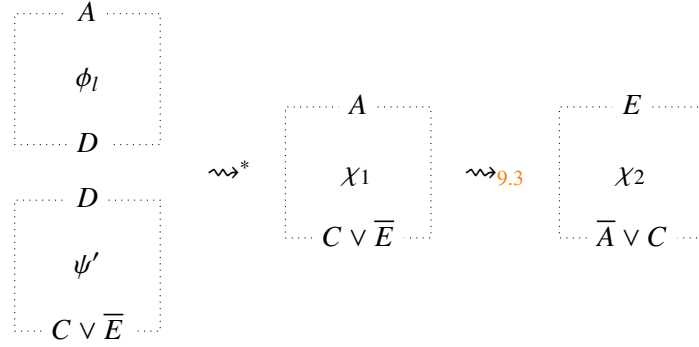
Proof. We proceed by induction on the cut formula B . If $B = D \wedge E$, first we write the left and right projections of $\phi: A \vdash D \wedge E$ in three-element normal form by Lemma 9.14 and from $\psi: D \wedge E \vdash C$ we obtain $\psi': D \vdash \overline{E} \vee C$ in three-element normal form by Lemma 9.3. Note that all three new

²Other constructions can be viewed in [49] where mix is present and the projection is shown on lax combinatorial proofs and in [6] with empire construction on MLL proof-nets.

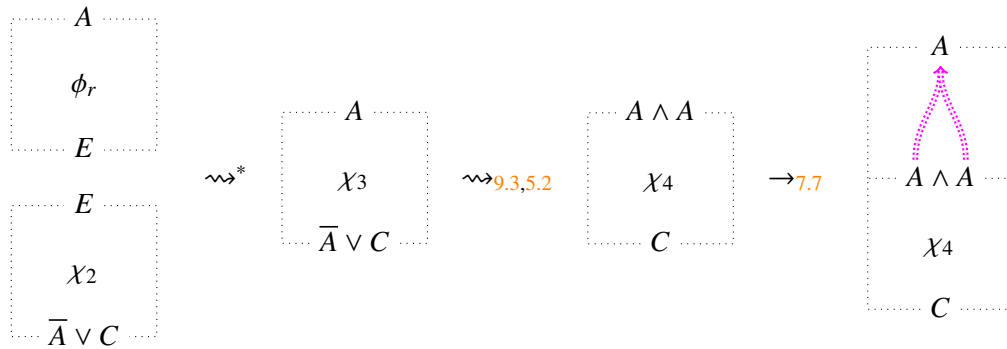
normal combinatorial flows are in three-element normal form obtained by Theorem 9.2. Below we show these combinatorial flows schematically:



Next, we compose ϕ_l with ψ' to get $\chi_1: A \vdash C \vee \bar{E}$ by induction hypothesis and by applying Lemma 9.3 on χ_1 , we obtain $\chi_2: E \vdash \bar{A} \vee C$. This procedure is shown below:



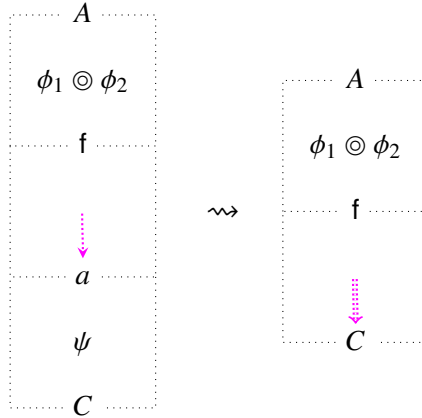
Next, the combinatorial flow χ_2 can be composed with ϕ_r inductively to obtain $\chi_3: A \vdash \bar{A} \vee C$ from which we obtain $\chi_4: A \wedge A \vdash C$ by using Lemma 9.3. As the last step we vertically compose χ_4 with a co-contraction flowbox (i.e. an a^\uparrow -flow flowbox which has the same shape as deep inference co-contraction rule c^\uparrow) from A to $A \wedge A$ to get χ . We show these steps in drawing below. Note that by composing a co-contraction with a normal combinatorial flow we obtain a normal combinatorial flow using the equivalences in (8.1).



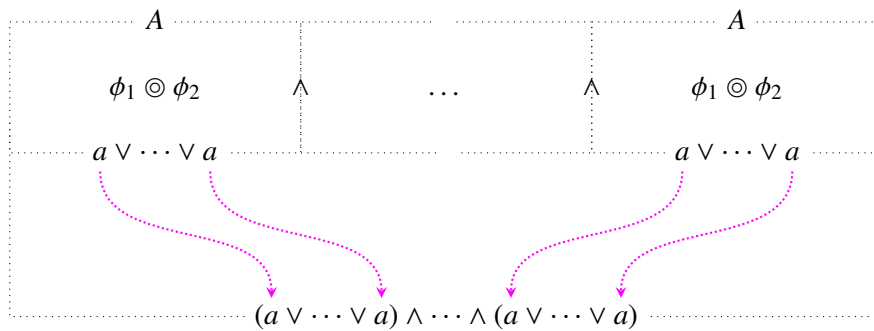
If $B = D \vee E$, we proceed dually to the conjunctive case. In the case that $B = \mathbf{t}$, then ϕ does not have an a^\downarrow -flow and ψ does not have an a^\uparrow -flow (unless they are identity flows or flows with hypothesis \mathbf{f} and conclusion \mathbf{t}), therefore the vertical composition is trivial by first doing the normalization on m-flows. The case that $B = \mathbf{f}$ is treated dually. If $B = a$ where a is an atom, then we write $\phi = \phi_1 \odot \phi_2 \odot \phi_3$ and $\psi = \psi_1 \odot \psi_2 \odot \psi_3$ where ϕ_1 and ψ_1 are a^\uparrow -flows, ϕ_2 and ψ_2 are m-flows,

and ϕ_3 and ψ_3 are a^\downarrow -flows. To continue we treat normalization based on different patterns of ϕ_3 shown below:

- if ϕ_3 is a weakening $\phi_3 = \langle f, a, f_{\phi_3}^\downarrow \rangle$, we use the rewriting step (9.5) on $\phi_3 \odot \psi$ to obtain a a^\downarrow -flow $\chi_1 = \langle f, C, f_{\chi_1}^\downarrow \rangle$ and then composing $\phi_1 \odot \phi_2$ with χ_1 by using the same pattern as shown in (9.5) which results in χ :

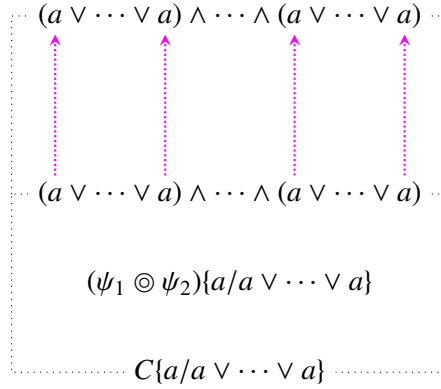


- If ϕ_3 is not a weakening but ψ_1 is a co-weakening $\psi_1 = \langle a, t, f_{\psi_1}^\uparrow \rangle$. This case gets resolved dual to the previous case.
- If the previous two cases do not hold, then ϕ_3 is a contraction $\phi_3 = \langle a \vee \dots \vee a, a, f_{\phi_3}^\downarrow \rangle$ and ψ is a co-contraction $\psi_1 = \langle a, a \wedge \dots \wedge a, f_{\psi_1}^\uparrow \rangle$ and we proceed as follows. Let n be the number of occurrences of a in hypothesis of ϕ_3 and m the number of such occurrences in the conclusion of ψ_1 . We proceed by first making m copies of $\phi_1 \odot \phi_2$ and composing them horizontally by conjunctive composition. Then we advance by composing the result with an identity flow as shown below:

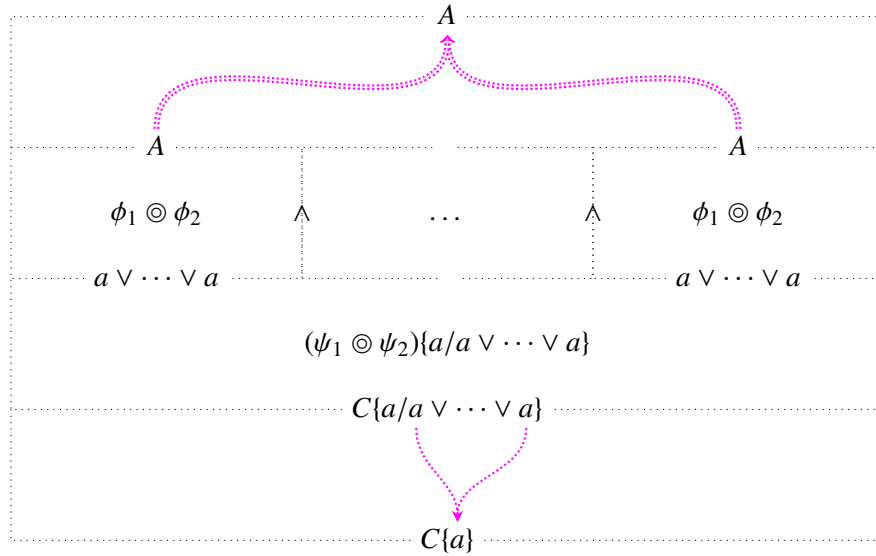


Next, we replace every link that starts with an instance of a in hypothesis in ψ with identity

flow $\text{id}_{a \vee \dots \vee a} = \langle a \vee \dots \vee a, a \vee \dots \vee a, \mathbb{B}_{\text{id}} \rangle$ shown below:



Now, we can observe that the composition of the two recent flows is easily possible since we turned the additive flows to identity flows therefore $(a \vee \dots \vee a) \wedge \dots \wedge (a \vee \dots \vee a)$ is not a cut formula. Next, we just need to add contractions $a \vee \dots \vee a \vdash a$ and co-contraction $A \vdash A \wedge \dots \wedge A$ to the flow and arrive at the resulting normal combinatorial flow, show below:



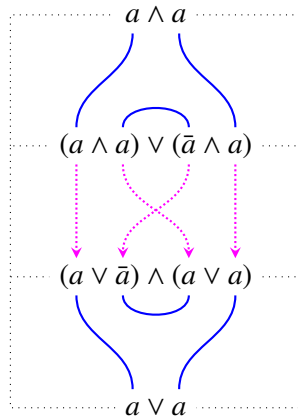
□

Theorem 9.16. *Let $\phi: A \vdash B$ be a combinatorial flows, then there exists combinatorial a combinatorial $\psi: A \vdash B$ which is the normalized version of ϕ .*

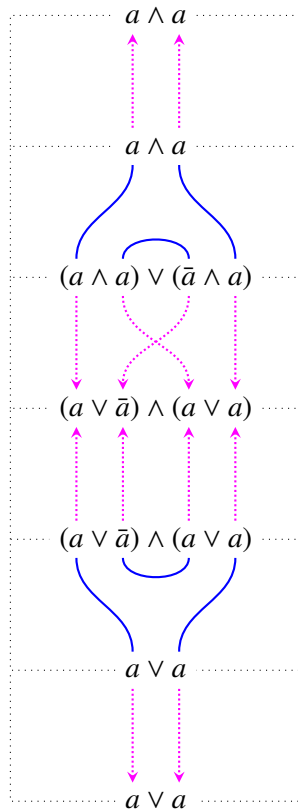
Proof. We can proceed by using definition of simplification in previous chapter (p. 73), Theorem 9.11, Equivalences on combinatorial flows (8.1) (p. 73), Theorem 9.4, and Lemma 9.15. If we apply all these steps in the same order as Figure 9.1, we can obtain ψ which is the normalized version of ϕ and ψ is in three-element normal form. □

Example 9.17. Below we depict a combinatorial flow $\phi: a \wedge a \vdash a \vee a$ which consists of vertical composition of two smaller normal combinatorial flows $\psi: a \wedge a \vdash (a \vee \bar{a}) \wedge (a \vee a)$ and $\chi: (a \vee$

$$\bar{a}) \wedge (a \vee a) \vdash a \vee a.$$

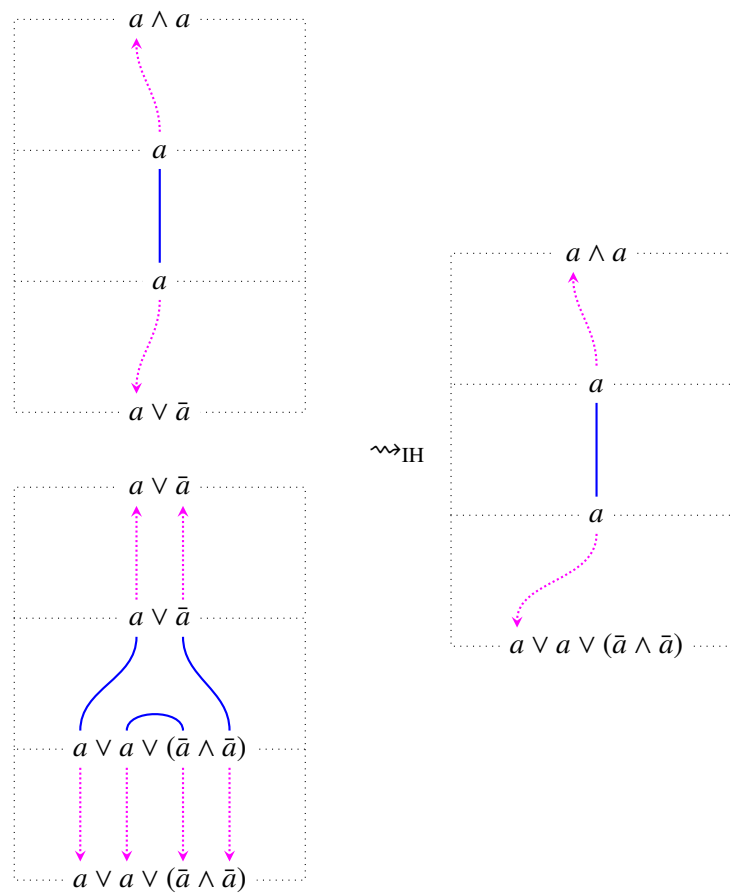


There is no need for first and second step in the example as it is simplified and pure. So we write the three-element normal forms for ψ and χ to form their composition ϕ as shown below:



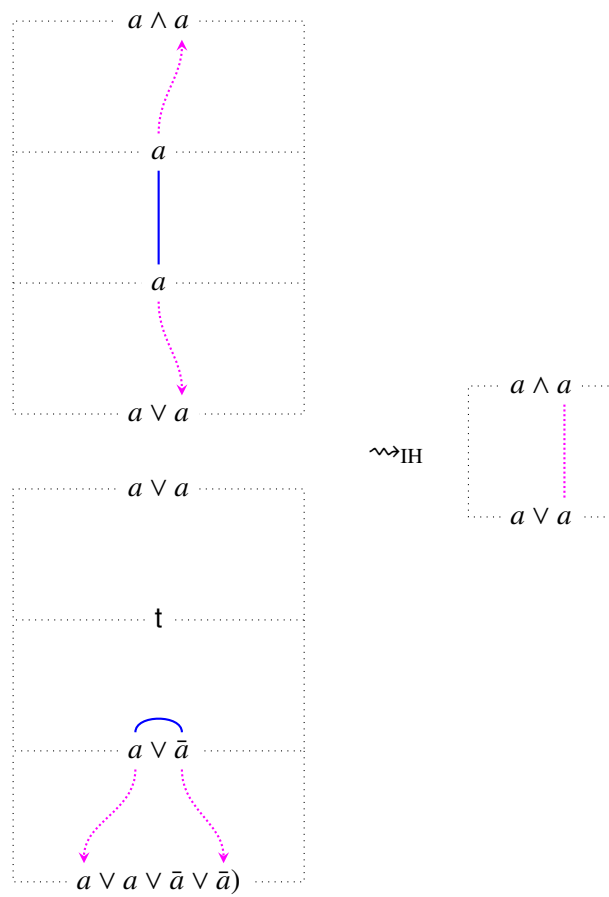
The first step is to write the projection ψ_l as depicted below in top left and do cut elimination inductively as well as applying Theorem 9.3 on χ (yanking of subformula $a \vee a$) depicted below bottom left to obtain the flow $\chi' : a \vee \bar{a} \vdash a \vee a \vee (\bar{a} \vee \bar{a})$. Next, we perform normalization inductively

on $\psi_l \odot \chi'$ to obtain the normal combinatorial flow π shown below right.



