HAL
open science

Randomized strategies in concurrent games

Daniel Stan

To cite this version:

Daniel Stan. Randomized strategies in concurrent games. Computer Science and Game Theory [cs.GT]. Université Paris Saclay (COmUE), 2017. English. NNT : 2017SACLN011 . tel-01519354

HAL Id: tel-01519354

https://theses.hal.science/tel-01519354v1

Submitted on 6 May 2017

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.

NNT : 2017SACLN011

Thèse de doctorat de l'Université Paris-Saclay préparée à l'École normale supérieure de Cachan (École normale supérieure Paris-Saclay)

Ecole doctorale n°580
Sciences et Technologies de l'Information et de la Communication
Spécialité de doctorat : Informatique
par

M. Daniel Stan

Stratégies randomisées dans les jeux concurrents

Thèse présentée et soutenue à Cachan, le 30 mars 2017.

Composition du Jury :

Mme.	Nathalie Bertrand	Chargée de recherche INRIA	(Rapporteuse)
M.	Antonín Kučera	Professeur Université Masaryk	(Rapporteur)
Mme.	Johanne Cohen	Directrice de recherche CNRS	(Examinatrice)
M.	Olivier Serre	Directeur de recherche CNRS	(Examinateur)
M.	Joost-Pieter Katoen	Professeur Université RWTH	(Président)
M.	Nicolas Markey	Directeur de recherche CNRS	(Directeur de thèse)
Mme.	Patricia Bouyer-Decitre	Directrice de recherche CNRS	(Directrice de thèse)

Laboratoire Spécification et Vérification
École Normale Supérieure Paris-Saclay, UMR 8643 du CNRS
61 avenue du Président Wilson, 94235 Cachan Cedex, France

Remerciements

Tout d'abord, je tiens à remercier les membres de mon jury, qui ont accepté d'assister à ma soutenance, et particulièrement Nathalie Bertrand et Antonín Kucera qui m'ont fourni de nombreux retours et suggestions de corrections, dans un délai très court. Nicolas et Patricia me connaissent depuis maintenant quatre ans, et je leur suis également profondément reconnaissant pour leur travail d'encadrement et leurs conseils, qui m'ont énormément appris sur la façon de présenter mes travaux, que ce soit à l'oral ou à l'écrit[1]. Plus encore, ils ont su, tout du long, m'apporter la motivation et surtout l'enthousiasme qui m'a permis d'apprécier la recherche, même lorsque je pensais, fréquemment et très souvent à tort, me trouver dans une impasse. J'ai bien sûr retrouvé cette dynamique avec d'autres chercheurs, tels qu'Arnaud Sangnier et Mickael Randour, avec qui j'ai apprécié les nombreuses discussions et découvertes enthousiastes devant un tableau tout griffonné. Une autre source d'inspiration non négligeable m'est venu de l'enseignement en tant que chargé de TD, que ce soit grâce à des encadrants dynamiques, autrefois mes professeurs, comme Serge Haddad, Paul Gastin, Stefan Schwoon, Sylvain Schmitz et Alain Finkel ainsi que des élèves, trop nombreux pour être cités ici mais toujours pleins de ressources.

Le LSV est un laboratoire très agréable à vivre où l'on s'intègre facilement [2] et les discussions et débats[3] se lancent facilement autour du café lors d'une pause. Je garde des souvenirs mémorables des thésards et stagiaires du « fond du couloir », passés ou actuels, avec lesquels j'ai passés de très bons moments : Anthony, Antoine, Samy, Simon, Rémy, Guillaume. Grâce à eux, j'ai pu rencontrer Nadine, et surtout Monsieur Pierre-Jacques Henri, qui m'ont apporté de nombreux conseils avisés. Je n'oublie pas non plus les autres doctorants, notamment Jérémy, Lucca, Yann, Patrick, Pierre C., François T., mais aussi Marie van Den Bogaard qui a guidé le groupe des doctorants pendant quasiment toute ma thèse, et m'a aussi fourni de précieux conseils avant ma propre rédaction et soutenance. Les chercheurs ayant parfois peur des formalités administratives, je tiens à remercier Catherine, Imane, et surtout Virginie pour leur travail, et régulièrement pour leur patience et leur pédagogie. Hugues et Francis ont également fait un travail remarquable au niveau informatique et je dois reconnaître que j'ai pris plaisir à discuter quelque fois avec eux de sujets techniques, que certains qualifieraient de « cambouis ».

En parlant de cambouis, il me serait difficile de faire l'impasse sur le Crans, qui m'a permis tout au long de ma scolarité à l'ENS de pratiquer une autre forme d'informatique, plus appliquée et complémentaire de toute la théorie déployée pendant ma thèse. Mais cette *grande association Cachanaise* ne serait rien sans la bande de « geeks » qui la compose, et que je ne nommerai que par leurs pseudonymes[4] : Nit, olasd, iota, Zelda, Harry, Tilgaht, Bernie, Chirac, Hamza, Guinness. C'est aussi de ce groupe d'amis qu'est née la colocation avec Pauline, PEB et 20-100 et tous les souvenirs inoubliables que nous avons partagé ces trois dernières années ensemble, et avec son « adhérence topologique », notamment Zadou et Corentin.

Merci aussi à Chloé, Baptiste, Tilgaht et Lucile, que je ne vois pas très souvent, mais dont j'apprécie toujours les retrouvailles, ainsi qu'à mes parents, et ma petite sœur, qui sans me pousser dans la voie que je poursuis aujourd'hui, ont toujours validé mes choix et m'ont

[1] Vive Tikz !

[2] même lorsque l'on refuse de goûter à la gastronomie du restaurant universitaire

[3] parfois même des trolls

[4] car il nous arrive de ne même plus savoir nous nommer autrement

encouragé à persévérer.

Le début de ma thèse a aussi été marquée par ma rencontre avec Maud, qui n'a jamais cessé de croire en moi, de me réconforter et de m'encourager, même après que nos chemins aient divergés, je chéris ces souvenirs. Je souris encore aux nombreux moments passés avec les « Grotas », mais aussi la « Trolloc », Ping, Marion, trolin, Ara, Nolwenn, Ariane, mais aussi Misc et Blupon. Ce dernier méritant ainsi d'être cité comme mon fidèle compagnon d'infortune (je lui souhaite bon courage pour sa future thèse) lors de cette dernière année de rédaction, de soutenance, et enfin de départ pour une nouvelle vi(ll)e.

Contents

1 Introduction **1**
1.1 Game theory and computer science . . . 2
1.2 Non-determinism, stochasticity and quantitative problems . . . 3
1.3 Strategic optimization . . . 3
1.4 Outlines . . . 4
1.5 Related work . . . 5
1.6 Structure of the Thesis . . . 6
1.7 Scientific publications . . . 6

2 Preliminaries **9**
2.1 Usual notations . . . 9
2.2 Operations over a monoid . . . 9
2.2.1 Monoid over words . . . 9
2.2.2 Relations over a set . . . 10
2.2.3 Paths . . . 10
2.2.4 Partial functions . . . 10
2.3 Sets and multisets . . . 10
2.4 Well-quasi-orders . . . 10
2.5 Probability theory . . . 11

I Mixed Strategies in Concurrent games **13**

3 Concurrent games **15**
3.1 Strategies . . . 16
3.2 Visibility of actions . . . 18
3.3 Semantics . . . 19
3.4 Outcome of a game . . . 19
3.5 Two-player zero-sum games . . . 21
3.6 Nash Equilibria . . . 25
3.6.1 Definition . . . 26
3.6.2 Sub-game characterization . . . 27
3.6.3 Subgame perfect equilibrium . . . 28
3.6.4 Example of equilibria . . . 29

4 Decidability of Nash Equilibria **31**
4.1 Tools . . . 32
4.1.1 One-shot games . . . 32
4.1.2 k-action matching-pennies games . . . 33
4.1.3 Embedded game . . . 34
4.2 Modules . . . 40
4.2.1 Rescale game . . . 40
4.2.2 Testing game . . . 43
4.2.3 Counting modules . . . 44
4.2.4 Description of the reduction . . . 46
4.3 Conclusions . . . 50
4.3.1 Unconstrained problem . . . 50
4.3.2 Qualitative objectives . . . 51
4.3.3 Summary . . . 55

5 Games that almost-surely terminate **57**
5.1 Avoiding cycling behaviours . . . 57
5.1.1 Non-cycling games . . . 59
5.1.2 Strong components . . . 61
5.1.3 Fixed point analysis . . . 61
5.2 Equilibria under imprecise deviations . . . 64
5.2.1 Restricting to memoryless deviations . . . 65
5.2.2 Existence theorem . . . 67
5.2.3 Discussions . . . 68
5.3 Computing stationary equilibria under imprecise deviations . . . 68

II Parametrized Stochastic Systems **71**

6 Interaction models **73**

7 Parametrized register protocols **79**
7.1 Non-deterministic transition system . . . 80
7.2 Parametrized reachability: a global picture . . . 82
7.3 Monotonicity . . . 82
7.3.1 Upward closed reachability objectives . . . 82
7.3.2 Non-atomicity . . . 83
7.4 Probabilistic transition system . . . 85
7.4.1 Qualitative analysis . . . 86
7.4.2 Cut-off property . . . 87

8 Probabilistic reachability and safety **89**
8.1 Existence . . . 89
8.2 Symbolic graph . . . 89

9 Almost-sure reachability **93**
9.1 First examples . . . 93
9.1.1 Atomicity prevents cut-off existence . . . 93
9.1.2 Symbolic graph is powerless . . . 94
9.2 Existence . . . 95
9.3 Bound examples . . . 97
9.3.1 Linear cut-off . . . 97
9.3.2 Counter machine . . . 99
9.3.3 PSPACE-hardness . . . 102
9.4 Decision procedure . . . 105
9.4.1 Refined symbolic graph . . . 105
9.4.2 Symbolic based algorithm . . . 106
9.4.3 Complexity bounds on covering . . . 107
9.4.4 General bounding scheme . . . 109

10 Extensions and discussions **113**
10.1 Model checking . . . 113
10.2 r-register protocol . . . 114
10.2.1 Tools enhancement . . . 114
10.2.2 Operations over r registers . . . 115
10.2.3 Discussion on the r-register extension . . . 118
10.2.4 Comparison with non-atomic protocols . . . 119
10.3 Process identifiers . . . 119
10.4 Conclusions . . . 119

11 Toward Strategy Synthesis **121**
11.1 Definitions . . . 121
11.1.1 Allowed actions and randomization . . . 121
11.1.2 Local strategies . . . 122
11.1.3 Semantics . . . 124
11.1.4 Cut-off property . . . 125
11.2 Reachability . . . 126
11.2.1 Mixed strategies . . . 126
11.2.2 Pure strategies . . . 126
11.2.3 Summary . . . 128
11.3 Safety . . . 129
11.4 Conclusions . . . 132

Bibliography **135**

List of figures **142**

List of tables **145**

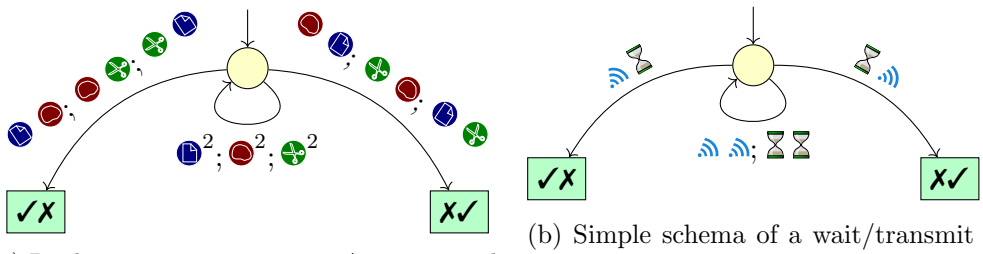
Résumé en français **146**

Chapter 1

Introduction

Stochastic behaviours are a particular form of uncertainty that also allow desynchronization and symmetry breaking mechanisms. Imagine for example two persons playing the well-known rock-paper-scissors game as in Figure 1.1a, on a daily basis. The game consists in a series of rounds, where each of them consists in the two persons picking a symbol among 🪨, 📄 and ✂. The game continues until a winner is determined, that is when a tie is not triggered. As these players often play this game against each other, they may start to learn from each other and infer over time how their opponent reasons. Each day that passes, the players can decide to change their mind according to what was played the day before, which will result in an endless oscillation of their played symbols, or keep playing the same way. This may require them to remember the beginning of the previous plays, or at least the beginning of the sequence of played symbols. One day, one of the players is bored of this game and instead of trying to "outsmart" their opponent as usual, decides at each round, to roll a dice and play 🪨 if the value is 1 or 2, 📄 if the value is 3 or 4 and ✂ otherwise. As soon he decides to play according to this scheme, he secures his win with probability $\frac{1}{2}$. No matter what the other player does, there is no point in trying to guess an opponent behaviour nor remembering what he has played the day before, thus the game becomes easier to play for both of them. Playing at random may also be a time-saver, as every day, players are now guaranteed to spend only $\frac{3}{2}$ rounds on average.

In this thesis, we study formal verification and synthesis of systems with stochastic behaviours. Randomization in the computer science field is indeed a key tool to desynchronise processes, to avoid deadlock situations or collisions in communication protocols. The global



(a) Rock-paper-scissors game. A tie resets the game.

(b) Simple schema of a wait/transmit (⌛/📶) mechanism for two entities that share the same medium. Any collision resets the game.

Figure 1.1 – Simple games that require randomization

picture in this context is very close to our previous rock-paper-scissors game, as shown in Figure 1.1b. Many protocols rely on randomization, like CSMA/CD [CSM00], which requires its participants to randomize their sending in case of a previous collision. As a consequence, considering stochasticity in this protocol is of fundamental feature for its analysis [DKN+12]. Another example of desynchronization through randomization occurs for the distributed assignment of link-local addresses as specified by the Zeroconf protocol [BSHV03, AGC05].

With the rise of open-source softwares and the progresses of code disassembly, but also because communication protocols are publicly described, it is reasonable to assume that the source code of any program is in fact a public information. This has security implications since several algorithms and data structures have a reasonable complexity on average, but a huge complexity in the worst case, that can be triggered by an attacker. For example, with limited computational power and bandwidth, an attacker could trigger many collisions in the deterministic hash-table of a server-side program and bring down a large computational structure [CW03]. In this context, randomization is not only a way to ensure fairness during a transmission, but also a natural counter-measure against an adversary.

1.1 Game theory and computer science

Game theory and strategic reasoning is a prolific domain, introduced by Von Neumann [Neu28] with his famous minimax theorem, and later more complete formalisms [VNM47]. This framework has been successfully applied to economics and sociology by Nash, Selten and others [Nas50, Sel65], as it provides a relevant model for the behaviours and interactions of people and entities, that are seen as *players* or *agents*, each one of them making decisions that affect all players. Decisions or *actions* are chosen by each player, according to their *strategy*, which can possibly take into account some *observation* of the game and its environment. Each player is given a *goal* or *objective*, usually a utility function or more generally a preference relation. The utility is rewarded to each player according to the actions played by all the players.

These games are usually given as payoff matrices, that is to say the game is described as a punctual event in time. The previous example and related communication protocols include a sequential aspect, that is often represented in an automata theoretical framework in the computer science point of view. For example, the Church's problem asks whether a binary relation on infinite sequences of words, represented as a logical formula, can be implemented in terms of a circuit that takes the first infinite sequence as an input, and produces one of the corresponding possible infinite sequence as an output. The problem can be represented by a game played on a graph between two players, the first one proposing an output of the game by choosing an edge labelled with the corresponding letter of the output sequence, while the other tries to exhibit an input sequence, letter by letter that doesn't correspond to the output. Implementing the circuit corresponds to a winning strategy for the first player [BL69], hence solving the game.

This methodology is typically well-suited for *reactive systems*, where we are interested in the synthesis of a controller that should satisfy a logical property against an unpredictable environment [ALW89]. Here, the first player is *existential*, and we shall name her "Eve", as we are interested in a single possible winning strategy, whereas the second player is *universal*, namely "Adam", and we consider all his possible behaviours.

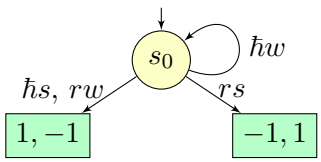


Figure 1.2 – $\mathcal{H}$ has no optimal strategy

1.2 Non-determinism, stochasticity and quantitative problems

When we say that a player is winning against all possible strategies, we adopt a *worst-case* approach since all possible actions are taken into account. One can legitimately argue that this form of uncertainty is very strong as we may consider irrelevant scenarios, that may not happen in practice. For example the simple transmission model depicted before, can be assumed to enjoy a fairness property, that is to say, the existence of infinitely many time slots available for emission.

An intermediate notion of uncertainty is stochasticity, namely the environment, which is usually introduced as a particular player that takes no decision but has a stochastic behaviour. As opposed to a universal player, the environment has no objective and can help or work against the existential player. Introduction of probabilities into a model has two major consequences: first of all, some sets of possible executions from the worst-case approach are discarded. For example, a tie loop is improbable in the rock-paper-scissors game as soon as one player plays at random. This can be related to certain forms of fairness properties, that are otherwise harder to model. On the other hand, stochasticity brings new quantitative questions. For example, we can ask whether a player can win with probability larger than a given threshold.

1.3 Strategic optimization

Several concepts of optimality for strategies have been proposed, depending on the desired properties. An *optimal* strategy is the first natural solution concept where the strategy of a given player has to maximize the utility of that player, no matter the actions of the other players. This concept leads to the notion of value [Sha53]. In this context, games are assumed to be played by two players with antagonistic objectives, namely *zero-sum*. Although the resulting values for both players do not necessarily sum to zero, *determinacy* results state that this intuition is correct for a wide range of games when players play in turn without randomization but with regular objectives [Mar75].

When allowing concurrent moves and randomization, optimal strategies may not exist even though the game is determined, as the value may be asymptotically reached [Mar98]. Consider for example the concurrent 2-player zero-sum game $\mathcal{H}$ (*hide-or-run*), depicted in Figure 1.2. Player 1 can either hide ($\hbar$) or run home (r) while player 2 tries to shoot him, with a snowball. If player 2 shoots while player 1 is hiding, she loses her snowball and loses the game. If player 2 shoots while the other player is running, she wins. This game has been shown by [KS81] to have no optimal strategy although player 1 can ensure to win with probability arbitrarily close to 1.

When the game involves more than two players, or when objectives are compatible from

one player to another, we may argue that the notions of optimal strategy and value are not suited anymore: players may not necessarily play against each other anymore. The equilibrium notion introduced by Nash [Nas50] plays a central role for the study of non-zero-sum games. In such an equilibrium, we are interested in particular strategies, one for each player, such that for each player, her strategy is optimal when played together with the particular strategies of the other players, that are already fixed. On a graph structure, an equilibrium forms a particular cooperation mechanism between the players, in order to visit profitable states. What happens when a player does not respect the agreement depends on the exact notion considered, as players in a Nash equilibrium prefer to retaliate to maintain their equilibrium, instead of continuing to optimize their own utility. This last observation justified the introduction of particular notions of Nash equilibria called subgame perfect equilibria [Sel65], where equilibrium must be ensured from any state.

The assumed behaviour of the other players, that may want to optimize their own utility is called *rationality*. This hypothesis is particularly sound in the computer science area, as players are programs, autonomous systems, or individual devices, that interact together, through a protocol or some predefined patterns. Hence, a strategy for a player can be seen as a program or firmware, that we may, depending on the studied problem, try to *synthesize* or simply assume to exist [KPV15].

1.4 Outlines

We focus in this thesis on the power provided by stochasticity in games represented by finite control-flow graphs, and mainly consider reachability and safety objectives of a particular state. This choice can be considered as a first step to more general verification studies.

We are particularly interested in the study of the decidability and computability statuses of solution concepts from the perspective of the following parameters: (a) the number of agents involved, (b) the way they interact with each other, (c) the allowed memory in strategies, (d) the allowed stochasticity in strategies, (e) the unpredictability of environment (stochastic, and/or non-deterministic), (f) the relaxation of the solution concept.

We consider first a classical framework of concurrent games played on graphs by a fixed number of agents, with stochastic behaviours coming from the environment and the strategies. More precisely, we study how concurrent actions and stochastic strategies together harden the decidability problem of the existence of a Nash equilibrium. In this general situation, which we will see is undecidable, we investigate relaxed notions of equilibria to regain decidability.

Despite their very general specification, concurrent games have a rigid structure that we argue is not suited to the representation of concrete problems. One main challenge that we address is the study of a model that is flexible enough to capture the stochastic interactions between an arbitrary number of agents, while remaining simple enough to keep decidability and computation possible and tractable.

Therefore, we consider a model where the number of agents is a *parameter*, and consider behaviours for large values. Because of this parametrized setting, the model of interaction has to be discussed and made precise. We proceed to the study of interactions made through a shared variable with finite domain, that each agent can access asynchronously. The system is somehow turn-based, however, the order is chosen according to a (non-)deterministic or stochastic scheduler, seen as an environment. On the one hand, this model is more general than games as it is parametrized, that is to say it involves an arbitrary number of players.

On the other hand, it is more restricted since the communication primitives are limited to a particular case of shared-register access between the players. Again, we study which role a stochastic environment plays in this context. Then, we address the existence and even the synthesis of strategies for players by characterizing the memory and randomization that they may require. To simplify the study in this second part, our work focuses only on qualitative reachability properties, that is to say reachability with probability 1, or strictly less than 1.

1.5 Related work

The seminal result of Nash [Nas50], shows that any one-shot game has an equilibrium in mixed strategies. The crucial point of the proof is the introduction of strategy randomization to ensure continuity then existence of a fixed point of a well-chosen function, for example by applying Brouwer's fixed-point theorem. Such arguments can be adapted to some games played on graphs, for example in the *finite horizon* case, when the length of the play is bounded. Discounted utility functions is a special case of *infinite horizon*, where reward on each state exponentially decreases over time. Players are therefore encouraged to apply their best strategies in the beginning of the play, which ensures continuity then existence of a Nash equilibrium [Fin64]. The undiscounted infinite horizon case is usually harder, even when the game involves only one player interacting with her stochastic environment. Such a game can be seen as a *Markov Decision Process*, for which a family of asymptotically optimal strategies is known to exist [Put94]. From a mathematical point of view, games with at least two players are also studied under the name *Competitive Markov Decision Process* with similar discounted and undiscounted objectives [FV96].

Here, our objectives consist in reachability and safety objectives on states, which can be seen as a particular notion of undiscounted objective. Existence is ensured for games with safety objectives for all players, that is to say staying inside a subset of safe states [SS01]. In the reachability case, the existence of a Nash equilibrium remains open, nonetheless, a relaxed notion of equilibria where each player can still deviate to win up to a fixed imprecision $\varepsilon > 0$ is known to exist for terminal-reachability objectives [CJM04]. Extensions to ω-regular objectives have been considered in the two-player case [Cha05], while the general case remains open. A more complete survey of known results of Nash equilibria and stochastic games in general can be found in [CH12].

When restricting to deterministic games without randomization, a study of ω-regular objective is possible [Bre12], by encoding any Nash equilibrium in a two-player game. Stochastic games have also been studied in the turn-based case by [Umm10], where the decision problem of the existence of a general Nash equilibrium is shown undecidable. More precisely, the author shows that it is undecidable to determine whether a game involving 11 players with terminal rewards and where a given player wins almost-surely has a Nash equilibrium. When restricting to memoryless strategies, we can decide in PSPACE whether there exists a Nash equilibrium whose average payoff profile lies between two given threshold profiles, which provides a method for approximating the resulting payoff. This has to be compared to the complexity of computing a Nash equilibrium in matrix games which is PPAD-complete, even with only two players [CDT09]. Note that the two studies focus on decision and computation problems since the existence of a Nash equilibrium is not ensured in their restriction. Indeed, randomization and concurrent actions are the main ingredient of the existence proof.

Considering an arbitrary number of agents is a form of parametrized system, for which

verification has been introduced by German and Sistla [GS92]. In general, systems that interact together, concurrently, can be modeled as a vector-addition system, or equivalently a Petri net. Reaching a particular control state by any agent can be seen as a coverability property, which is known to be decidable and EXPSPACE-complete [Rac78, Lip76]. Therefore we realize that parametrized verification may be an already difficult problem and has mainly been studied in the non-stochastic case. However, some studies of population protocols, like [EGLM15], include fairness properties and ask for reachability of a strongly connected component, which relates to almost-sure reachability, that is to say a qualitative property. Petri nets and population protocols are nonetheless very powerful models, that we may want to restrict. Several restricted communication primitives are presented in [Esp14], and their impact on the complexity of verification problem is discussed. The non-atomic shared register protocols, chosen as the communication scheme of the second part of the thesis, have been for example studied in [EGM13, DEGM15]. The authors consider only the non-stochastic case, but do not restrict their analysis to finite control-flow graph and also consider agents described by pushdown machines. Similar models as reconfigurable broadcast networks [BFS14], can also be studied with their stochastic behaviours, while local distributed strategies are introduced in [BFS15].

1.6 Structure of the Thesis

After providing in Chapter 2 the basic concepts, tools and notations used along the thesis, the rest of the manuscript is divided into two independent parts:

- The first one is devoted to the study of concurrent stochastic games with terminal-reachability objectives and a fixed number of agents. Chapter 3 defines the whole game framework, from stochastic arena to state-of-the-art results on Nash equilibria. Chapter 4 develops a proof of the undecidability of the existence of Nash equilibria and discusses several implications. Finally, Chapter 5 introduces a relaxed notion of equilibria that is proven to always exist and enjoy computability properties.

- In the second part, we shift to a more realistic model where the number of involved agents is not fixed *a priori* and information is only partial. Here, we mainly focus on qualitative reachability properties of our systems, assuming that the behaviour of agents is fixed. Chapter 6 introduces the parametric verification concepts and relates the studied model to the different communication classes already considered in the literature. Chapter 7 introduces formally the model and describes the considered semantics. Chapter 8 recalls some preliminary results on the deterministic case, or equivalently, on the stochastic case with positive probability. Chapter 9 studies the almost-sure reachability question. The techniques are then generalized to some extensions in Chapter 10. The last Chapter 11 tries to bridge the gap with the first part, by introducing a local strategy concept for agents, and presents decidability results in some restricted cases.

1.7 Scientific publications

Several results presented here complete previous conference publications that the author of this thesis has co-authored: Chapter 4 describes the undecidability result of [BMS14], and exposes further discussions on the invisibility of played actions as well as several implications to

the study of Nash equilibria in games with qualitative reachability and safety objectives. The existence and computation of equilibria under imprecise deviations developed in Chapter 5 have been published in [BMS16]. Finally, the second part of the thesis extends [BMR+16], whose main theorem is presented in Chapter 9, to the case of a distinguished agent, called a leader, and to more general reachability objectives. Extensions to multiple registers and strategy concepts are also studied as further improvements.

Chapter 2

Preliminaries

We briefly describe in this chapter some notations and concepts that will be intensively used in the sequel.

2.1 Usual notations

We respectively denote by $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ the sets of natural, relative, rational and real numbers, with the usual comparison relation $\leq$.

For a set $A \subseteq \mathbb{R}$, we will write $A_{\geq 0} = \{x \in A \mid x \geq 0\}$, $A_{>0} = \{x \in A \mid x > 0\}$.

2.2 Operations over a monoid

Let $(R, \cdot)$ a monoid with identity element 1, and $A, B \subseteq R$, two subsets.

We extend notations on $\cdot$ to sets by $A \cdot B = \{a \cdot b \mid a \in A, b \in B\}$. For $x \in R$ and $n \in \mathbb{N}$,

$$x^0 = 1 \quad A^0 = \{1\} \quad \text{and} \quad x^{n+1} = x^n \cdot x \quad A^{n+1} = A^n \cdot A$$

We define the *Kleene plus* of A as the smallest subset of R containing A, and stable under $\cdot$ operation, and denoted $A^+ = \cup_{n>0} A^n$. Similarly, the *Kleene star* of A is written $A^* = A \cup \{1\} = \cup_{n \geq 0} A^n$.

We give several examples of monoids that will be considered through this thesis.

2.2.1 Monoid over words

For a finite set Σ called an alphabet, we consider R the set of finite sequences of elements of Σ, namely words. In particular, ε is the empty sequence, or empty word.

For a word $w \in R$, we denote by $|w| \in \mathbb{N}$ its length, and for any position i, for $1 \leq i \leq |w|$, we write $w[i]$ the i-th element of w. $\cdot$ is defined as the concatenation of two words: for any w_1, w_2, $w_1 \cdot w_2$ is defined as

$$\begin{aligned}
&|w_1 \cdot w_2| = |w_1| + |w_2| \\
&\forall 1 \leq i \leq |w_1| \ (w_1 \cdot w_2)[i] = w_1[i] \\
&\forall |w_1| + 1 \leq i \leq |w_1| + |w_2| \ (w_1 \cdot w_2)[i] = w_2[i - |w_1|]
\end{aligned}$$

We can check that $(R, \cdot)$ is indeed a monoid with identity element ε, and that $\Sigma^* = R$. For a word $w \in \Sigma^+$, we write $\mathit{first}(w) = w[1]$ and $\mathit{last}(w) = w[|w|]$. For $w, w' \in \Sigma^*$, we write $w \sqsubseteq w'$ whenever there exists $w'' \in \Sigma^*$ such that $w' = w \cdot w''$.