Next, we write the projection ψ_r as depicted below left as well as π' which is obtained by yanking the subformula $\bar{a} \wedge \bar{a}$ to the top on and yank all the hypothesis to the bottom in π . Finally we normalize combinatorial flow $\psi_r \odot \omega$ by induction to obtain the normalized version of the flow ϕ

shown below right.



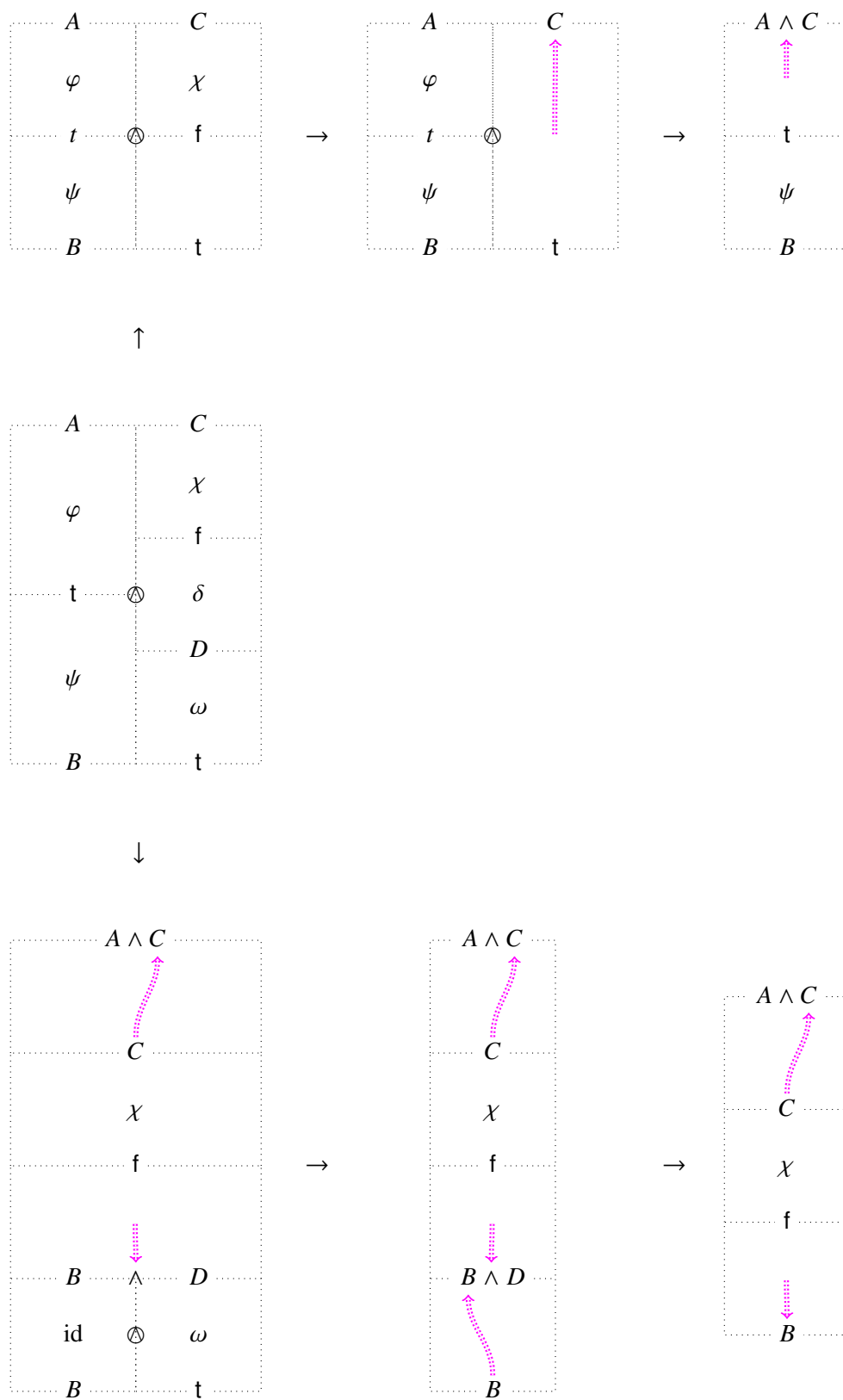


Figure 9.6: purification is not confluent

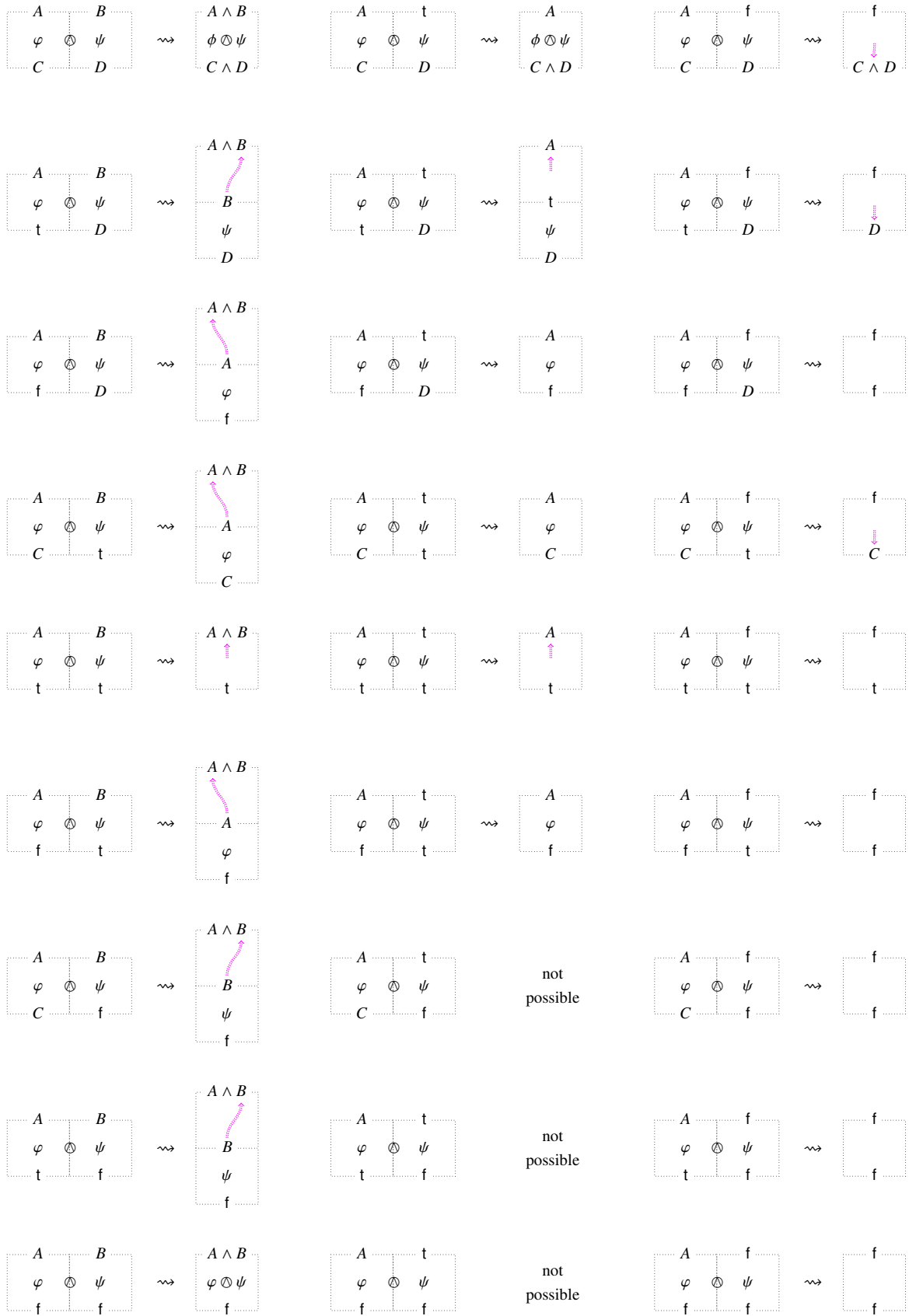


Figure 9.7: Purification cases for conjunction where the hypothesis of φ is unit-free formula A

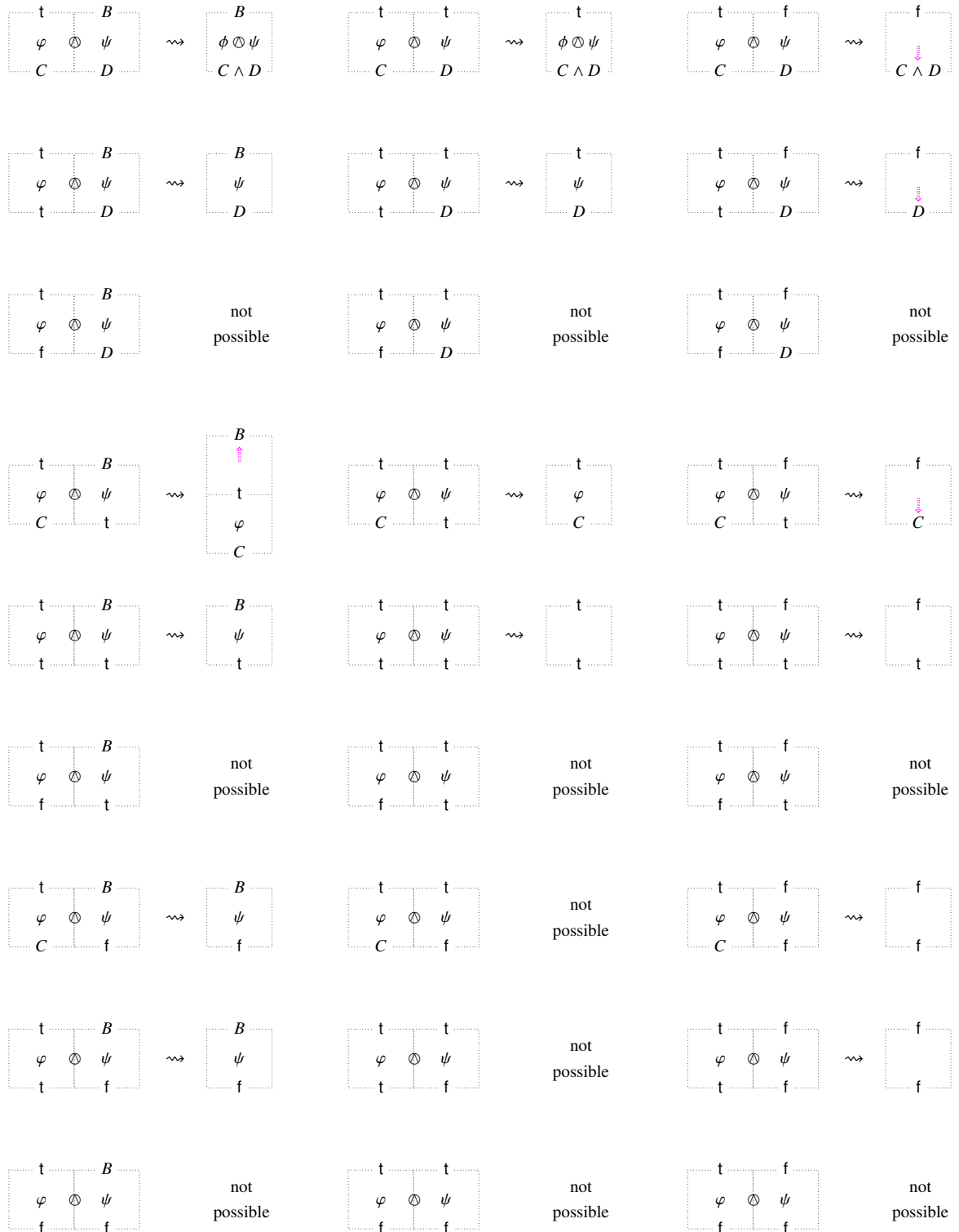


Figure 9.8: Purification cases for conjunction where the hypothesis of φ is t

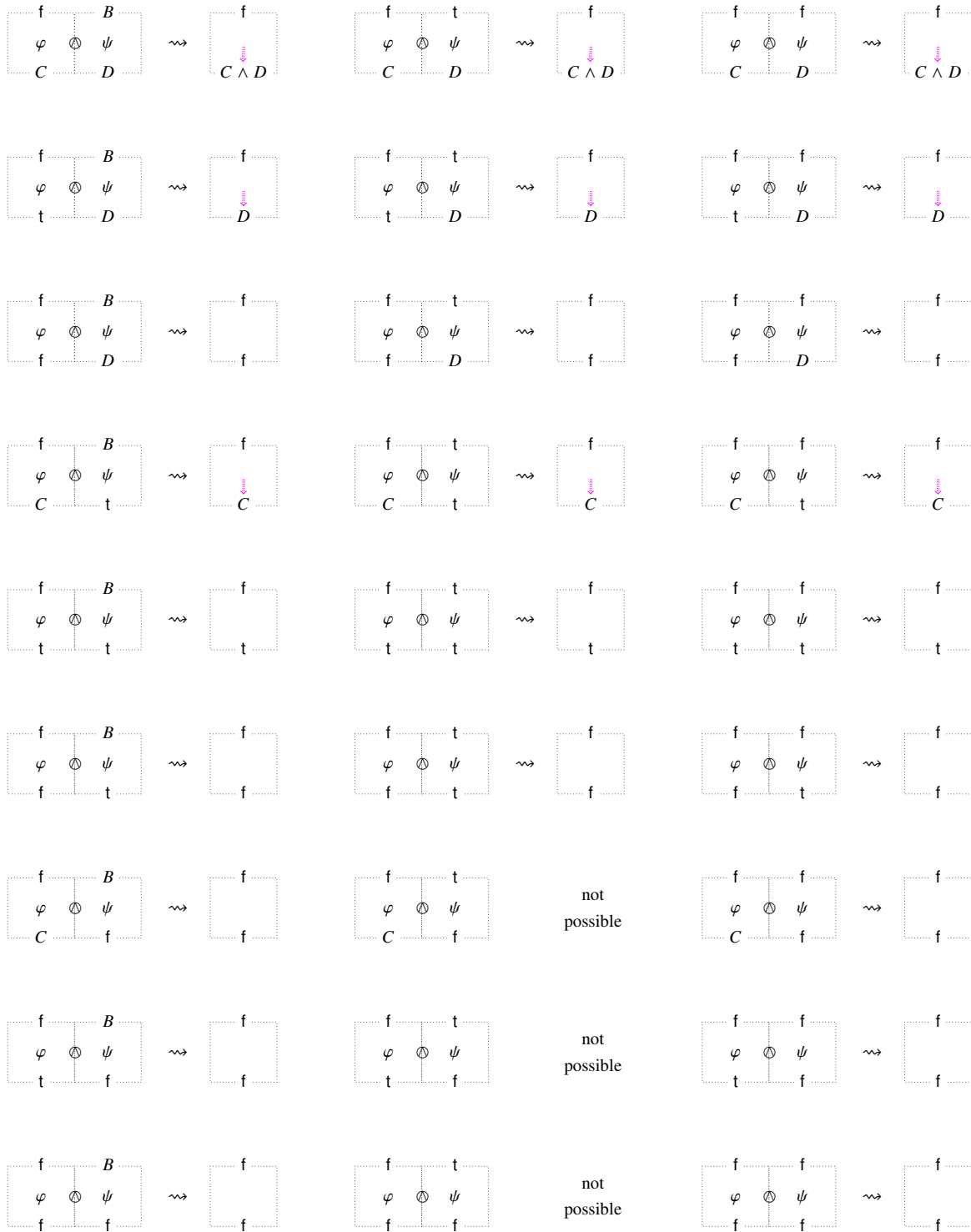


Figure 9.9: Purification cases for conjunction where the hypothesis of φ is f

CHAPTER 10

Conclusion and Future Work

In our investigation into the problem of proof identity, we introduced combinatorial flows as a graphical representation of classical proofs, enhancing our understanding of proof structures. In the initial segment of this thesis, we laid the groundwork by defining the formal language and proof systems suitable for classical logic. This encompassed natural deduction, sequent calculus, and deep inference. Additionally, we introduced key concepts such as atomic flows [32] and RB-cographs [73], and we leveraged them combinatorial proofs [47] as foundational elements for the development of combinatorial flows.

In the second part, we delved deeper into the interplay between the multiplicative and additive fragments of classical logic within our setting. We explored the possibility of incorporating units into the language, acknowledging that this choice would impact the compositional nature of proofs. We demonstrated that with specific adjustments, both multiplicative and additive flows could adhere to the same correctness criteria established in combinatorial proofs. For multiplicative flows, we focused on ensuring α -acyclicity in RB-cographs while accommodating the existence of the mix rule and units. In the case of additive flows, we introduced the concept of a skew fibration and enhanced it to accommodate empty flows, which mimic the behavior of weakening. Collaborating with Willem Heijltjes and Lutz Straßburger, we employed additive nets [38] to delve into the distinctions between the two fragments and explore the potential for a more localized, terminating normalization process, though this endeavor presented notable challenges.

In the third part of this thesis, we unified these foundational elements to form combinatorial flows, providing insights into their relationship with various proof systems, including combinatorial proofs with cuts, natural deduction, sequent calculus, and deep inference. While the elusive notion of proof identity remains a complex challenge, combinatorial flows offer valuable insights into the structures of classical logic proofs. Notably, we observed that combinatorial flows do not encompass a notion of rule permutation, as proofs with such permutations can be translated into the same combinatorial flow. The quest for local normalization remains an open problem, but we have made significant strides, incorporating techniques such as purification (tailored for compositions on combinatorial flows with units), as well as horizontal and vertical composition normalization, to establish general normalization procedures.

The notion of combinatorial proofs has been introduced for the logics listed below as a gathering effort of mainly the following people: Matteo Acclavio, Willem Heijltjes, Dominic Hughes, and Lutz Straßburger.

- Propositional Classical Logic [47]
- First-order Logic [52, 50, 46]
- Intuitionistic Logic [41]

- Modal Logics [4]
- Constructive Modal Logic [2]
- Logics of Relevance and Entailment [3]
- Multiplicative Linear Logic [48]
- Additive Linear Logic [40]
- Multiplicative Exponential Linear Logic [1]

Extending the definition for these logics into combinatorial flows remains to be investigated. In fact, with Acclavio we have started on the development of a satisfactory definition of additive flows for modal logics. Moreover, we believe that notions of proof identity and proof comparison could be achieved by further studies on the properties of combinatorial flows.

Bibliography

- [1] Matteo Acclavio. “Exponentially Handsome Proof Nets and Their Normalization”. In: *Proceedings Second Joint International Workshop on Linearity & Trends in Linear Logic and Applications, Linearity&TLLA@IJCAR-FSCD 2020, Online, 29-30 June 2020*. Ed. by Ugo Dal Lago and Valeria de Paiva. Vol. 353. EPTCS. 2020, pp. 1–25. doi: 10.4204/EPTCS.353.1. URL: <https://doi.org/10.4204/EPTCS.353.1>.
- [2] Matteo Acclavio and Lutz Straßburger. “Combinatorial Proofs for Constructive Modal Logic”. In: *Advances in Modal Logic, AiML 2022, Rennes, France, August 22-25, 2022*. Ed. by David Fernández-Duque, Alessandra Palmigiano, and Sophie Pinchinat. College Publications, 2022, pp. 15–36.
- [3] Matteo Acclavio and Lutz Straßburger. “On Combinatorial Proofs for Logics of Relevance and Entailment”. In: *26th Workshop on Logic, Language, Information and Computation (WoLLIC 2019)*. Ed. by Rosalie Iemhoff and Michael Moortgat. Springer, 2019.
- [4] Matteo Acclavio and Lutz Straßburger. “On Combinatorial Proofs for Modal Logic”. In: *Automated Reasoning with Analytic Tableaux and Related Methods - 28th International Conference, TABLEAUX 2019, London, UK, September 3-5, 2019, Proceedings*. Ed. by Serenella Cerrito and Andrei Popescu. Vol. 11714. Lecture Notes in Computer Science. Springer, 2019, pp. 223–240. doi: 10.1007/978-3-030-29026-9_13. URL: https://doi.org/10.1007/978-3-030-29026-9%5C_13.
- [5] Peter B. Andrews. “Refutations by Matings”. In: *IEEE Transactions on Computers C-25.8* (1976), pp. 801–807.
- [6] Gianluigi Bellin and Jacques van de Wiele. “Subnets of Proof-nets in MLL⁻”. In: *Advances in Linear Logic*. Ed. by J.-Y. Girard, Y. Lafont, and L. Regnier. Vol. 222. London Mathematical Society Lecture Notes. Cambridge University Press, 1995, pp. 249–270.
- [7] Richard Blute et al. “Natural Deduction and Coherence for Weakly Distributive Categories”. In: *J. of Pure and Applied Algebra* 113 (1996), pp. 229–296.
- [8] Bernard Bolzano. “Purely analytic proof of the theorem that between any two values which give results of opposite sign, there lies at least one real root of the equation”. In: *Abhandlungen der königlichen böhmischen Gesellschaft der Wissenschaften* 5 (1817), pp. 225–48.
- [9] Kai Brünnler. “Nested Sequents”. Habilitationsschrift. Universität Bern, 2010.
- [10] Kai Brünnler and Alwen Fernanto Tiu. “A Local System for Classical Logic”. In: *LPAR 2001*. Ed. by R. Nieuwenhuis and A. Voronkov. Vol. 2250. LNAI. Springer, 2001, pp. 347–361.
- [11] Paola Bruscoli et al. “A Quasipolynomial Cut-Elimination Procedure in Deep Inference via Atomic Flows and Threshold Formulae”. In: *LPAR-16*. Vol. 6355. LNCS. Springer-Verlag, 2010, pp. 136–153.

-
- [12] Samuel R. Buss. “The undecidability of k -provability”. In: *Annals of Pure and Applied Logic* 53.1 (1991), pp. 72–102.
- [13] Alessandra Carbone. “Interpolants, cut elimination and flow graphs for the propositional calculus”. In: *Annals of Pure and Applied Logic* 83 (1997), pp. 249–299.
- [14] Alessandra Carbone. “Turning Cycles into Spirals”. In: *Ann. Pure Appl. Logic* 96.1-3 (1999), pp. 57–73.
- [15] Kaustuv Chaudhuri, Nicolas Guenot, and Lutz Straßburger. “The Focused Calculus of Structures”. In: *CSL’11*. Ed. by Marc Bezem. Vol. 12. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2011, pp. 159–173.
- [16] Alonso Church. “A Formalism of the Simple Theory of Types”. In: *The Journal of Symbolic Logic* 5 (1940), pp. 56–68.
- [17] Stephen A. Cook and Robert A. Reckhow. “The Relative Efficiency of Propositional Proof Systems”. In: *The Journal of Symbolic Logic* 44.1 (1979), pp. 36–50.
- [18] Derek G. Corneil, Yehoshua Perl, and Lorna K. Stewart. “A Linear Recognition Algorithm for Cographs”. In: *SIAM J. Comput.* 14.4 (1985), pp. 926–934.
- [19] Vincent Danos and Laurent Regnier. “The structure of multiplicatives”. In: *Arch. Math. Log.* 28.3 (1989), pp. 181–203.
- [20] Anupam Das. “Rewriting with Linear Inferences in Propositional Logic”. In: *24th International Conference on Rewriting Techniques and Applications (RTA)*. Ed. by Femke van Raamsdonk. Vol. 21. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2013, pp. 158–173.
- [21] Gottlob Frege. *Begriffsschrift*. English Translation in: J. van Heijenoort (ed.), *From Frege to Gödel*, Harvard University Press: 1977. Louis Nebert, Halle, 1879.
- [22] Gottlob Frege. *Grundgesetze der Arithmetik*. H. Pohle, 1893.
- [23] Harvey Friedman. “Some systems of second order arithmetic and their use”. In: *Proceedings of the International Congress of Mathematicians* 1 (1975).
- [24] Gerhard Gentzen. “Neue Fassung des Widerspruchsfreiheitsbeweises für die reine Zahlentheorie, Forschungen zur Logik und zur Grundlegung der exakter Wissenschaften”. In: *Neue Folge* 4 (1938), pp. 19–44.
- [25] Gerhard Gentzen. “Untersuchungen über das logische Schließen. I.” In: *Mathematische Zeitschrift* 39 (1935), pp. 176–210.
- [26] Gerhard Gentzen. “Untersuchungen über das logische Schließen. II.” In: *Mathematische Zeitschrift* 39 (1935), pp. 405–431.
- [27] Jean-Yves Girard. “Linear Logic”. In: *Theoretical Computer Science* 50 (1987), pp. 1–102.
- [28] Jean-Yves Girard. *Proof Theory and Logical Complexity, Volume I*. Vol. 1. Studies in Proof Theory. Bibliopolis, edizioni di filosofia e scienze, 1987.
- [29] Kurt Gödel. “Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I”. In: *Monatshefte für Mathematik und Physik* 38 (1931), pp. 173–198.
- [30] Alessio Guglielmi. *A Calculus of Order and Interaction*. Tech. rep. WV-99-04. Now obsolete and replaced by [31]. Technische Universität Dresden, 1999.
- [31] Alessio Guglielmi. “A System of Interaction and Structure”. In: *ACM Transactions on Computational Logic* 8.1 (2007), pp. 1–64.
- [32] Alessio Guglielmi and Tom Gundersen. “Normalisation Control in Deep Inference Via Atomic Flows”. In: *Logical Methods in Computer Science* 4.1:9 (2008), pp. 1–36.
-

- [33] Alessio Guglielmi, Tom Gundersen, and Michel Parigot. “A Proof Calculus Which Reduces Syntactic Bureaucracy”. In: *21st International Conference on Rewriting Techniques and Applications, RTA 2010*. Ed. by Christopher Lynch. Vol. 6. Leibniz International Proceedings in Informatics (LIPIcs). United Kingdom: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2010, pp. 135–150. doi: 10.4230/LIPIcs.RTA.2010.135. URL: <https://hal.science/hal-00529301>.
- [34] Alessio Guglielmi and Lutz Straßburger. “Non-commutativity and MELL in the Calculus of Structures”. In: *Computer Science Logic, CSL 2001*. Ed. by Laurent Fribourg. Vol. 2142. LNCS. Springer-Verlag, 2001, pp. 54–68.
- [35] Tom Gundersen. “A General View of Normalisation through Atomic Flows”. PhD thesis. University of Bath, 2009.
- [36] Michel Habib and Christophe Paul. “A simple linear time algorithm for cograph recognition”. In: *Discrete Applied Mathematics* 145.2 (2005). Structural Decompositions, Width Parameters, and Graph Labelings, pp. 183–197. ISSN: 0166-218X. doi: <https://doi.org/10.1016/j.dam.2004.01.011>. URL: <https://www.sciencedirect.com/science/article/pii/S0166218X04002446>.
- [37] Allen P. Hazen and Francis Jeffrey Pelletier. “Gentzen and Jaśkowski Natural Deduction: Fundamentally Similar but Importantly Different”. In: *Studia Logica: An International Journal for Symbolic Logic* 102.6 (2014), pp. 1103–1142.
- [38] Willem Heijltjes. “Proof Nets for Additive Linear Logic with Units”. In: *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*. IEEE Computer Society, 2011, pp. 207–216.
- [39] Willem Heijltjes and Robin Houston. “No proof nets for MLL with units: proof equivalence in MLL is PSPACE-complete”. In: *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*. Ed. by Thomas A. Henzinger and Dale Miller. ACM, 2014, 50:1–50:10.
- [40] Willem Heijltjes and Dominic J. D. Hughes. “Complexity Bounds for Sum-Product Logic via Additive Proof Nets and Petri Nets”. In: *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015*. IEEE Computer Society, 2015, pp. 80–91.
- [41] Willem B. Heijltjes, Dominic J. D. Hughes, and Lutz Straßburger. “Intuitionistic proofs without syntax”. In: *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*. IEEE, 2019, pp. 1–13. doi: 10.1109/LICS.2019.8785827. URL: <https://doi.org/10.1109/LICS.2019.8785827>.
- [42] David Hilbert. “Mathematische Notizbücher”. Niedersächsische Staats- und Universitätsbibliothek, Cod. Ms. D. Hilbert 600:3, S.25.
- [43] David Hilbert. “Mathematische Probleme”. In: *Nachrichten der Königlichen Gesellschaft der Wissenschaften zu Göttingen, mathematisch-physikalische Klasse* 3 (1900), pp. 253–297.
- [44] David Hilbert. *The Foundations of Mathematics*. In J. van Heijenoort (ed.), From Frege to Gödel, Harvard University Press: 1967. Louis Nebert, Halle, 1967.
- [45] W. A. Howard. “The Formulae-as-Types Notion of Construction”. In: *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Ed. by J. P. Seldin and J. R. Hindley. Academic Press, 1980, pp. 479–490.
- [46] Dominic Hughes. “First-order proofs without syntax”. Berkeley Logic Colloquium. 2014.
- [47] Dominic Hughes. “Proofs Without Syntax”. In: *Annals of Mathematics* 164.3 (2006).