With the exception of the *last* operator, we extend previous notions on words to ω-words, defined on set Σ^ω.

2.2.2 Relations over a set

Let us now consider a set V and the class R of binary relations over V that is to say elements of the form $E \subseteq V \times V$. For $x, y \in V$, we write xEy whenever $(x, y) \in E$. For E_1 and E_2 two binary relations, we denote by $E_1 \cdot E_2 = \{(x, y) \in V^2 \mid \exists z \in V \ xE_1z \wedge zE_2y\}$. We can then define for any class $\mathcal{C}$ of binary relations $\mathcal{C}^*$ and $\mathcal{C}^+$. By analogy with the previous notations, we write $E^{-1} = \{(y, x) \mid (x, y) \in E\}$. Notice that in general, $1 \subset E^{-1} \cdot E$ but $1 \neq E^{-1} \cdot E$. E^* and E^+ are called respectively the transitive and strict transitive closure of E.

2.2.3 Paths

Let E be a binary relation over V. We denote by $\mathrm{paths}\,(E)$ the set of paths on E, that is to say:

$$\mathrm{paths}\,(E) = \left\{\pi \in V^+ \;\middle|\; \forall i \in \mathbb{N}\ 1 \leq i < |\pi| \Rightarrow (\pi[i], \pi[i+1]) \in E\right\}$$

For $x, y \in V$ and $\pi \in V$, we write $\pi : \quad x\ E^*\ y$ whenever $\pi \in \mathrm{paths}\,(E)$, $\pi[1] = x$ and $\pi[|\pi|] = y$.

2.2.4 Partial functions

Partial functions from V to V are particular binary relations E over V, that are stable by $\cdot$ operations, hence that form a sub-monoid.

For a property φ, we will write $\mathbb{1}_\varphi$ for the value 1 if φ holds and 0 otherwise. When X is a set, we also write $\mathbb{1}_X$ for the function $x \mapsto \mathbb{1}_{x \in X}$. For any function f defined on a monoid R, and any $x \in R$, we write $f(x \cdot -)$ for the function: $y \in R \mapsto f(x \cdot y)$.

2.3 Sets and multisets

If A is an arbitrary set, $\sharp\,[A]$ denotes its cardinal which is a natural number or ∞.

A multiset over A is a function $\mu : A \to \mathbb{N}$. Its cardinality is denoted by $|\mu| = \sum_{a \in A} \mu(a) \in \mathbb{N} \uplus \{\infty\}$ while its support is $\overline{\mu} = \{a \in A \mid \mu(a) > 0\}$. The set of multiset of cardinality n is written $\mathbb{N}_n^A$, while the whole set of multisets is $\mathbb{N}^A = \uplus_n \mathbb{N}_n^A$. $\oplus$ and $\ominus$ respectively denote point-wise addition and subtraction of multisets.

2.4 Well-quasi-orders

Let A be a set equipped with a quasi-order $\preceq$, that is to say a reflexive and transitive binary relation over A. We say that $(A, \preceq)$ is a well-quasi-order (wqo for short), whenever any infinite sequence $w \in A^\omega$ contains two elements $w[i]$ and $w[j]$ with $1 \leq i < j$ such that $w[i] \preceq w[j]$.

A subset U of A is upward closed whenever for all $x \preceq y$ and $x \in U$, we also have $y \in U$. When $(A, \preceq)$ is a wqo, we see that any upward closed set U can be represented by a unique

finite subset $B \subseteq U$, in the following way: $U = \{y \mid \exists x \in B.x \preceq y\}$. Remark that cardinality $\sharp\,[U]$ is potentially infinite, although U can be finitely represented.

2.5 Probability theory

In order to give semantics to our different models, we will equip them with a probability distribution over their possible runs. A run will usually be represented as a maximal word, possibly of infinite length, whose all finite prefixes are paths in the considered model.

Theorem 2.1 (Carathéodory's criterion [RF10])**.** *Let $\mathbb{P}$ a function of the form $\mathbb{P} : 2^{\Sigma^+} \to \mathbb{R}^{\geq 0}$ such that for all $w \in \Sigma^+$,*

$$\sum_{a \in \Sigma} \mathbb{P}(w \cdot a \cdot (\Sigma^* \uplus \Sigma^\omega)) \in \{0,\ \mathbb{P}(w \cdot (\Sigma^* \uplus \Sigma^\omega))\}$$

Then, $\mathbb{P}$ can be uniquely extended as a measure on the σ-algebra $\mathbb{B}(\Sigma)$ generated by the class of languages with a common prefix, namely $\{w \cdot (\Sigma^* \uplus \Sigma^\omega) \mid w \in \Sigma^*\}$. For any finite word $w \in \Sigma^+$, we usually identify notations $\mathbb{P}(w)$ with $\mathbb{P}(w \cdot (\Sigma^* \uplus \Sigma^\omega))$

If $w \sqsubseteq w'$ and $\mathbb{P}(w') > 0$, then by induction, we show that $\mathbb{P}(w) > 0$. Finite words w such that $\mathbb{P}(w) > 0$ and $\forall a\ \mathbb{P}(w \cdot a) = 0$, are called *terminal words*. For any infinite sequence of words of the form $(w_i)_{i>0}$ with $\forall i\ w_i \sqsubseteq w_{i+1}$ with $\forall i\ \mathbb{P}(w_i) > 0$, the limit of such sequence is a finite or infinite word $w \in \Sigma^\omega \uplus \Sigma^+$, which has measure $\lim_{i>0} \mathbb{P}(w_i)$.

Our main goal is to study probabilistic properties. As the notation $\mathbb{P}$ suggests it, for a given prefix $w_0 \in \Sigma^+$, if $\mathbb{P}(w_0) > 0$, then $\forall X \in \mathbb{B}(\Sigma) \mapsto \mathbb{P}(w_0 \cdot X)$ is a probability measure. In the following, if A is an at most denumerable set, $Dist(A)$ will denote the set of probability distributions over A. If μ is such a distribution, $\overline{\mu}$ denotes the support of μ, that is the subset $\{a \in A \mid \mu(a) > 0\}$. Pointwise addition for distributions will be written $+$, and multiplication by a scalar is written $\cdot$, so that for any two distributions μ and μ' on the same set A, and for any $p \in [0, 1]$, $p \cdot \mu + (1 - p) \cdot \mu'$ is still a distribution on A.

Part I

Mixed Strategies in Concurrent games

Chapter 3

Concurrent games

In this chapter, we introduce the game framework used in the first part. We describe how players can interact together, by the means of an arena, which roughly corresponds to a finite directed and labelled graph. Then, several classes of strategies are defined, that will later induce an outcome, seen as a probability measure over the possible runs in the arena. Game theory is focused on how players can enforce an outcome that is profitable for them, so we have to define how outcomes can be compared, that is to say to define payoffs for the players. Finally, we consider game theoretical concepts such as optimal strategies and Nash equilibria in our framework and discuss known results.

Definition 3.1. A concurrent arena $\mathcal{A}$ is a tuple $\mathcal{A} = (\mathsf{States}, \mathsf{Agt}, \mathsf{Act}, \mathsf{Tab}, (\mathsf{Allow}_i)_{i \in \mathsf{Agt}})$ where

- States is a finite set of states;
- Agt is a finite set of players;
- Act is a finite set of actions;
- For all $i \in \mathsf{Agt}$, $\mathsf{Allow}_i \colon \mathsf{States} \longrightarrow 2^{\mathsf{Act}}$ is a function describing authorized actions in a given state for Player i;
- $\mathsf{Tab} \colon \mathsf{States} \times \mathsf{Act}^{\mathsf{Agt}} \to \mathit{Dist}(\mathsf{States})$ is the transition function.

  A state $s \in \mathsf{States}$ is said *terminal* (or *final*) if it allows no action for any player: $\prod_{i \in \mathsf{Agt}} \mathsf{Allow}_i(s) = \varnothing$. We write $\mathtt{F}_{\mathcal{A}}$ (or simply $\mathtt{F}$ when the underlying arena is clear from the context) for the set of terminal states of $\mathcal{A}$.

We argue below that this model is very general, and captures several frameworks from the litterature.

We say that the arena is *deterministic* whenever the transition function is deterministic *i.e.*, only makes use of Dirac distributions, *stochastic* otherwise. When refering to an arena with $n = \sharp\,[\mathsf{Agt}]$ players, we can refer to the probabilistic behaviour as a last player, usually called the *environment*, who resolves the probabilistic transitions. As a consequence, we will say that an arena is a $n + 1/2$-player arena if it is stochastic, or n-player otherwise.

We say that the arena is *turn-based* whenever for any $s \in \mathsf{States}$, there is at most one player $i \in \mathsf{Agt}$ who has more than one allowed action: $\forall j \neq i \; \sharp\,[\mathsf{Allow}_j(s)] \leq 1$.

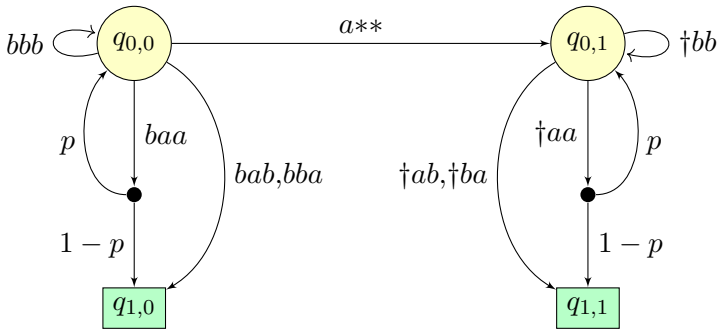


Figure 3.1 – Graphical representation of a 3+1/2-player arena. Player 1 can enforce transition from $q_{0,0}$ to $q_{0,1}$ by playing a. Player 1 has no move left from this state since $\mathsf{Allow}_1(q_{0,1}) = \{\dagger\}$, on the other hand, players 2 and 3 can trigger a probabilistic transition if they both play action a.

Graphical representation. In the sequel, we usually identify each agent $i \in \mathsf{Agt}$, with a positive integer from 1 to $\sharp[\mathsf{Agt}]$, that is to say $\mathsf{Agt} = [\![1, \sharp[\mathsf{Agt}]]\!]$.

Thus, an *action profile* of all players is a word $A \in \mathsf{Act}^{\mathsf{Agt}}$, of length $\sharp[\mathsf{Agt}]$, and for any player $i \in \mathsf{Agt}$, $A[i]$ is the action played by player i.

An example of stochastic arena[1] is represented in Figure 3.1. We provide several graphical conventions:

- Non-terminal states are drawn with non-empty circles filled with their state label.
- Rectangle states are terminal.
- If $\mathsf{Tab}(s_1, A)$ is a distribution containing a unique state s_2, we write an arrow from s_1 to s_2 labelled with the word A.
- If several action profiles link a state s_1 to another state s_2, we can draw a single arrow labelled with the language describing the corresponding set of action profiles.
- The symbol $*$ is a shortcut for the language of the whole alphabet Act.
- The action $\dagger$ is used whenever it is the only action allowed for a given player.
- If $\mathsf{Tab}(s_1, A)$ is a distribution, we write a first A-labelled arrow to a *stochastic node* then several arrows to each state s_2 in the distribution support, labelled with the corresponding probability in the distribution.

3.1 Strategies

During a play, players in Agt choose their next moves in Act *concurrently and independently* of each other. They can even sample the played actions from a chosen distribution.

[1] In fact, this arena can model a system of three agents that try to emit at the same time, inducing interferences, but where agent 1 has higher priority (or more transmission power). Probability of a collision when agents 2 and 3 are emitting simultaneously is denoted by p. The terminal states correspond to situations where at least player 2 or 3 have successfully transmitted their own message.

In order to make their choice, players must be aware of the current location of the game since the set of allowed actions may differ for each state. In the general setting, players are even given the complete sequence of visited states from the beginning of the play. Such sequence $h \in \mathsf{States}^+$ will be called an *history*, and the current state will be denoted by $last(h) = h[|h|]$. The choices operated by each agent are represented by strategies, that we define below.

Definition 3.2. A *mixed strategy* for player $i \in \mathsf{Agt}$ is a mapping $\sigma_i \colon \mathsf{States}^+ \to Dist(\mathsf{Act})$, with the requirement that for all $h \in \mathsf{States}^+$, $\overline{\sigma_i(h)} \subseteq \mathsf{Allow}_i(last(h))$.
We denote by $\mathbb{S}_i^{\mathcal{A}}$ the class of (general, mixed) strategies for player i in arena $\mathcal{A}$. We will usually simply write $\mathbb{S}_i$ when $\mathcal{A}$ is clear in the context.

We consider several subclasses of strategies, defined below:

Definition 3.3. Let $\sigma_i \in \mathbb{S}_i$ and $k \in \mathbb{N}$, we define $\mathbb{M}(k)_i$, $\mathbb{M}_i$, $\mathbb{F}_i$ as follows:

- $\sigma_i \in \mathbb{M}(k)_i$ if the strategy requires a *memory of size* k, that is to say, if there exists a deterministic finite-state automaton with
  - state-space $\mathfrak{M}$ of cardinality $k+1$;
  - transition function[2] $\delta : \mathfrak{M} \times \mathsf{States} \to \mathfrak{M}$, with initial state $m_0 \in \mathfrak{M}$;
  - for all $hq, h'q \in \mathsf{States}^+$, if $\delta^*(m_0, h) = \delta^*(m_0, h')$, then $\sigma_i(hq) = \sigma_i(h'q)$.

  Intuitively, such a strategy stores $m = \delta^*(m_0, h)$ the current state in the automaton, seen as a memory information, that can be updated iteratively. The decision of σ_i from history h and memory m is expressed as a function of $(last(h), m)$.

- $\sigma_i \in \mathbb{M}_i = \mathbb{M}(0)_i$, if σ_i is a *stationary* (or *memoryless*) strategy, that is to say $\forall h \in \mathsf{States}^+ \ \sigma_i(h) = \sigma_i(last(h))$.

- $\sigma_i \in \mathbb{F}_i$ if σ_i requires *finite memory*, that is to say $\mathbb{F}_i = \bigcup_{k' \geq 0} \mathbb{M}(k')_i$.

- We also define the class of *pure strategies* S_i in which all probability distributions are Dirac functions, that is, strategies are deterministic:

  $$\sigma_i \in S_i \Leftrightarrow \forall h \in \mathsf{States}^+ \ \sharp \left[\overline{\sigma_i(h)}\right] = 1$$

  Each previously listed class can also be restricted to its pure counterpart: $M(k)_i = \mathbb{M}(k)_i \cap S_i$, $M_i = \mathbb{M}_i \cap S_i$, $F_i = \mathbb{F}_i \cap S_i$.

A strategy profile is a tuple $\sigma = (\sigma_i)_{i \in \mathsf{Agt}}$, in which σ_i is a strategy for player i. Following the definitions introduced above, we consider the full class $\mathbb{S}^{\mathcal{A}}$, or simply $\mathbb{S}$, of mixed strategy profiles. More generally, given a strategy class C_i for each player $i \in \mathsf{Agt}$, we will write C the corresponding class of strategy profiles.

For $a \in \mathsf{Act}$, $i \in \mathsf{Agt}$ and $h \in \mathsf{States}^+$, we write $\sigma_i(a \mid h)$ for the quantity $\sigma_i(h)(a)$. We also identify a with the Dirac distribution $\mathbb{1}_{\{a\}}$, allowing us to write $\sigma_i(h) = a$ whenever $\sigma_i(a \mid h) = 1$.

[2] δ and m_0 induce a general transition function δ^* defined for any $m \in \mathfrak{M}$ by $\delta^*(\varepsilon, m) = m_0$ and for any $hq \in \mathsf{States}^+$, $\delta^*(m, hq) = \delta(\delta^*(m, h), q)$

A strategy profile σ naturally induces a distribution over $\mathsf{Act}^{\mathsf{Agt}}$ for any history h by

$$\forall A = (a_i)_{i \in \mathsf{Agt}} \in \mathsf{Act}^{\mathsf{Agt}},\ \sigma(A \mid h) = \prod_{i \in \mathsf{Agt}} \sigma_i(a_i \mid h)$$

For an action profile $A \in \mathsf{Act}^{\mathsf{Agt}}$, $\sigma(A \mid h)$ and $\sigma(h) = A$ are defined similarly to the actions and strategies for a specific player.

Remark 3.4. *A strategy profile σ for a game $\mathcal{G}$ can be seen as a strategy σ_C for a unique player C in a game $\mathcal{G}_{\mathsf{Agt}}$ where all agents are seen as only one agent C, a* coalition*, with set of allowed actions from state s, $\mathsf{Allow}_C(s) = \prod_{i \in \mathsf{Agt}} \mathsf{Allow}_i(s)$. Notice however the converse is not true: a strategy for player c may not be a strategy profile in the original game as the unique player may introduce dependencies between concurrently played actions. For example, from a state s, the coalition strategy $\sigma_C(s) = \frac{1}{2}\mathbb{1}_{\{ab,ba\}}$ cannot be decomposed as a strategy profile for two players with two allowed actions a and b.*

3.2 Visibility of actions

Note that strategies, as defined above, can only observe the sequence of visited states along the history, but they may not depend on the exact distributions chosen by the players along the history, nor on the actual sequence of actions played by the players. Usually, actions are made visible in the game models considered in the literature—see for instance [Umm08] and [BBMU11, section 6] or [CD14] for discussions—and the results presented here may differ.

One motivation for action-visible models comes from the fact that invisibility of actions is a particular case of partial information, which is a concept usually inducing undecidable models [BBG08]. However, we allow this little source of unobservability in our model, as it makes it more general, and recent results like [Bre12] tend to prove that such a lack of observation is not crucial.

Moreover, we can still consider subclasses of arenas for which the observation of history determines which action has been played. Such arenas are defined below:

Definition 3.5. An arena $\mathcal{A}$ is *action-visible* if, for every non-final states $s_1, s_2 \in \mathsf{States} \backslash \mathtt{F}$, there exists at most one action profile $A \in \mathsf{Act}^{\mathsf{Agt}}$ such that $\mathsf{Tab}(s_1, A)(s_2) > 0$.

Remark 3.6. *We can see that any arena $\mathcal{A}$ can be transformed into $\mathcal{A}^v$ which is action-visible. Indeed, we can encode the played actions inside the current state, by setting $\mathsf{States}^v = \mathsf{States} \times \mathsf{Act}^{\mathsf{Agt}}$ and $\mathsf{Tab}^v((s_1, A_1), A)((s_2, A_2)) = \mathsf{Tab}(s_1, A)(s_2) \cdot \mathbb{1}_{A=A_2}$. A strategy $\sigma \in \mathbb{S}^{\mathcal{A}}$ can be seen as a strategy $\sigma^v \in \mathbb{S}^{\mathcal{A}^v}$ that ignores the second coordinate of each state that appears in any history.*

When considering solution concepts like Nash Equilibria, or algorithms based on concurrent games, our framework is more general as it does not require action-visibility. Hence, existential and algorithmic results from Chapter 5 are stronger in this framework.

However, when considering hardness results, as the undecidability proof of Chapter 4, exploiting the lack of visibility can be seen as a weakness, as imperfect information in stochastic games usually leads to undecidability. Although similar undecidability results have been developed for example by [UW09] without this hypothesis, we argue that this extra hypothesis allows us to narrow the gap between decidability and undecidability problems, since the undecidability result only leaves open the two-player case, which improves our understanding of concurrent games.

3.3 Semantics

For a given strategy profile σ in arena $\mathcal{A}$, we can define a distribution over histories, that we will write $\mathbb{P}^\sigma$. This section is devoted to its definition and its related properties.

Definition 3.7. Let $\sigma \in \mathbb{S}$ a strategy profile for $\mathcal{A}$. We define $\mathbb{P}^\sigma(h)$ by induction on $h \in \mathsf{States}^+$.

- For any $s \in \mathsf{States}$, $\mathbb{P}^\sigma(s) = 1$
- For any $h \in \mathsf{States}^+$ and $s \in \mathsf{States}$,

$$\mathbb{P}^\sigma(h \cdot s) = \begin{cases} \mathbb{P}^\sigma(h) \cdot \sum_{A \in \mathsf{Act}^{\mathsf{Agt}}} \sigma(A \mid h) \cdot \mathsf{Tab}(\mathit{last}(h), A)(s) & \text{if } \mathit{last}(h) \notin \mathtt{F}, \\ 0 & \text{otherwise.} \end{cases}$$

If $\mathit{last}(h) \notin \mathtt{F}$, we check that

$$\sum_{s \in \mathsf{States}} \mathbb{P}^\sigma(h \cdot s) = \sum_{s} \mathbb{P}^\sigma(h) \cdot \sum_{A \in \mathsf{Act}^{\mathsf{Agt}}} \sigma(A \mid h) \cdot \mathsf{Tab}(\mathit{last}(h), A))(s) = \mathbb{P}^\sigma(h) \cdot \sum_{A} \sigma(A \mid h) = \mathbb{P}^\sigma(h)$$

Therefore, we can apply Carathéodory's criterion stated in Theorem 2.1 to extend $\mathbb{P}^\sigma$ as a measure to any finite or infinite sequence on States. Notice that maximal sequences belong to the language $(\mathsf{States} \backslash \mathtt{F})^* \cdot \mathtt{F} \uplus \mathsf{States}^\omega$.

Intuitively, for any history or infinite run $h \in \mathsf{States}^+ \uplus \mathsf{States}^\omega$, $\mathbb{P}^\sigma(h)$ will denote the probability of generating h with strategy (profile) σ, when starting from state $h[1]$.

Definition 3.8 (Conditional probability)**.** Let $h \in \mathsf{States}^+$ be a history, we define the conditional probability measure $\mathbb{P}^\sigma(- \mid h)$ for any finite history or infinite run $h' \in \mathsf{States}^+ \uplus \mathsf{States}^\omega$, by

- $\mathbb{P}^\sigma(h' \mid h) = \frac{\mathbb{P}^\sigma(h')}{\mathbb{P}^\sigma(h)}$ if $\mathbb{P}^\sigma(h) > 0$ and $h \sqsubseteq h'$;
- $\mathbb{P}^\sigma(h' \mid h) = 0$ otherwise.

We can check that such a conditional measure is indeed always a probability measure over $(\mathsf{States} \backslash \mathtt{F})^* \cdot \mathtt{F} \uplus \mathsf{States}^\omega$, as the choice of h fixes the initial state.

Remember that when considering a finite history $h \in \mathsf{States}^+$, we are in fact considering the probability of the whole cylinder $h \cdot ((\mathsf{States} \backslash \mathtt{F})^* \cdot \mathtt{F} \uplus \mathsf{States}^\omega)$. This property also applies to conditional properties.

Another interpretation of $\mathbb{P}^\sigma(h' \mid h)$ is to say that arena and strategies are "initialized" with history h, and we are considering the probability that the resulting stochastic process generates a run that starts with h'.

3.4 Outcome of a game

An arena as previously described, only gives information about how the agents can interact, to make the game progress. However, at this point, no information was given about the incentive of each player. Several choices can be made at this point to define such a concept:

- When the arena involves at most two players, we can define a set Ω of states that have to be reached. First player (if he exists) will try to enforce reaching such a state, whereas second player (if he exists) will try to avoid it.
- In the same setting, we can define a set $W \subseteq \mathsf{States}^\omega \uplus \mathsf{States}^* \cdot \mathtt{F}$ of winning runs for first player. In the previous setting, this approach boils down to $W = \mathsf{States}^* \cdot \Omega \cdot (\mathsf{States}^\omega \uplus \mathsf{States}^* \cdot \mathtt{F})$.
- More generally, we can provide a total ordering relation $\preceq$ over $\mathsf{States}^\omega \uplus \mathsf{States}^* \cdot \mathtt{F}$, which will be called a *preference relation* for the first player. With the previous approach, this is equivalent to define $\forall r, r'\ r \preceq r' \Leftrightarrow r' \in W$.
- When the game involves more than two players, the roles of the players is not necessarily antagonistic, so we can give to each player a set of target states, a set of runs, or a preference relation, to define their objectives.

In a deterministic setting, we have listed above any possible objective: when considering a deterministic arena $\mathcal{A}$, a pure strategy profile σ and an initial state s_0, we can define precisely the *outcome* of the game as the unique finite or infinite maximal path starting from s_0, denoted $\mathsf{Out}_{s_0,\mathcal{A}}(\sigma)$. The last preference relation is in some sense the most general one, as it allows to compare any run, seen as an outcome, for each player.

However, this approach is not suited anymore when introducing probabilities in our model, as the outcome cannot be defined as runs in the arena. More precisely, the outcome of a game starting from s_0 in arena $\mathcal{A}$ with arbitrary distributions consists in the resulting probability measure, that is to say $\mathsf{Out}_{s_0,\mathcal{A}}(\sigma) = \mathbb{P}^\sigma(- \mid s_0)$. As a consequence, comparing outcomes in a stochastic setting boils down to defining preference relations over probability measures of paths.

Instead of providing such a general and complex definition of objectives, we limit ourselves to the first and second settings where objectives are given as measurable functions from the set of possible runs to the real numbers.

Definition 3.9. $\mathcal{G} = (\mathcal{A}, s, (\Phi_i)_{i \in \mathsf{Agt}})$ is given by an arena $\mathcal{A}$, an initial state s, and for every player $i \in \mathsf{Agt}$, a real-valued function $\Phi_i : \mathsf{States}^* \uplus \mathsf{States}^\omega \to \mathbb{R}$, which is measurable over the set of runs of $\mathcal{A}$.

As explained before, we will not define a general preference relation over outcomes, which are general distributions over runs, but give a more concrete characterization based on reward functions. For any history h, and any function measurable over the set of runs of $\mathcal{A}$, we denote by $\mathbb{E}^\sigma(\phi \mid h)$ the expected value of the reward function ϕ induced by the probability measure $\mathbb{P}^\sigma(- \mid h)$.

As for strategy profiles, we write by extension $\mathbb{E}^\sigma(\Phi \mid h)$ for the tuple $(\mathbb{E}^\sigma(\Phi_i \mid h))_{i \in \mathsf{Agt}}$ when $(\Phi_i)_{i \in \mathsf{Agt}}$ is a family of measurable functions, one for each player.

Finally we say that an history h is *activated*, or *enabled* by a strategy profile whenever it is visited with positive probability under that profile.

Some measurable reward functions

In this thesis, we will be particularly interested in simple objective functions that consists in assigning a reward to a player only if she manages to reach a final state, or complementary, if she manages to visit as few different states as possible.

Definition 3.10. An objective function ϕ is:

- a *reachability objective* if for any run r, $\phi(r) = \max\{\phi(r[k]) \mid 1 \leq k < |r| + 1\}$;
- a *safety objective* if for any run r, $\phi(r) = \min\{\phi(r[k]) \mid 1 \leq k < |r| + 1\}$;
- a *terminal-reachability objective* if for any run r,

$$\phi(r) = \begin{cases} \phi(last(r)) & \text{if } r \in \mathsf{States}^* \cdot \mathtt{F}, \\ 0 & \text{otherwise.} \end{cases}$$

We can easily check that terminal-reachability is a particular case of reachability objective where the payoff profile is determined by the final state reached in the game, and that each non-final state has reward 0.

The study of terminal reachability objectives is crucial for the understanding of general reachability objectives, so our study will mainly focus on this particular class of objectives. When considering games where all players have terminal-reachability objectives, the payoff functions are entirely characterized by the vectors $\Phi(f) \in \mathbb{R}^{\mathsf{Agt}}$ for each $f \in \mathtt{F}$. Thus, a final state f will usually be directly denoted and graphically represented by its payoff profile $\Phi(f)$.

Definition 3.11. A reward objective ϕ is *non-negative* whenever each ϕ is a non-negative function. It is *qualitative* whenever each ϕ takes values in $\{0, 1\}$, in the opposite case, it is *quantitative*.

A reachability or safety objective ϕ, can be represented by a subset of states $G \subseteq \mathsf{States}$ representing the *goal of the player*, that is to say the states that she has to reach or, respectively, to not leave.

Qualitative reachability and safety objectives for Nash equilibria have been studied intensively in the general case of ω-regular objectives in [Bre12]. In the quantitative case, terminal reachability objectives can be generalized to limit-average rewards, as in [UW11a], where reward values are attached to each state and where the reward function consists in computing the upper or lower limit of the mean value of visited states. At the expense of a exponential blow-up[3], average-reward objectives can even encode general quantitative reachability and safety objectives.

3.5 Two-player zero-sum games

In this section, we develop known results in the $2 + 1/2$-player case. More precisely, we revisit results of the zero-sum fragment. A 2-player stochastic game $\mathcal{G}$ with payoff function $\Phi = (\Phi_1, \Phi_2)$ is called *zero-sum* if $\Phi_1 + \Phi_2 \equiv 0$, that is to say $\forall r \in \mathsf{States}^\omega \uplus \mathsf{States}^+ \ \Phi_1(r) + \Phi_2(r) = 0$.

In such games, the roles of players 1 and 2 are totally antagonistic. When considering qualitative objectives, player 1 usually has a qualitative terminal reachability objective, while the second player goal has to avoid the same reachability set of states, that is to say ensure a safety condition.