- [48] Dominic Hughes. “Simple Multiplicative Proof Nets with Units”. Preprint. 2005. URL: <http://arxiv.org/abs/math.CT/0507003>.
- [49] Dominic Hughes. “Towards Hilbert’s 24th Problem: Combinatorial Proof Invariants: (Preliminary version)”. In: *Electr. Notes Theor. Comput. Sci.* 165 (2006), pp. 37–63.
- [50] Dominic J. D. Hughes. *First-order proofs without syntax*. arXiv preprint 1906.11236. 2019. arXiv: 1906.11236 [math.LO].
- [51] Dominic J. D. Hughes and Rob J. van Glabbeek. “Proof nets for unit-free multiplicative-additive linear logic”. In: *ACM Trans. Comput. Log.* 6.4 (2005), pp. 784–842.
- [52] Dominic J. D. Hughes, Lutz Straßburger, and Jui-Hsuan Wu. “Combinatorial Proofs and Decomposition Theorems for First-order Logic”. In: *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*. IEEE, 2021, pp. 1–13. DOI: 10.1109/LICS52264.2021.9470579. URL: <https://doi.org/10.1109/LICS52264.2021.9470579>.
- [53] Stanisław Jaśkowski. “On the Rules of Suppositions in Formal Logic”. In: (1934).
- [54] H.A Jung. “On a class of posets and the corresponding comparability graphs”. In: *Journal of Combinatorial Theory, Series B* 24.2 (1978), pp. 125–133. ISSN: 0095-8956. DOI: [https://doi.org/10.1016/0095-8956\(78\)90013-8](https://doi.org/10.1016/0095-8956(78)90013-8). URL: <https://www.sciencedirect.com/science/article/pii/0095895678900138>.
- [55] Gregory Maxwell Kelly and Saunders Mac Lane. “Coherence in Closed Categories”. In: *J. of Pure and Applied Algebra* 1 (1971), pp. 97–140.
- [56] O. Ketonen. *Untersuchungen zum Prädikatenkalkül*. Annales Academiae scientiarum fennicae. Series A. I. Mathematica-Physica. 23. Suomalainen Tiedeakatemia, 1944. URL: <https://books.google.fr/books?id=yhrSSAAACAAJ>.
- [57] Stephen Cole Kleene. *Introduction to Metamathematics*. Vol. 1. Bibliotheca Mathematica, A Series of Monographs on Pure and Applied Mathematics. North-Holland Publishing Co., 1952.
- [58] Ulrich Kohlenbach. “Applied foundations: proof mining in analysis (Final version)”. In: *Newsletter of the Danish Mathematical Society* 13 (2002).
- [59] Yves Lafont. “Equational reasoning with 2-dimensional diagrams”. In: *Term Rewriting*. Vol. 909. LNCS. 1995, pp. 170–195.
- [60] François Lamarche and Lutz Straßburger. “From Proof Nets to the Free *-Autonomous Category”. In: *Logical Methods in Computer Science* 2.4:3 (2006), pp. 1–44. URL: <http://arxiv.org/abs/cs/0605054>.
- [61] François Lamarche and Lutz Straßburger. “Naming Proofs in Classical Propositional Logic”. In: *TLCA’05*. Ed. by Paweł Urzyczyn. Vol. 3461. LNCS. Springer, 2005, pp. 246–261. URL: <http://www.lix.polytechnique.fr/~lutz/papers/namingproofsCL.pdf>.
- [62] Olivier Laurent. “Polarized proof-nets: proof-nets for LC (Extended Abstract)”. In: *Typed Lambda Calculi and Applications (TLCA 1999)*. Ed. by Jean-Yves Girard. Vol. 1581. LNCS. Springer, 1999, pp. 213–227.
- [63] Ross M. McConnell and Fabien de Montgolfier. “Linear-time modular decomposition of directed graphs”. In: *Discrete Applied Mathematics* 145.2 (2005). Structural Decompositions, Width Parameters, and Graph Labelings, pp. 198–209. ISSN: 0166-218X. DOI: <https://doi.org/10.1016/j.dam.2004.02.017>. URL: <https://www.sciencedirect.com/science/article/pii/S0166218X04002458>.

- [64] Rolf H. Möhring. “Computationally tractable classes of ordered sets”. In: *Algorithms and Order*. Ed. by I. Rival. Kluwer Acad. Publ., 1989, pp. 105–194.
- [65] Giti Omidvar. “Syntactic and Combinatorial Proofs”. MA thesis. University of Paris Cite, 2020.
- [66] Giti Omidvar and Lutz Straßburger. “Combinatorial Flows as Bicolored Atomic Flows”. In: *Logic, Language, Information, and Computation - 28th International Workshop, WoLLIC 2022, Iași, Romania, September 20-23, 2022, Proceedings*. Ed. by Agata Ciabattoni, Elaine Pimentel, and Ruy J. G. B. de Queiroz. Vol. 13468. Lecture Notes in Computer Science. Springer, 2022, pp. 141–157. doi: [10.1007/978-3-031-15298-6_9](https://doi.org/10.1007/978-3-031-15298-6_9). URL: https://doi.org/10.1007/978-3-031-15298-6%5C_9.
- [67] Dag Prawitz. “Ideas and Results in Proof Theory”. In: *Proceedings of the Second Scandinavian Logic Symposium*. Ed. by J. E. Fenstad. North-Holland Publishing Co., 1971, pp. 235–307.
- [68] Dag Prawitz. *Natural Deduction, A Proof-Theoretical Study*. Almqvist and Wiksell, 1965.
- [69] Benjamin Ralph and Lutz Straßburger. “Towards a Combinatorial Proof Theory”. In: *Automated Reasoning with Analytic Tableaux and Related Methods - 28th International Conference, TABLEUX 2019, London, UK, September 3-5, 2019, Proceedings*. Ed. by Serenella Cerrito and Andrei Popescu. Vol. 11714. Lecture Notes in Computer Science. Springer, 2019, pp. 259–276. doi: [10.1007/978-3-030-29026-9_15](https://doi.org/10.1007/978-3-030-29026-9_15). URL: https://doi.org/10.1007/978-3-030-29026-9%5C_15.
- [70] Greg Restall. “Normal Proofs, Cut Free Derivations and Structural Rules”. In: *Studia Logica: An International Journal for Symbolic Logic* 102.6 (2014), pp. 1143–1166.
- [71] Christian Retoré. “Calcul de Lambek et Logique Linéaire”. In: *Traitement Automatique des Langues* 37.2 (1996), pp. 39–70.
- [72] Christian Retoré. “Handsome proof-nets: perfect matchings and cographs”. In: *Theoretical Computer Science* 294.3 (2003), pp. 473–488.
- [73] Christian Retoré. *Handsome proof-nets: R&B-graphs, perfect matchings and series-parallel graphs*. Rapport de recherche 3652. Appeared as [72]. INRIA, 1999.
- [74] Christian Retoré. “Perfect matchings and series-parallel graphs: multiplicatives proof nets as R&B-graphs.” In: *Electronic Notes in Theoretical Computer Science* 3 (1996).
- [75] Edmund P. Robinson. “Proof Nets for Classical Logic”. In: *Journal of Logic and Computation* 13 (2003), pp. 777–797.
- [76] Bertrand Russell. *The Principles of Mathematics*. Cambridge, England: Allen & Unwin, 1903.
- [77] D. Seinsche. “On a property of the class of n-colorable graphs”. In: *Journal of Combinatorial Theory, Series B* 16.2 (1974), pp. 191–193. ISSN: 0095-8956. doi: [https://doi.org/10.1016/0095-8956\(74\)90063-X](https://doi.org/10.1016/0095-8956(74)90063-X). URL: <https://www.sciencedirect.com/science/article/pii/009589567490063X>.
- [78] Peter Selinger. “A Survey of Graphical Languages for Monoidal Categories”. In: *New Structures for Physics*. Ed. by Bob Coecke. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 289–355. ISBN: 978-3-642-12821-9. doi: [10.1007/978-3-642-12821-9_4](https://doi.org/10.1007/978-3-642-12821-9_4). URL: https://doi.org/10.1007/978-3-642-12821-9_4.
- [79] Lutz Straßburger. “A Characterisation of Medial as Rewriting Rule”. In: *Term Rewriting and Applications, RTA’07*. Ed. by Franz Baader. Vol. 4533. LNCS. Springer, 2007, pp. 344–358. URL: <http://www.lix.polytechnique.fr/~lutz/papers/CharMedial.pdf>.

-
- [80] Lutz Straßburger. *Combinatorial Flows and Proof Compression*. Research Report RR-9048. Inria Saclay, 2017. URL: <https://hal.inria.fr/hal-01498468>.
- [81] Lutz Straßburger. “Combinatorial Flows and Their Normalisation”. In: *2nd International Conference on Formal Structures for Computation and Deduction, FSCD 2017, September 3-9, 2017, Oxford, UK*. Ed. by Dale Miller. Vol. 84. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017, 31:1–31:17. ISBN: 978-3-95977-047-7.
- [82] Lutz Straßburger. “From deep inference to proof nets via cut elimination”. In: *Journal of Logic and Computation* 21.4 (2011), pp. 589–624.
- [83] Lutz Straßburger. “Linear Logic and Noncommutativity in the Calculus of Structures”. PhD thesis. Technische Universität Dresden, 2003.
- [84] Lutz Straßburger. “The problem of proof identity, and why computer scientists should care about Hilbert’s 24th problem”. In: *Philosophical Transactions of the Royal Society A* 377.2140 (2019), p. 20180038. ISSN: 1364-503X.
- [85] David P. Sumner. “Dacey Graphs”. In: *Journal of the Australian Mathematical Society* 18.4 (1974), pp. 492–502. doi: 10.1017/S1446788700029232.
- [86] Rüdiger Thiele. “Hilbert’s Twenty-Fourth Problem”. In: *American Mathematical Monthly* 110 (2003), pp. 1–24.
- [87] Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Second. Cambridge University Press, 2000.
- [88] W. T. Tutte. “The Factorization of Linear Graphs”. In: *Journal of the London Mathematical Society* s1-22.2 (Apr. 1947), pp. 107–111. ISSN: 0024-6107. doi: 10.1112/jlms/s1-22.2.107.
- [89] Alfred North Whitehead and Bertrand Russell. *Principia Mathematica*. Ed. by Bertrand Russell and Alfred North Whitehead. New York: Cambridge University Press, 1962.

APPENDIX A

Graph Theory Proofs

Proof of Proposition 4.7. we proceed by induction on the construction of \mathcal{G} :

- If \mathcal{G} is a single vertex, it is clear that $\mathcal{H} = \mathcal{G}$ holds the conclusion.
- If $\mathcal{G} = \mathcal{I} \uplus \mathcal{J}$, by induction hypothesis there exist graphs \mathcal{K} and \mathcal{L} which are isomorphic to \mathcal{I} and \mathcal{J} and constructed from single vertices using \uplus and \bowtie . Hence, $\mathcal{H} = \mathcal{K} \uplus \mathcal{L}$ has the same property.
- If $\mathcal{G} = \overline{\mathcal{I}}$, by induction hypothesis there exists a graph \mathcal{J} which is isomorphic to \mathcal{I} and it is constructed by operations \bowtie and \uplus over single vertices. If we define \mathcal{H} by the same construction as \mathcal{I} with the change that we switch \bowtie and \uplus , we only need to prove that \mathcal{H} is isomorphic to \mathcal{G} . By switching the two operations disjoint union and conjunction, we are complementing the edge relation in the graph. So \mathcal{H} is isomorphic to $\overline{\mathcal{I}}$ which is equal to \mathcal{G} by hypothesis. \square

Proof of Proposition 4.8. See [64]. (\Rightarrow) We prove this by induction on the structure of \mathcal{G} :

- If \mathcal{G} is a single vertex, it is indeed a \mathcal{P}_4 -free graph.
- If $\mathcal{G} = \mathcal{H} \uplus \mathcal{I}$, by induction hypothesis we know that \mathcal{H} and \mathcal{I} are \mathcal{P}_4 -free. So $\mathcal{H} \uplus \mathcal{I}$ is also \mathcal{P}_4 -free since $R_{\mathcal{G}} = R_{\mathcal{H}} \cup R_{\mathcal{I}}$.
- If $\mathcal{G} = \overline{\mathcal{H}}$, by induction we know \mathcal{H} is \mathcal{P}_4 -free. Let us assume by way of contradiction that $\overline{\mathcal{H}}$ is not a \mathcal{P}_4 -free so it contains a \mathcal{P}_4 . This leads to \mathcal{H} containing a \mathcal{P}_4 since the complement of a \mathcal{P}_4 is also a \mathcal{P}_4 which is a contradiction.

(\Leftarrow) We proceed by induction on the number of vertices in \mathcal{G} . The goal in this proof is to divide \mathcal{G} into two induced subgraphs \mathcal{H} and \mathcal{I} with either $\mathcal{G} = \mathcal{H} \bowtie \mathcal{I}$ or $\mathcal{G} = \mathcal{H} \uplus \mathcal{I}$ which results to \mathcal{G} being a cograph by Proposition 4.7. If \mathcal{G} has only a single vertex then it is obvious that it is constructed from a single vertex. Otherwise, let v be a vertex of \mathcal{G} . Let V_1 be the set of all neighbors of v in \mathcal{G} .

If $V_1 = \emptyset$, then \mathcal{G} is $\mathcal{H} \uplus \mathcal{I}$ where \mathcal{H} is the induced subgraph of \mathcal{G} over $\{v\}$ and \mathcal{I} is the induced subgraph of \mathcal{G} over $V_{\mathcal{G}} \setminus \{v\}$. Using induction hypothesis we know that \mathcal{I} is constructed from single vertices using \bowtie and \uplus so \mathcal{I} is a cograph which implies that \mathcal{G} is also a cograph.

Now we look into the case where $V_1 \neq \emptyset$.

- If for every vertex $u \in V_1$, it only exists one incident (which is the edge vRu) then let \mathcal{H} be the induced subgraph of \mathcal{G} over the set of vertices $\{v\} \cup V_1$ and $\mathcal{I} = \mathcal{G} \setminus \mathcal{H}$. Since no vertex in \mathcal{H} has an edge to any vertex in \mathcal{I} , we have $\mathcal{G} = \mathcal{H} \uplus \mathcal{I}$. Induction hypothesis shows us that \mathcal{H} and \mathcal{I} are both cographs so \mathcal{G} is a cograph as well.

- If there is at least one vertex such as $u \in V_1$ such that there exists w in V_G where $w \neq v$, $w \notin V_1$, and uRw , then we define the set of vertices $V_2 = \{u | (v, u) \notin R \text{ and } \exists w \in V_1. uRw\}$. Vertices in V_2 cannot have any neighbors outside of $\{v\} \cup V_1 \cup V_2$, otherwise \mathcal{G} is not \mathcal{P}_4 -free. Now let \mathcal{H} be the induced subgraph of \mathcal{G} over the vertices $\{v\} \cup V_1 \cup V_2$ and $\mathcal{I} = \mathcal{G} \setminus \mathcal{H}$. By induction hypothesis we know that \mathcal{I} is a cograph. It only remains to show that \mathcal{H} can be constructed from single vertices using operations \bowtie and \uplus . Let x be a vertex in V_2 . Let V_3 be all vertices in V_1 connected to x . We prove that $\mathcal{H} = \mathcal{J} \bowtie \mathcal{K}$ where \mathcal{J} is the induced subgraph of \mathcal{H} over V_3 and \mathcal{K} is the induced subgraph of \mathcal{H} over the vertices $\{v\} \cup V_2 \cup (V_1 \setminus V_3)$. We know that every vertex in V_3 is connected to v and x . It only remains to show that they are also connected to $V_1 \setminus V_3$ and $V_2 \setminus x$. Let w be a vertex in $V_1 \setminus V_3$ and y a vertex in V_3 . By looking at x, y, v , and w we can see a \mathcal{P}_4 which is contradiction with the fact that \mathcal{H} is \mathcal{P}_4 -free. So w is connected to y since none of the edges xR_Hv and xR_Hw can exist. Now Let z be a vertex in $V_2 \setminus x$ which is one of the incidents to w . Considering the vertices x, y, w , and z shows us that either we have to have the edge yR_Hz or yR_Hx . The former case would be the edge we need but if we have the later edge, then we can consider the four vertices z, x, y, v . We know that the edges vR_Hx and vR_Hz cannot exist so the edge yR_Hz should exist. So every vertex in $V_2 \setminus x$ is connected to every vertex in V_3 as well. As a result we have the two induced subgraphs \mathcal{J} and \mathcal{K} of \mathcal{H} that connect together with \bowtie and both are cographs by induction hypothesis. \square

Proof of Theorem 4.19. Let us assume that $P_{\mathcal{G}}$ has an alternating elementary cycle c by way of contradiction. Then we can write an alternating elementary cycle in \mathcal{G} by using the edges and links in c . For the new cycle, we keep the axiom links in c and add an edge if to atoms are connected in c without any interruption of another atom. The new cycle created is chordless. Hence the RB-cograph \mathcal{G} is no \mathfrak{a} -acyclic which is in contradiction with the hypothesis. \square

APPENDIX B

Decomposition and Composition of Skew Fibrations

This chapter of Appendix is dedicated to the proof of Lemma 7.7 copied directly from [65].

Note that the composition of two skew fibrations is not a skew fibration for arbitrary graphs. Figure B.1 is an example where \mathcal{H} and \mathcal{I} are not cographs and $f: \mathcal{G} \hookrightarrow \mathcal{H}$ and $g: \mathcal{H} \hookrightarrow \mathcal{I}$ are skew fibrations. However, $f \circ g$ is not a skew fibration.

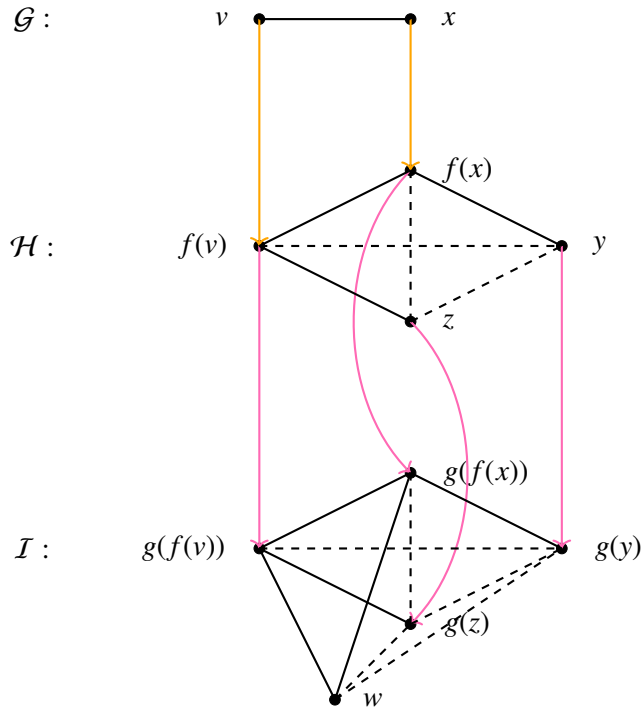


Figure B.1: Vertical composition of skew fibrations is not a skew fibration

Let $v \in V_{\mathcal{G}}$, $w \in V_{\mathcal{I}}$, and $g(f(v))R_{\mathcal{I}}w$. If $f \circ g$ was a skew fibration, there should have existed a vertex $u \in V_{\mathcal{G}}$ such that:

$$vR_{\mathcal{G}}u \text{ and } g(f(u))R_{\mathcal{I}}w$$

One can observe that such a vertex does not exist in $V_{\mathcal{G}}$ so $f \circ g$ is not a skew fibration. In this example, one can notice that \mathcal{H} and \mathcal{I} are not cographs. We must add more edges to $R_{\mathcal{H}}$ and $R_{\mathcal{I}}$ to make sure they become cographs. Note that, adding edges can lead to adding new nodes because of the skew fibration property for f and g .

Theorem B.1. Let $f: \mathcal{G} \hookrightarrow \mathcal{H}$ and $g: \mathcal{H} \hookrightarrow \mathcal{I}$ be skew fibrations where \mathcal{H} , and \mathcal{I} are cographs. The composition of $f \circ g$ is a skew fibration (\mathcal{G} can be an arbitrary graph).

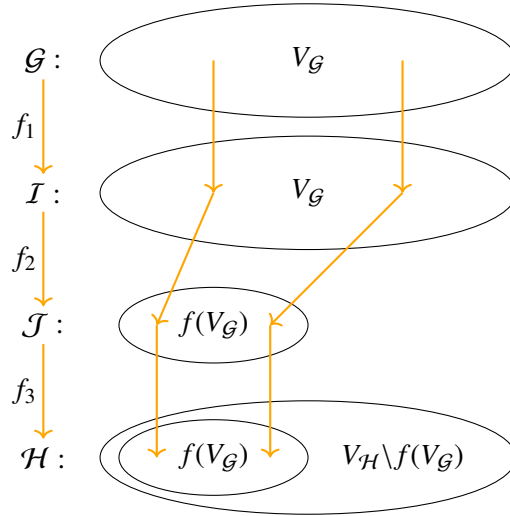
Outline of the proof: First, decompose g into three skew fibrations that are full injective, full surjective, and bijective. Next, we prove that the composition of each decomposed part of g with f is a skew fibration. The following lemmas are proving these statements.

Lemma B.2. Let $f: \mathcal{G} \hookrightarrow \mathcal{H}$ be a skew fibration. Then we can decompose f to the following three parts: $f_1: \mathcal{G} \twoheadrightarrow \mathcal{I}$ which is a bijective skew fibration, $f_2: \mathcal{I} \twoheadrightarrow \mathcal{J}$ which is a full surjective skew fibration, and $f_3: \mathcal{J} \hookrightarrow \mathcal{H}$ which is a full injective skew fibration. Moreover, \mathcal{I} and \mathcal{J} are cographs. ($f = f_1 \circ f_2 \circ f_3$)

Proof. This lemma follows from [69]. But below is the direct proof. We construct the graphs \mathcal{I} and \mathcal{J} as follows:

$$\begin{aligned} V_{\mathcal{J}} &:= f(V_{\mathcal{G}}) \\ R_{\mathcal{J}} &:= \{(u, v) \mid u, v \in f(V_{\mathcal{G}}) \text{ and } uR_{\mathcal{H}}v\} \\ V_{\mathcal{I}} &:= V_{\mathcal{G}} \\ R_{\mathcal{I}} &:= \{(u, v) \mid f(u)R_{\mathcal{J}}f(v)\} \end{aligned}$$

As we can see in the picture below, the intuition is to first add the extra edges that \mathcal{H} has for the set $f(V_{\mathcal{G}})$ to $R_{\mathcal{I}}$. With this kind of definition f_1 can be bijective. Then we make \mathcal{J} in such a way that only consists of the images of $V_{\mathcal{G}}$ so f_2 will be surjective. Lastly, we can see that the vertices $V_{\mathcal{H}} \setminus f(V_{\mathcal{G}})$ are new in \mathcal{H} so f_3 can be injective. In this picture we have only shown the outline of the vertices of each graph and yellow arrows are showing the functions $f_1, f_2,$ and f_3 . The detailed proof follows the picture.



Let $f_3(u) := u$ for all $u \in V_{\mathcal{J}}$. It is trivial that f_3 is injective. By looking at the definition of $R_{\mathcal{J}}$, we can see that f_3 is a full homomorphism. Now we move on to proving that f_3 is a skew fibration. Graph \mathcal{J} is a cograph since $f(V_{\mathcal{G}})$ is a subgraph of $V_{\mathcal{H}}$ and \mathcal{H} is a cograph. Assume $v \in V_{\mathcal{J}}$, $w \in V_{\mathcal{H}}$, and $f_3(v)R_{\mathcal{H}}w$. Since $V_{\mathcal{J}} = f(V_{\mathcal{G}})$, there exists a node $u \in V_{\mathcal{G}}$ with $f(u) = v$. From the fact that f is a skew fibration, we can deduce that:

$$\exists x \in V_{\mathcal{G}}. uR_{\mathcal{G}}x \wedge (f(x), w) \notin R_{\mathcal{H}} \quad (\text{B.1})$$

In addition, from $f(u)f(x) = v f(x) \in R_{\mathcal{H}}$ and f_3 being a full injective function we can derive the following conclusion:

$$vR_{\mathcal{J}}f(x) \quad (\text{B.2})$$

Equations (B.1) and (B.2) show that f_3 is a skew fibration.

For all vertices $v \in V_{\mathcal{G}}$, let $f_1(v) := v$ and $f_2(v) := f(v)$. Because of the definitions for f_1 , f_2 , \mathcal{I} , and \mathcal{J} , it is obvious that f_1 is bijective and f_2 is full surjective. The graph \mathcal{I} is a cograph as well since having a \mathcal{P}_4 means that \mathcal{J} has the same (f_2 is full) \mathcal{P}_4 which contradicts the fact that \mathcal{J} is a cograph (we proved this before). The function f_2 is a skew fibration because it is full and surjective. Since f_1 and f_3 are identity functions and $\forall v \in V_{\mathcal{G}}$. $f_2(v) = f(v)$, one can observe that $\forall v \in V_{\mathcal{G}}$. $f(v) = f_3(f_2(f_1(v)))$. The only remaining part is to prove that f_1 is a skew fibration.

Assume $v \in V_{\mathcal{G}}$, $w \in V_{\mathcal{I}}$, and $f_1(v)w = vw \in R_{\mathcal{I}}$. f_2 and f_3 are homomorphisms so:

$$f_3(f_2(f_1(v)))f_3(f_2(w)) = f(v)f_3(f_2(w)) \in R_{\mathcal{H}}$$

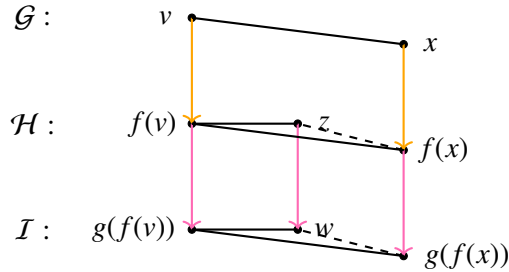
Since f is a skew fibration, there exists a vertex x in $V_{\mathcal{G}}$ which:

$$vR_{\mathcal{G}}x \wedge f(x)f_3(f_2(w)) = f_3(f_2(f_1(x)))f_3(f_2(w)) \notin R_{\mathcal{H}}$$

Using the fact that f_3 and f_2 are homomorphisms we can deduce that $(f_1(x), w) \notin R_{\mathcal{I}}$. This means that f_1 is a skew fibration, too. \square

Lemma B.3. *If $f: \mathcal{G} \hookrightarrow \mathcal{H}$ is a skew fibration and $g: \mathcal{H} \rightarrow \mathcal{I}$ is a full surjective skew fibration where \mathcal{H} and \mathcal{I} are cographs, then $f \circ g$ is a skew fibration.*

Proof. Suppose we have $v \in V_{\mathcal{G}}$, $w \in V_{\mathcal{I}}$ and $g(f(v))R_{\mathcal{I}}w$. Since g is surjective, there exists $z \in V_{\mathcal{H}}$ which $g(z) = w$. We can deduce that $f(v)R_{\mathcal{H}}z$ from the fact that g is full. Following the fact that f is a skew fibration, it can be deduced that there exists $x \in V_{\mathcal{G}}$ with $vR_{\mathcal{G}}x$ and $(f(x), z) \notin R_{\mathcal{H}}$. Using this result together with the fact that g is full, we can deduce that $(g(f(x)), w) \notin R_{\mathcal{I}}$.