[3] by encoding inside each state the set of already visited states

Borel determinacy

In his seminal work, [Mar75] studied Gale-Stewart games [GS53] and established their determinacy in a very general setting. Such games are played by two players, who pick in turn one an element at a time from a possibly infinite set A, which generates an infinite word in A^ω. Winning condition for player 1 is given as a subset $\Omega \subseteq A^\omega$ of infinite runs. Whenever Ω is a Borel set, the game is shown to be determined, that is to say there exists a deterministic winning strategy either for player 1, or for player 2.

This result can be rephrased in our framework to state that any deterministic, 2-player, qualitative zero-sum, turn-based game with qualitative measurable payoff function ϕ is *determined*, that is to say:

- either there exists a *winning strategy for player* 1, $\sigma_1 \in S_1$, such that for all $\sigma_2 \in S_2$, $\mathbb{E}^{(\sigma_1,\sigma_2)}(\phi \mid s_0) = 1$,
- or there exists a *winning strategy for player* 1, $\sigma_2 \in S_2$, such that for all $\sigma_1 \in S_1$, $\mathbb{E}^{(\sigma_1,\sigma_2)}(\phi \mid s_0) = 0$.

Notice this notion does not involve any probabilistic property, as the game and its strategies are assumed deterministic: all possible outcomes of such a game are therefore single runs. Notice however that a winning strategy is still winning against a randomized adversary, which allows us to generalize the result to randomized strategies.

However, the result is not valid anymore when considering games with stochastic transition functions, as both players may win with positive probability. A similar phenomenon occurs when allowing concurrent games. Such models form first examples of imperfect information as each player is not aware of the action played by the other player at a given step. Moreover, objectives are not necessary qualitative.

Nonetheless, some another notion of determinacy may be introduced, that captures the quantitative features and imperfect information of our model.

Value determinacy

In the quantitative setting, an interesting concept to introduce is the maximal expected value that a player can reach no matter what her opponent is doing. A dual approach consists in looking at the minimal expected payoff this opponent can achieve, when strategy of player 1 is not known in advance.

Note that "maximal" and "minimal" are here to be interpreted in an asymptotic manner, as we are not guaranteed that these optimal values can be reached.

Definition 3.12. Let $\mathcal{G}$ be a stochastic two-player game, such that the reward function Φ_1 of player 1, is measurable. We define the *upper value* and the *lower value* from state s as the quantities:

$$\overline{\nu}_1(s) = \inf_{\sigma_2 \in \mathbb{S}_2} \sup_{\sigma_1 \in \mathbb{S}_1} \mathbb{E}^{(\sigma_1,\sigma_2)}(\Phi_1 \mid s) \qquad \underline{\nu}_1(s) = \sup_{\sigma_1 \in \mathbb{S}_1} \inf_{\sigma_2 \in \mathbb{S}_2} \mathbb{E}^{(\sigma_1,\sigma_2)}(\Phi_1 \mid s)$$

In general $\overline{\nu}_1(s) \geq \underline{\nu}_1(s)$. The converse inequality is the so-called *value determinacy of a game*. This unique value is then called *value from state* s, denoted by $\nu_1(s)$.

Value determinacy is achieved for a large variety of zero-sum games including Blackwell games, that is to say concurrent games with a Borel measurable payoff function [Mar98].

In [Mar98], the author suggests that the corresponding strategies somehow require as much memory as needed to compute the payoff function: the more complex the reward, the more complex the strategy.

When the reward function is a terminal reachability objective, results from [Sec97] and [KS81] allow to consider only memoryless strategies to asymptotically achieve the game value. This means there exist pairs of ε-optimal strategies that are memoryless, for both players and for any $\varepsilon > 0$:

Theorem 3.13. *Let ϕ be a terminal reachability payoff function of an action-visible zero-sum game $\mathcal{G}$. Then for any state s, $\overline{\nu}_1(s) = \underline{\nu}_1(s)$, denoted from now on $\nu_1(s)$, and there exists a family of memoryless strategies $(\sigma_1^\varepsilon)_{\varepsilon>0}$ (resp. $(\sigma_2^\varepsilon)_{\varepsilon>0}$) such that for any state s:*

$$\forall \varepsilon > 0 \ \inf_{\sigma_2 \in \mathbb{S}_2} \mathbb{E}^{(\sigma_1^\varepsilon, \sigma_2)}(\phi \mid s) \geq \nu_1(s) - \varepsilon$$

and respectively,

$$\forall \varepsilon > 0 \ \sup_{\sigma_1 \in \mathbb{S}_1} \mathbb{E}^{(\sigma_1, \sigma_2^\varepsilon)}(\phi \mid s) \leq \nu_1(s) + \varepsilon$$

Action invisibility

The previous theorem is stated in the particular case of action-visible games. Although the proof can be adapted to the case of invisible actions, we give here a simpler argument in our setting, based on the fact that memoryless ε-optimal strategies are not aware of played actions.

Let us consider a stochastic game $\mathcal{G}$ and a terminal reachability payoff function ϕ. We define $\mathcal{A}'$ the corresponding action-visible arena defined from $\mathcal{A}$ as in Remark 3.6. For a run r of $\mathcal{A}'$, we define $\phi'(r) = \phi(p(r[1]) \cdot p(r[2]) \cdots)$ where p is the projection on the first coordinate[4].

From now on, we reason on a game $\mathcal{G}'$ with arena $\mathcal{A}'$ and objective ϕ', so $\mathbb{S} = \mathbb{S}^{\mathcal{G}'}$ denotes the class of strategies that see actions in the history. We define, the subclass of strategy profiles $\mathbb{S}^u \subseteq \mathbb{S}$, that do not see actions, that is to say $\sigma \in \mathbb{S}^u$ if, and only if,

$$\begin{aligned}
&\forall h \in \mathsf{States}^+, \ \forall \alpha, \alpha' \in (\mathsf{Act}^{\mathsf{Agt}})^{|h|}, \\
&\sigma((h[1], \alpha[1]) \cdot (h[2], \alpha[2]) \cdots (h[|h|], \alpha[|h|])) = \sigma((h[1], \alpha'[1]) \cdot (h[2], \alpha'[2]) \cdots (h[|h|], \alpha'[|h|]))
\end{aligned}$$

We define the $\overline{\nu}_1^u$ and $\underline{\nu}_1^u$ by the following mapping over any $s \in \mathsf{States}$ and $A \in \mathsf{Act}^{\mathsf{Agt}}$:

$$\overline{\nu}_1^u((s, A)) = \inf_{\sigma_2 \in \mathbb{S}_2^u} \sup_{\sigma_1 \in \mathbb{S}_1^u} \mathbb{E}^{(\sigma_1, \sigma_2)}(\phi' \mid (s, A)) \qquad \underline{\nu}_1^u((s, A)) = \sup_{\sigma_1 \in \mathbb{S}_1^u} \inf_{\sigma_2 \in \mathbb{S}_2^u} \mathbb{E}^{(\sigma_1, \sigma_2)}(\phi' \mid (s, A))$$

We can see that these quantities does not depend on A, and correspond in fact to the values in the original game $\mathcal{G}$. Indeed, the reward function Φ' and transition table do not observe any action.

Also, we still have $\overline{\nu}^u((s, A)) \geq \underline{\nu}^u((s, A))$.

Moreover, $\mathbb{S}^u \subseteq \mathbb{S}$ so by introducing a pair of ε-optimal memoryless strategies $(\sigma_1^\varepsilon, \sigma_2^\varepsilon)$ for each $\varepsilon > 0$, from Theorem 3.13, we have:

$$\forall \varepsilon > 0 \ \inf_{\sigma_2 \in \mathbb{S}_2^u} \mathbb{E}^{(\sigma_1^\varepsilon, \sigma_2)}(\phi \mid (s, A)) \geq \inf_{\sigma_2 \in \mathbb{S}_2} \mathbb{E}^{(\sigma_1^\varepsilon, \sigma_2)}(\phi \mid (s, A)) \geq \nu_1((s, A)) - \varepsilon$$

[4] That is to say $\forall s \in \mathsf{States}, \forall A \in \mathsf{Act}^{\mathsf{Agt}}, \ p((s, A)) = s$.

and respectively,

$$\forall \varepsilon > 0 \ \sup_{\sigma_1 \in \mathbb{S}_1^u} \mathbb{E}^{(\sigma_1, \sigma_2^\varepsilon)}(\phi \mid (s, A)) \leq \sup_{\sigma_1 \in \mathbb{S}_1} \mathbb{E}^{(\sigma_1, \sigma_2^\varepsilon)}(\phi \mid (s, A)) \leq \nu_1((s, A)) + \varepsilon$$

Moreover, these ε-optimal strategies are *uniform*, as they try to optimize the expected payoff from any state. Since this payoff does not depend on the played action, we argue that we can assume, without loss of generality, that $\sigma_1^\varepsilon(s, A)$ and $\sigma_2^\varepsilon(s, A)$ do not depend on A for any state s, that is to say $\sigma^\varepsilon \in \mathbb{M}^u = \mathbb{S}^u \cap \mathbb{M}$.

As a consequence, when ε goes to 0, we get the inequality $\underline{\nu}_1^u((s, A)) \geq \nu_1((s, A)) \geq \overline{\nu}_1^u((s, A))$ hence $\underline{\nu}_1^u((s, A)) = \nu_1((s, A)) = \overline{\nu}_1^u((s, A))$.

We conclude that action-visibility hypothesis can be omitted from Theorem 3.13.

Values may not be achieved

We emphasize here on the fact that the value of a game may not be achieved exactly, and only ε-optimal values may exists. Consider the 2-player zero-sum game $\mathcal{H}$ (*hide-or-run*), depicted in Figure 1.2 on page 3 in the introduction.

First of all, we show that $\nu(s_0) = (1, -1)$, by considering a fixed strategy for player 1 who is hiding at each round with probability $\varepsilon > 0$. As a matter of fact, player 2 can only shoot, which generates average payoff $(1 - 2\varepsilon, 2\varepsilon - 1)$, or wait, which yields payoff $(1, -1)$ or iterate the game again. In both cases, the game almost-surely terminates, with a payoff for player 1 larger than $1 - 2\varepsilon$. Hence, $\nu_1(s_0) = \overline{\nu}_1(s_0) \geq 1 - 2\varepsilon$ for any $\varepsilon > 0$.

However, no optimal strategy σ_1 exists for player 1, as it would require to avoid state $(-1, 1)$ almost-surely for any strategy of player 2. This would imply that player 1 plays deterministically in each history s_0^n. Moreover, such a strategy would require reaching $(1, -1)$ almost-surely, that is to say, player 1 should not play hide action $\hbar$ forever. However, when playing action run r, player 2 has positive chances to shoot her (for some strategy) so reaching $(1, -1)$ almost-surely no matter the strategy of player 2 is impossible.

Nevertheless, we can identify some game classes that allow existence of optimal strategies for both players. For example, it is well-known that an optimal policy exists for Markov Decision Processes with reachability objectives [Put94]. In our terminology, a *policy* is a strategy, and a Markov Decision Process is a $1 + 1/2$-player game. In a multi-player setting, such results have been extended to the case of turn-based zero-sum games, with any limit average reward objective. For example, [LL69] shows that game values are achieved by pure memoryless strategies, for both players. Again, visibility of actions makes little difference to the result, and can be omitted.

As a consequence, deciding whether the value of a player in such a game is larger than a given threshold is in NP $\cap$ coNP (see [Con92] for the qualitative reachability case, or [ZP96] for the study in the deterministic average payoff case), by guessing pure memoryless strategies achieving or refuting a value larger than the threshold. This has to be compared with parity games: where deciding which of the two players has a winning strategy is in NP $\cap$ coNP, as well as in UP $\cap$ coUP [Jur98].

Notice also another consequence of the existence of pure memoryless strategies in the turn-based case: the values become rational, whenever the game involves rational numbers for discounted factors, probability transitions and rewards. On the contrary, concurrent games may have irrational values, even for terminal qualitative reachability objectives with rational probabilities [AM04, page 382].

Qualitative analysis

Let us consider qualitative terminal reachability objectives. Even when $\nu_1(s_0) = 1$, that is to say when player 1 can yield her maximal payoff, she can win in several manners:

- Limit-surely, if there exists a family of strategies ensuring payoff $1 - \varepsilon$ for any $\varepsilon > 0$,
- Almost-surely, if there exists a strategy ensuring an average payoff of 1,
- Surely, if furthermore this strategy ensures the game terminates in finite time.

The qualitative analysis focus on the existence of strategies for such winning modes, while the goal of the opposite player is only to *spoil* the winning condition. Notice that in the limit-sure case, a unique *spoiling strategy* should be resilient against a whole family of strategies that try to achieve a reachability objective with probability arbitrarily close to one, hence this context is slightly different from the existence of ε-optimal strategies for both players, as presented before.

In [AHK07], the authors showed determinacy for each three winning conditions. When a winning strategy exists for player 1, no memory is required for winning (although randomization may be required). When a winning strategy for player 2 exists, that is to say a *spoiling* strategy, counting[5] memory is required only for almost-sure reachability, whereas memoryless strategies suffice to spoil sure and limit-sure winning modes. This last result should be kept in mind as it relates to the value of a game, which in turn connects to Nash equilibrium concepts.

3.6 Nash Equilibria

We shift now to the study of games with more than two players. By analogy with the previous study, for any game with reachability and safety objectives, we denote by $\nu_i(s)$ the value from state s for player i, in a new game played by i against the coalition composed of all agents except i.

Definition 3.14. Let $\mathcal{G}$ be a stochastic $n + 1/2$-player game, with payoff function ϕ, s a state, and $i \in \mathsf{Agt}$ one of the players. Assume that Φ_i is a terminal reachability or safety reward function, then we define $\nu_i(s)$ by:

$$\nu_i(s) = \inf_{\sigma \in \mathbb{S}} \sup_{\sigma'_i \in \mathbb{S}_i} \mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s) = \sup_{\sigma'_i \in \mathbb{S}_i} \inf_{\sigma \in \mathbb{S}} \mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s)$$

Where $\sigma[i/\sigma'_i]$ denotes the strategy profile where strategy for player i has been changed to σ'_i.

Furthermore, $\nu(s)$ can be seen as a vector of $\mathbb{R}^{\mathsf{Agt}}$ representing for each player, the supremum of her achievable payoffs, no matter what the other players do.

As opposed to the two-player case, it is not clear how actions from one player can be opposed to the actions of the other player, so the zero-sum condition is usually omitted, that is to say, there may exist two runs r_1, r_2 such that $\sum_i \Phi_i(r_1) \neq \sum_i \Phi_i(r_2)$.

[5] A strategy requires counting memory whenever it depends on the current state and the length of the history, this is a particular case of infinite memory.

3.6.1 Definition

To address this issue, John Nash introduced a new concept [Nas50], where emphasis is put on each individual objective, and strategy for each player should be *stable* with respect to the strategies of the others.

Definition 3.15. Let $\mathcal{G} = (\mathcal{A}, s, \Phi)$ be a stochastic game. A strategy profile σ forms a *Nash equilibrium* for $\mathcal{G}$, when no player has a profitable deviation; in other terms, for all $i \in \mathsf{Agt}$ and for all $\sigma'_i \in \mathbb{S}_i$, it holds

$$\mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s) \leq \mathbb{E}^{\sigma}(\Phi_i \mid s)$$

Equivalently, the average payoff of an equilibrium σ, denoted by the vector $\mathbb{E}^{\sigma}(\Phi \mid s)$, satisfies, for all player $i \in \mathsf{Agt}$:

$$\mathbb{E}^{\sigma}(\Phi_i \mid s) = \sup_{\sigma'_i \in \mathbb{S}_i} \mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s)$$

This last form immediately implies that the average payoff of an equilibrium is component-wise larger than the value of the game from the same state.

Deterministic deviations

When defining the Nash equilibrium concept, we used the term *deviation* to denote a single strategy $\sigma'_i \in \mathbb{S}_i$ for player i that may increase her payoff. Such deviation σ'_i is called *deterministic*, or *pure*, whenever $\sigma'_i \in S_i$. An important remark when analyzing Nash equilibria is that randomization in our context is not required for deviations: if a profitable deviation has to be found, a deterministic one also exists. Randomizing its strategy is necessary only to ensure stability, that is to say prevent the other players from deviating. This observation is summarized in the following proposition:

Proposition 3.16. *Let $\mathcal{G} = (\mathcal{A}, s, \Phi)$ be a stochastic concurrent game with safety and reachability objectives and σ be a strategy profile. Then σ is a Nash equilibrium if, and only if, for all $i \in \mathsf{Agt}$ and all* deterministic *deviation $\sigma'_i \in S_i$, it holds $\mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s) \leq \mathbb{E}^{\sigma}(\Phi_i \mid s)$.*

Proof. This result is similar to [UW11c, proposition 3.1] for turn-based games with qualitative Borel objectives (the payoff is 1 if the run belongs to the designed objective and 0 otherwise). Intuitively, for a given $\sigma \in \mathbb{S}$, the composed game $\mathcal{G}\langle\sigma\rangle_{-i}$ where the actions of all players but i are fixed, is a one-player stochastic game (with possibly countably many states), that can be seen as a Markov Decision Process. Our safety and reachability payoffs can be interpreted as a total-reward payoff that can be taken only once (by encoding in each state the set of already visited states). Under our hypothesis, this Markov Decision Processes are known to have an ε-optimal pure memoryless[6] strategy, for any $\varepsilon > 0$, see for example [Put94, Theorems. 7.2.7 and 7.3.7]. As a consequence, if σ is not a Nash equilibrium, there exists a strategy improving the payoff of some player i by some positive quantity $\varepsilon > 0$, and a deterministic one that improves her payoff by $\varepsilon/2 > 0$. □

Remark 3.17. *It may not be possible, in general, to achieve the best profitable value in a deviation, and randomization or memory may be required. First remark that if the initial strategy profile σ has finite memory, the resulting Markov Decision Process has also a finite state space*

[6]The strategy is memoryless in the composed game, hence, may require memory in the original game.

and an optimal deterministic and memoryless strategy exists in this case. However, the result does not hold in the case of infinite memory: consider for example the game composed of a unique non-terminal state s, with $\mathsf{Tab}(s, a*) = s$, $\mathsf{Tab}(s, ba) = (1, 0)$, $\mathsf{Tab}(s, bb) = (0, 0)$, and strategy profile $\forall n \geq 1,\ \sigma_1(a \mid s^n) = 1/2, \sigma_2(b \mid s^n) = 1/2^{n-1}$. If player 1 deviates by deciding to always play b with probability $p > 0$, then he yields payoff $\sum_{n\geq 0} p(1-p)^n \cdot (1 - \frac{1}{2^n}) = \frac{1-p}{1+p}$. Thus, the optimal value that player 1 can yield by deviating is 1, although this value cannot be achieved by any deviation. Moreover, if $\sigma_1' \in M(n)_1$ is a deterministic deviation using memory of size n, then he plays his first b action after at most n steps, which yields a payoff of at most $1 - 1/2^n$. In particular, there is no profitable memoryless deterministic deviation in this case.

3.6.2 Sub-game characterization

We explore some further characterization of a Nash equilibrium in terms of its sub-game. In order to do so, we extend the notion of Nash equilibrium to any history, in the following way:

Definition 3.18. Let $\mathcal{G} = (\mathcal{A}, s, \Phi)$ be a stochastic game. A strategy profile σ forms a *Nash equilibrium* after a history $h \in \mathsf{States}^+$ when the following conditions are met:

- $h \in \mathsf{States}^+$ is enabled by σ from *first*(h);
- No player has a profitable deviation; in other terms, for all $i \in \mathsf{Agt}$ and for all $\sigma_i' \in \mathbb{S}_i$ such that h is enabled from $\sigma[i/\sigma_i']$, it holds

$$\mathbb{E}^{\sigma[i/\sigma_i']}(\Phi_i \mid h) \leq \mathbb{E}^{\sigma}(\Phi_i \mid h)$$

We then write that (σ, h) is a Nash equilibrium for $\mathcal{G}$.

This notion generalizes a classical Nash equilibrium σ in a game $(\mathcal{A}, s, \Phi)$ as a Nash equilibrium (σ, s). Intuitively, we can assume that strategy σ_i is not changed for any history $h' \notin h\mathsf{States}^+$, and that the game is initialized with an initial history h.

In his seminal work, Nash concluded that Nash equilibria always exists for one-shot games, that is to say, games that terminates after one iteration. The result can be extended to games played on finite trees, as we can prove the result by induction. We have indeed the following results:

Lemma 3.19. *Let $\mathcal{G}$ be a stochastic concurrent game, and (σ, h) be a Nash equilibrium. If (σ, h) enables h', then, (σ, h') is a Nash equilibrium.*

The proof is immediate by expressing conditional probabilities from history h.

A converse property is harder to define in the general case because we cannot assume that any history h' that extends h is enabled by σ. More precisely, whenever an history h' is not enabled by a strategy σ but can be enabled in a new strategy profile $\sigma[i/\sigma_i']$ by the deviation of some player i, the other players but i have no incentive to improve their payoff, so (σ, h') is not necessarily a Nash equilibrium.

When the game is action-visible, the other players have a perfect knowledge of the player that started a deviation and can retaliate to ensure as much stability of the strategy profile as possible. We conclude on the following property:

Proposition 3.20. *Let $\mathcal{G}$ be a stochastic action-visible concurrent game, σ be a strategy profile and h an history enabled by σ. We construct the following one-shot game $\mathcal{G}^h = (\mathcal{A}^h, last(h), \Phi^h)$ that starts from $last(h)$, with same actions, and same transition functions but where any state s is supposed to be final, with payoff $\Phi^h(hs)$ that equals:*

- *$\nu(s)$ whenever hs is not enabled by σ in the original game;*
- *$\mathbb{E}^\sigma(\Phi \mid hs)$ otherwise.*

Assume moreover that

- *for any disabled history hs, $\sigma(hs-)$ is an optimal strategy that achieves value $\nu_i(s)$ where i is the unique player who can perform transition from $last(h)$ to s,*
- *for any enabled history hs, (σ, hs) is a Nash equilibrium,*
- *σ is a Nash equilibrium of $\mathcal{G}^h$.*

Then, (σ, h) is a Nash equilibrium of $\mathcal{G}$.

Again, the property is proven immediately by expressing conditional expectations from history h and possible deviations for any player i.

In the case of action invisibility, the situation is less clear, as players may not be immediately aware of the player actually deviating, hence we cannot directly consider the value function previously defined. This suggested some new constructions like the suspect game defined in [Bre12], that encodes in a new 2-player game the set of possible deviating players. The first player wins if she manages to construct a Nash equilibrium while the other player tries to dismiss it by exhibiting a potential deviation, that can be done by one of the remaining players in set of suspects. However, this technique does not extend well when considering mixed strategies, as the resulting suspect game would contain a continuous set of allowed actions (distributions) from each state.

3.6.3 Subgame perfect equilibrium

An alternative and stronger notion consists in considering strategy profiles that yield a Nash equilibrium, when the game is initialized with any history, even if not enabled. With this approach, detecting deviations is pointless, as players always have to stick to the Nash equilibrium from any history. From a game theoretical point-of-view, players are more eager to pursue their own objective, instead of threatening each other to prevent deviations. Note however that such an equilibrium is less likely to occur, as it requires more stability than a Nash equilibrium.

Definition 3.21. Let $\mathcal{G} = (\mathcal{A}, s, \Phi)$ be a stochastic game. A strategy profile σ forms a *subgame perfect equilibrium* for $\mathcal{G}$, when for any history $hs' \in \mathsf{States}^+$, $(\sigma(h \cdot -), s')$ is a Nash equilibrium in $\mathcal{G}^{hs'} = (\mathcal{A}, s', \Phi(h \cdot -))$.
Equivalently, no player has a profitable deviation from hs': for all $i \in \mathsf{Agt}$, $h \in \mathsf{States}^+$ and $\sigma'_i \in \mathbb{S}_i$, it holds

$$\mathbb{E}^{\sigma[i/\sigma'_i](h\cdot -)}(\Phi_i(h \cdot -) \mid s') \leq \mathbb{E}^{\sigma(h\cdot -)}(\Phi_i(h \cdot -) \mid s')$$

Where for any function f, $f(h \cdot -)$ is the function that maps each h' to $f(h \cdot h')$.

For prefix independent objectives, like terminal-reachability, and whenever the considered history hs is enabled by σ, $(\sigma(h \cdot -), s')$ is a Nash equilibrium in $\mathcal{G}^{hs'}$ if, and only if, (σ, hs') is a Nash equilibrium in $\mathcal{G}$.

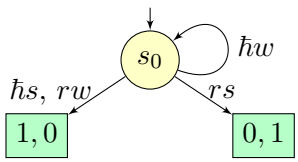


Figure 3.2 – A shifted version of game of Figure 1.2 on page 3.

3.6.4 Example of equilibria

In the following section, we recall some known facts about the existence status of Nash equilibria, for games with terminal reachability objectives.

In general, *several* Nash equilibria may coexist (see *e.g.* later Figure 4.4a on page 39), in finite, countable, or uncountable number. They may *require finite memory* as depicted in a turn-based example of [Umm10].

Moreover, there exist games that admit no Nash equilibria.

Lemma 3.22. *The game hide-or-run $\mathcal{H}$ of Figure 1.2 on page 3 has no Nash equilibria.*

Proof. Since $\mathcal{H}$ is zero-sum, any Nash equilibrium σ is also an optimal strategy, which has been proven to not exist for this game. □

Even if $\mathcal{H}$ is a terminal reachability game, it involves negative values as payoffs. A possible way to remove these negative values consists in shifting then rescaling the payoff profiles, which has no effect on the dynamics of the game, since expectation operator $\mathbb{E}$ is linear. Let us define $\phi' = \frac{\phi+1}{2}$, which assigns $(1, 0)$ (resp. $(0, 1)$) to any history ending in the leftmost (resp. rightmost) final state, but also assigns $(1, 1)$ to the infinite run s_0^ω. As a consequence, the new game admits no Nash equilibrium, but is not terminal-reachability anymore.

In order to have only non-negative terminal-reachability payoffs, we could remove this last reward on loop s_0^ω, which corresponds to the game $\mathcal{H}'$ depicted on Figure 3.2. But then one easily sees that the strategies $\sigma_0(\hbar \mid s_0^n) = 1$ and $\sigma_1(s \mid s_0^n) = 1$ form a Nash equilibrium, contrary to a claim in [CJM04]. In fact, it is not known whether there always exists a Nash equilibria in concurrent game with nonnegative terminal-reachability (Chapter 5 tries unsuccessfully to solve this issue).

However, the intuition behind this example remains valid as the authors of [CJM04] successfully establish the existence of a relaxed notion of Nash equilibrium, where players can deviate to improve their payoff by at most ε, for a fixed positive ε. The existence of such an *ε-Nash equilibrium* is proven with memoryless strategies, the take-away message being that as for the value of a zero-sum game, considering ε-optimality (here ε-Nash equilibrium) suffices to establish the existence of memoryless strategies.

Chapter 4

Decidability of Nash Equilibria

In this chapter, we study the existence problem of a Nash Equilibrium with arbitrary memory and precision in games with reachability and safety objectives. We will prove that the following general problem is undecidable:

<u>ENE</u>

INPUT: A stochastic terminal reward game $\mathcal{G}$ with rational probability distributions and rewards.

QUESTION: Whether there exists a Nash equilibrium for $\mathcal{G}$.

Notice that from a complexity point of view, we need to ensure that $\mathcal{G}$ can be finitely represented to make the decision problem clear. This is why we require distributions and rewards to be rational, hence finitely representable.

We will focus on a more precise problem in this chapter, that we can describe as follows:

<u>CNE</u>

INPUT: A deterministic terminal-reachability game $\mathcal{G}$, with non-negative integral terminal rewards, and three players.

QUESTION: Whether there exists a Nash equilibrium for $\mathcal{G}$, which is 1-*maximal*, that is $\mathbb{E}^{\sigma}(\Phi_1 \mid h) = \max_r \Phi_1(r)$.

One may notice our second problem CNE is way more restrictive, as we allow only three players, rewards ranging over $\mathbb{N}$, and deterministic transition functions only. However, we require in this second problem a *constraint*, the 1-maximality, so CNE does not reduce immediately to ENE. This extra requirement can be justified by several consideration

- First of all, we are often not only interested in knowing whether a Nash equilibrium exists, but also in computing one. Given two payoff profiles $x_1, x_2 \in \mathbb{R}^{\mathsf{Agt}}$, we may be interested in checking whether there exists a Nash equilibrium, with payoff profile lying between x_1 and x_2. In this particular setting, 1-maximality is a special case of constraints.

- When fixing to its maximal value the payoff of the first agent, we are interested in synthezing a strategy for this particular player, interacting with other components which are non-cooperative, but still behave rationally according to a Nash equilibrium. Thus, this problem is similar to the rational synthesis problem, studied in the non-deterministic setting by [KPV15] and [CFGR16].

Undecidability of ENE will be derived from CNE by replacing 1-maximality with gadgets with possibly negative rewards.

This chapter is devoted to the proof of undecidability of CNE, and its consequences. The core argument is a reduction of the non-halting problem of a Minsky machine [Min67], namely a 2-counter machine. This proof is composed first of the study of some game constructions that help manipulate equilibrium payoffs and observation properties of strategies. Then, we will describe several modules, or *gadgets*, that will be able to carry on the encoding of a 2-counter machine. By plugin together these modules, any control structure of such machine will be encoded and correctness of the reduction will be proven. Finally, we derive undecidability results for several classes of existence problems.