The picture shows the skew fibrations g and f described in the proof where the yellow arrow lines from vertices in $V_{\mathcal{G}}$ to $V_{\mathcal{H}}$ represent f and the pink arrow lines from $V_{\mathcal{H}}$ to $V_{\mathcal{I}}$ show g . \square

Lemma B.4. *If $f: \mathcal{G} \hookrightarrow \mathcal{H}$ is a skew fibration and $g: \mathcal{H} \rightarrow \mathcal{I}$ is a full injective skew fibration where \mathcal{H} and \mathcal{I} are cographs, then $f \circ g$ is a skew fibration.*

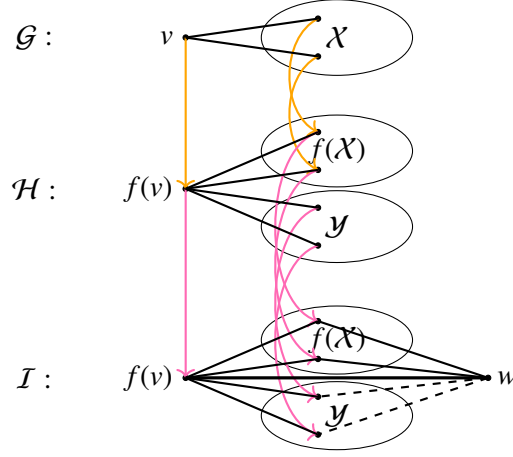
Proof. Let $v \in V_{\mathcal{G}}$, $w \in V_{\mathcal{I}}$ and $g(f(v)) \in R_{\mathcal{I}}$. Since g is injective, we can assume without loss of generality that $\forall u \in V_{\mathcal{H}}$. $g(u) = u \in V_{\mathcal{I}}$. Let us define the sets \mathcal{X} and \mathcal{Y} as follows:

$$\begin{aligned} \mathcal{X} &:= \{x \in V_{\mathcal{G}} | vR_{\mathcal{G}}x\} \\ f(\mathcal{X}) &:= \{f(x) | x \in \mathcal{X}\} \\ \mathcal{Y} &:= \{y \in V_{\mathcal{H}} | f(v)R_{\mathcal{H}}y \text{ and } y \notin f(\mathcal{X})\} \end{aligned}$$

Assume by contradiction that $f \circ g$ is not a skew fibration. As a result:

$$\forall x \in \mathcal{X}. f(x)R_{\mathcal{I}}w$$

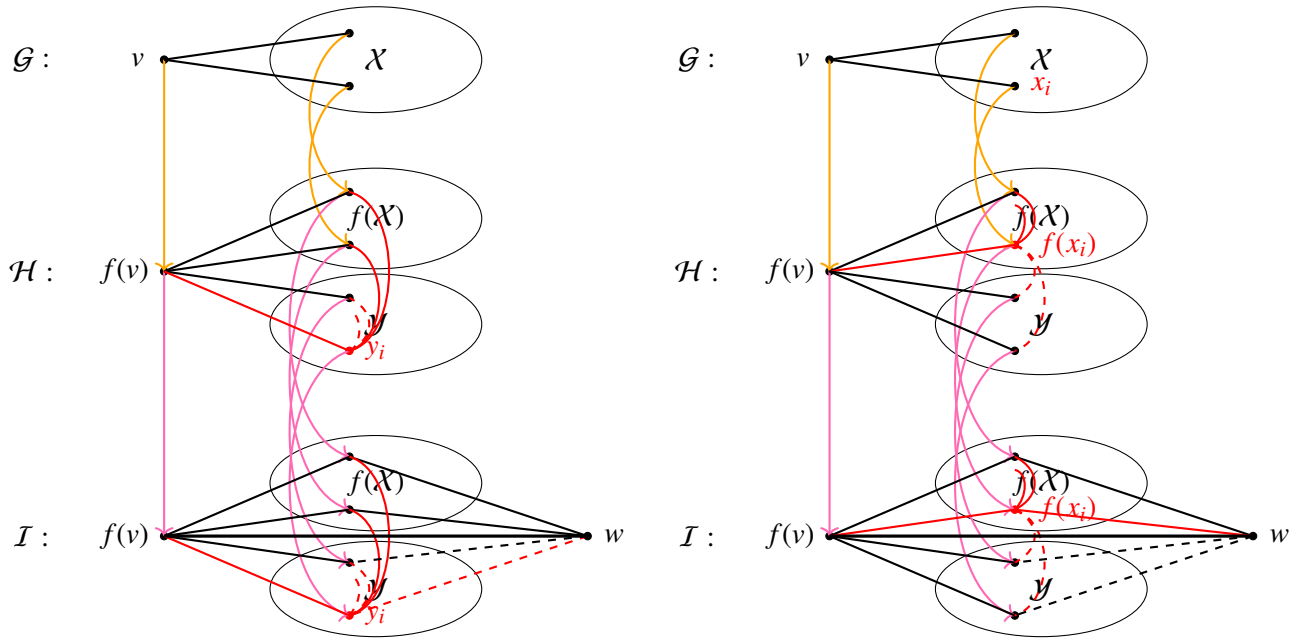
This is indicated in the picture below.



In order to preserve the skew fibration property for g and f we will create an infinite sequence of vertices $y_1, x_1, y_2, x_2, \dots$ such that:

- $y_i \in \mathcal{Y}$ and $x_i \in \mathcal{X}$
- $\forall j = 1 \dots i - 1. y_i \neq y_j \wedge x_i \neq x_j$
- $\forall j = 1 \dots i - 1. (y_i, y_j) \notin R_{\mathcal{I}}$
- $\forall j = 1 \dots i - 1. y_i R_{\mathcal{H}} f(x_j)$
- $(y_i, w) \notin R_{\mathcal{I}}$
- $\forall j = 1 \dots i - 1. f(x_i) R_{\mathcal{I}} f(x_j)$
- $\forall j = 1 \dots i. (f(x_i), y_j) \notin R_{\mathcal{I}}$
- $f(x_i) R_{\mathcal{I}} w$

The following pictures illustrate these conditions for each node in the sequence. First, each vertex is either a member of \mathcal{X} or \mathcal{Y} . To the right, we can see the edges, whether they exist or not, in red for the vertex x_i according to the conditions above. For example, there is no edge between $f(x_i)$ and y_j where $j = 1 \dots i$ in $R_{\mathcal{H}}$ and $R_{\mathcal{I}}$. The left figure demonstrates the conditions for y_i with red lines or dashed lines. There is no edge from y_i to all previous y 's as well as w . On the other hand, y_i has edge to all previous x 's.

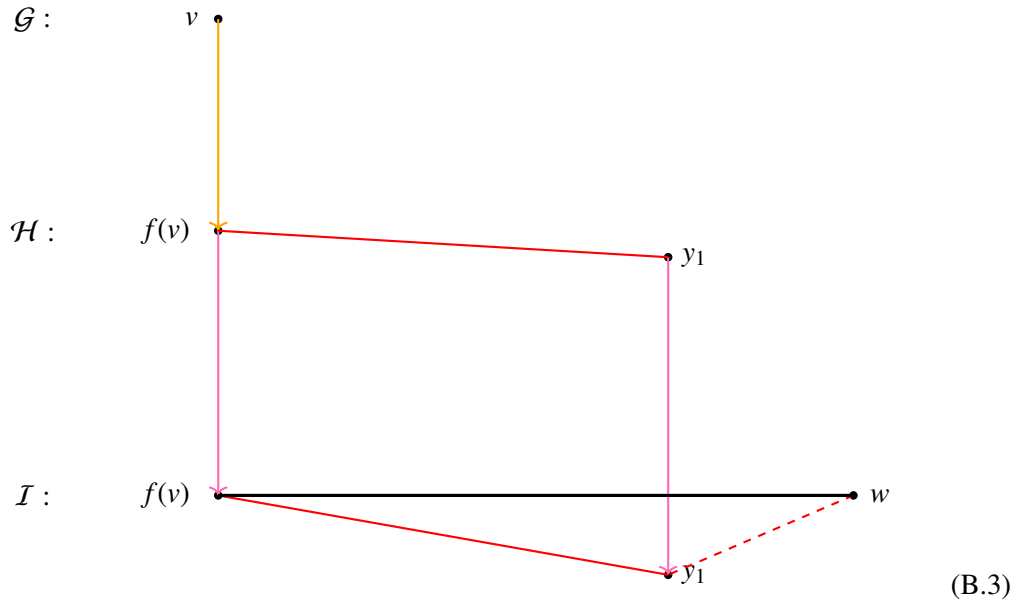


we proceed by induction on the length of the sequence:

- **Step 1.** Since $f(v)R_I w$, we have:

$$\exists y_1 \in V_{\mathcal{H}}. f(v)R_{\mathcal{H}}y_1 \wedge (y_1, w) \notin R_I$$

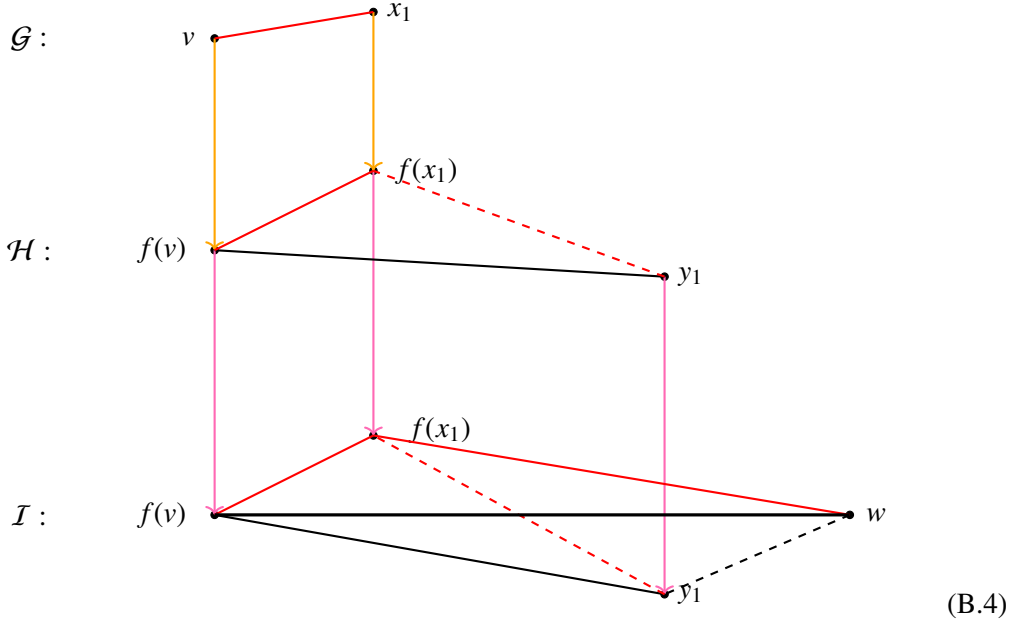
Hence, with respect to the definition of \mathcal{X} , $y_1 \notin \mathcal{X}$ so $y_1 \in \mathcal{Y}$. All the properties hold for y_1 trivially. This is indicated below:



- **Step 2.** Since $f(v)R_{\mathcal{H}}y_1$, we have:

$$\exists x_1 \in \mathcal{X}. (f(x_1), y_1) \notin R_{\mathcal{H}}$$

It is trivial that x_1 has all the claimed properties since g is injective:



- **Step $2m + 1$.** Assume we already have a sequence $y_1, x_1, \dots, y_m, x_m$. In this step, we only consider edges in R_I . Note that these edges are replicated in R_H since g is full. By induction hypothesis, we know that:

- a1. $\forall i = 1 \dots m - 1. f(x_m)R_I f(x_i)$
- b1. $\forall i = 1 \dots m. (f(x_m), y_i) \notin R_I$
- c1. $f(x_m)R_I w$
- d1. $\forall i = 1 \dots m. \forall j = 1 \dots i - 1. (y_i, y_j) \notin R_I$
- e1. $\forall i = 1 \dots m. \forall j = 1 \dots i - 1. y_i R_I f(x_j)$
- f1. $\forall i = 1 \dots m. (y_i, w) \notin R_I$
- g1. $\forall i = 1 \dots m. y_i \in \mathcal{Y} \wedge x_i \in \mathcal{X}$

It follows from a1 and b1 that there exists a new vertex y_{m+1} such that $y_{m+1}R_I f(x_m)$ and $(y_{m+1}, w) \notin R_I$. Using the induction hypothesis b1 and f1, we have for all $i = 1 \dots m$:

$$(f(x_m), y_i) \notin R_I \wedge (y_i, w) \notin R_I \wedge (w, y_{m+1}) \notin R_I \quad (\text{B.5})$$

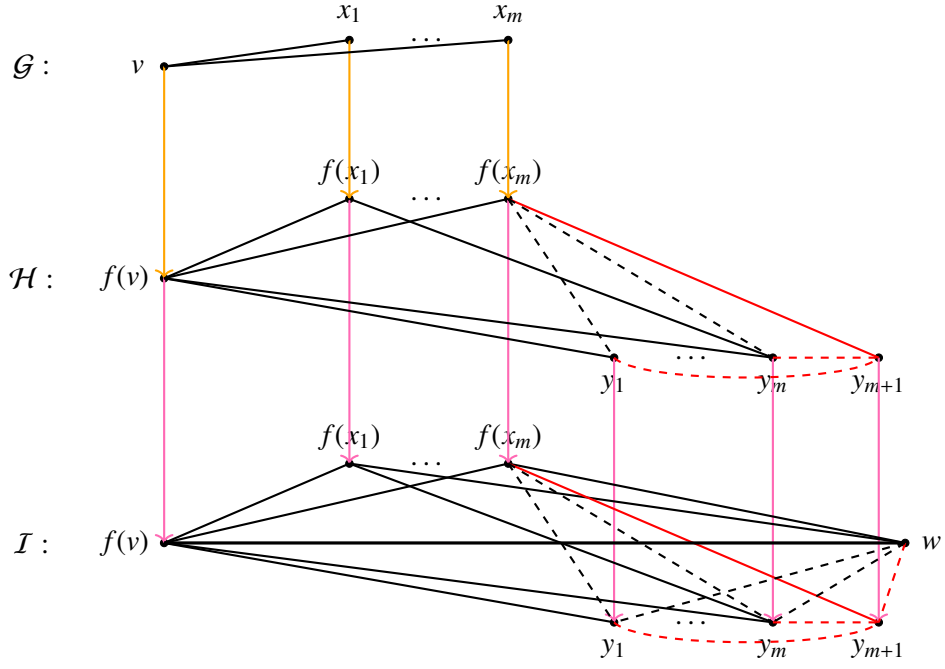
Also, we know that:

$$f(x_m)R_I w \wedge y_{m+1}R_I f(x_m) \quad (\text{B.6})$$

As a result of Equations (B.5) and (B.6) plus the fact that C is a cograph:

$$\forall i = 1 \dots m. (y_{m+1}, y_i) \notin R_I \quad (\text{B.7})$$

The following figure shows what happens when we add the vertex y_{m+1} to the sequence we already have. The red dashed lines from y_{m+1} to y_1 and y_m indicate the result from (B.7). It means that y_{m+1} doesn't have any edge to y_i where $i = 1 \dots m$. The red line between $f(x_m)$ and y_{m+1} demonstrates the last part of (B.5). Moreover, the red dashed line from w to y_{m+1} in I indicates the last part of (B.6).



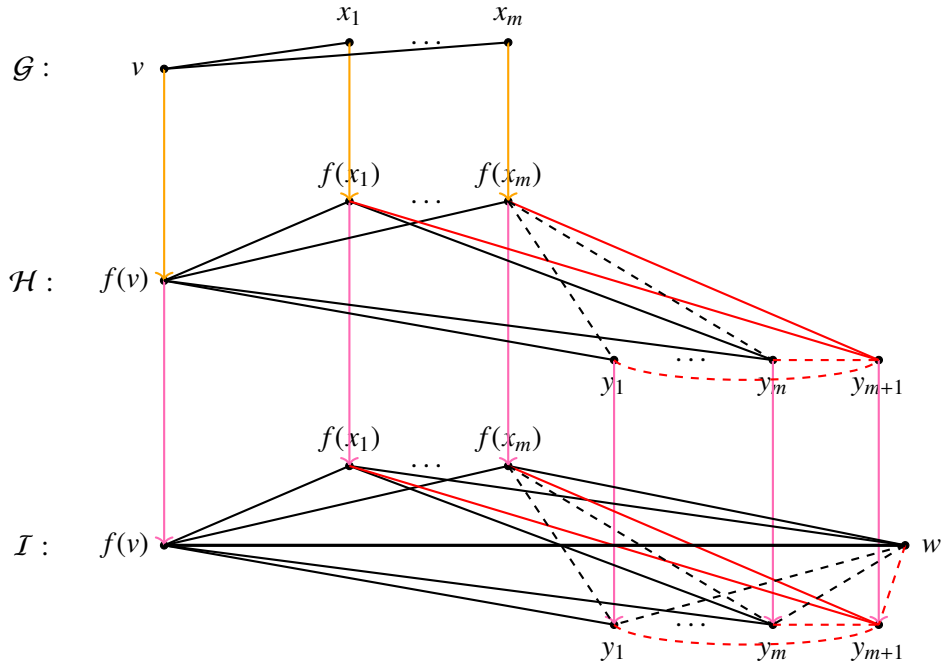
The properties a1 and b1 for x_m and e1 for y_m (which is the last vertex in the sequence) show that:

$$(\forall i = 1 \dots m - 1. f(x_i)R_I f(x_m) \wedge f(x_i)R_I y_m) \wedge (y_m, f(x_m)) \notin R_I \quad (\text{B.8})$$

Since C is a cograph, it follows from (B.7) and (B.8) that:

$$\forall i = 1 \dots m. f(x_i)R_I y_{m+1} \quad (\text{B.9})$$

as indicated below:



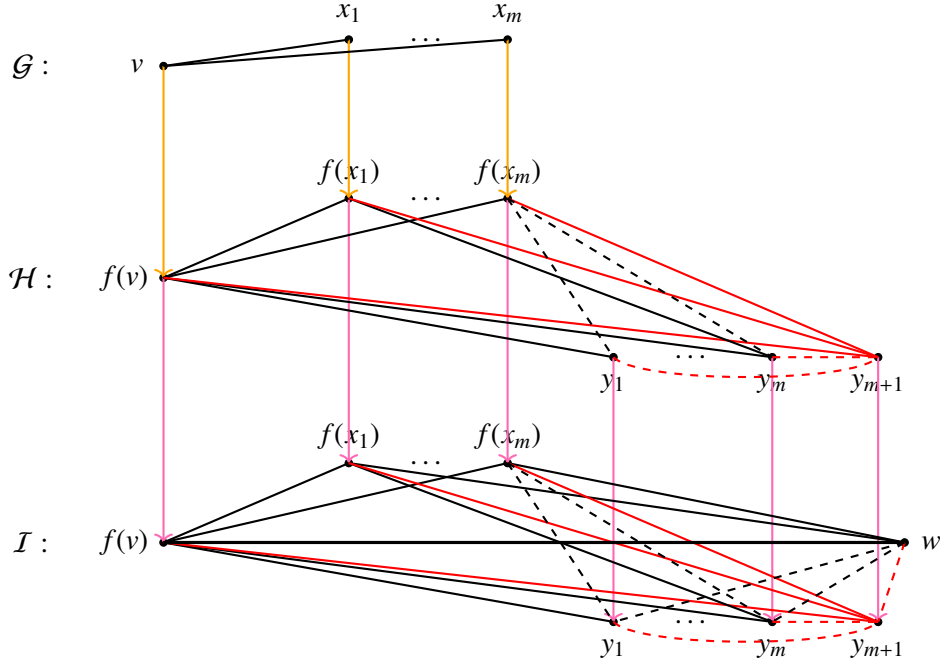
Now we have:

$$y_{m+1}R_I f(v) \quad (\text{B.10})$$

Because, otherwise $y_m f(v) f(x_m) y_{m+1}$ would be a \mathcal{P}_4 (because of (B.7), (B.8), and (B.9)). Since g is full, $y_{m+1} R_{\mathcal{H}} f(v)$ should hold. By the definition of y_{m+1} and \mathcal{Y} , it is obvious that:

$$y_{m+1} \in \mathcal{Y} \quad (\text{B.11})$$

The following figure shows that y_{m+1} is a neighbor of $f(v)$ with a red line between the two.



- **Step $2m + 2$.** The proof of this step is given in Appendix C. Assume we have a sequence $y_1, x_1, \dots, y_m, x_m, y_{m+1}$. The last edge we added in the previous step was $f(v) R_{\mathcal{I}} y_{m+1}$. Since g is a full homomorphism:

$$f(v) R_{\mathcal{H}} y_{m+1}$$

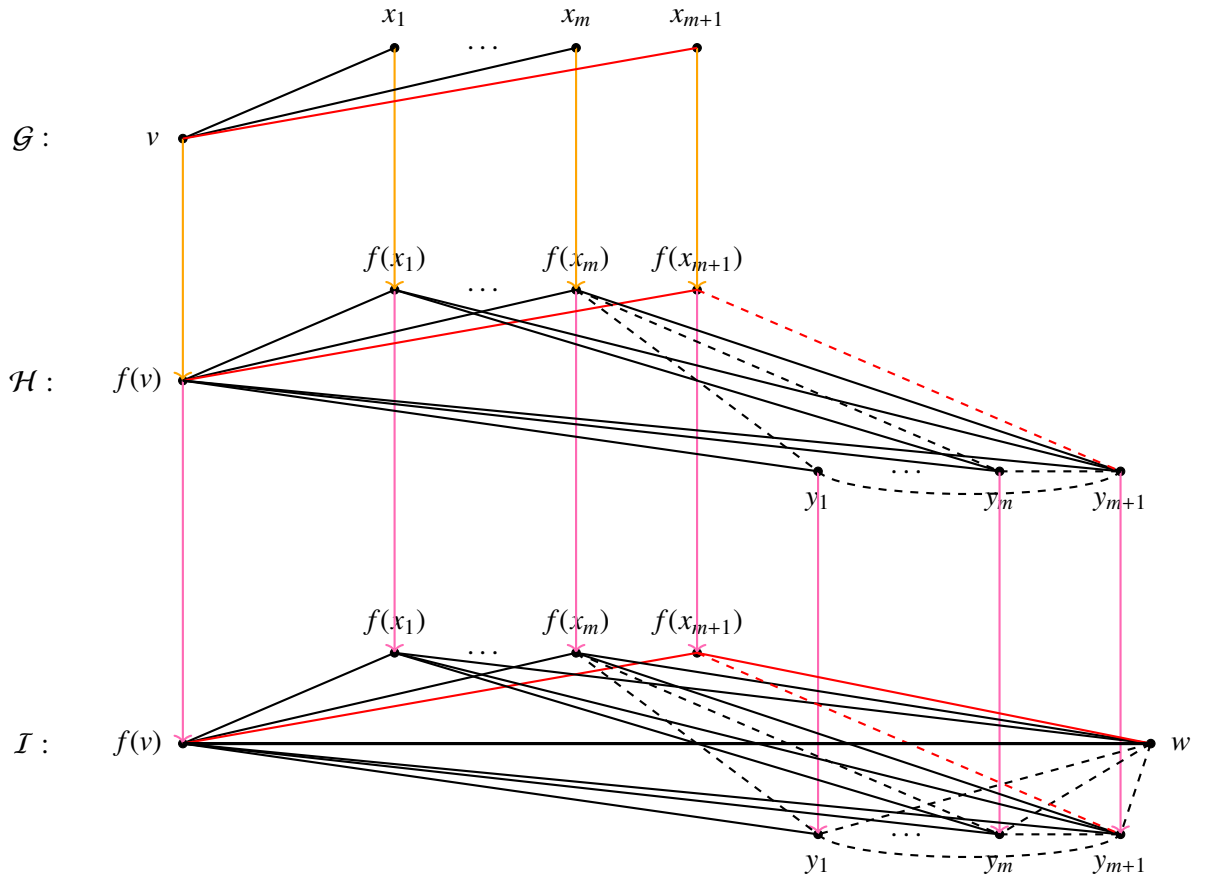
To preserve skew fibration property for f , we need a vertex $x_{m+1} \in \mathcal{X}$ such that:

$$(f(x_{m+1}), y_{m+1}) \notin R_{\mathcal{H}} \quad (\text{B.12})$$

On the other hand, the induction hypothesis for y_{m+1} shows that $\forall i = 1 \dots m, y_{m+1} R_{\mathcal{H}} f(x_i)$ which leads to $x_{m+1} \notin \{x_1 \dots x_m\}$. Moreover, from (B.12) and the fact that we assumed $f \circ g$ is not a skew fibration, we can deduce that:

$$f(x_{m+1}) R_{\mathcal{I}} w \quad (\text{B.13})$$

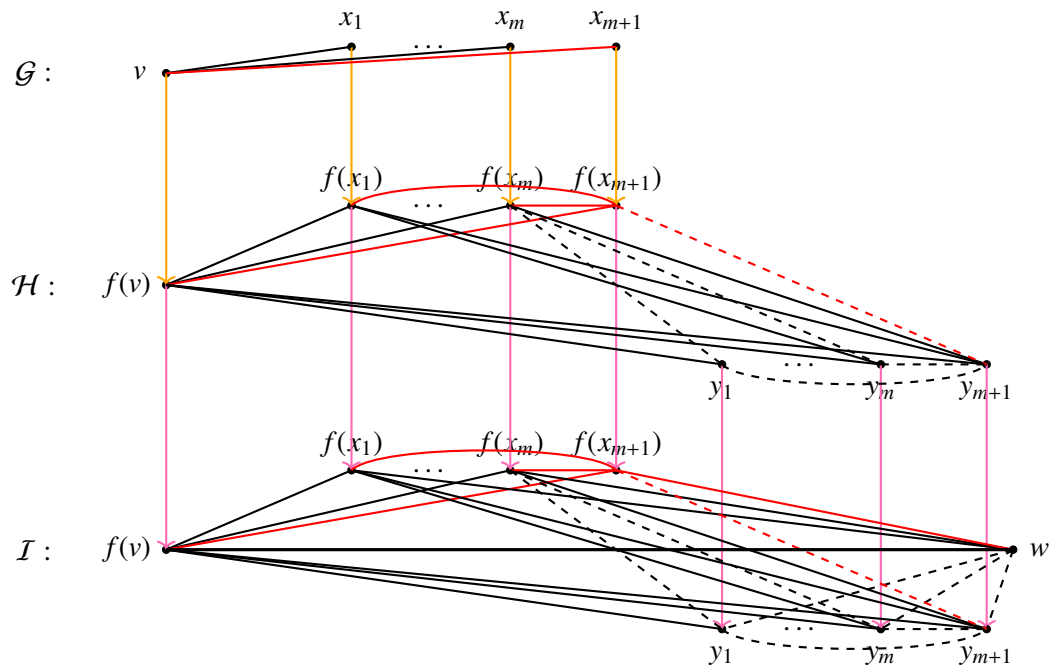
This is indicated below.



Now we have:

$$\forall i = 1 \dots m. f(x_{m+1})R_{\mathcal{I}}f(x_i) \tag{B.14}$$

because otherwise, $y_{m+1}R_{\mathcal{I}}f(x_i)R_{\mathcal{I}}wR_{\mathcal{I}}f(x_{m+1})$ would be a \mathcal{P}_4 (because of (B.5), (B.6), (B.9), and (B.13)). The edges added to $R_{\mathcal{I}}$ and $R_{\mathcal{H}}$ are indicated in the picture below:



The induction hypothesis b1 for x_m shows:

$$\forall i = 1 \dots m. (f(x_m), y_i) \notin R_{\mathcal{I}} \quad (\text{B.15})$$

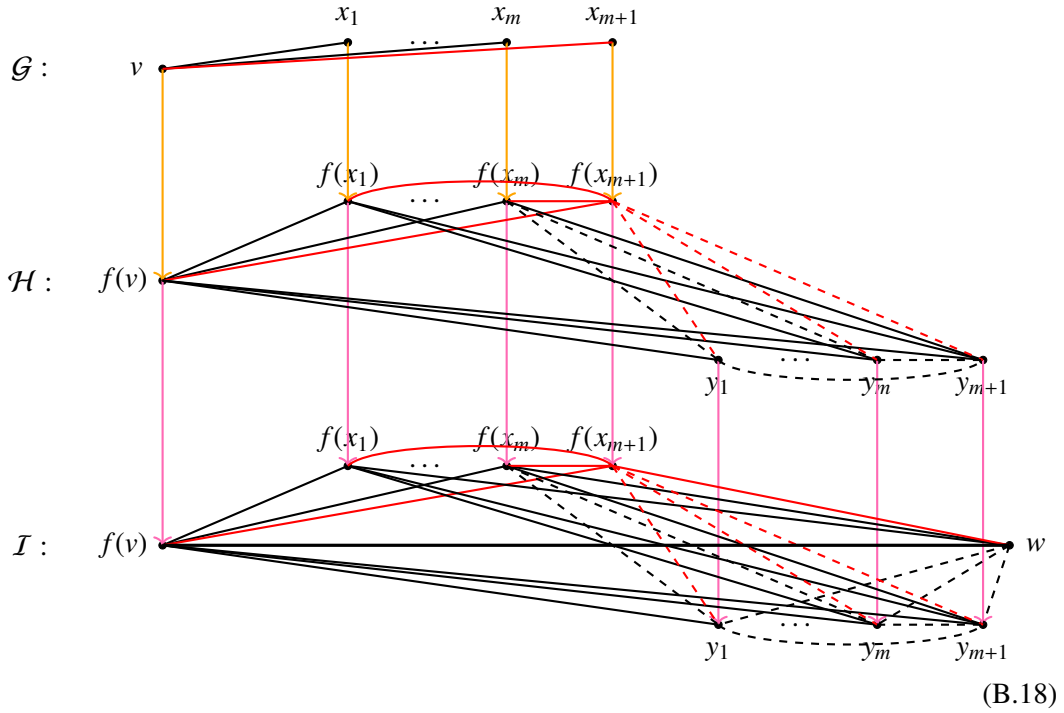
Additionally, from the induction hypothesis for y_{m+1} we have:

$$\forall i = 1 \dots m. (y_{m+1}, y_i) \notin R_{\mathcal{I}} \wedge y_{m+1} R_{\mathcal{I}} f(x_i) \quad (\text{B.16})$$

One can see the edges we talked about in the last two equations in the last figure. The equations (B.12), (B.15), (B.16) and (B.14) show that \mathcal{I} should always remain a cograph:

$$\forall i = 1 \dots m. (f(x_{m+1}), y_i) \notin R_{\mathcal{I}} \quad (\text{B.17})$$

This equation states that there should exist no edge between $f(x_{m+1})$ and y_i for all $i = 1 \dots m$. Otherwise, $y_{m+1} R_{\mathcal{I}} f(x_m) R_{\mathcal{I}} f(x_{m+1}) R_{\mathcal{I}} y_i$ would be a \mathcal{P}_4 in \mathcal{I} which contradicts the fact that \mathcal{I} is a cograph. The following picture shows the completed sequence after adding x_{m+1} in this step.