Similar proofs based on the non-halting problem of a Minsky machine appear in the litterature: first of all, [Umm08] proved a similar undecidability result for deterministic turn-based games, with 14 players and 1-maximality condition. This result was later improved by [DKM$^+$15] to 1-maximal equilibria with finite memory and pure strategies in turn-based games with at least 5 players. In the concurrent setting, [UW11a] adapted his proof to the existence of a Nash equilibrium in 14-player concurrent deterministic games, without 1-maximality condition.

All these techniques rely on the encoding of any pair of counter values (c_1, c_2) by an average payoff of $1 + \frac{1}{2^{c_1}3^{c_2}}$ for some player, while another player plays an antagonistic role and gets an average payoff of $1 - \frac{1}{2^{c_1}3^{c_2}}$. The reduction proceeds to construct several gadgets in order to test and update the counters and move to the next state of the counter machine. These constructions usually rely on the maximality condition, that is placed on a third player.

Keeping counter values as the simulated machine makes a transition to the next state usually requires to duplicate the two antagonistic players that encodes the counter. As a consequence, the resulting reductions cited before require at least 5 players. Here, the reduction exploits the concurrent action framework in order to avoid the duplication of the players and to keep reduction valid with 3 players. The proof is done at the expense of more complex gadgets, that we show to be reducible to single states each, thanks to the lack of action visibility.

4.1 Tools

4.1.1 One-shot games

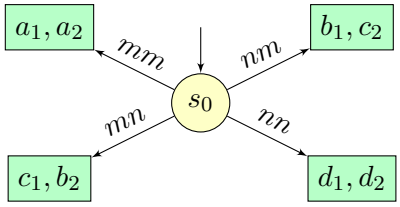
In this section, we state useful properties of Nash equilibria in two-player two-action one-shot games (that is, in one step the game ends up in a terminal state).

Such games can be represented by a graph as shown in Figure 4.1a, where terminal states are labeled with their payoff functions ϕ for both players. Alternatively, these games, also known as one-shot games, can be represented as a payoff matrix as in Table 4.1b.

Lemma 4.1. *Consider the two-player two-action one-shot concurrent game $\mathcal{G}$ of Figure 4.1, and pick some strategy profile σ. If (σ, s_0) is a Nash equilibrium, then for every player $i \in \{1, 2\}$, it holds*

$$\sigma_i(m \mid s_0) < 1 \quad \Rightarrow \quad [(d_i - c_i) + (a_i - b_i)] \cdot \sigma_{3-i}(m \mid s_0) \leq d_i - c_i \tag{4.1}$$

$$\sigma_i(m \mid s_0) > 0 \quad \Rightarrow \quad [(d_i - c_i) + (a_i - b_i)] \cdot \sigma_{3-i}(m \mid s_0) \geq d_i - c_i \tag{4.2}$$



(a) A generic two-player two-action one-shot game

2 \ 1	m	n
m	a_1, a_2	b_1, c_2
n	c_1, b_2	d_1, d_2

(b) Associated matrix representation

Figure 4.1 – Representations of a one-shot game

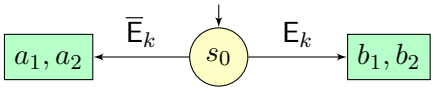


Figure 4.2 – Matching-pennies game

Proof. (σ, s_0) is a Nash equilibrium if, and only if, it is resilient to deterministic deviations (Proposition 3.16). Considering the deterministic deviation of player 1 returning move m, we get (omitting to mention s_0 in $\sigma_i(m) = \sigma_i(m|s_0)$):

$$a_1\sigma_1(m)\sigma_2(m) + b_1\sigma_1(n)\sigma_2(m) + c_1\sigma_1(m)\sigma_2(n) + d_1\sigma_1(n)\sigma_2(n) \geq a_1\sigma_2(m) + c_1\sigma_2(n).$$

As $\sigma_1(m) + \sigma_1(n) = 1$, we get $b_1\sigma_1(n)\sigma_2(m) + d_1\sigma_1(n)\sigma_2(n) \geq a_1\sigma_1(n)\sigma_2(m) + c_1\sigma_1(n)\sigma_2(n)$, which, assuming $\sigma_1(n) > 0$ (or, equivalently, $\sigma_1(m) < 1$), gives

$$[(a_1 - c_1) - (b_1 - d_1)] \cdot \sigma_2(m) \leq d_1 - c_1.$$

The other cases are similar. □

4.1.2 k-action matching-pennies games

The classical matching-pennies games are a special case of the games of the previous section, where $a_i = d_i$ and $b_i = c_i$: basically, there are two outcomes, depending on whether the players propose the same action or not. This game can be generalized to $k \geq 2$ actions, as depicted on Figure 4.2. In this figure (and in the sequel), E_k (resp. $\overline{\mathsf{E}}_k$) is a shorthand for all pairs of identical (resp. different) actions taken from a set of k actions $\Sigma_k = \{c_1, \ldots, c_k\}$. In other terms, E_k represents the set $\{c_ic_i \mid 1 \leq i \leq k\}$.

Lemma 4.2. *In the k-action matching-pennies game, playing uniformly at random for the two players defines a Nash equilibrium. This is the unique Nash equilibrium of the game if, and only if, either ($a_1 < b_1$ and $b_2 < a_2$), or ($a_1 > b_1$ and $b_2 > a_2$). The payoff profile of this Nash equilibrium is $\left(\frac{1}{k} \cdot b_1 + \left(1 - \frac{1}{k}\right) \cdot a_1, \frac{1}{k} \cdot b_1 + \left(1 - \frac{1}{k}\right) \cdot a_1\right)$.*

Proof. First of all, we check that when $\sigma_i(s_0)$ is uniform for both $i = 1$ and $i = 2$, σ is always a Nash Equilibrium. Indeed, for any $i \in \mathsf{Agt}$ and σ'_i a deviation, $\mathbb{P}^{\sigma[i/\sigma'_i]}(s_0 \cdot (b_1, b_2)) = \mathbb{P}^{\sigma}(s_0 \cdot (b_1, b_2)) = \frac{1}{k}$. Thus, the payoff profiles of σ and $\sigma[i/\sigma'_i]$ are equal to the announced result.

We prove now that this equilibrium is the only one if, and only if, ($a_1 < b_1$ and $b_2 < a_2$), or ($a_1 > b_1$ and $b_2 > a_2$).

($\Rightarrow$) We prove the contraposition, by assuming the condition on payoffs is not satisfied, that is to say ($a_1 \geq b_1$ or $b_2 \geq a_2$), and ($a_1 \leq b_1$ or $b_2 \leq a_2$), and exhibit a pure Nash equilibrium in each case.

- If $a_1 \geq b_1$ and $a_1 \leq b_1$, we let $\sigma_1(s_0) = c_1$ and one of the best profitable actions for player 2 (c_1 if $b_2 \geq a_2$, c_2 otherwise). Player 1 has no incentive to deviate since she always wins a_1, and player 2 already yields her maximal payoff.
- If $a_1 \geq b_1$ and $b_2 \leq a_2$, then $\sigma(s_0) = c_1c_1$ is a Nash equilibrium.
- The two last cases are symmetric.

($\Leftarrow$) Assume $a_1 < b_1$ and $b_2 < a_2$ (the other case is symmetric) and σ is a Nash equilibrium. By shifting payoff of player 1 (resp. 2) by $-a_1$ (resp. $-b_2$) then by rescaling by $b_1 - a_1 > 0$ (resp. $a_2 - b_2 > 0$), we can assume without loss of generality that $(a_1, a_2) = (0, 1)$ and $(b_1, b_2) = (1, 0)$. The resulting game is of constant-sum and has value $\left(\frac{k-1}{k}, \frac{1}{k}\right)$. Similarly, if player 1 does not play uniformly, some action c_α occur with probability smaller than $\frac{1}{k}$, then player 2 can ensure winning with probability bigger than $\frac{1}{k}$ by avoiding playing c_α. If player 2 does not play uniformly, some action c_α occur with probability smaller than $\frac{1}{k}$, then player 1 can ensure winning with probability bigger than $\frac{k-1}{k}$ by playing purely c_α. Both cases are incompatible with the value payoff, thus are not Nash Equilibria.

($\Rightarrow$) We prove the converse property, by noticing a Nash equilibrium that agrees on a final state exists whenever $a_1 = b_1$ or $a_2 = b_2$ or $a_1 \leq b_1$ and $a_2 \leq b_2$ or $a_1 \geq b_1$ and $a_2 \geq b_2$.

□

4.1.3 Embedded game

In this section, we present another technical construction that will be useful for our reduction: indeed, our reduction is modular, and consists in plugging several modules, sequentially, to encode the counter value updates. Some modules may require several branching transitions which are expected to lead to the same continuation of the simulation, from where a new Nash equilibrium encodes the updated values of the counters.

However, since our strategies can observe the visited states, they can distinguish such branchings, that in turn can represent different Nash equilibria hence different counter values. We are willing to prevent this phenomenon to occur, to ensure a proper encoding of the simulation, by hiding the different histories that may occur inside a module.

The basic idea consists in transforming each module, seen as a terminal-reachability game $\mathcal{G}$, into a one-shot game $\mathcal{G}'$, where all but one state are terminal, embedding all the histories and possible strategies in simple transitions. Intuitively, the states of $\mathcal{G}'$ will be the maximal runs of $\mathcal{G}$, which we will assume to be finitely many.

This method is presented in a refined manner, since our modules are in fact plugged together to form an entire simulation of a Turing machine. Some issues may arise when applying such a transformation:

- The whole game may have cycles, hence an infinite set of strategies and histories, that cannot be finitely described.

- Each agent now plays a whole strategy at a time. This means she can introduce a probabilistic correlation between her past sampled actions, and the new played actions, that were not possible otherwise. This may introduce new Nash equilibria that are irrelevant for the encoding.

In order to circumvent these problems, we will perform the embedding with respect to some subset of states, that we want to make undistinguishable, namely a sub-arena. We will see that Nash equilibria are indeed preserved when we additionally assume the sub-arena to be *action-visible* and *without cycle*, which is not true for the whole game.

Sub-arena

We proceed first tho define a sub-arena, which is intuitively just a subset of the states of the original arena, with the same structure, interpreted as a concurrent arena.

Definition 4.3. Let $\mathcal{G} = (\mathcal{A}, s, \phi)$ be a terminal-reachability game. A *sub-arena* $\mathcal{B}$ of arena $\mathcal{A}$ (and game $\mathcal{G}$) denoted $\mathcal{B} \subseteq \mathcal{A}$ (or $\mathcal{B} \subseteq \mathcal{G}$) is an arena $\mathcal{B} = (\mathsf{States}^{\mathcal{B}}, \mathsf{Agt}, \mathsf{Act}, \mathsf{Tab}, \mathsf{Allow}^{\mathcal{B}})$, with a subset of states $\mathsf{States}^{\mathcal{B}}$ but the same set of agents, actions and transition function, such that for all $s \in \mathsf{States}^{\mathcal{B}}$, we have

- Either $\mathsf{Allow}^{\mathcal{B}}(s) = \mathsf{Allow}(s)$ and $\forall A \in \mathsf{Allow}(s),\ \overline{\mathsf{Tab}(s, A)} \subseteq \mathsf{States}^{\mathcal{B}}$,
- Or $\mathsf{Allow}^{\mathcal{B}}(s) = \varnothing$ (that is $s \in \mathtt{F}^{\mathcal{B}}$) and $\forall A \in \mathsf{Allow}(s),\ \overline{\mathsf{Tab}(s, A)} \subseteq \mathsf{States} \backslash \mathsf{States}^{\mathcal{B}}$.

Where $\overline{\mathsf{Tab}(s, A)}$ denotes the support of distribution $\mathsf{Tab}(s, A)$. Any latter quantity or object relative to $\mathcal{B}$ will be denoted with an upper-script $\mathcal{B}$.

By restricting the set of allowed actions, a previously internal state can become final in the sub-arena. When a state is not final, the sub-arena structure requires all its successors to be in the sub-arena.

In the sequel, the non-final nodes of a sub-arena $\mathcal{B}$ are marked on the figures with a different color, to differentiate them, as shown on the examples on Figure 4.3. For notational purpose, we write $\mathsf{Int}^{\mathcal{B}} = \mathsf{States}^{\mathcal{B}} \backslash \mathtt{F}^{\mathcal{B}}$ the set of internal states of $\mathcal{B}$.

Note that we do not require the sub-arena to be connected, and may have several disjoint connected components. This will appear to be useful as we want to mark all possibly problematic states in all the modules used by our reduction.

Embedded game

Our main goal, as stated before, is to hide from a strategic point of view, the events occurring in a sub-arena, and give information to the players, only about the entry point of the sub-arena, and the resulting exit point, which will be a final state of the sub-arena. Note that the normal game continues from those states.

Example 4.4. *As an intuition of the construction, let us consider the games of Figure 4.3. In the left game, players can enter the sub-arena from s_0 or s_1. In the first case, agent 1 playing action b leads to state s_1 still in the sub-arena, then to states (x, y) or t_1, which are both terminal in the sub-arena. In the right-hand side of the figure, state s_0 directly leads to (x, y) and t_1 without any intermediate state of the sub-arena. In order to do so, possible actions for player 1 have been replaced by a word consisting of both her actions from s_0 and s_1,*

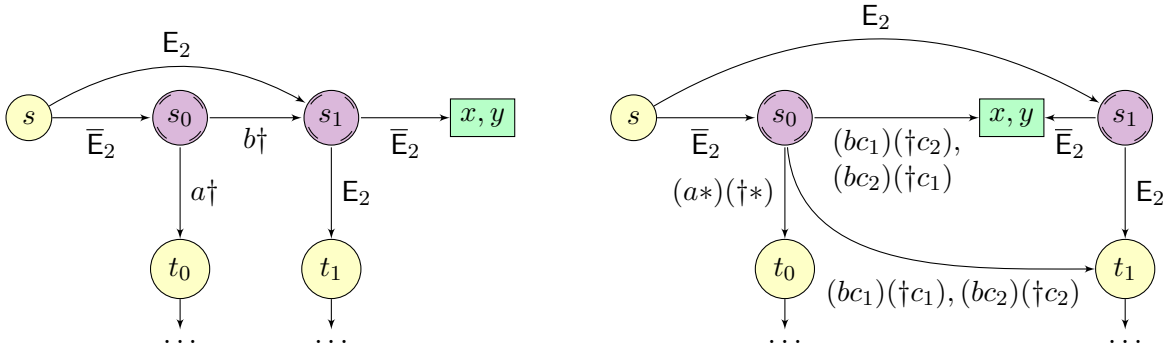


Figure 4.3 – Example of arenas, with differentiated states for a sub-arena composed of set of states $\{s_i, t_i \mid i \in \{0,1\}\}$. Only s_0 and s_1 are marked, as internal states. Notice the only internal transition in the left arena is from s_0 to s_1, which is action-visible. In the second arena, we exit immediately the arena when entering s_0 by hiding transitions that could have occur through s_1.

that is to say $\mathsf{Allow}_1(s_0) = \{ac_1, ac_2, bc_1, bc_2\}$. *Similarly, both actions for player* 2 *are encoded simultaneously from* s_0*:* $\mathsf{Allow}_2(s_0) = \{\dagger c_1, \dagger c_2\}$. *Whenever the played action of player* 1 *starts with letter* a*, the game continues to state* t_0*. Otherwise, the second letter of her played action has to be compared to the second letter of agent* 2*, which determines whether state* (x, y) *or* t_1 *is reached. We conclude that both games behave similarly, in the sense that probabilities to reach one of the final states of the sub-arena* t_0*,* t_1 *or* (x, y) *are the same. Moreover, there exists a single history connecting* s_0 *to any of them, once the transformation has been applied.*

The construction is made precise in the following definition:

Definition 4.5. Let $\mathcal{G}$ be a terminal-reachability game and $\mathcal{B}$ a sub-arena. We construct $\mathcal{G}' = \mathcal{G}/\mathcal{B}$ *the quotient game* $\mathcal{G}$ by $\mathcal{B}$, by extending the set of states to

- $\mathsf{States}' = \mathsf{States} \uplus \{\bot\}$
- If $s \in \mathsf{States} \backslash \mathsf{Int}^{\mathcal{B}}$, $\mathsf{Allow}'(s) = \mathsf{Allow}(s)$ and $\mathsf{Tab}(s, -) = \mathsf{Tab}'(s, -)$.
- If $s \in \mathsf{Int}^{\mathcal{B}}$, $\forall i$ $\mathsf{Allow}'_i(s) = M_i^{\mathcal{B}}$ which is finite, and $\forall \sigma \in M^{\mathcal{B}}$ $\forall s' \in \mathsf{F}^{\mathcal{B}}$ $\mathsf{Tab}'(s, \sigma)(s') = \mathbb{P}^{\sigma}_{\mathcal{B}}((\mathsf{Int}^{\mathcal{B}})^* \cdot s' \mid s)$ is the probability to eventually reach final state s' from s in arena $\mathcal{B}$ under strategy profile σ, considered here as an action profile in game $\mathcal{G}'$.
  We normalize by adding a transition to $\bot$: $\forall \sigma \in M^{\mathcal{B}}$ $\mathsf{Tab}'(s, \sigma)(\bot) = \mathbb{P}^{\sigma}_{\mathcal{B}}((\mathsf{Int}^{\mathcal{B}})^{\omega} \mid s)$ the probability to never reach a final state.

More informally, each non-terminal state of the sub-arena $\mathcal{B}$, is converted into a bigger state, where each allowed action corresponds to the whole strategy to a final state of the sub-arena. In the resulting game, what happens from the entrance to the exit of the sub-arena is encoded in one single state (for each entrance state). As a consequence, we can say that the strategy, when exiting the state, cannot distinguish between the possible plays inside the sub-arena. We formalize this notion below.

Definition 4.6. We define, $\varepsilon/\mathcal{B} = \varepsilon$ and for any $h \cdot s \in \mathsf{States}^+$,

$$(h \cdot s)/\mathcal{B} = \begin{cases} h/\mathcal{B} \text{ if } h \neq \varepsilon \text{ and } \mathit{last}(h) \cdot s \in (\mathsf{Int}^{A^u})^2 \\ (h/\mathcal{B}) \cdot s \text{ otherwise} \end{cases}$$

Intuitively, $h/\mathcal{B}$ only keeps from history h states that enter sub-arena $\mathcal{B}$ and terminal states that leave this sub-arena.

For example, back to Figure 4.3, we have for both games, $s \cdot s_0 \cdot s_1 \cdot t_1/\mathcal{B} = s \cdot s_0 \cdot t_1$.

Definition 4.7. We say a strategy σ is *blind to a sub-arena* $\mathcal{B}$, or $\mathcal{B}$*-blind* for short, if for all $h, h_1, h_2 \in \mathsf{States}^+$, such that $(h_1/\mathcal{B}) \cdot h = (h_2/\mathcal{B}) \cdot h$, we have $\sigma(h_1 \cdot h) = \sigma(h_2 \cdot h)$.

As we can expect, the previous notion allows us to consider Nash equilibria with blindness to some sub-arena, from the perspective of a standard Nash equilibrium, thanks to the following theorem:

Theorem 4.8. *Let $\mathcal{G}$ be a terminal-reachability game and $\mathcal{B}$ an action-visible sub-arena and payoff vector $v \in \mathbb{R}^{\mathsf{Agt}}$. Assume that for two internal states $s, s' \in \mathsf{Int}^{\mathcal{B}}$, there exists at most one path (with actions) from s to s'.*

Then, there exists a Nash equilibrium with value v in $\mathcal{G}/\mathcal{B}$ if, and only if, there exists a Nash equilibrium with value v in $\mathcal{G}$ which is blind to $\mathcal{B}$.

Remark 4.9. *Notice the hypothesis on $\mathcal{B}$ is very strong, it implies first that the sub-arena is action-visible, but also that it has no cycle. In particular, this means that $\perp$ state won't be reachable. If the sub-arena is only action-visible and cycle free, one can convert it to add finite memory inside the current state (as for the conversion from action-invisible to action-visible arena). This transformation is not developed further as Definition 4.5 would require encoding finite memory strategies as actions (instead of memoryless), and this more general result is not useful in our reduction.*

Before proving the lemma directly, we introduce some notations and justify why the action-visible property is crucial:

Lemma 4.10. *Let h be an history of $\mathcal{B}$. Let $\mathsf{Prev}(h) = (\{\sigma \in M^{\mathcal{B}} \mid \mathbb{P}^{\sigma}_{\mathcal{B}}(h) > 0\})$ and assume $\mathcal{B}$ is action-visible, then for any $i \in \mathsf{Agt}$, there exists a set $\mathsf{Prev}_i(h)$ such that $\mathsf{Prev}(h) = \prod_{i \in \mathsf{Agt}} \mathsf{Prev}_i(h)$.*

Proof. We prove the result by induction on $|h|$.

- If $|h| = 1$, the result is immediate, as $\mathsf{Prev}(h) = M^{\mathcal{B}}$ and $\forall i\ \mathsf{Prev}_i(h) = M^{\mathcal{B}}_i$.
- Let $h \cdot s$ be an history such that $\mathsf{Prev}_i(h)$ are already proven to exist. Let $\sigma \in M^{\mathcal{B}}$, then $\mathbb{P}^{\sigma}(h \cdot s) > 0$ is equivalent to $\mathbb{P}^{\sigma}(h) > 0$ and $\mathbb{P}^{\sigma}(last(h) \cdot s) > 0$. However, there exists at most one action profile $A \in \mathsf{Act}^{\mathsf{Agt}}$ such that $\mathsf{Tab}(last(h), A)(s) > 0$. If A does not exist, we have $\mathsf{Prev}(hs) = \varnothing$, otherwise, $\mathsf{Prev}(hs) = \{\sigma \in \mathsf{Prev}(h) \mid \sigma(last(h)) = A\} = \prod_{i \in \mathsf{Agt}} \{\sigma_i \in \mathsf{Prev}_i(h) \mid \sigma_i(last(h)) = A_i\}$, which concludes the induction step.

□

Thanks to this lemma, we can locally translate an embedded strategy profile into strategies for each player:

Lemma 4.11. *Let $\mathcal{B}$ an action-visible sub-arena, such that for any $s_1, s_2 \in \mathsf{Int}^{\mathcal{B}}$ two internal states, there exists at most a path (with actions)from s_1 to s_2. Let $\sigma^u \in \mathbb{S}^{\mathcal{G}/\mathcal{B}}$, and $\sigma \in \mathbb{S}^{\mathcal{G}}$ such that for any $h \in (\mathsf{Int}^{\mathcal{B}})^+$*

$$\forall i \in \mathsf{Agt}\ \forall a \in \mathsf{Act}\ \sigma_i(a \mid h) = \sigma^u_i(first(h))\left(\left\{\sigma'_i \in \mathsf{Prev}_i(h) \;\middle|\; \sigma'_i(h) = a\right\}\right)$$

Then, for all $s_1 \in \mathsf{States}^{\mathcal{B}}$, $s_f \in \mathsf{F}^{\mathcal{B}}$, *we have*

$$\mathbb{P}^{\sigma^u}_{\mathcal{G}/\mathcal{B}}(s_1 \cdot s_f) = \sum_{h \in (\mathsf{States}^{\mathcal{B}})^*} \mathbb{P}^{\sigma}_{\mathcal{B}}(h \cdot s_f \mid s_1)$$

Proof. If $s_1 \in \mathsf{F}^{\mathcal{B}}$ then both sides are equal to 0.

We assume now that $s_1 \in \mathsf{Int}^{\mathcal{B}}$.

For any $s_2 \in \mathsf{Int}^{\mathcal{B}}$, we write $w^{s_2} \in (\mathsf{Act}^{\mathsf{Agt}})^*$ the unique sequence of action profiles, if any, leading from s_1 to s_2 in $\mathcal{B}$, denoted by history h^{s_2}.

Then, any $h \in (\mathsf{States}^{\mathcal{B}})^*$, such that $\mathbb{P}^{\sigma}_{\mathcal{B}}(h \cdot s_f \mid s_1) > 0$ can be uniquely described as $h = h^{s_2}$ for some $s_2 \in \mathsf{Int}^{\mathcal{B}}$.

Previous sum can now be rewritten:

$$\sum_{h \in (\mathsf{States}^{\mathcal{B}})^*} \mathbb{P}^{\sigma}_{\mathcal{B}}(h \cdot s_f \mid s_1) = \sum_{s_2} \mathbb{P}^{\sigma}_{\mathcal{B}}(h^{s_2} \cdot s_f) = \sum_{s_2} \mathbb{P}^{\sigma}_{\mathcal{B}}(h^{s_2}) \mathbb{P}^{\sigma(h^{s_2}-)}_{\mathcal{B}}(s_2 \cdot s_f) = \mathbb{P}^{\sigma^u}_{\mathcal{G}/\mathcal{B}}(s_1 \cdot s_f)$$

□

We now turn to the proof of Theorem 4.8.

Proof. We describe first how to convert a strategy profile σ^u for $\mathcal{G}/\mathcal{B}$ into a strategy profile σ which is $\mathcal{B}$-blind.

For any $h \in \mathsf{States}^+$, let h' be its maximal suffix in $\mathsf{Int}^{\mathcal{B}}$, and define $\sigma_i(a \mid h)$ by

$$\sigma_i(a \mid h) = \begin{cases} \sigma^u_i(a \mid h/\mathcal{B}) & \text{if } h' = \varepsilon \\ \sigma^u_i(h/\mathcal{B})\left(\{\sigma'_i \in \mathsf{Prev}_i(h') \mid \sigma'_i(h) = a\}\right) & \text{otherwise} \end{cases}$$

We can check that the corresponding strategy profile σ is indeed $\mathcal{B}$-blind by applying the definition. Notice that the translation can be done the other way around, to obtain a strategy for $\mathcal{G}/\mathcal{B}$ from a blind strategy profile σ.

For such pairs of strategy profiles, we show that they have the same pay-off. In order to do so, we introduce the measure $\mu(h) = \sum_{h' \mid h' \notin \mathsf{States}^*(\mathsf{Int}^{\mathcal{B}})^2 \wedge h'/\mathcal{B} = h} \mathbb{P}^{\sigma}_{\mathcal{G}}(h')$, which sums together all equivalent history probabilities in $\mathcal{G}$ for $\mathcal{B}$-blindness. We prove that $\forall h \in \mathsf{States}^+$ $\mu(h) = \mathbb{P}^{\sigma^u}(h)$ by induction on $|h|$. For the inductive case, we have to distinguish the case where $last(h) \notin \mathsf{Int}^{\mathcal{B}}$ which is immediate, to the opposite case, where we apply Lemma 4.11.

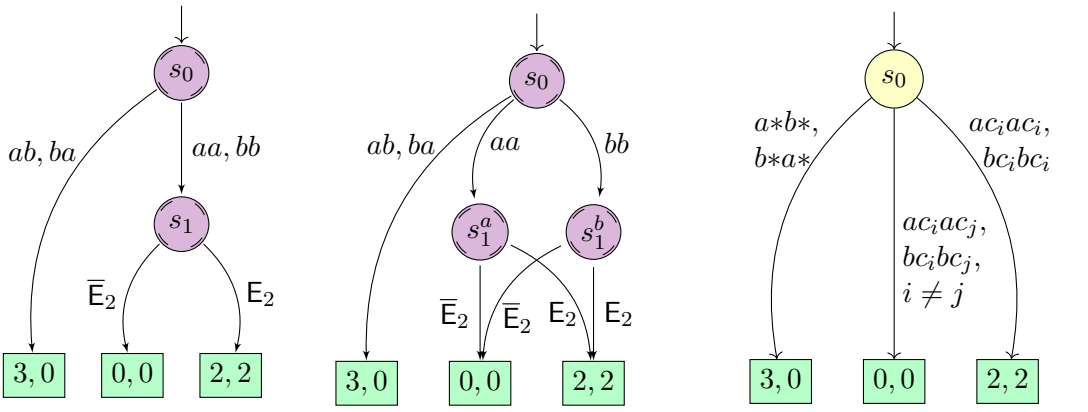
Since we are considering terminal-reachability games, and final states are preserved by embedding, we conclude that σ and σ^u yield the same payoff vector v.

Finally, we have to show σ is a Nash equilibrium if, and only if, σ^u is a Nash equilibrium. The reverse implication is immediate as we can extract from a deviation of σ^u a deviation of σ which is still $\mathcal{B}$-blind. The direct implication is concluded by noticing that a deviation can be assumed to be deterministic, hence, we can build a $\mathcal{B}$-blind deviation, which can then be converted into a deviation in the embedded game. □

On the action-visible hypothesis

We develop here why the restriction to action-visible games is crucial by considering the game structures depicted in Figure 4.4 on the next page.

Game $\mathcal{C}'$ is obtained from game $\mathcal{C}$ by making actions from s_0 visible, as discussed in Remark 3.6. Game $\tilde{\mathcal{C}}$ will be shown to correspond both to games $\mathcal{C}/\mathcal{A}$ and $\mathcal{C}'/\mathcal{A}'$.



(a) $\mathcal{C}$, two equilibrium payoffs: $\left\{\left(2, \frac{1}{2}\right), \left(\frac{5}{2}, 1\right)\right\}$

(b) $\mathcal{C}'$, with an additional equilibrium payoff: $\left(\frac{7}{3}, \frac{2}{3}\right)$

(c) $\widetilde{\mathcal{C}}$: Equivalent embedded game.

Figure 4.4 – $\mathcal{C}$ is not action-visible whereas $\mathcal{C}'$ is and has another equilibrium. This second can be replaced by a single node which has the same Nash equilibria.

2 \ 1	a	b
a	$1,1$	$3,0$
b	$3,0$	$2,2$

(a) Payoff matrix from state s_0, assuming the players agree on a deterministic action from state s_1^bbut play uniformly at random from state s_1^a.

$$\begin{cases} \sigma_1(a \mid s_0) = \dfrac{2}{2+1} = \dfrac{2}{3} \\ \sigma_2(a \mid s_0) = \dfrac{-1}{-1-2} = \dfrac{1}{3} \\ \mathbb{E}^\sigma(\Phi_i \mid s_0) = \left(\dfrac{7}{3}, \dfrac{2}{3}\right) \end{cases}$$

(b) Associated equilibrium

Figure 4.5 – Computation of an additional equilibrium payoff in game $\mathcal{C}'$.

There are only two possible equilibria in s_1: either both players agree on playing the same action, yielding payoff $(2, 2)$, or both players play uniformly yielding payoff profile $(1, 1)$. From state s_0, the second player has always an incentive to go to s_1 whereas the first player is always better off with profile $(3, 0)$. We conclude that game $\mathcal{C}$ has two equilibrium :

- $\sigma_i(s_0)$ uniform and $\sigma_0(s_1) = \sigma_1(s_1)$, with payoff profile $\left(\frac{5}{2}, 1\right)$;
- $\sigma_i(s_0)$ and $\sigma_i(s_1)$ uniform, with payoff profile $\left(2, \frac{1}{2}\right)$.

The previous equilibria can be achieved in the action-visible variant $\mathcal{C}'$, but a new equilibrium can occur: agents can play uniformly from s_1^α and agree on an action in the other state s_1^β ($\alpha \neq \beta$). Let us consider the case where s_1^a is played uniformly, the other case being symmetric. From state s_0, game can now be seen as a one-shot game with the following payoff table represented in Figure 4.5a. This game has a unique equilibrium that can be computed by Lemma 4.1 as seen in Figure 4.5b.

However, both embedded games $\mathcal{C}/\mathcal{A}$ and $\mathcal{C}'/\mathcal{A}'$ are somehow equivalent: let us denote the actions $\sigma_i \in M^{\mathcal{A}}$ from s_0 in $\mathcal{C}/\mathcal{A}$ as a word $w = \sigma_i(s_0)\sigma_i(s_1) \in \mathsf{Act}^2$. We represented in the same way actions in $\mathcal{C}'/\mathcal{A}'$ as a word $w = \sigma_i(s_0)\sigma_i(s_0^a)\sigma_i(s_0^b)$.

Even if $\mathcal{C}'/\mathcal{A}'$ allows more actions, we argue that these extra actions are equivalent. Indeed, for any $x, y \in \mathsf{Act}$, embedded action axy (resp bxy) in $\mathcal{C}'/\mathcal{A}'$ is equivalent to embedded action

ax (resp by) in $\mathcal{C}/\mathcal{A}$ from the point of view of transition function. This equivalence relation is transposed to the expected payoffs of both players, and their possible deviations.

We conclude that $\mathcal{C}/\mathcal{A}$, $\mathcal{C}'/\mathcal{A}'$ and $\mathcal{C}'$ have the same set of Nash Equilibria. This implies that Theorem 4.8 cannot apply on $\mathcal{C}$ and sub-arena $\mathcal{A}$ which is not action-visible.

To summarize, we have in this section that several branchings in a module can be hidden from strategies thanks to an embedding of all possible strategies seen as simple actions. The reduction is shown to be sound whenever action are visible in the module and that there exists a unique path from every two states in the module. Notice we can replace the latter condition by a simple cycle-free hypothesis by using finite memory strategies as actions instead of simply memoryless strategies. However, the action-visible hypothesis is crucial for the correction of the proof. Notice also that paradoxically, the resulting embedded state has multiple actions (one for each strategy) pointing to the same final state in the sub-arena. This means that strategies should not see actions, which is the case in our model, in order to hide these branchings.

4.2 Modules

In this section, we present a game that will be a building block in our undecidability proof, which is based on an encoding of a two-counter machine.

4.2.1 Rescale game

We discuss here different ways of updating the values of a game when the strategy profile forms a 1-maximal Nash equilibrium. This will be useful later for updating counter values in the undecidability reduction. Assume the continuation of a game forms a Nash Equilibrium of payoff $(1, 4+x, 4-x)$ for some $x \geq 0$. We want to produce a new 1-maximal Nash Equilibrium with payoff $(1, 4+\alpha \cdot x, 4-\alpha \cdot x)$ for some constant α. A game achieving such a property is described below.

Consider the 3-player game $\mathcal{R}_k$ depicted on Figure 4.6: in this game, player 1 has two available actions a and b from r_0, s_k and s_l, while the other two players can either continue (action c), or unilaterally decide to stop the game (action s) and go to a terminal state (where player 1 will have payoff 0). Notice that s_k and s_l have the same structure, but with different values for their immediate terminal successors. In node t_k, only players 2 and 3 have a choice: they can either continue to final state n (when both of them play c), or decide to stop and possibly go to a k-action matching-pennies game. In Figure 4.6, we write S as a shorthand to represent any combination of moves of players 2 and 3 where at least one of them decides to stop (action s). Node n is a final node with some payoff $(1, 4+x, 4-x)$ that will be later replaced by the continuation of the game, when considering the actual reduction. Notice that the sub-arena described by the set of states $\{r_0, s_k, s_l, t_k, (0,4,4), (0,5,3), (0,5+k,3-k), (1,4+k,4-k), (0,5+l,3-l), (1,4+l,4-l)\}$ is action-visible. Notice that t_k is a terminal node in this sub-arena, therefore not marked. Indeed, if t_k was marked, there would have been two paths from r_0 to t_k in the sub-arena, and Theorem 4.8.

We relate 1-maximal Nash equilibria from r_0 and those from n:

Proposition 4.12. *Consider game $\mathcal{R}_k$ from Figure 4.6, with some fixed parameter $x \in \mathbb{R}$. Then there exists a 1-maximal Nash equilibrium σ from r_0 if, and only if, $0 \leq k \cdot x \leq 1$.*

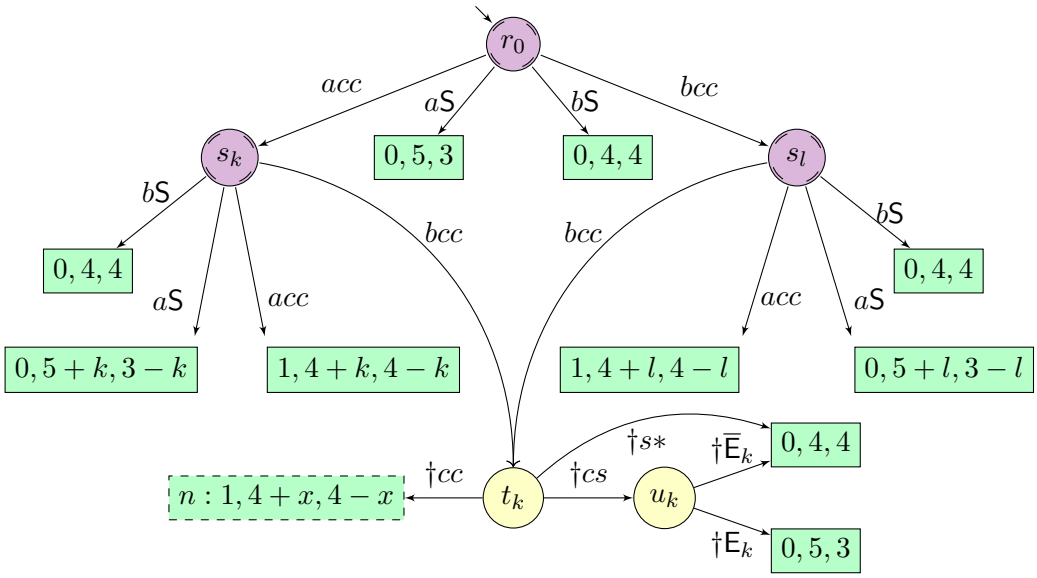


Figure 4.6 – The rescale game $\mathcal{R}_k$ for $k \in \{1, 2, 3\}$ and $l = k - 1$

Moreover, for any such Nash equilibrium, terminal state n is reached with positive probability and expected payoff profile equals $(1, 4 + k \cdot x, 4 - k \cdot x)$.

We first give the proof of the necessary conditions:

Proof. Since the strategy profile has to be 1-maximal, any terminal state rewarding agent 1 with payoff 0 should be avoided. One of the two states s_k or s_l (or both) should be enabled by σ. However, action profile *acc* cannot be played with probability 1 by all three players, otherwise player 2 can deviate to win $5 + k$ (resp $5 + l$) instead of $4 + k$ (resp $4 + l$). This means that t_k is also enabled by strategy profile σ. Players should then agree to go to final state n from t_k in order to achieve 1-maximality.

Player 2 has no incentive to deviate from t_k, so $4 + x \geq 4$, that is $x \geq 0$. Value of state u_k can be computed and equals $v(u_k) = (0, 4 + 1/k, 4 - 1/k)$. Since player 3 has no incentive to deviate from t_k to enable state u_k, we also have $4 - x \geq 4 - 1/k$ that is $k \cdot x \leq 1$. □

The sufficient condition, and the exact computation value is carried on in the following lemma. We assume that $0 \leq k{\cdot}x \leq 1$, and start to build a strategy profile which is $1-$*maximal* Nash equilibrium. We can already state that such profile must enable n with probability 1 from t_k and play uniformly from state u_k to avoid any deviation, from t_k.

Lemma 4.13. *Assume that there is a Nash equilibrium from $s_k{\cdot}t_k$ with payoff $(1, 4{+}x, 4{-}x)$. If (σ, s_k) is a 1-maximal Nash equilibrium from s_k, then the state t_k is activated by σ, and the expected payoff of σ from s_k is $(1, 4 + (k + 1) \cdot \frac{x}{x+1}, 4 - (k + 1) \cdot \frac{x}{x+1})$. Furthermore such a 1-maximal Nash equilibrium from s_k exists and consists for player 1 in playing a with probability $x/(x{+}1)$ and b with probability $1/(x{+}1)$, and for the other two players, in playing c almost-surely in s_k, and then follow the given equilibrium from $s_k t_k$.*

Proof. Let (σ, s_k) be a 1-maximal Nash equilibrium. Because the equilibrium is 1-maximal, players 2 and 3 do not play action s: we have $\sigma_2(c \mid s_k) = \sigma_3(c \mid s_k) = 1$. Considering this

2 \ 1	a	b
s	$0, 5+k$	$0, 4$
c	$1, 4+k$	$1, 4+x$

3 \ 1	a	b
s	$0, 3-k$	$0, 4$
c	$1, 4-k$	$1, 4-x$

Table 4.1 – Two-player projections of $\mathcal{R}_k$ in s_k assuming player 3 (left), resp. player 2 (right), plays c almost-surely

fixed action for player 3, we can look at the 2-player game between players 1 and 2, which is represented in matrix form in Table 4.1 (left).

Applying Lemma 4.1, using the fact that $\sigma_2(s \mid s_k) < 1$, we get that $(x+1) \cdot \sigma_1(a \mid s_k) \leq x$. The same argument applied to the projection to players 1 and 3 (see Table 4.1 (right)) gives $(x+1) \cdot \sigma_1(a \mid s_k) \geq x$. Hence $(x+1) \cdot \sigma_1(a \mid s_k) = x$. This entails that $x \neq -1$ (actually, x will be forced to be nonnegative in the sequel), so that the action profile bcc has probability $1/(x+1)$ in the Nash equilibrium (σ, s_k). We conclude that t_k is reached with positive probability, and that

$$\mathbb{E}^\sigma(\phi \mid s_k) = \left(1, \frac{(4+k)x + (4+x)}{x+1}, \frac{(4-k)x + (4-x)}{x+1}\right).$$

Conversely we check that the strategy profile from s_k where player 1 plays a with probability $x/(x+1)$ and b with probability $1/(x+1)$ in s_k, and where the other two players play c almost-surely, is a 1-maximal Nash equilibrium. □

Notice the same reasoning (and lemma) can be done from state $s_l = s_{k-1}$, with the same conclusions.

We now consider the global game $\mathcal{R}_k$ from its initial state r_0, in order to conclude.

Lemma 4.14. *Assume that $0 \leq k \cdot x \leq 1$. Then, any 1-maximal Nash equilibrium from r_0 has payoff $(1, 4 + k \cdot x, 4 - k \cdot x)$. Moreover, such a strategy profile exists.*

Proof. Since the equilibrium is 1-maximal, it holds $\sigma_2(c \mid r_0) = \sigma_3(c \mid r_0) = 1$. We first consider the cases when only one of the states s_k and s_l is enabled:

- if only s_k is enabled, *i.e.* $\sigma_1(a \mid r_0) = 1$, then from the previous lemma, we have $\mathbb{E}^\sigma(\Phi_2) = 4 + (k+1) \cdot \frac{x}{x+1}$. This quantity should be greater than 5 (otherwise player 2 would better deviate), so that $k \cdot x \geq 1$, and using our hypothesis, $k \cdot x = 1$. It follows that the payoff of σ from r_0 is $(1, 5, 3)$ in this case.

- if only s_l is enabled, *i.e.* $\sigma_1(a \mid r_0) = 1$, the value for player 3 is $4 - (l+1) \cdot \frac{x}{x+1}$. This must be greater than or equal to 4 (otherwise player 2 has a profitable deviation). We get $x = 0$, and the expected payoff of σ is $(1, 4, 4)$.

We now consider the case where both states s_k and s_l are enabled, *i.e.* $0 < \sigma_1(a \mid r_0) < 1$. We again separately consider the strategies of players 2 and 3, as shown in Table 4.2. Let us fix y_i (for $i \in \{k, l\}$) and y such that $\mathbb{E}^\sigma(\phi \mid r_0 s_i) = (1, 4 + y_i, 4 - y_i)$, and $\mathbb{E}^\sigma(\phi \mid r_0) =$

2 \ 1	a	b
s	$0,5$	$0,4$
c	$1,4+y_k$	$1,4+y_l$

3 \ 1	a	b
s	$0,3$	$0,4$
c	$1,4-y_k$	$1,4-y_l$

Table 4.2 – Two-player projections of $\mathcal{R}_k$ in r_0 assuming player 3 (left), resp. player 2 (right), plays c almost-surely

$(1, 4+y, 4-y)$. Applying Lemma 4.1 twice, we get $(y_l + 1 - y_k) \cdot \sigma_0(a \mid r_0) = y_l$. Then the expected payoff for player 2 is

$$\mathbb{E}^\sigma(\Phi_2 \mid r_0) = 4 + y = \sigma_1(a \mid r_0) \cdot (4 + y_k) + (1 - \sigma_1(a \mid r_0)) \cdot (4 + y_l).$$

This simplifies as $y = y_l/(y_l - y_k + 1)$. Replacing y_l and y_k with their values (Lemma 4.13), we end up with $y = k \cdot x$.

Now to reconstruct a 1-maximal Nash equilibrium from r_0, it is sufficient for player 1 to play a at r_0 with probability $k \cdot x$, and for the other two players, to play c almost-surely at r_0, and then to follow the 1-maximal Nash equilibrium from s_k and s_l. This yields a 1-maximal Nash equilibrium from r_0, with the announced expected payoff profile. □

To summarize this section, we have built a module that takes a sub-game as an input (from state n), which is assumed to have an equilibrium of payoff $(1, 4+x, 4-x)$, and returns a game which has an equilibrium of payoff $(1, 4+k \cdot x, 4-k \cdot x)$ for a given integer k. Notice the reverse operation (dividing by k) is more natural, as it is sufficient to implement a k-matching-pennies state to implement it. This way of the rescaling game was harder, as it requires somehow to increase the gap of expected rewards between players 2 and 3. This has been achieved thanks to the extra 1-maximality requirement.

4.2.2 Testing game

We present in this section the construction of a game for comparing the expected payoffs in different nodes. This will be useful in our reduction to encode the zero-tests of our two-counter machine. Once again, this construction heavily relies on the 1-maximal character of our studied Nash Equilibria.

Consider the game $\mathcal{T}$ of Figure 4.7. Any non-terminal state is now marked, since the whole game is action-visible. Two terminal states, n_1 and n_2, will be later replaced by continuation of another reduction, but are given here with 1-maximal Nash equilibria payoff profiles of the form $(1, 4+x, 4-x)$ and $(1, 4-y, 4+y)$ for some $x, y \in \mathbb{R}_{\geq 0}$.

Proposition 4.15. *Consider the game $\mathcal{T}$ of Figure 4.7, for some parameters $x, y \geq 0$. There exists a 1-maximal Nash equilibrium from s_0 if, and only if, $x = y$.*

Moreover, when this condition holds, the expected payoff its equals $(1, 4+x/2, 4-x/2)$. Additionnaly, n_1 and n_2 are both enabled whenever $x = y > 0$.

Proof. If $x = y$, we build σ that plays uniformly from states s_2^a and s_2^b, yielding an equilibrium of value $(1, 4, 4)$. Playing uniformly from s_0 then playing action profile $\dagger cc$ from s_1^a and s_1^b

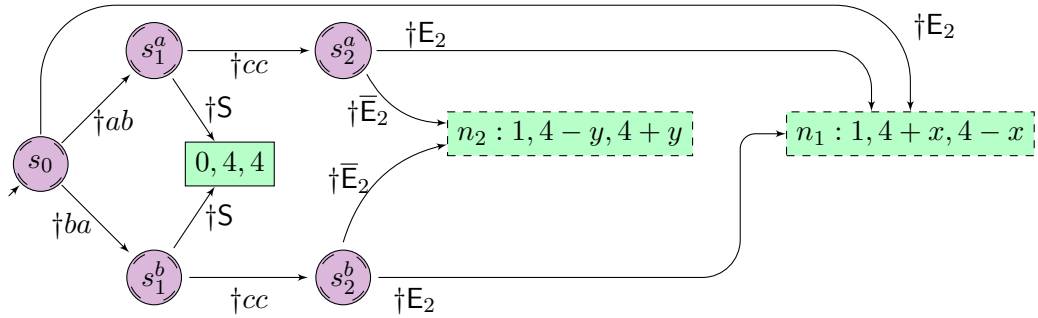


Figure 4.7 – Testing module $\mathcal{T}$

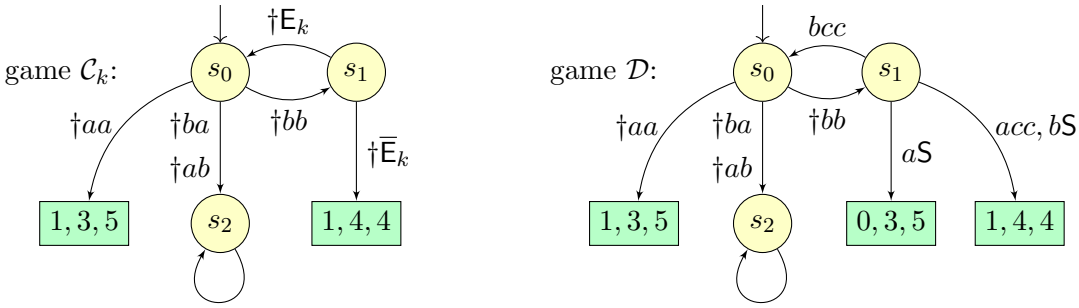


Figure 4.8 – The modules $\mathcal{C}_k$ (for $k \geq 2$) and $\mathcal{D}$. Notice that state s_2 should be considered terminal, as it only carries a self-loop. We could replace it by a two-state loop. We could also see it as a terminal state with reward $(0, 0, 0)$, but for technical reasons (in Section 4.3.2), we want the terminal rewards of players 1 and 2 to always sum to 8, which we could not achieve easily in this case.

generates the announced payoff, and we can check that no deviation to $(0, 4, 4)$ is profitable for any player.

Conversely, if σ is a 1-maximal Nash equilibrium from s_0,

- if n_1 (resp n_2) is enabled alone, then $x = 0$ (resp $y = 0$), otherwise player 3 (resp 2) could deviate to $(0, 4, 4)$.
- Otherwise, both n_1 and n_2 are enabled, then at least one of the two states s_1^a or s_1^b is visited, which means by stability that $x = y$ (otherwise one of the two players can deviate). Then, by stability, both players should play uniformly from s_0.

□

4.2.3 Counting modules

Previous testing module allows us to test two sub-games which have similar (when reversing player 2 and 3) payoff profiles. Our reduction strategy will consist in simulating on the one hand a 2-counter machine, and on the other hand, check that the resulting encoding has a particular form. In order to check for this particular form, we introduce below games that can generate some specific families of Nash equilibria with a particular expected payoffs.

Lemma 4.16. *Consider the games of Figure 4.8. For $n \in \mathbb{N} \uplus \{+\infty\}$, we define*

$$r_k(n) = \left(1, 4 - \frac{1}{k^n}, 4 + \frac{1}{k^n}\right) \qquad s(n) = \left(1, 4 - \frac{1}{n+1}, 4 + \frac{1}{n+1}\right)$$

Then the set of 1-maximal Nash equilibrium values is $\{s(n) \mid n \in \mathbb{N} \uplus \{\infty\}\}$ in $\mathcal{D}$, and $\{r_k(n) \mid n \in \mathbb{N} \uplus \{\infty\}\}$ in $\mathcal{C}_k$ for all $k \geq 2$.

Proof. Fix $k \geq 2$. We begin with proving that these values are indeed the payoffs of Nash equilibria. For this, we define the witnessing strategy profiles. For all history h, we let

$$\gamma^\infty(hs_0) = abb \qquad \gamma^\infty(hs_1) = ac_1c_2 \qquad \delta^\infty(hs_0) = abb \qquad \delta^\infty(hs_1) = bss$$

One easily observes that (γ^∞, s_0) and (δ^∞, s_0) are Nash equilibria with payoff $(1, 4, 4)$ in $\mathcal{C}_k$ and in $\mathcal{D}$, respectively.

For $n \in \mathbb{N}$, we define γ^n and δ^n inductively. First, we let $\gamma^0(s_0) = \delta^0(s_0) = aaa$, which gives rise to Nash equilibria with payoff $(1, 3, 5)$ from s_0 both in $\mathcal{C}_k$ and in $\mathcal{D}$.

Then, for $n \in \mathbb{N}$, we define the strategies inductively on the length of the history. The base cases are $\gamma^{n+1}(s_0) = \delta^{n+1}(s_0) = abb$ and

$$\gamma_1^{n+1}(s_0s_1) = a \qquad \forall 1 \leq j \leq k.\ \gamma_2^{n+1}(c_j \mid s_0s_1) = \gamma_3^{n+1}(c_j \mid s_0s_1) = 1/k$$

$$\delta_1^{n+1}(a \mid s_0s_1) = \frac{1}{n+2} \qquad \delta_2^{n+1}(s_0s_1) = \delta_3^{n+1}(s_0s_1) = c.$$

The inductive case is

$$\forall 1 \leq i \leq 3, \gamma_i^{n+1}(s_0s_1h) = \gamma_i^n(h) \qquad \delta_i^{n+1}(s_0s_1h) = \delta_i^n(h)$$

We can indeed check that these strategy profiles form Nash equilibria with the expected payoffs.

Conversely, let us fix a 1-maximal Nash equilibrium (σ, s_0), and let us show that the expected payoff of that Nash equilibrium is one of the above values. By 1-maximality, players 2 and 3 have to play deterministically the same actions after all histories ending up in s_0 (state s_2 should not be enabled under (σ, s_0)). We can reason on the number of histories enabled from s_0. For that, we define $N_{s_0}(\sigma) = \sup\{|h| \mid h \text{ enabled by } \sigma \text{ and } \mathit{last}(h) = s_0\}$ (this is somehow the maximal number of visits of s_0 enabled by σ).

- If $N_{s_0}(\sigma) = +\infty$, then the transition $\dagger aa$ from s_0 is never taken (since it would then be played deterministically and it would then stop the game immediately). Since (σ, s_0) is 1-maximal, it then means that the terminal state $(1, 4, 4)$ is reached almost-surely (only possibility for Player 1 to get payoff 1).
- Otherwise, $N_{s_0}(\sigma)$ is finite and we reason by induction on this number:
  - first if $N_{s_0}(\sigma) = 1$, the game ends up immediately in $(1, 3, 5)$.
  - if $N_{s_0}(\sigma) > 1$, then a transition to s_1 occurs with probability 1. If the path $s_0 \cdot s_1 \cdot (1, 4, 4)$ has probability 1 under σ, then the results holds; otherwise $s_0s_1s_0$ is enabled from s_0, and (σ', s_0), with $\sigma' \colon h \mapsto \sigma(s_0s_1h)$, is another 1-maximal Nash

equilibrium such that $N_{s_0}(\sigma') < N_{s_0}(\sigma)$. By induction, it has an expected payoff of the form $(1, 4-x, 4+x)$, with either $x = \frac{1}{k^n}$ (for $\mathcal{C}_k$) or $x = \frac{1}{n+1}$ (for $\mathcal{D}$) for some $n \in \mathbb{N} \uplus \{\infty\}$. If $x = 0$, the results holds immediately, as the expected payoff of (σ, s_0) is also $(1, 4, 4)$. Now assume $x > 0$, and consider the game $\mathcal{C}_k$, and the distributions proposed by the strategies σ_2 and σ_3 after $s_0 s_1$. For this to be a Nash equilibrium, both distributions must be uniform; this leads to payoff $(1, 4-\frac{x}{k}, 4+\frac{x}{k})$, and proves the result. For $\mathcal{D}$, if it were $\sigma_1(s_0 s_1) = b$, then Player 2 would have a profitable deviation. Hence $\sigma_1(a \mid s_0 s_1) > 0$, and the best response for players 2 and 3 is to play c. We can then analyze the projections on agents $1, 2$ and $1, 3$ (as done in Section 4.2.1) and apply Lemma 4.1, which yields $(x+1)\sigma_1(a \mid s_0 s_1) \geq x$ and $(x+1)\sigma_1(a \mid s_0 s_1) \leq x$; it follows $\sigma_1(a \mid s_0 s_1) = \frac{1}{n+2}$ and $\mathbb{E}^\sigma(\phi \mid s_0 s_1) = (1, 4-\frac{1}{n+2}, 4+\frac{1}{n+2})$. □

4.2.4 Description of the reduction

We now turn to the global undecidability proof of the constrained-existence problem in three-player games. The proof is a reduction from the recurring problem of a two-counter machine.

We first recall the definition of a two-counter machine as a tuple $\mathcal{M} = (Q, q_0, \Delta)$ where:

- Q is a finite set of states,
- $q_0 \in Q$ is an initial state,
- $\Delta \subseteq Q \times \Gamma \times Q$ is the transition table with $\Gamma = \{inc(j), dec(j), zero(j), !zero(j) \mid j \in \{1, 2\}\}$ is the set of operations on counters.

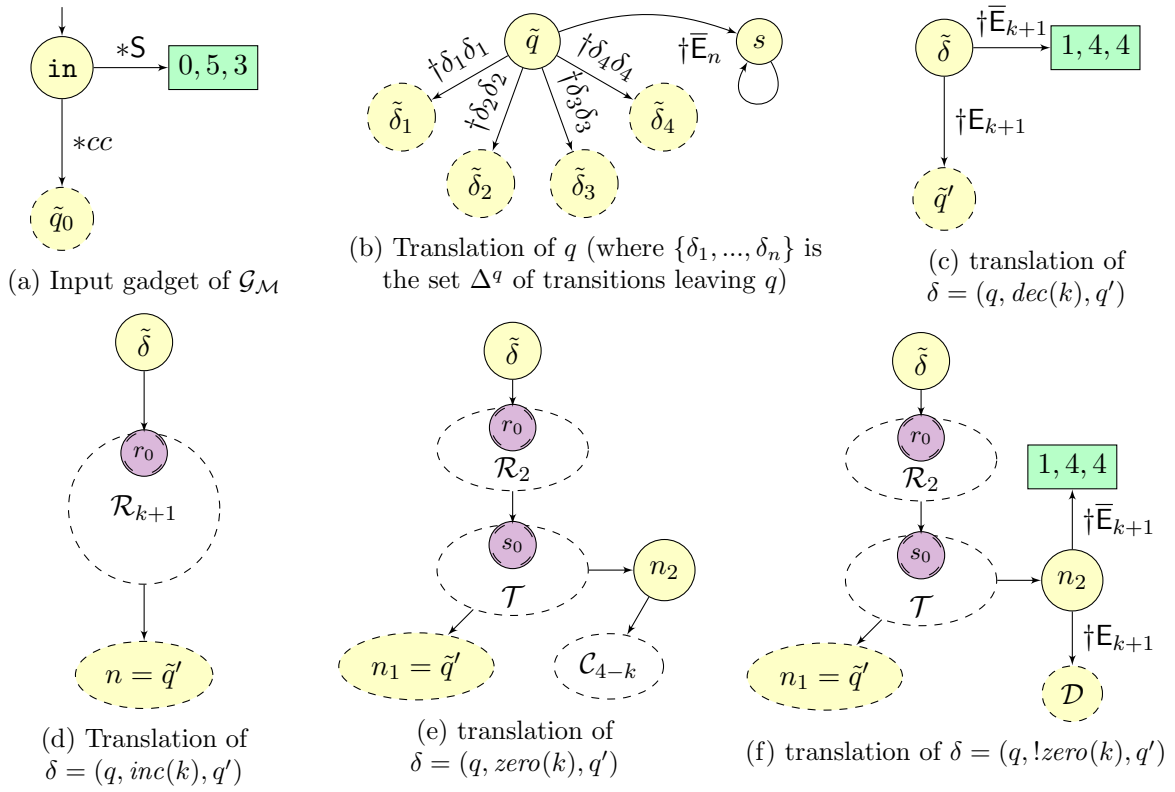
Without loss of generality, we assume that any considered machine $\mathcal{M}$ never decreases a counter with value 0. This can be syntactically enforced by placing a non-zero test before any decrement operation.