With these induction steps we have shown under the assumption that $g \circ f$ is not a skew fibration, we can create an infinite sequence of disjoint vertices. Since \mathcal{I} is finite, we have a contradiction. \square

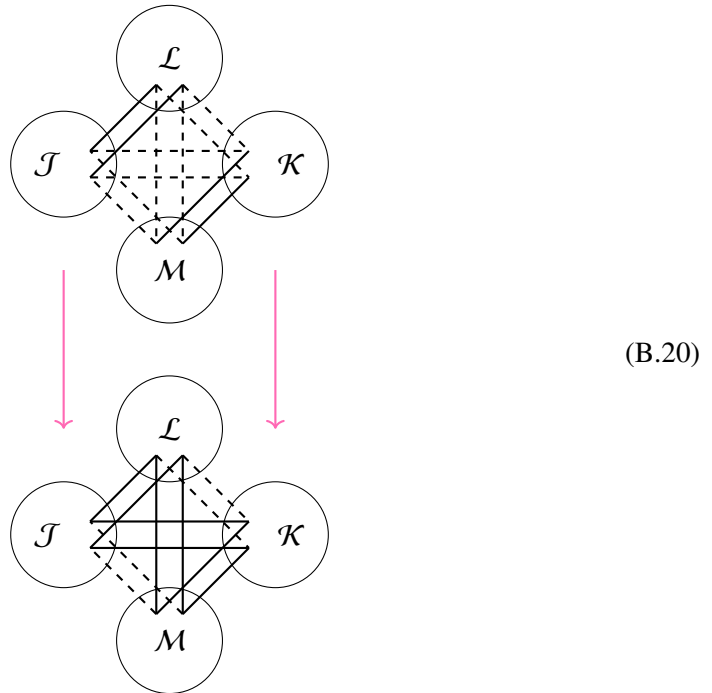
Lemma B.5. *If $f: \mathcal{G} \hookrightarrow \mathcal{H}$ is a skew fibration and $g: \mathcal{H} \twoheadrightarrow \mathcal{I}$ is a bijective skew fibration where \mathcal{H} and \mathcal{I} are cographs, then $f \circ g$ is a skew fibration.*

Proof. Proposition 6.7 in [69] shows that every bijective skew fibration is nothing but a series of consecutive medial rules from SKS shown in Figure 2.4. Since g is a bijection, assume without loss of generality that $\forall v \in V_{\mathcal{H}}. g(v) = v$. We proceed the proof by induction on the number of medial rules.

- **number of medial rules = 1:** Assume by way of contradiction that $f \circ g$ is not a skew fibration. This means that:

$$\exists v \in V_{\mathcal{G}}. \exists w \in V_{\mathcal{I}}. g(f(v))R_{\mathcal{I}}w \wedge \forall x \in V_{\mathcal{G}}. vR_{\mathcal{G}}x \rightarrow g(f(x))R_{\mathcal{I}}w \quad (\text{B.19})$$

In the case that $f(v)R_{\mathcal{H}}w$, we can make an infinite sequence of nodes $x_1, y_1, x_2, y_2, \dots$ with the same edges in $R_{\mathcal{H}}$ as the sequence created in Lemma B.3. The proof for this case is also the same as the one in Lemma B.3. The only difference is that we state everything for \mathcal{H} instead of \mathcal{I} . If $(f(v), w) \notin R_{\mathcal{H}}$, one can observe that $f(v)$ and w are in two different modules participating in the medial rule because an edge between them is being added in $R_{\mathcal{I}}$. Assume $\mathcal{J}, \mathcal{K}, \mathcal{L}$, and \mathcal{M} are the modules affected by the medial rule where $f(v) \in \mathcal{I}$ and $w \in \mathcal{K}$. Let $\mathcal{J} = \mathcal{G}(A)$, $\mathcal{K} = \mathcal{G}(B)$, $\mathcal{L} = \mathcal{G}(C)$, and $\mathcal{M} = \mathcal{G}(D)$. This means that the medial is changing $(A \wedge C) \vee (B \wedge D)$ to $(A \vee B) \wedge (C \vee D)$. We show this transformation below.



In (B.20), we show the change from $\mathcal{G}((A \wedge C) \vee (B \wedge D))$ to $\mathcal{G}((A \vee B) \wedge (C \vee D))$ using the medial rule (in other words using g). At first, all the nodes in $\mathcal{G}(A)$ (respectively $\mathcal{G}(B)$) are connected to all the nodes in $\mathcal{G}(C)$ (respectively $\mathcal{G}(D)$). With applying the medial rule or g , we obtain the same vertices in the graph but the edges between vertices in $\mathcal{G}(A)$ (resp. $\mathcal{G}(C)$) and $\mathcal{G}(B)$ (resp. $\mathcal{G}(D)$) will be added to this graph, where the pink downward arrows show the application of the medial rule or the bijective homomorphism g .

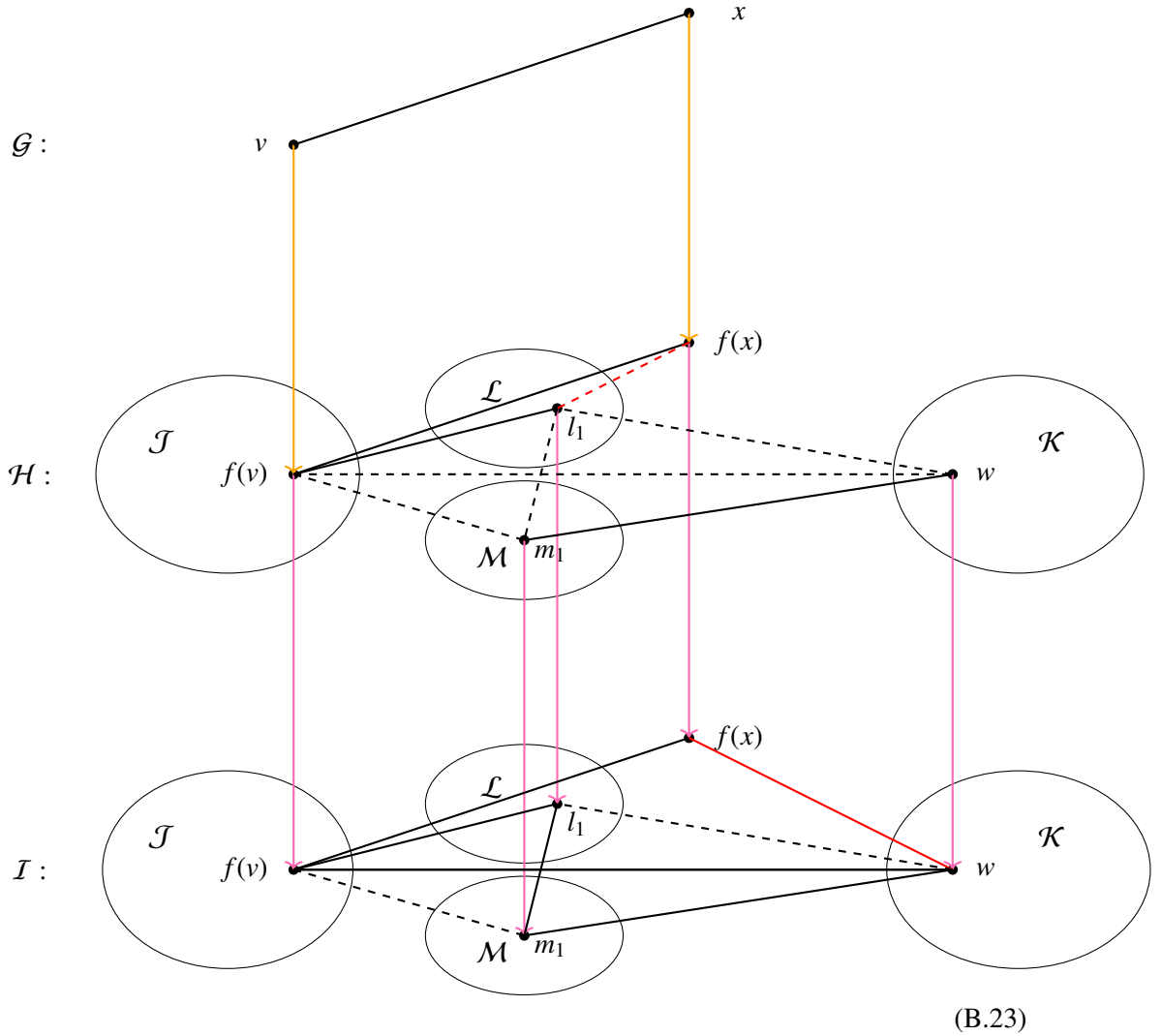
Since f is a skew fibration and there exists an edge between $f(v)$ and each vertex in \mathcal{L} (because $f(v) \in \mathcal{J}$), one of the neighbors of v in $V_{\mathcal{G}}$ (let us name it x) should have the following property (for $l_1 \in \gamma$):

$$(f(x), l_1) \notin R_{\mathcal{H}} \quad (\text{B.21})$$

Moreover, following from the fact that x is a neighbor of v in \mathcal{G} and contradiction hypothesis we can deduce that:

$$f(x)R_{\mathcal{I}}w \quad (\text{B.22})$$

The following figure demonstrates the situation of the graphs in the initial case of induction hypothesis where we first found x . The red dashed line refers to (B.21) and the red line shows (B.22). I also have outlined the modules $\mathcal{J}, \mathcal{K}, \mathcal{L}$, and \mathcal{M} .



We can notice that $f(x)l_1wf(v) \notin R_{\mathcal{H}}$ and $f(x)g_1f(v)l_1 \in R_{\mathcal{H}}$. In order to preserve the cograph property for \mathcal{H} we need to state:

$$(f(x), w) \notin R_{\mathcal{H}} \quad (\text{B.24})$$

Equations (B.22) and (B.24) show that an edge has been added between $f(x)$ and w when we applied the medial rule (or g) on \mathcal{H} . However, we know that $f(x)$ is not one of the vertices affected by medial rule ($f \circ g$ is not a skew fibration) so g is not able to add this edge which contradicts with the process of finding x . Consequently, at least one neighbor of v in $V_{\mathcal{G}}$ is in \mathcal{L} after applying f and this vertex is the one we need to show $f \circ g$ is a skew fibration.

- **number of medial rules** = $n + 1$: By Proposition 6.7 in [69] we know that the n first medial rules are equal to a bijective skew fibration $g_1: \mathcal{H} \rightsquigarrow \mathcal{H}'$ where \mathcal{H} and \mathcal{H}' are cographs. By induction hypothesis, we know that $f \circ g_1$ is a skew fibration. Assume $g_2: \mathcal{H} \rightsquigarrow \mathcal{I}$ is the bijective skew fibration corresponding to the $n + 1$ th medial rule. By applying the induction hypothesis once again, we can deduce that $f \circ g_1 \circ g_2$ is a skew fibration.

□

Proof for Theorem B.1. Using Lemmas B.2, B.3, B.4, and B.5, it is immediate that $f \circ g$ is a skew fibration. □



Titre : Flux Combinatoires

Mots clés : preuves combinatoires; inférence profonde; théorie de la démonstration; flux atomiques; calcul séquentiel; déduction naturelle; normalisation; élimination des coupures

Résumé : Dans cette thèse, nous introduisons des flux combinatoires pour les preuves en logique propositionnelle classique comme réponse possible au problème de l'identité des preuves et de la comparaison des preuves que nous appellerons le 24^{ème} problème de Hilbert. Les flux combinatoires sont une représentation graphique de preuves inspirés des propriétés des flux atomiques et des preuves combinatoires. Des flux atomiques, introduits par Guglielmi et Gundersen, ils héritent de la correspondance étroite avec la déduction ouverte, de la possibilité de retracer les occurrences des atomes dans une dérivation et de la compositionnalité libre. Des preuves combinatoires, introduites par Hughes, elles héritent du critère de correction qui permet la

reconstruction de la dérivation à partir du flux et la différenciation entre la partie linéaire et la partie gestion des ressources de la preuve. En fait, nous montrerons que les flots combinatoires forment un système de preuve au sens de Cook et Reckhow. De plus, nous établissons une traduction entre les dérivations de différents systèmes de preuve (à savoir la déduction naturelle, le calcul séquentiel et l'inférence profonde) et les flux combinatoires, ainsi que leur lien avec les preuves combinatoires avec coupures. Enfin, nous discuterons des formes normales et des procédures de normalisation sur les flux combinatoires tout en introduisant la purification comme sous-procédure concernée par la présence d'unités dans le langage.

Title : Combinatorial Flows

Keywords : combinatorial proofs; deep inference; proof theory; atomic flows; sequent calculus; natural deduction; normalization; cut elimination

Abstract : In this thesis, we introduce combinatorial flows for proofs in Classical propositional logic as a possible answer to the problem of proof identity and proof comparison which we refer to as the 24th problem of Hilbert. Combinatorial flows are a graphical representation of proofs inspired by properties of atomic flows and combinatorial proofs. From atomic flows, introduced by Guglielmi and Gundersen, they inherit the close correspondence with open deduction, the possibility of tracing the atom occurrences in a derivation, and the free compositionality. From combinatorial proofs, introduced by Hughes, they inherit the correctness criterion that allows the reconstruction of

the derivation from the flow and the differentiation between the linear and the resource management part of the proof. In fact, we will show that combinatorial flows form a proof system in the sense of Cook and Reckhow. Moreover, we will show how to translate between derivations from different proof systems (namely natural deduction, sequent calculus, and deep inference) and combinatorial flows, and we will show their relation with combinatorial proofs with cuts. Finally, we will discuss normal forms and normalization procedures on combinatorial flows while introducing purification as a sub-procedure concerned with the presence of units in language.