The semantics of $\mathcal{M} = \langle Q, q_0, \Delta \rangle$ is given as a transition system where configurations are tuples $C = (q, c_1, c_2) \in Q \times \mathbb{N} \times \mathbb{N}$ and for any two configurations $C = (q, c_1, c_2)$ and $C' = (q', c_1', c_2')$, for every $\delta = (q, \gamma, q') \in \Delta$, there is a transition $C \to_\delta C'$ if, and only if:

- $c_k' = c_k + 1$ and $c_{3-k}' = c_{3-k}$, if $\gamma = inc(k)$;
- $c_k' = c_k - 1$ and $c_{3-k}' = c_{3-k}$, if $\gamma = dec(k)$;
- $c_k = 0$ and $(c_1', c_2') = (c_1, c_2)$, if $\gamma = zero(k)$;
- $c_k > 0$ and $(c_1', c_2') = (c_1, c_2)$, if $\gamma = !zero(k)$.

We fix for the rest of this section a two-counter machine $\mathcal{M} = (Q, q_0, \Delta)$, and we build a terminal-reachability game $\mathcal{G}_\mathcal{M}$ as depicted in Figure 4.9 on the facing page: for every machine state $q \in Q$ (resp. every $\delta \in \Delta$) we have a corresponding state $\tilde{q}$ (resp. state $\tilde{\delta}$) in the game $\mathcal{G}_\mathcal{M}$, as depicted on Figure 4.9b (resp. Figures 4.9c to 4.9f). These states are connected together thanks to the described translation which takes into account the transition table of $\mathcal{M}$. We write ϕ for the terminal-reachability payoff function that is given by $\mathcal{G}_\mathcal{M}$.

The states of the game $\mathcal{G}_\mathcal{M}$ are $\tilde{Q} \uplus \tilde{\Delta} \uplus T$, where T represents the set of states that were introduced by our reduction, as part of one of the different gadgets, with multiplicity. Moreover, we have seen that these modules may have some branching, that will be hidden



(a) Input gadget of $\mathcal{G}_{\mathcal{M}}$

(b) Translation of q (where $\{\delta_1, ..., \delta_n\}$ is the set Δ^q of transitions leaving q)

(c) translation of $\delta = (q, dec(k), q')$

(d) Translation of $\delta = (q, inc(k), q')$

(e) translation of $\delta = (q, zero(k), q')$

(f) translation of $\delta = (q, !zero(k), q')$

Figure 4.9 – Description of the encoding $\mathcal{G}_{\mathcal{M}}$. Modules are $\mathcal{R}_k$ and $\mathcal{T}$ are copied several times. $n = \tilde{q}$ means the terminal state n of a module is replaced by further encoded state $\tilde{q}$.

by considering the embedded game construction with respect to $\mathcal{B}$, which is the sub-arena composed of the marked states of the modules. For the sake of simplicity, we assume, from now on, that $\mathcal{G}_{\mathcal{M}}$ has already been quotiented by this sub-arena. In the sequel, any considered Nash equilibrium or strategy profile will therefore become implicitly a $\mathcal{B}$-blind strategy profile in the original game.

We will evaluate the existence of a 1-maximal Nash equilibrium from all the main nodes of the game. The relation between $\mathcal{M}$ and $\mathcal{G}_{\mathcal{M}}$ is made explicit thanks to the following predicate. Let $s \in Q \uplus \Delta$, and $c_1, c_2 \in \mathbb{N}$. We denote by $P(s, c_1, c_2)$ the predicate:

$$\exists \sigma \in \mathbb{S}. \left[\sigma \text{ is a Nash equilibrium from } \tilde{s} \text{ and } \mathbb{E}^{\sigma}(\phi \mid \tilde{s}) = \left(1, 4 + \frac{1}{2^{c_1}3^{c_2}}, 4 - \frac{1}{2^{c_1}3^{c_2}}\right)\right]$$

Lemma 4.17. *Assume $C = (q, c_1, c_2)$ is a configuration of $\mathcal{M}$ such that $P(q, c_1, c_2)$ holds. Then there are a transition δ and a configuration $C' = (q', c_1', c_2')$ such that (i) $C \to_{\delta} C'$ and (ii) $P(\delta, c_1, c_2)$ and $P(q', c_1', c_2')$ hold.*

Proof. Write σ for a 1-maximal Nash equilibrium witnessing the truth of predicate $P(q, c_1, c_2)$. In particular,

$$\mathbb{E}^{\sigma}(\phi \mid \tilde{q}) = \left(1, 4 + \frac{1}{2^{c_1}3^{c_2}}, 4 - \frac{1}{2^{c_1}3^{c_2}}\right)$$

In $\mathcal{G}_{\mathcal{M}}$ from $\tilde{q}$, only one state $\delta \in \Delta^q$ is activated, otherwise σ would not be 1-maximal, as with positive probability the play would end up in state s of Figure 4.9b. Hence $P(\delta, c_1, c_2)$ holds as well.

We write $x = \frac{1}{2^{c_1}3^{c_2}}$, and we distinguish the different cases for δ.

- First assume $\delta = (q, dec(k), q')$. Since $x > 0$, the next state q' has to be activated by σ, and $(\sigma, \tilde{q}\tilde{\delta}\tilde{q}')$ needs to be a 1-maximal Nash equilibrium as well. Applying the analysis of k-action matching-pennies games of Section 4.1.2, it must be the case that the payoff of σ after $\tilde{q}\tilde{\delta}\tilde{q}'$ is $(1, 4 + y, 4 - y)$ with $y = (k + 1) \cdot x$. If $k = 1$, it is the case that $y = \frac{1}{2^{c_1-1}3^{c_2}}$, and if $k = 2$, it is the case that $y = \frac{1}{2^{c_1}3^{c_2-1}}$. Writing $c'_k = c_k - 1$ and $c'_{3-k} = c_{3-k}$, we get that $(q, c_1, c_2) \to_\delta (q', c'_1, c'_2)$ and that $P(q', c'_1, c'_2)$.

- Then assume $\delta = (q, inc(k), q')$. Applying Proposition 4.12, we get that $\tilde{q}'$ is activated by σ, and that there is a 1-maximal Nash equilibrium from $\tilde{q}'$ whose expected payoff is $(1, 4 + y, 4 - y)$ with $y = x/(k + 1)$. As in the previous case, writing $c'_k = c_k + 1$ and $c'_{3-k} = c_{3-k}$, we get that $(q, c_1, c_2) \to_\delta (q', c'_1, c'_2)$ and that $P(q', c'_1, c'_2)$.

- Assume $\delta = (q, zero(k), q')$. Applying Proposition 4.12, we get that the first node s_0 of $\mathcal{G}_t$ is activated by σ, and that there is a 1-maximal Nash equilibrium from that node whose payoff is $(1, 4+x/2, 4-x/2)$. Then, applying Proposition 4.15, we get that $\tilde{q}'$ and the initial node of $\mathcal{C}_{k+1}$ are activated, and that σ after $\tilde{q}'$ and σ after entering $\mathcal{C}_{k+1}$ are 1-maximal Nash equilibrium. Furthermore, writing $(1, 4 + z, 4 - z)$ and $(1, 4 - y, 4 + y)$ for the payoffs of those equilibria respectively, we should have $z = y$ and $x/2 = z/2$. In particular, $P(q', c_1, c_2)$ holds. Now thanks to Lemma 4.16, we know that there exists m such that $y = \frac{1}{(4-k)^m}$. This implies that $c_k = 0$: $(q, c_1, c_2) \to_\delta (q', c_1, c_2)$.

- Finally, assume that $\delta = (q, !zero(k), q')$. Again applying Proposition 4.12, we get that the first node of $\mathcal{G}_t$ is activated by σ, and that there is a 1-maximal Nash equilibrium from that node whose payoff is $(1, 4 + x/2, 4 - x/2)$. Then, applying Proposition 4.15, we get that $\tilde{q}'$ and node n_2 are activated, and that σ after $\tilde{q}'$ and σ after entering n_2 are 1-maximal Nash equilibria. Furthermore, writing $(1, 4 + z, 4 - z)$ and $(1, 4 - y, 4 + y)$ for the payoffs of those equilibria respectively, we should have $z = y$ and $x/2 = z/2$. In particular, $P(q', c_1, c_2)$ holds.

  Note that there is a 1-maximal Nash equilibrium from n_2 of payoff $(1, 4 - y, 4 + y)$ with $y > 0$ if, and only if, there is a 1-maximal Nash equilibrium in $\mathcal{D}$ of payoff $(1, 4-y', 4+y')$ with $y' = (k+1) \cdot y$. Now, thanks to Lemma 4.16, we know that there exists h such that $y' = \frac{1}{h+1}$, hence $y = \frac{1}{(k+1)\cdot(h+1)}$. This implies that $c_k > 0$: $(q, c_1, c_2) \to_\delta (q', c_1, c_2)$. □

We can now show the following correspondence between $\mathcal{M}$ and $\mathcal{G}_{\mathcal{M}}$:

Proposition 4.18. *The two-counter machine $\mathcal{M}$ has an infinite valid computation if, and only if, there is a 1-maximal Nash equilibrium from state* **in** *in game $\mathcal{G}_{\mathcal{M}}$.*

Proof. We use the reduction from the non-halting problem of a two-counter machine we have described. Given a two-counter machine $\mathcal{M}$, we construct game $\mathcal{G}_{\mathcal{M}}$ as on Figure 4.9. For technical reasons, we syntactically require that each incrementation is followed by a non-zero test: since decrements are preceded with a zero-test, this enforces infinitely many visits to module $\widetilde{\mathcal{G}}_t$ in $\mathcal{G}_{\mathcal{M}}$ along any infinite run, so that infinite runs will have probability zero in strategy profiles we will build (we will see that later).

Assume a 1-maximal Nash equilibrium exists from the initial state `in`. Then it must be the case that its payoff profile is at least $(1, 5, 3)$, hence equal to $(1, 5, 3)$ as the payoffs sum of player 2 and 3 always sum to 8 and payoffs of player 1 are at most 1. As the predicate $P(q_0, 0, 0)$ is true, we can inductively apply Lemma 4.17 to build the corresponding valid infinite run of the counter machine.

Conversely assume that $\mathcal{M}$ has an infinite (valid) run $C_0 \to_{\delta_0} C_1 \to_{\delta_1} \dots$ with $C_i = (q_i, c_1^i, c_2^i)$, and $c_1^0 = c_2^0 = 0$. We build a strategy profile σ inductively as follows:

- in state `in`, players 2 and 3 should play c almost-surely;
- we assume we have built σ for the prefix $C_0 \to_{\delta_0} C_1 \to_{\delta_1} \dots C_i$, and that the "main stream" of σ traverses successively the gadgets starting in $\texttt{in}, \tilde{q}_0, \tilde{\delta}_0, \dots, \tilde{\delta}_{i-1}$ and arrives in state $\tilde{q}_i$, from which we now need to define the strategy profile σ. In state $\tilde{q}_i$, the players should select transition δ_i almost-surely, and then enter gadget state $\tilde{\delta}_i$. We now distinguish the possible cases for δ_i:
  - if $\delta_i = (q_{i-1}, dec(k), q_i)$, then players 2 and 3 should play uniformly at random among the $k+1$ actions;
  - if $\delta_i = (q_{i-1}, inc(k), q_i)$, then, in $\mathcal{R}_{k+1}$, the players should follow the strategy described in Proposition 4.12 with $x = \frac{1}{(k+1)\cdot 2^{c_1^{i-1}} \cdot 3^{c_2^{i-1}}}$;
  - if $\delta_i = (q_{i-1}, zero(k), q_i)$, then, in $\mathcal{R}_2$, the players should follow the strategy described in Proposition 4.12 with $x = \frac{1}{2\cdot 2^{c_1^{i-1}} \cdot 3^{c_2^{i-1}}}$; in $\mathcal{G}_t$, the players should follow the strategy described in Proposition 4.15, and in $\mathcal{C}_{4-k}$, they should follow the strategies described in the proof of Lemma 4.16, for the correct values of the counters;
  - if $\delta_i = (q_{i-1}, !zero(k), q_i)$, then we apply a strategy as described in the previous item (except that we replace the strategy in $\mathcal{C}_{4-k}$ by that in $\mathcal{D}$).

First, due to the hypothesis on (non-)zero-tests in every syntactic loop of the machine, under the above strategy profile, the game ends up almost-surely in a terminal state, where Player 1 has reward 1. This is because in any state $\tilde{\delta}$ where δ is a test-to-zero, or a test-to-nonzero, the game ends up in gadget $\mathcal{C}_{4-k}$ or $\mathcal{D}$ with probability 1/2, and the previously defined strategies from these gadgets ensure almost-sure termination.

More formally, for every $\varepsilon > 0$, there is a length N_ε that we can easily compute such that

$$\mathbb{P}^\sigma(\text{reach terminal in no more than } N_\varepsilon \text{ steps} \mid \texttt{in}) \geq 1 - \varepsilon.$$

We write $\mathcal{G}_\mathcal{M}(N_\varepsilon)$ the game $\mathcal{G}_\mathcal{M}(N_\varepsilon)$ truncated after N_ε computation steps of $\mathcal{M}$, in which we replace any outgoing transition by a terminal node with reward $(1, 0, 0)$. We note σ_ε the truncated strategy profile.

We have that

$$\mathbb{E}^{\sigma_\varepsilon}(\Phi_j \mid \texttt{in}) \leq \mathbb{E}^\sigma(\Phi_j \mid \texttt{in}) \leq \mathbb{E}^{\sigma_\varepsilon}(\Phi_j \mid \texttt{in}) + 8\varepsilon$$

We can now show by induction on $i \leq N_\varepsilon$ that $\mathbb{E}^{\sigma_\varepsilon}(\phi \mid \texttt{in})$ is $(1, u_1^\varepsilon, u_2^\varepsilon)$ with $|5 - u_1^\varepsilon| \leq 8\varepsilon$ and $|3 - u_2^\varepsilon| \leq 8\varepsilon$, which entails that $\mathbb{E}^\sigma(\phi \mid \texttt{in}) = (1, 5, 3)$. We can then show inductively that for every i,

$$\mathbb{E}^\sigma(\phi \mid \texttt{in} \cdot T^* \cdot \tilde{q}_0 \cdot T^* \cdot \tilde{\delta}_0 \cdot T^* \cdots \tilde{\delta}_{i-1} \cdot T^* \cdot \tilde{q}_i) = \left(1, 4 + \frac{1}{2^{c_1^i} 3^{c_2^i}}, 4 - \frac{1}{2^{c_1^i} 3^{c_2^i}}\right)$$

Assume that σ is not a Nash equilibrium, and pick σ'_j a deviation of Player j (with $j \in \{2, 3\}$) which improves her payoff. By Proposition 3.16 we can assume that σ'_j is deterministic. Under $\sigma' = \sigma[j/\sigma'_j]$, we can first notice that the same gadgets are visited as under σ, since Player j cannot improve her payoff by switching the choice of the transitions (gadgets starting from $\tilde{q}$ for any $q \in Q$). We also realize that in all states of the game, under σ, the choice of Player j is either deterministic (play c) or she plays matching-pennies games uniformly at random against Player $5 - j$. Switching the choice of Player j in a matching-penny game does not change the probabilities of the two output-edges. So only a switch from action c to action s can possibly improve the payoff of Player j.

This is not the case, since by construction, we have a local Nash equilibrium in every gadget. Hence, no deterministic deviation of Player j can improve her payoff. □

4.3 Conclusions

We summarize our construction as follows: for any 2-counter machine $\mathcal{M}$, we built a game $\mathcal{G}_{\mathcal{M}}$ which has the following properties:

- Deterministic with three players;
- Reward functions are terminal, with non-negative integers;
- Rewards of agents 2 and 3 always sum up to 8;
- Rewards of agent 1 are either 0 or 1, and agent 1 is always winning, its only purpose is to balance the game for the two other players;
- $\mathcal{M}$ has an infinite run if, and only if, $\mathcal{G}_{\mathcal{M}}$ has a Nash equilibrium which is 1-maximal.

The immediate consequence of our reduction is the following undecidability theorem:

Theorem 4.19. *The constrained existence problem with non-negative rewards (*CNE*) and three players is undecidable.*

We consider in the sequel several extensions of this result. We first state two straightforward corollaries in this section, and develop more involved extensions to terminal-reachability and safety games in the next sections.

First of all, we realize that in the reduction, there is a 1-maximal Nash equilibrium from in if, and only if, there is a Nash equilibrium with social welfare larger than or equal to 9, where the social welfare is defined as the sum of the expected payoffs of all players. As an immediate corollary, we get:

Corollary 4.20. *We cannot decide whether there exists a Nash equilibrium with some lower bound on the social welfare (or with optimal social welfare) in three-player terminal-reachability games with non-negative payoffs.*

4.3.1 Unconstrained problem

We discuss in this section how to get rid of the 1-maximality condition. The main idea is similar to [Bre12] where the initial state of the game is replaced by a branching to a submodule which is a game known to not have a Nash equilibrium in the considered model.

Here, the considered module is the game $\mathcal{H}$ of Section 3.6.4, which is known to not have a Nash equilibrium even in mixed strategies with arbitrary memory. Note however this game has negative terminal reward values, so the reduction is done at the expense of introducing negative terminal rewards.

Lemma 4.21. *Let $\mathcal{G}$ be a terminal-reachability game. We can build a terminal-reachability game $\mathcal{G}'$ such that $\mathcal{G}$ has a 1-maximal Nash equilibrium if, and only if, $\mathcal{G}'$ has a Nash equilibrium.*

Proof. Without loss of generality, we can rescale payoffs in $\mathcal{G}$ and assume the maximal reward of player 1 is 1. The game $\mathcal{G}'$ is depicted on Figure 4.10, where player 1 can decide in s'_0 whether to go to $\mathcal{H}$ or to $\mathcal{G}$. Assume there is a Nash equilibrium in $\mathcal{G}'$. Since $\mathcal{H}$ has no Nash equilibrium, in any Nash equilibrium of $\mathcal{G}$ from s'_0, player 1 will play action continue (with probability 1) in s'_0. This entails that $\mathcal{G}$ has a Nash equilibrium (since the payoffs are prefix-independent). Moreover, the payoff of player 1 in this Nash equilibrium must be 1, as otherwise player 1 could secure a better payoff by going to $\mathcal{H}$ (see proof of Lemma 3.22). Conversely, if there is a 1-maximal Nash equilibrium in $\mathcal{G}$, then it gives rise to a Nash equilibrium in $\mathcal{G}'$ by letting player 1 move to $\mathcal{G}$ in s'_0. This is easily seen to be a Nash equilibrium, in particular because deviating to $\mathcal{H}$ in s'_0 cannot benefit to player 1. □

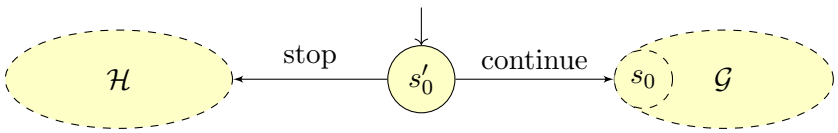


Figure 4.10 – A game that has a Nash equilibrium if, and only if, $\mathcal{G}$ has a 1-maximal Nash equilibrium

It follows:

Corollary 4.22. *The existence of a Nash equilibrium in a deterministic game with only three players and (relative) integral terminal rewards is undecidable. In particular* ENE *is undecidable.*

One drawback of the above proof is the use of the game $\mathcal{H}$ which has negative rewards so removing a constraint has been done at the expense of introducing negative rewards. In fact, game $\mathcal{H}$ can be replaced by any game with no Nash equilibria, such that player 1 can still secure a payoff $1 - \varepsilon$ for every $\varepsilon > 0$. For instance, one could use a game with limit-average payoff and nonnegative rewards only as shown in [UW11b], hence giving undecidability results for the limit-average payoff case. However, it is not known if there exists a concurrent deterministic game with at most three players and non-negative rewards which has no Nash equilibrium. Existence of such game would imply undecidability of the existence of a Nash equilibrium with non-negative terminal rewards.

4.3.2 Qualitative objectives

In order to strengthen our result, we are interested in qualitative objectives, that is to say objective functions ranging over $\{0, 1\}$. More precisely, we study here how we can transform our previous reductions in order to use only qualitative terminal reachability and safety objectives.

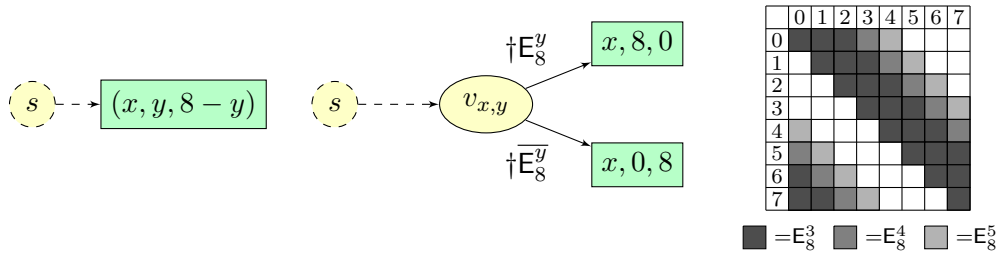


Figure 4.11 – Transformation of a terminal node $(x, y, 8-y)$ with an intermediate node $v_{x,y}$. The table on the right gives the value of E_8^y for some values of y (notice that $\mathsf{E}_8^y \subseteq \mathsf{E}_8^{y'}$ when $y \leq y'$, so that for instance $\sharp\left[\mathsf{E}_8^4\right] = 32$).

Terminal-reachability games

We now explain how to extend our main theorem to games with terminal-reachability objectives (in other terms, with terminal payoffs in $\{0, 1\}$). The crucial point to achieve this is that in all our terminal states of $\mathcal{G}_{\mathcal{M}}$, the sum of the rewards of players 2 and 3 is 8. Our construction amounts to replacing these terminal rewards with a simple module in which the payoffs of players 2 and 3 are $(8, 0)$ and $(0, 8)$.

Proposition 4.23. *Let $\mathcal{G} = (\mathcal{A}, s_0, (\Phi_i)_i)$ be a 3-player terminal-reachability game such that in any final state s, the terminal payoff $\Phi(s) = (x, y, z)$ satisfies the following conditions:*

$$x \in \{0, 1\} \qquad y, z \in \mathbb{N} \qquad y + z = 8$$

Then we can construct an arena $\mathcal{A}'$, and a qualitative terminal-reachability reward ϕ'_i for each player i, such that (σ, s_0) is a 1-maximal Nash equilibrium in $\mathcal{G}$ if, and only if, it is a 1-maximal Nash equilibrium in $\mathcal{G}' = (\mathcal{A}', s_0, (\phi'_i)_i)$.

Proof. We replace every final node $(x, y, 8-y)$ with a constant-sum game as depicted in Figure 4.11. In this figure, for all $i \in \{1, 2\}$, the set $\mathsf{Allow}_i(v_{x,y})$ of allowed actions is the set $\{c_i \mid i \in [\![0, 7]\!]\}$, and for any $k, y \in \mathbb{N}$, $\mathsf{E}_k^y = \{c_i \cdot c_j \mid \exists 0 \leq r < y.\ i - j = r \mod k\}$. Notice this definition generalizes the k-action construction of Section 4.1.2, where $\mathsf{E}_k = \mathsf{E}_k^0$.

By playing uniformly at random, player 2 can ensure winning (*i.e.* reaching state $(x, 8, 0)$) with probability $y/8$, whatever player 3 does. She then gets payoff y. Similarly, player 3 can ensure winning with probability $1 - y/8$ by playing uniformly. We conclude that $(x, y, 8-y)$ is the only equilibrium payoff.

We built this way a new game $\mathcal{G}'$ where all final payoffs are of the form $(x, 8, 0)$ or $(x, 0, 8)$. Every Nash equilibrium in $\mathcal{G}$ can be converted into a Nash equilibrium with the same payoff (by playing uniformly in every new node $v_{x,y}$) in $\mathcal{G}'$.

Conversely, if (σ, s_0) is a Nash equilibrium in $\mathcal{G}'$, then for every $hv_{x,y} \in \mathsf{States}^+$,

- If $hv_{x,y}$ is enabled by σ, we have $\mathbb{E}^\sigma(\phi \mid hv_{x,y}) = (x, y, 8-y)$;
- Otherwise, $hv_{x,y}$ is not enabled and we can assume $\sigma_i(hv_{x,y})$ is the uniform distribution for both $i \in \{1, 2\}$. This assumption does not change the final reward of the game (as $hv_{x,y}$ is not enabled) and preserves the equilibrium because a deviation of player 2 in this branch can already ensure at least payoff y (respectively at least $8-y$ for player 3).

Finally, every branch ending up in $v_{x,y}$ has payoff $(x, y, 8-y)$ so (σ, s_0) is in fact an equilibrium in $\mathcal{G}$ with the same value.

To conclude, we can divide every terminal reward for players 2 and 3 by 8, so that every final state satisfies $\phi'(s) \in \{0,1\}^{\mathsf{Agt}}$. By linearity, every 1-maximal Nash equilibrium in the original game is a Nash equilibrium in $\mathcal{G}'$ with average payoffs for players 2 and 3 divided by 8. □

We conclude with the following corollary:

Corollary 4.24. *The existence of a* 1*-maximal Nash equilibrium in a* 3*-player game with qualitative terminal rewards is undecidable.*

Safety games

We explain now how the previous proof can also be transformed to prove the undecidability of the constrained existence of a Nash equilibrium in a safety game.

This result is interesting since [SS01] shows that there always exists a Nash equilibrium in safety games (called stay-in-a-set games), without constraints. We must remark that the existence result is established for action-visible games only. Nevertheless, the authors notice that the constructed strategies require relatively low memory, as players only have to remember the set of players that already lost their objective. This information does not depend on the played actions, and when encoded inside the current state, the authors remark that memoryless Nash equilibria are proven to exists. Therefore we conclude that the action-visible hypothesis can in fact be omitted.

The following result comes in contrast:

Corollary 4.25. *The existence of a Nash equilibrium in a* 3*-player game, with qualitative safety objectives, where player* 1 *loses almost surely, is undecidable.*

By analogy, a strategy profile losing for player 1 will be called 1-minimal.

Proof. Proposition 4.23 along with the reduction presented in Proposition 4.18 allows us to compute for every two-counter machine $\mathcal{M}$ a concurrent qualitative terminal-reachability game $\mathcal{G}$ such that $\mathcal{M}$ does not halt if, and only if, $\mathcal{G}$ has a 1-maximal Nash equilibrium.

For each player i, we can write her payoff function as a function $\phi_i^r = \mathbb{1}_{\mathsf{States}^* \cdot R_i}$ with $R_i \subseteq \mathtt{F}$.

We now define safety conditions for this arena by:

$$G_1 = \mathsf{States} \backslash R_1$$
$$\forall i \in \{2,3\}.\ G_i = \mathsf{States} \backslash \mathtt{F} \uplus R_i$$

We define reward function ϕ_i^s, for each player i, by $\phi_i^s = \mathbb{1}_{G_i^\omega \uplus G_i^*}$.

First remark that we defined all internal states as winning for all players so an infinite run is a possible Nash equilibrium for the safety game. Let us now consider the constraint $\mathbb{E}^\sigma(\phi_1^s \mid s_0) = 0$. In the following, we will say that a $\langle \sigma, s_0 \rangle$ is an 1-minimal Nash equilibrium if it is a Nash equilibrium of the safety game which satisfies the above constraint.

Let us notice that $\phi_1^s = 1 - \phi_1^r$ and for $i \in \{2,3\}$, $\phi_i^s = \mathbb{1}_{\mathsf{States}^* R_i} + \mathbb{1}_{(\mathsf{States} \backslash \mathtt{F})^\omega}$ The following analysis is mostly concerned with the term $\mathbb{1}_{(\mathsf{States} \backslash \mathtt{F})^\omega}$ that is the difference between terminal-reachability objectives and safety objectives. Based on the reduction of Proposition 4.18, we

will show that this term can be neglected, in the original profile σ, but also when considering any deviation.

- Assume σ is a 1-minimal Nash equilibrium. We have $\mathbb{E}^{\sigma}(\phi_1^r \mid s_0) = 1$ so σ is 1-maximal for the reachability objective.

  Moreover, $R_1 \subseteq \mathtt{F}$ is reached with probability 1 so $\mathbb{E}^{\sigma}(\mathbb{1}_{(\mathsf{States}\setminus\mathtt{F})^\omega} \mid s_0) = 0$. So $\forall i \in \{2,3\}$, $\mathbb{E}^{\sigma}(\phi_i^s \mid s_0) = \mathbb{E}^{\sigma}(\phi_i^r \mid s_0)$.

  For i and any deviation $\sigma_i' \in S_i$, let $\sigma' = \sigma[i/\sigma_i']$, then $\mathbb{E}^{\sigma'}(\phi_i^s \mid s_0) = \mathbb{E}^{\sigma'}(\mathbb{1}_{\mathsf{States}^* R_i} \mid s_0) + \mathbb{E}^{\sigma'}(\mathbb{1}_{(\mathsf{States}\setminus\mathtt{F})^\omega} \mid s_0) \geq \mathbb{E}^{\sigma'}(\phi_i^r \mid s_0)$. So $\mathbb{E}^{\sigma'}(\phi_i^r \mid s_0) \leq \mathbb{E}^{\sigma'}(\phi_i^s \mid s_0) \leq \mathbb{E}^{\sigma}(\phi_i^s \mid s_0) = \mathbb{E}^{\sigma}(\phi_i^r \mid s_0)$.

  We conclude that σ is a 1-maximal Nash equilibrium for the reachability objectives.

- Conversely, assume there exists a 1-maximal Nash equilibrium. Thanks to the reduction proof, we know this corresponds to an infinite run of $\mathcal{M}$. Without loss of generality, we can assume such run has infinitely many counter tests, so that the underlying Nash equilibrium enables the testing module $\widetilde{\mathcal{G}}_t$ infinitely often. This strategy profile σ makes both players 2 and 3 play uniformly at random so even if one decides to deviate, there is still a fixed positive probability $\frac{1}{4}$ to branch to submodules n_2 and eventually reach a final state. We conclude from this analysis, that for every deviation $\sigma_i \in S_i$ ($i \in \{2,3\}$), $\mathbb{E}^{\sigma[i/\sigma_i]}(\mathbb{1}_{(\mathsf{States}\setminus\mathtt{F})^\omega} \mid s_0) = 0$. Hence σ is resilient to deviations of 2 and 3 for safe objectives.

  It remains to check that player 1 has no incentive to deviate, so we have to carefully look at the states where several allowed actions are given to her. Such constructions happen only in modules $\mathcal{R}_k$ and $\mathcal{D}$, where player 1 can play actions a or b. However, since σ encodes a correct simulation of $\mathcal{M}$, we know by construction that players 2 and 3 always play action c concurrently. Hence, a deviation for player 1 is not profitable, since it always yield the same payoff 1, for the reachability objective ϕ_1^r, or 0 for the safety objective ϕ_1^s.

  We conclude that the strategy profile σ is also a 1-minimal Nash equilibrium. □

Unconstrained problem revisited

The previous game $\mathcal{H}$ used to get rid of 1-maximality condition in Corollary. 4.22 can be seen as a zero-sum 2-player game, where player 1 has a terminal reachability objective and second player has the complementary safety objective.

We introduce a family of decision problems, where we allow a given number of terminal reachability and several safety objectives:

Definition 4.26. For $w \in \{R, S\}^*$, we define the Nash equilibrium decision problem for w-qualitative objectives by:

<u>QNE(w)</u>

INPUT: A $|w|$-player game $\mathcal{G}$ such that for all $i \in [1, |w|]$,

- If $w[i] = R$, player i has a qualitative terminal reachability objective: there exists $G_i \subseteq \mathtt{F}$ such that $\Phi_i(r) = \mathbb{1}_{r \in \mathsf{States}^* G_i}$.

- If $w[i] = S$, player i has a qualitative safety objective: there exists $G_i \subseteq \mathsf{States}$ such that $\Phi_i(r) = \mathbb{1}_{r \in G_i^\omega}$.

R \ S	0	1	≥ 2
0	**Ensured** (Stay-in-a-set games)		
1	**Ensured** (MDP)	??	
≥ 2	??	**Undecidable**	

Table 4.3 – Summary of the decidability status of exact Nash Equilibria in games with several agents with qualitative reachability or safety objectives.

QUESTION: Whether there exists a Nash equilibrium for $\mathcal{G}$.

In the previous proof of Corollary. 4.25, we replaced reachability objectives of all players by safety objectives. We argue that this replacement can be done one by one for each player, with the same proof. We conclude that we can build games $\mathcal{G}_{\mathcal{M}}$ for any $\mathcal{M}$, that have reachability objectives for player 1 (resp. 1 and 2) and safety objectives for players 2 and 3 (resp. 3), such that $\mathcal{G}_{\mathcal{M}}$ admits a 1-maximal Nash equilibrium if, and only if, $\mathcal{M}$ does not halt.

Mixing at least one reachability player with a safety player allows us to apply the same reduction as in Corollary. 4.22, to derive the following result:

Corollary 4.27. *Both problems* $\text{QNE}(RS^2)$ *and* $\text{QNE}(R^2S)$ *are undecidable.*

As a comparison, two particular classes of decision problems are trivial, since Nash equilibria always exist:

- Games with at most one reachability/safety player can be seen as Markov decision processes. Existence of an optimal policy is ensured, which corresponds to a Nash equilibrium.
- When only safety objectives are at stake, we can rephrase the main existence results of [SS01] on stay-in-a-set games.

Theorem 4.28. *For any* $n \leq 1$*, the decision problem* $\text{QNE}(R^n)$ *is trivial (always true). For any* $n \in \mathbb{N}$*, the decision problem* $\text{QNE}(S^n)$ *is trivial (always true) hence decidable.*

A summary of these qualitative results is depicted in Table 4.3. Notice two cases remain open:

- $\text{QNE}(RS)$ contains problems such as whether the value of a zero-sum reachability game has 0-optimal strategy. As $\mathcal{H}$ does not satisfy this property, we can already say that the decision class is not trivial. Moreover, the class contains instances of games that are not zero-sum.
- For $n \geq 2$, $\text{QNE}(R^n)$, no examples of instances (games) without Nash equilibria are known.

4.3.3 Summary

In this chapter we have shown the undecidability of the existence of a constrained Nash equilibrium in a three-player concurrent game with terminal-reachability or safety objectives.

Several similar results were shown in the literature, for example by [UW11a], where the reduction is made for 14 players in a turn-based setting with arbitrary (relative integers) terminal rewards. Variants of the reduction are reducing the problem to the existence of a finite (arbitrary) memory Nash equilibrium. On the other side, our proof exploits intensively the concurrent aspects of the framework, as long as the action-invisibility property, to lower the undecidability bound to three players and qualitative objectives.

This lets open the two-player positive rewards case, where little is known. In fact, even the existence of Nash equilibria in such games is an open problem: it was believed until recently that there are two-player games with nonnegative terminal rewards having no Nash equilibrium [CJM04, UW11a], but the proposed example was actually wrong (as explained in Section 3.6.4). On the one hand, if one can find such a game with no Nash equilibrium, then Corollary. 4.22 extends to nonnegative terminal-reachability games, and possibly to qualitative terminal-reachability games.

On the other hand, the existence of exact Nash equilibria remains open, as classical proof techniques based on fixed point theorem fail to apply. As a matter of fact, terminal reachability cannot preserve continuity of payoff functions when cyclic behaviours occur. A classical example of such discontinuity is exhibited in game $\mathcal{H}$, with strategy patterns consisting in waiting more and more in the internal state. Such phenomena are usually tackled by introducing relaxed equilibrium notions, as discussed in the next chapter.

Chapter 5

Games that almost-surely terminate

In the previous chapter, we have seen that deciding the existence of an arbitrary Nash equilibrium is undecidable in the general case. Moreover, there exist games without Nash equilibria, as soon as we consider three players and at least one reachability and one safety qualitative objective.

In the case of non-negative reachability objectives, the decidability status is still open, however undecidability can still be derived when looking at computability questions, since the constrained problem remains undecidable.

Among the causes of undecidability, we may invoke:

- The arbitrary precision for the strategies and payoffs.
- The arbitrary memory for the strategies.

In this chapter, we focus on a relaxed notion of Nash equilibrium, with limited precision. Such a relaxation ensures existence, even under memoryless strategies. As a consequence, we are able to develop efficient computation techniques, and bring back decidability for this model. We are mainly focusing on games with terminal-reachability objectives, with possibly negative rewards, although the presented results may be generalized to other objectives.

The chapter is divided into three sections: first of all we study a restriction of the strategy space to particular memoryless strategies, that ensures the existence of a "stable" point in the restricted space. Then, we characterize properties of such profiles, against arbitrary strategies, and interpret it as a particular notion of equilibrium. Finally, we explore computational questions on the newly introduced equilibrium notion.

5.1 Avoiding cycling behaviours

The classical approach for the proof of equilibrium profiles consists in defining a *best-response* function, that maps each strategy profile σ, to a set of memoryless strategy profiles where each player has an optimal value against the rest of the players in σ. As we can characterize the equilibrium profiles as fixed points of this best-response, it is sufficient to prove the existence of such fixed point, thanks to some regularity properties of the function. Usually, such regularity properties are derived from structural properties of the game, that ensures the run to eventually terminate. We can cite for example the following cases:

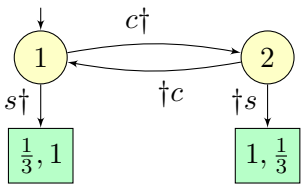


Figure 5.1 – The first player to quit the cycle loses

- The existence of exact Nash equilibria in the one-shot case, of the original work of [Nas50]. Termination is ensured after one round. The existence of Nash equilibria in surely (within a fixed number of steps) eventually terminating games can be proven the same way, or by induction.

- For qualitative safety objectives, [SS01] showed that exact Nash equilibria exist, with linear memory. The key argument consists in bounding from below the probability for any player to lose, whenever players cannot agree on a run that stays safe for all of them.

- In [CJM04], a discounted version of terminal reachability objectives is considered. The discounted factor λ can be interpreted as a probability to finish the game, with probability λ and reward 0, at each step. In this context, the authors showed the existence of a relaxed notion of Nash equilibrium in terminal-reachability games, namely ε-Nash equilibrium, where deviations may only improve the payoffs by at most $\varepsilon > 0$.

In general, the best-response function is not continuous when considering terminal reachability objectives (see later Remark 5.11 on page 63), as the probability to not terminate the game can be arbitrarily close to 0, as seen on the hide-or-run game of Figure 1.2. As opposed to [CJM04], where termination is forced from the game structure, we explore here a dual approach, by considering a restriction on strategies, that enforces players to leave cycles of the game, hence terminating.

Example 5.1. *Consider Figure 5.1 which displays an example of a turn-based game, where state $i \in \{1, 2\}$ is controlled by player i. The number labelling each node corresponds to the player controlling the state. She can decide whether to stop (action s) or to continue (action c) playing the game. However, each player has an incentive to wait for the other player to terminate the game. Again, the payoff is $(0, 0)$ if the play does not reach a terminal state. This game has pure Nash equilibria: for instance, the memoryless strategy profile where player 1 plays c and player 2 plays s is an equilibrium, with payoff $(1, 1/3)$. Another solution concept would allow a trade-off between players who will commit a fixed probability each to exit the game (for example $\varepsilon > 0$). In general, such a trade-off is not a Nash equilibrium as the other player can change her mind (and continue to play c to yield payoff 1).*

We fix for the rest of this section a stochastic concurrent game with terminal-reachability payoffs $\mathcal{G} = (\mathcal{A}, s_0, (\Phi_i)_{i\mathsf{Agt}})$, with $\mathcal{A} = (\mathsf{States}, \mathsf{Agt}, \mathsf{Act}, (\mathsf{Allow}_i)_{i\in\mathsf{Agt}}, \mathsf{Tab})$

5.1.1 Non-cycling games

Definition 5.2. A state s of $\mathcal{A}$ is said *cycling* if there exists a mixed strategy profile $\sigma \in \mathbb{S}$ such that no single player can enforce (by deviating) reaching a final state, that is:

$$\forall i \in \mathsf{Agt}\ \forall \sigma'_i \in \mathbb{S}_i,\ \mathbb{P}^{\sigma[i/\sigma'_i]}(\mathsf{States}^*\mathtt{F} \mid s) = 0.$$

The arena $\mathcal{A}$ (and by extension, the game $\mathcal{G}$) is said *cycle-free* if it contains no cycling state.

Computing the set of cycling states can be done in polynomial time, for example by computing the set of states that are almost-surely winning for the first player, in the following turn-based zero-sum qualitative safety game where:

- Player 1 suggests an action profile $\sigma(s)$ from any state s;
- Player 2 agrees or starts a deviation $\sigma_i(s)$ for an agent i that is now fixed;
- The game continues to the next state determined by both action propositions.
- Player 2 has a qualitative reachability objective equal to $\mathtt{F}$.

Moreover, this game is determined by pure memoryless strategies, so we can assume in the previous definition that $\sigma, \sigma' \in M$.

We further notice that from any cycling state, there is a Nash equilibrium with payoff zero for all the players (playing profile σ from the definition). They are therefore somehow pathological behaviours, that we may want to remove. This is formalized as follows:

Proposition 5.3. *Assume $\mathcal{G}$ contains a final state with payoff profile 0^{Agt}. One can construct a new transition function $\widetilde{\mathsf{Tab}}$ such that the resulting game $\tilde{\mathcal{G}}$:*

- *has the same set of states, final states, rewards and allowed actions,*
- *is cycle-free,*
- *and has "fewer equilibria" than $\mathcal{G}$, from any state s: for any $\tilde{\sigma} \in \mathbb{S}$, there exists $\sigma \in \mathbb{S}$ such that:*

$$\mathbb{E}^{\tilde{\sigma}}_{\tilde{\mathcal{G}}}(\phi \mid s) = \mathbb{E}^{\sigma}_{\mathcal{G}}(\phi \mid s)$$

and

$$\forall i \in \mathsf{Agt}\ \forall \sigma'_i \in \mathbb{S}_i.\ \mathbb{E}^{\sigma[i/\sigma'_i]}_{\mathcal{G}}(\phi \mid s) = \mathbb{E}^{\tilde{\sigma}[i/\sigma'_i]}_{\tilde{\mathcal{G}}}(\phi \mid s)$$

Notice that since only the transition function is modified, strategies in $\mathcal{G}$ and $\tilde{\mathcal{G}}$ coincide, which allows us to consider $\tilde{\sigma}[i/\sigma'_i]$ as a strategy profile of $\tilde{\mathcal{G}}$. Moreover, this proposition implies that any Nash equilibrium of $\tilde{\mathcal{G}}$ can be converted into an equilibrium of $\mathcal{G}$ with the same payoff. The theorem is stated in a more general context in the hope to be applied to later notions of equilibria. In particular, keeping the same strategy space allows us to keep more easily further relations between a strategy profile and its allowed deviations.

Proof. Let $f \in \mathtt{F}$ such that $\phi(f) = 0^{\mathsf{Agt}}$ and $C = \{s \in \mathsf{States} \mid s \text{ cycling in } \mathcal{A}\}$.

We construct $\widetilde{\mathsf{Tab}}$ from Tab by immediately ending the game to state f from any cycling state. More precisely, for every $A \in \mathsf{Act}^{\mathsf{Agt}}$, and $s \notin \mathtt{F}$: $\widetilde{\mathsf{Tab}}(s, A) = f$ if $s \in C$;
$\widetilde{\mathsf{Tab}}(s, A) = \mathsf{Tab}(s, A)$ otherwise
We can easily see that the underlying defined arena has the same set of final states, and is cycle-free.

Let $\sigma^C \in \mathbb{M}$ be a (partial) stationary strategy profile in $\mathcal{G}$ which allows to stay within C (and hence prohibits reaching a final state from every $s \in C$, even under single-player deviations). This is possible since C is the set of cycling states of $\mathcal{A}$.

We prove that

$$\forall \tilde{\sigma} \in \mathbb{S}.\ \mathbb{E}^{\tilde{\sigma}}(\phi \mid h) = \mathbb{E}^{\sigma}(\phi \mid h)$$

whenever

- $last(h) \in \mathtt{F}$, since we are considering the same reward function, or
- $last(h) \in C$, since both sides are equal to 0^{Agt}.

Let $s \in \mathsf{States}$ a fixed state. Moreover, we assume that $s \in \mathsf{Int}$ (otherwise the result is immediate). We decompose all maximal runs starting from s into the following disjoint union:

$$s \cdot (\mathsf{Int}^{\omega} \uplus \mathsf{Int}^* \cdot \mathtt{F}) = s \cdot (\mathsf{Int} \backslash C)^{\omega} \uplus \biguplus_{h \in s \cdot (\mathsf{Int} \backslash C)^* \cdot \mathtt{F}} h \uplus \biguplus_{h \in s \cdot (\mathsf{Int} \backslash C)^* \cdot C} h \cdot (\mathsf{Int}^{\omega} \uplus \mathsf{Int}^* \cdot \mathtt{F})$$

Since Tab and $\widetilde{\mathsf{Tab}}$ functions coincide on $\mathsf{Int} \backslash C$, we can show by induction that for any $n \in \mathbb{N}$,

- If $A_n = s \cdot (\mathsf{Int} \backslash C)^n \cdot (\mathsf{Int}^{\omega} \uplus \mathsf{Int}^* \cdot F)$, then $\mathbb{P}^{\sigma}_{\mathcal{G}}(A_n) = \mathbb{P}^{\tilde{\sigma}}_{\tilde{\mathcal{G}}}(A_n)$. $(A_n)_n$ is a decreasing sequence whose limit equals $s \cdot (\mathsf{Int} \backslash C)^{\omega}$, hence $\mathbb{P}^{\sigma}_{\mathcal{G}}(s \cdot (\mathsf{Int} \backslash C)^{\omega}) = \mathbb{P}^{\tilde{\sigma}}_{\tilde{\mathcal{G}}}(s \cdot (\mathsf{Int} \backslash C)^{\omega})$;
- For any $h \in s \cdot (\mathsf{Int} \backslash C)^n \cdot \mathtt{F}$, $\mathbb{P}^{\sigma}_{\mathcal{G}}(h) = \mathbb{P}^{\tilde{\sigma}}_{\tilde{\mathcal{G}}}(h)$;
- For any $h \in s \cdot (\mathsf{Int} \backslash C)^n C$, let $A = h \cdot (\mathsf{Int}^{\omega} \uplus \mathsf{Int}^* \cdot F)$, then $\mathbb{P}^{\sigma}_{\mathcal{G}}(A) = \mathbb{P}^{\tilde{\sigma}}_{\tilde{\mathcal{G}}}(A)$ (or $\mathbb{P}^{\sigma}_{\mathcal{G}}(h) = \mathbb{P}^{\tilde{\sigma}}_{\tilde{\mathcal{G}}}(h)$ when using usual shorthand notations).

We conclude by applying the total probability formula:

$$\mathbb{E}^{\sigma}_{\mathcal{G}}(\phi \mid s) = \sum_{h \in s \cdot (\mathsf{Int} \backslash C)^* \cdot \mathtt{F}} \underbrace{\mathbb{E}^{\sigma}_{\mathcal{G}}(\phi \mid h)}_{\mathbb{E}^{\tilde{\sigma}}_{\tilde{\mathcal{G}}}(\phi \mid h)} \cdot \underbrace{\mathbb{P}^{\tilde{\sigma}}(h)}_{\mathbb{P}^{\sigma}(h)} + \sum_{h \in s \cdot (\mathsf{Int} \backslash C)^* \cdot C} \underbrace{\mathbb{E}^{\sigma}_{\mathcal{G}}(\phi \mid h)}_{\mathbb{E}^{\tilde{\sigma}}_{\tilde{\mathcal{G}}}(\phi \mid h)} \cdot \underbrace{\mathbb{P}^{\tilde{\sigma}}(h)}_{\mathbb{P}^{\sigma}(h)} = \mathbb{E}^{\tilde{\sigma}}_{\tilde{\mathcal{G}}}(\phi \mid s)$$

Same reasoning applies to $\tilde{\sigma}[i/\sigma'_i]$ for any $i \in \mathsf{Agt}$ and any deviation $\sigma'_i \in \mathbb{S}_i$. □

Thanks to this proposition, we restrict our analysis to the case of cycle-free games. In this context, we aim at developping an equilibrium concept that is ensured for cycle-free games, and that can be later extended to arbitrary games.

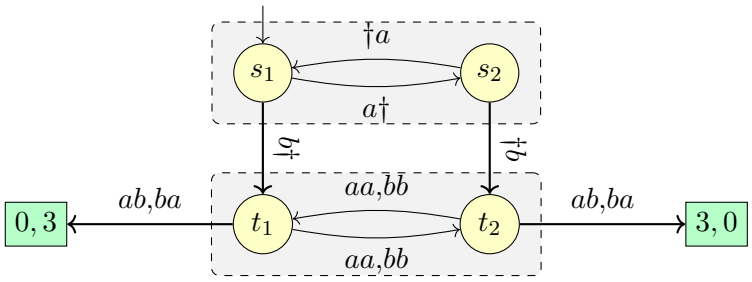


Figure 5.2 – Example of an arena with some exiting actions

5.1.2 Strong components

The main argument of our existence theorem relies on the structure of the strategy profiles, which can be forced to terminate the game, even in the presence of deviations. We define in this section a set of *constraints* we impose on our strategies. These constraints should be tight enough for the game to terminate, thus implying the existence theorem of a *stable* profile, but should also be loose enough to allow "interesting" deviations.

Definition 5.4. Let C be a non-empty set of non-terminal states of $\mathcal{A}$, and $\sigma \in \mathbb{M}$ be a stationary strategy profile. We say that σ stabilizes C if for every $s \in C$, for every $s' \in \mathsf{States}$, $\mathbb{P}^{\sigma}(\mathsf{States}^* \cdot s' \mid s) > 0$ if, and only if, $s' \in C$. When such a profile exists for C, we say that C is a *strong component*, and write SC the set of strong components.

Notice that for defining the stabilization property, one could equivalently require the probability to be equal to 1, as C would be a recurring set under $\mathbb{P}^{\sigma}$.

Definition 5.5. Let $C \in \mathrm{SC}$ be a strong component, and $s \in C$. An action $a \in \mathsf{Act}$ is an *exiting action* from C for a state s and player i if there exists $\sigma \in \mathbb{S}$ that stabilizes C such that:

$$\mathbb{P}^{\sigma[i/(s \mapsto a)]}\left(s \cdot (\mathsf{States} \setminus C)\right) > 0.$$

We set $\mathrm{Exit}(C) = \{(a, i, s) \mid a \text{ is an exiting action from } C \text{ for a state } s \text{ and player } i\}$.

As an illustration, let us consider Figure 5.2, whose game has two strong components:

$$\mathrm{Exit}(\{s_1, s_2\}) = \{(b, 1, s_1), (b, 2, s_2)\} \quad \mathrm{Exit}(\{t_1, t_2\}) = \{a, b\} \times \{1, 2\} \times \{t_1, t_2\}$$

5.1.3 Fixed point analysis

We now restrict the set of strategy profiles in which we search for equilibria. Under this restriction, we will show that each play almost-surely reaches a final state, which will provide the main argument for the existence of a "stable strategy profile", that is yet to be defined.

For the rest of this section, we assume that $\mathcal{A}$ is cycle-free, hence for any $C \in \mathrm{SC}$, $\mathrm{Exit}(C) \neq \varnothing$.

Definition 5.6. Let $\varepsilon > 0$. For every strong component $C \in \mathrm{SC}$, we define the set of (ε, C)-*exiting stationary strategy profiles* as follows:

$$\Delta_{\varepsilon}(C) = \{\sigma \in \mathbb{M} \mid \forall (a, i, s) \in \mathrm{Exit}(C)\ \sigma_i(a \mid s) \geq \varepsilon\}$$

We also let $\Delta_{\varepsilon} = \bigcap_{C \in \mathrm{SC}} \Delta_{\varepsilon}(C)$.

Note that, to be properly defined (non-empty), Δ_ε requires the assumption that the game arena is cycle-free:

Lemma 5.7. *For all $\varepsilon \leq \frac{1}{\sharp[\mathsf{Act}]}$ and $\mathcal{A}$ cycle-free, it holds $\Delta_\varepsilon \neq \varnothing$.*

Proof. Consider the stationary strategy profile σ^u which makes each player play uniformly at random over the set of allowed actions, at each state.

For any $C \in \mathrm{SC}$, since $\mathrm{Exit}(C)$ is non-empty, this strategy profile is in $\Delta_\varepsilon(C)$. Hence $\sigma^u \in \Delta_\varepsilon$. □

Under a strategy profile of Δ_ε, almost-sure reachability of a final state is ensured, but even more precisely, the probability to reach a final state, after a finite number of steps, is bounded from below by a positive constant:

Proposition 5.8. *Assume $\varepsilon > 0$. Then there exist $0 < p < 1$ and $k \in \mathbb{N}$ such that for every $\sigma \in \Delta_\varepsilon$, for every $s \in \mathsf{States}$, for every $n \geq 0$, $\mathbb{P}^\sigma(\mathsf{States}^{k\cdot n} \cdot \mathtt{F}^\omega \mid s) \geq 1 - p^n$.*

We remark that under different hypothesis, the conclusion of this proposition is similar to the one of [SS01, lemma 2.1, p483] for stay-in-a-set games. The proof is a direct consequence of the following lemma:

Lemma 5.9. *Let $\varepsilon > 0$ and $X \subseteq \mathsf{States}\backslash\mathtt{F}$. For every $s \in X$, there exists $p_X^s > 0$ such that for every $\sigma \in \Delta_\varepsilon$, $\mathbb{P}^\sigma(X^{\leq\sharp[X]} \cdot \overline{X} \mid s) \geq p_X^s$.*

Proof. Notice first that for any $\sigma \in \mathbb{M}$, the absence of memory ensures that $\mathbb{P}^\sigma(X^{\leq\sharp[X]} \cdot \overline{X} \mid s) > 0$ if, and only if, $\mathbb{P}^\sigma(X^* \cdot \overline{X} \mid s) > 0$.

Assume that this were not the case: that is to say, for any k, there exists $\sigma^k \in \Delta_\varepsilon$ such that

$$\mathbb{P}^{\sigma^k}(X^{\leq\sharp[X]} \cdot \overline{X} \mid s) < \frac{1}{k}.$$

One can assume by extraction that σ^k converges to some $\sigma \in \Delta_\varepsilon$ (Δ_ε is compact).

By continuity, we get

$$\mathbb{P}^\sigma(X^{\leq\sharp[X]} \cdot \overline{X} \mid s) = 0$$

Hence, $\mathbb{P}^\sigma(X^* \cdot \overline{X} \mid s) = 0$.

Let $s_0 \in X$ be such that $\mathbb{P}^\sigma((X^* s_0)^\omega \mid s) > 0$, that is to say one of the states that can appear infinitely often with positive probability under σ from s (it is in X due to $\mathbb{P}^\sigma(X^* \cdot \overline{X} \mid s) = 0$). We consider $C \subseteq X$ minimal subset containing s_0 and such that if $s_1 \in C$ and $\mathbb{P}^\sigma(s_1 s_2 \mid s_1) > 0$ then $s_2 \in C$. One can verify that such a set is a strong component stabilized by σ. However, $\sigma \in \Delta_\varepsilon$ hence a contradiction (by definition of Δ_ε). □

Our proof will rely on the following well-known fixed-point theorem, that we will apply to a well-adapted set of strategy profiles.

Theorem 5.10 ([Kak41])**.** *Let X be a non-empty, compact and convex subset of some Euclidean space. Let $f\colon X \to 2^X$ be a set-valued function on X with a closed graph and the property that $f(x)$ is non-empty and convex for all $x \in X$. Then f has a fixed point.*

A Nash equilibrium σ can be characterized as a strategy profile such that for each player i, σ_i lies in the set of the *best-response strategies* against the other players, denoted by $\mathrm{BR}_i(\sigma)$. By extension, the *best-response function*, which returns the profile of sets of best-responses for a given strategy profile, is written BR and its possible fixed points correspond to the Nash equilibria of the game ([Nas50]).

Remark 5.11. *Nevertheless, for games over graphs, continuity of this best-response function is not ensured. More precisely, the graph of the function is not closed. Let us consider for example the game of Figure 5.1, and write any stationary strategy profile σ in this game as the tuple $(\sigma_1(s \mid 1), \sigma_2(s \mid 2))$. Then, if one player decides to stop the game with any positive probability, the other player has all incentive to deterministically continue the game, until reaching the terminal state (almost-surely), hence:* $\mathrm{BR}((x,y)) = \{(0,0)\}$ *for every* $x, y > 0$, *where* BR *denotes the best-response function. In particular, let* $(x_n)_n$ $(y_n)_n$ *two sequences of positive probability numbers, both converging to* 0, *and let* $\sigma'^n \in \mathrm{BR}((x_n, y_n))$ *for any* n. *Then* σ'^n *equals (and converges to)* $(0,0)$. *However, if one player decides to continue the game with probability* 1, *the only way to win some positive payoff* $1/3$ *is to play the stopping action with positive probability, hence:* $\mathrm{BR}((0,0)) = \{(x,y) \mid x, y > 0\}$. *We conclude that the graph is not closed, so Theorem 5.10 cannot apply to the best-response function in this strategy profile space. This is not surprising as we know that Nash equilibria need not always exist (recall the example given in Figure 1.2).*

In the following we will see that the (standard) best-response function will fit well in our setting, with strategy restrictions:

Definition 5.12. We consider $T \subseteq \mathbb{M}$ a subset of stationary strategy profiles. Let $\mathrm{BR}_T \colon T \to 2^T$ with

$$\mathrm{BR}_T(\sigma) = \left\{ \sigma' \in T \;\middle|\; \forall i \in \mathsf{Agt}.\ \forall s \in \mathsf{States}.\ \sigma'_i \in \mathrm{argmax}_{\sigma''_i \text{ s.t. } \sigma'[i/\sigma''_i] \in T}\ \mathbb{E}^{\sigma[i/\sigma''_i]}(\Phi_i \mid s) \right\}$$

Note that $\mathrm{BR}_{\mathbb{M}}$ is the usual notion of best response function.

The fundamental application of Theorem 5.10 combined with our study of almost-sure termination is given below:

Lemma 5.13. *Assume that $\mathcal{A}$ is cycle-free and $0 < \varepsilon \leq \frac{1}{\sharp[\mathsf{Act}]}$. Then $\mathrm{BR}_{\Delta_\varepsilon}$ has a fixed point.*

Proof. We show that Theorem 5.10 applies:

- First notice that $T = \Delta_\varepsilon$ can be viewed as a non-empty compact convex subset of $\mathbb{R}^N$ where $N = \mathsf{Act} \times \mathsf{Agt} \times \mathsf{States}$. Moreover, T can be decomposed in a product of individual strategy sets for each player $T = T_1 \times \ldots T_{\sharp[\mathsf{Agt}]}$ where

  $$\forall i \in \mathsf{Agt}\ T_i = \{\sigma_i \mid \forall (a,s)\ (a,i,s) \in \mathrm{Exit}(C) \Rightarrow \sigma_i(a \mid s) \geq \varepsilon\}$$

  Hence, for every $(\sigma, \sigma') \in T^2$, and $i \in \mathsf{Agt}$, we still have $\sigma[i/\sigma'_i] \in T$.

- Let k and p be the constants appearing in the statement of Proposition 5.8. For every $n \geq 0$, we define g_n for the function assigning to every pair of strategy profiles $(\sigma, \sigma') \in T^2$ the following vector in $\mathbb{R}^{\mathsf{Agt} \times \mathsf{States}}$:

  $$\left(\sum_{j=0}^{k \cdot n} \sum_{f \in \mathsf{F}} \mathbb{P}^{\sigma[i/\sigma'_i]}\big((\mathsf{States} \setminus \mathsf{F})^j \cdot f \mid s\big) \cdot \nu_i(f) \right)_{i \in \mathsf{Agt}, s \in \mathsf{States}}$$

  Then, we obviously see that for every $(i,s) \in \mathsf{Agt} \times \mathsf{States}$, $\lim_{n \to \infty} g_n(\sigma, \sigma')_{i,s} = \mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s)$. Furthermore, as an application of Proposition 5.8, we get:

  $$|\mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s) - g_n(\sigma, \sigma')_i| \leq K \cdot p^n$$

  where $K = \max_{i \in \mathsf{Agt}, f \in \mathsf{F}} |\nu_i(f)|$. This implies that the above convergence is indeed uniform, and that $g_\infty : (\sigma, \sigma') \mapsto \left(\mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s)\right)_{i,s}$ is therefore continuous on T^2.

- Let us now show that the graph of BR_T is closed. In order to do so, we consider a converging sequence of strategy profiles $(\sigma^k)_{k>0}$ with limit σ^∞ and for each $k > 0$, $\sigma'^k \in \mathrm{BR}_T(\sigma^k)$ converging to σ'^∞. We will prove that $\sigma'^\infty \in \mathrm{BR}_T(\sigma^\infty)$. For a fixed σ', we have $\mathbb{E}^{\sigma^k[i/\sigma'_i]}(\Phi_i \mid s) \leq \mathbb{E}^{\sigma^k[i/\sigma'^k_i]}(\Phi_i \mid s)$, hence by continuity, $\mathbb{E}^{\sigma^\infty[i/\sigma'_i]}(\Phi_i \mid s) \leq \mathbb{E}^{\sigma^\infty[i/\sigma'^\infty_i]}(\Phi_i \mid s)$.

- It remains to show that $\mathrm{BR}_T(\sigma)$ is convex. We fix $i \in \mathsf{Agt}$ and show that $(\mathrm{BR}_T(\sigma))_i$ is convex hence the result. Let $0 < \lambda < 1$ and $\sigma', \sigma'' \in \mathrm{BR}_T(\sigma)$: this means that both vectors $(\mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s))_s$ and $(\mathbb{E}^{\sigma[i/\sigma''_i]}(\Phi_i \mid s))_s$ are maximal, and equal to some vector m_i. Indeed, if two different maximal vectors exist, we take the combined strategy that uses best action in each state, this new strategy is still in T_i.

  By convexity of $T = \Delta_\varepsilon$, $\sigma^\lambda = \sigma[i/\lambda \cdot \sigma'_i + (1-\lambda) \cdot \sigma''_i] \in T$, so $\forall s$, $\mathbb{P}^{\sigma^\lambda}(\mathsf{States}^*\mathtt{F} \mid s) = 1$. This implies that the payoff vector $(\mathbb{E}^{\sigma^\lambda}(\Phi_i \mid s))_s$ is the *unique* solution of the equation

$$\begin{cases} \forall f \in \mathtt{F} & \mathbb{E}^{\sigma^\lambda}(\Phi_i \mid f) = \nu_i(f) \\ \forall s \notin \mathtt{F} & \mathbb{E}^{\sigma^\lambda}(\Phi_i \mid s) = \sum_{s'} \mathsf{Tab}(s, \sigma^\lambda(s))(s') \mathbb{E}^{\sigma^\lambda}(\Phi_i \mid s') \\ & \qquad = \sum_{s'} \left[\lambda \mathsf{Tab}(s, \sigma[i/\sigma'_i](s)) + (1-\lambda)\mathsf{Tab}(s, \sigma[i/\sigma''_i](s))\right](s') \cdot \mathbb{E}^{\sigma^\lambda}(\Phi_i \mid s') \end{cases}$$

  On the other hand, m_i satisfies the following equation:

$$\begin{cases} \forall f \in \mathtt{F} & m_{i,f} = \nu_i(f) \\ \forall s \notin \mathtt{F} & m_{i,s} = \sum_{s'} \mathsf{Tab}(s, \sigma[i/\sigma'_i](s))(s') m_{i,s'} = \sum_{s'} \mathsf{Tab}(s, \sigma[i/\sigma''_i](s))(s') m_{i,s'} \end{cases}$$

  We can check that $(\mathbb{E}^{\sigma^\lambda}(\Phi_i \mid s))_{s \in \mathsf{States}} = m_i$ is a valid solution, hence the actual value, so $\sigma^\lambda_i \in \mathrm{BR}_T(\sigma)_i$. □

5.2 Equilibria under imprecise deviations

We have proven that the best-response function admits a fixed point, when restricting the strategy profiles to subset Δ_ε. This set is somehow close to the whole class of memoryless strategies, as ε goes to 0: intuitively, the "volume" of strategy spaces $\mathbb{M}$ and $\cup_{\varepsilon>0}\Delta_\varepsilon$ are equal.

However, such restriction does not allow us to prove the existence of a Nash equilibrium: for a given fixed point $\sigma \in \Delta_\varepsilon$ of BR_ε, and a deviation $\sigma'_i \in \mathbb{M}_i$ of player i, it is not guaranteed that $\sigma[i/\sigma'_i] \in \Delta_\varepsilon$. Nonetheless, there exists another deviation $\sigma''_i \in \mathbb{M}_i$ such that $\sigma[i/\sigma''_i] \in \Delta_\varepsilon$, which is close to σ'_i. More precisely, σ''_i and σ'_i can differ from each other by at most ε for each involved probability value.

Thanks to this observation, we introduce the following equilibrium concept, where deviations are required to have a certain form of *robustness*:

Definition 5.14. An *equilibrium under ε-imprecise deviations* from state s_0 is a strategy profile $\sigma \in \mathbb{S}$ such that:

$$\forall i \in \mathsf{Agt}.\ \forall \sigma'_i \in \mathbb{S}_i.\ \exists \sigma''_i \in \mathbb{S}_i \text{ s.t. } \mathbb{E}^{\sigma[i/\sigma''_i]}(\Phi_i \mid s_0) \leq \mathbb{E}^\sigma(\Phi_i \mid s_0) \text{ and } d(\sigma'_i, \sigma''_i) \leq \varepsilon$$

where $d(\sigma_i, \sigma'_i)$ is the supremum distance between the two distributions:

$$d(\sigma_i, \sigma'_i) = \sup_{h \in \mathsf{States}^+, a \in \mathsf{Act}} |\sigma(a \mid h) - \sigma'(a \mid h)|$$

The intuition behind that definition is that, to have an incentive to deviate, a player should ensure to improve her payoff, even if her deviation is perturbed by ε (this corresponds to some noise the other players can add, or to a lack of precision in playing distributions). Said differently, a deviation is only considered profitable when all the surrounding (up to a distance of ε) strategies are also profitable.

5.2.1 Restricting to memoryless deviations

The notion of equilibria under imprecise deviations has been introduced in a very general setting with arbitrarily strategies and deviations. Lemma 5.13 provides the existence of a memoryless strategy profile, that is resilient to memoryless deviations, even when perturbed by a distance ε, in a memoryless manner. In other words, previous fixed-point lemma cannot directly conclude on the existence of an equilibrium under imprecise deviations.

Intuitively, we can even wonder if we can, as in the case of a memoryless Nash Equilibria, only consider pure memoryless deviations, which are in finite number. This subsection is devoted to this question, through the proof of the following key lemma:

Lemma 5.15. *Let s_0 be a state of a stochastic concurrent game $\mathcal{G}$ with terminal-reachability payoffs. For any stationary strategy profile $\sigma \in \mathbb{M}$, it holds: σ is an equilibrium under ε-imprecise deviations from s_0 if, and only if,*

$$\forall i \in \mathsf{Agt}.\ \forall \sigma'_i \in M_i.\ \exists \sigma''_i \in \mathbb{M}_i.\quad d(\sigma'_i, \sigma''_i) \leq \varepsilon \wedge \mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s_0) \leq \mathbb{E}^{\sigma}(\Phi_i \mid s_0)$$

In other terms, it is sufficient to consider memoryless deviations when checking if a stationary strategy profile is an equilibrium under imprecise deviations.

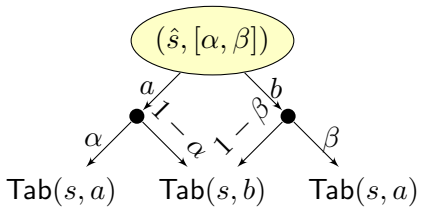
We prove this lemma by considering an intermediate $2 + 1/2$-player game to represent deviations of player i and her counter-deviations at distance ε.

Let $\mathcal{G}$ a game, σ a stationary strategy profile and $i \in \mathsf{Agt}$ a player. We write $\mathcal{G}\langle\sigma\rangle_{-i}$ for the $1+1/2$-player game obtained from $\mathcal{G}$ by assigning to all players, but player i, her strategy in σ. Note that for any $\sigma'_i \in \mathbb{S}_i$, we have $\mathbb{E}_{\mathcal{G}}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s) = \mathbb{E}_{\mathcal{G}\langle\sigma\rangle_{-i}}^{\sigma'_i}(\Phi_i \mid s)$ In the following, we are mainly interested in the possible ε-imprecise deviations of player i alone in this new game.

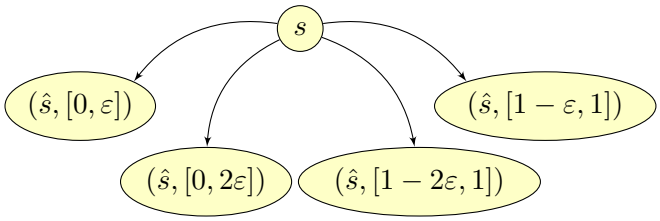
In order to make the reduction clear, we consider in the following the particular case of games where each player is allowed at most two actions. When exactly two distinct actions are allowed, they will be noted a and b. The general case will be discussed in Remark 5.20.

For a stationary profile σ, we consider the $1 + 1/2$-player game $\mathcal{G}\langle\sigma\rangle_{-i}$ as defined above (with player i alone, all other strategies being fixed) and construct a $2 + 1/2$-player turn-based game with an additional antagonistic player $\hat{i}$, whose role is to "change" the strategy of player i by a distance at most ε. Formally, for any state s where player i has two allowed actions a and b (resulting in distributions $\delta(s, a)$ and $\delta(s, b)$, resp.), we modify the game as follows:

- from s, player i is given the opportunity to move to one of the following four states: $(s, [0, \varepsilon])$, $(s, [0, 2\varepsilon])$, $(s, [1 - 2\varepsilon, 1])$ and $(s, [1 - \varepsilon, 1])$.



(a) Simple $(\hat{s}, I)$ node with I interval of $[0,1]$ played by $\hat{i}$, ensuring any distribution $p\mathsf{Tab}(s,a) + (1-p)\mathsf{Tab}(s,b)$ for any $p \in I$, hence a *deviation range* I.



(b) Replaced node s, where i can choose between 4 ranges of probabilities to play action a.

Figure 5.3 – Translation of a node s with initial allowed actions a and b.

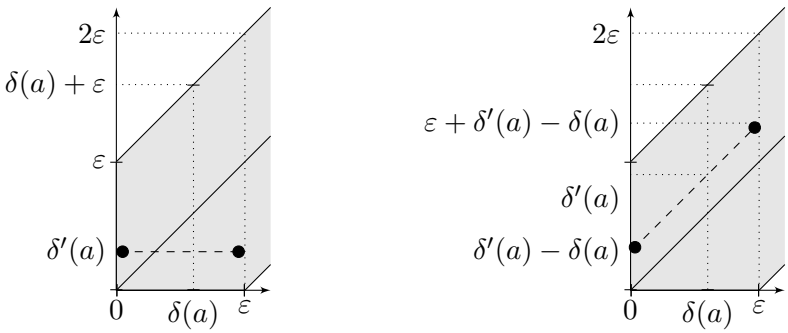


Figure 5.4 – Intuition of the construction for $\delta(a) \leq \varepsilon$: seeing $\delta(a)$ as a convex combination of 0 and ε, we obtain $\delta'(a)$ as the same convex combination of the black dots.

- from each state $(s, [\alpha, \beta])$, player $\hat{i}$ has two actions, leading to distributions $\alpha \cdot \mathsf{Tab}(s,a) + (1-\alpha) \cdot \mathsf{Tab}(s,b)$ and $\beta \cdot \mathsf{Tab}(s,a) + (1-\beta) \cdot \mathsf{Tab}(s,b)$, respectively. If player $\hat{i}$ plays action a with probability p, then the final distribution is $[p\alpha + (1-p)\beta] \cdot \mathsf{Tab}(s,a) + [(1-p)(1-\alpha) + p(1-\beta)] \cdot \mathsf{Tab}(s,b)$.

For a $1 + 1/2$-player game $\mathcal{G}$ for i, we denote by $\widehat{\mathcal{G}}^{\varepsilon}$ the previous transformation. Our aim is to provide a correspondence between (stochastic) moves of player i from s in $\mathcal{G}$, and her move from the corresponding state $\hat{s}$ in $\widehat{\mathcal{G}}^{\varepsilon}$. Our notion of correspondence is defined as follows:

Definition 5.16. Let $\sigma_i, \sigma_i' \in \mathbb{S}$ two strategies for the $1 + 1/2$-player game $\mathcal{G}$ (played by i) such that $d(\sigma_i, \sigma_i') \leq \varepsilon$, and $\hat{\sigma}$ a strategy profile in $\widehat{\mathcal{G}}^{\varepsilon}$. We say that (σ_i, σ_i') corresponds to $\hat{\sigma}$ if the following holds for any history $\hat{h}$ ending in state s of $\widehat{\mathcal{G}}^{\varepsilon}$:

$$\mathsf{Tab}(s, \sigma_i'(\pi_{\mathsf{States}}(\hat{h}))) = \widehat{\mathsf{Tab}}(s, \hat{\sigma}(\hat{h}))$$

where $\pi_{\mathsf{States}}(h)$ is the projection on the letters corresponding to the original states States.

We now make explicit the purpose of the construction by establishing a correspondence between strategies in the original game and strategies in our $2 + 1/2$-player version.

Lemma 5.17. *For any σ_i strategy of $\mathcal{G}$, there exists a strategy $\hat{\sigma}_i$ in $\widehat{\mathcal{G}}^{\varepsilon}$ for player i, such that, for any strategy σ_i' of $\mathcal{G}$ with $d(\sigma_i, \sigma_i') \leq \varepsilon$, there exists $\hat{\sigma}_{\hat{i}}$ such that (σ_i, σ_i') corresponds to $\hat{\sigma}$.*

Moreover, any pure memoryless strategy profile of $\widehat{\mathcal{G}}^{\varepsilon}$ corresponds to some pair of strategies (σ_i, σ'_i) in $\mathcal{G}$ where σ_i is pure memoryless and σ'_i is stationary.

The constructed game is a turn-based stochastic game with a quantitative terminal reachability objective, which can be interpreted as a special case of limit-average objective. Hence, thanks to a result of [LL69], such a game is determined with pure memoryless optimal strategies for both players.

As a consequence of this construction, we can infer two possible characterizations of imprecise deviations in stationary profiles:

Corollary 5.18. *The value of $\widehat{\mathcal{G}}^{\varepsilon}$ at state $\widehat{s}$ can be expressed as the following quantity on game $\mathcal{G}$:*

$$\sup_{\sigma \in M^{\mathcal{G}}} \inf_{\substack{\sigma' \in \mathbb{M}^{\mathcal{G}} \\ d(\sigma,\sigma') \leq \varepsilon}} \mathbb{E}^{\sigma'}(\Phi_i \mid s)$$

Corollary 5.19. *A stationary strategy profile $\sigma \in \mathbb{M}^{\mathcal{G}}$ in $\mathcal{G}$ is an equilibrium under ε-imprecise deviations from state s_0 if, and only if,*

$$\forall i \in \mathsf{Agt}.\ \forall \sigma'_i \in M_i^{\mathcal{G}}.\ \exists \sigma''_i \in \mathbb{M}_i^{\mathcal{G}} \text{ s.t. } \mathbb{E}^{\sigma[i/\sigma''_i]}(\Phi_i \mid s_0) \leq \mathbb{E}^{\sigma}(\Phi_i \mid s_0) \text{ and } d(\sigma'_i, \sigma''_i) \leq \varepsilon$$

Remark 5.20. *One can notice the construction of the deviation game and inferred results have been applied to states with only two allowed actions. In fact, the same reasoning can be generalized to an arbitrary number of allowed actions at the expense of an exponential blowup: player i has to announce simultaneously, for each allowed action a, if its probability in the expected distribution will be larger than ε and/or smaller than $1 - \varepsilon$. Note however that for a given fixed bound on the number of actions, the size of $\widehat{\mathcal{G}}^{\varepsilon}$ is still polynomial.*

5.2.2 Existence theorem

We finally conclude with the following existence theorem of an ε-imprecise:

Theorem 5.21. *Let $\mathcal{G}$ be a stochastic concurrent game with terminal-reachability payoffs, and let s_0 be a state of $\mathcal{G}$. For every $\varepsilon > 0$, there always exists an equilibrium under ε-imprecise deviations from state s_0.*

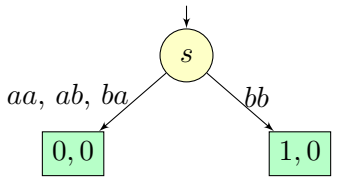
The proof consists in compiling together the several previous results:

Proof. We first consider a cycle-free game $\tilde{\mathcal{G}}$ obtained from $\mathcal{G}$ through Proposition 5.3.

Let $\varepsilon' = \min(1/\sharp\,[\mathsf{Act}]\,, \varepsilon)$. Let us consider a fixed-point $\tilde{\sigma} \in \mathbb{S}^{\tilde{\mathcal{G}}}$ of $\mathrm{BR}_{\Delta_{\varepsilon'}}$, which exists thanks to Lemma 5.13.

Such a profile $\tilde{\sigma}$ is an equilibrium under ε'-imprecise deviations from every state s, thanks to Corollary. 5.19. Since $\varepsilon' \leq \varepsilon$, this is also an equilibrium under ε-imprecise deviations.

Finally, we apply last implication of Proposition 5.3 to $\tilde{\sigma}$ to obtain an equilibrium under ε-imprecise deviations of $\mathcal{G}$. □



$\sigma_1(a \mid s) = 1$
$\sigma_2(b \mid s) = \varepsilon$

Figure 5.5 – A game with an ε-Nash equilibrium which is not an equilibrium under ε'-imprecise deviation, for any $\varepsilon' > 0$.

5.2.3 Discussions

We discuss the newly introduced notion of equilibrium under ε-deviation and compare it to ε-Nash equilibrium as introduced in [CJM04].

Both notions generalizes Nash equilibria, that is to say any Nash equilibrium is both an ε-Nash equilibrium and an equilibrium under ε-deviation.

First of all, consider the game of Figure 5.5, and the strategy profile (σ_1, σ_2): the payoff is then $(0, 0)$, and player 1 can improve her payoff by ε by playing action b from s. So (σ_1, σ_2) is an ε-Nash equilibrium but not an equilibrium under ε-imprecise deviations: any strategy at distance ε from σ_1' strictly improves the payoff of player 1.

In the previous game of Figure 5.1, the strategy profile where each player plays s with probability ε yields payoffs $1 - 2/(6 - 3\varepsilon)$ for player 1 and $1 - (2 - 2\varepsilon)/(6 - 3\varepsilon)$ for player 2 from the initial state. It is an equilibrium under ε-imprecise deviations. The only way to really improve the payoff for a player is to play with higher probability action c. But with the lack of precision, she might lose some payoff anyway. The payoff values get arbitrarily close to $2/3$ as ε goes to 0. Such an equilibrium is neither a Nash equilibrium, neither a ε-Nash equilibrium, since the pure deviation c allows an improvement of almost $1/3$.

This concludes that the two concepts are incomparable.

Remark 5.22. *For any $\varepsilon > 0$ small enough, there exists $\sigma^\varepsilon \in \mathbb{M}$ fixed point of $\mathrm{BR}_{\Delta_\varepsilon}$. As ε goes to 0, one can extract a converging sequence of $(\sigma^\varepsilon)_\varepsilon$, whose limit can be denoted by σ^0.*

When $\mathcal{G}$ is a one-shot game, one can prove that σ^0 is still a fixed point of $\mathrm{BR}_{\Delta_0} = \mathrm{BR}_{\mathbb{M}}$, that is to say σ^0 is a Nash equilibrium. Such particular Nash equilibrium is called trembling-hand equilibria (in [Sel75]) because of its extra properties. However, this converging procedure does not work anymore in the sequential games framework, as depicted again on Figure 5.1 and strategy profile where each player exits with probability ε. At the limit, strategy profile σ^0, which stays in the game almost-surely, is not a Nash equilibrium.

An interesting open question consists in restricting again the strategy space $\mathbb{M}$ in order to force convergence to a trembling-hand Nash equilibrium, if any. As a matter of fact, such proof may require additional assumptions on $\mathcal{G}$, as some games require finite memory for a Nash equilibrium to exist (see [Umm10]), so looking at an equilibrium in $\mathbb{M}$ may fail otherwise.

5.3 Computing stationary equilibria under imprecise deviations

We describe a polynomial-space algorithm for computing stationary equilibria under imprecise deviations for non-negative terminal reward games. A similar proof for Nash equilibria in

turn-based stochastic games is given in [UW11c]. We briefly describe the latter proof, which will help understand our current encoding.

The algorithm proceeds by encoding a Nash Equilibrium as an existential first-order formula over the reals, whose satisfiability can be decided in PSPACE. The formula quantifies over all stationary strategy profiles and payoffs at each state, and expresses that:

1. the strategy profile σ under consideration is properly defined;
2. the payoff in each state corresponds to the real payoff of the strategy profile;
3. for any i, player i cannot benefit from deviating in $\mathcal{G}\langle\sigma\rangle_{-i}$.

These properties cannot, *in general*, be expressed locally, but in the setting of [UW11c], one can first, non-deterministically, guess the support of the strategy. On the one hand, this allows us to compute (in linear time) the set of states from which `F` is never reached. Those states have payoff 0 for all agents, and the payoff in the other states (from which `F` is reachable with some positive probability) can be expressed as a combination of the payoff values of the successor states and the (local) strategy profile. On the other hand, we can also compute (still in linear time) the set of states that are reachable from s_0. It is easy to see that player i has an incentive to deviate if, and only if, her payoff can be increased by deviating locally from such a reachable state. Hence we can express stability of the Nash Equilibrium as a (polynomial size) conjunction of inequalities.

Another way of expressing this stability property is by saying that for any player i, s_0 should yield a payoff in the equilibrium that is larger than the optimal value $v_i(s_0)$ in the Markov decision process representing the possible deviations of player i, namely $\mathcal{G}\langle\sigma\rangle_{-i}$. Since the initial guess can be done in NPSPACE and the generated formula is of polynomial size, the whole algorithm runs in PSPACE.

In the case of equilibria under ε-imprecise deviations, we apply a similar technique but deviations are now to be considered as strategies for player i in $\widehat{\mathcal{G}\langle\sigma\rangle_{-i}}^{\varepsilon}$ against the worst strategies of player $\widehat{i}$. In fact, we want to check that s_0 has a payoff (in the equilibrium) larger for player i than the maximal value she could get by imprecisely deviating. Thanks to Corollary. 5.18, this optimal value is the same as in $\mathcal{G}_i = \widehat{\mathcal{G}\langle\sigma\rangle_{-i}}^{\varepsilon}$, denoted by $v_{\varepsilon,i}(s)$. In order to compute these values for each game $\mathcal{G}_i$, we non-deterministically compute optimal strategies for players i and $\hat{i}$. These strategies can be supposed to be pure memoryless. In order to do so, we first guess a strategy for player i in the game game $\widehat{\mathcal{G}\langle\sigma\rangle_{-i}}^{\varepsilon}$. Without knowing the exact probability values of this game (which depends on σ), we can still derive its structure since the support is known, thus we can compute the set of states for which player $\hat{i}$ can totally spoil i's payoff, that is, enforce a non-terminating run; such a run has payoff 0, which is optimal for player $\hat{i}$. We later guess a pure memoryless strategy for player $\hat{i}$ keeping in mind that $\hat{i}$ has to play such a cycling strategy from any state where she is able to. From the other states, for which player i can still ensure positive probability to terminate, the value of the game can again be expressed locally as a combination of the guessed strategy profile and the values of the successor states. As for the previous algorithm for Nash Equilibrium in $\mathcal{G}$, the optimality of both strategies can be expressed as stability by local deviations. Finally, stability by imprecise deviations in $\mathcal{G}$ consists in coding the fact that payoff in $\mathcal{G}$ for player i should be larger than the optimal value $v_{\varepsilon,i}(s_0)$.

We now make precise the result and the algorithm.

Theorem 5.23. *Let $k > 0$. Let $\mathcal{G} = (\mathcal{A}, \Phi_\nu)$ be a stochastic concurrent game with non-negative terminal rewards with $\sharp\,[\mathsf{Act}] \leq k$. Let $s_0 \in \mathsf{States}$ and $\varepsilon > 0$. For every $i \in \mathsf{Agt}$, we fix $x_i, y_i \in \mathbb{R}_+$ two real numbers. We can decide in* PSPACE *whether there is a stationary equilibrium under ε-imprecise deviations σ from s_0, such that for every $i \in \mathsf{Agt}$, $x_i \leq \mathbb{E}^\sigma(\Phi_i \mid s_0) \leq y_i$.*

Remark 5.24. *The previous theorem can be applied to compute* some *equilibria in the case of negative payoffs by considering the new payoff function $\nu' = \nu - \min \nu \geq 0$. However, $\phi' = \Phi_\nu - \min \nu$ and $\Phi_{\nu'}$ coincide only on runs that reach a final state since ϕ' assigns positive value $-\min \nu$ to non-terminating runs. A possible work-around is to first compute the cycle-free arena $\widetilde{\mathcal{A}}$ and exiting conditions Δ_ε, whose size is bounded by the number of pairs $(a, i, s) \in \mathsf{Act} \times \mathsf{Agt} \times \mathsf{States}$. Then we can apply the previous theorem on game $\langle \widetilde{\mathcal{A}}, \Phi_{\nu'} \rangle$ with the extra formula $\sigma \in \Delta_\varepsilon$. Thanks to this last constraint, we ensure that the run always terminates, thus the payoff functions coincide. Finally we conclude the computation by applying Proposition 5.3 to get back an equilibrium on $\mathcal{G}$.*

Part II

Parametrized Stochastic Systems

Chapter 6

Interaction models

In the first part, we studied systems with a fixed finite number of agents, that have in each round arbitrary interactions. Such agents may have different roles, objectives and strategies. Moreover, as noticed before, introducing partial information may lead to undecidability, therefore our previous model allows all agents to have a complete view (apart from action visibility) of the current global state.

We may argue that more realistic models should include more structural properties, namely:

- All agents, except possibly one, for the environment, a server or some other leader mechanism, should have a similar structure, entailing some symmetry for the global system.
- Interaction and communication between agents should have some particular structure.
- Agents should be aware of their own current state only, and only be able to deduce others' statuses thanks to interaction and communication.
- Even the number of initial agents should *a priori* be unknown, potentially large.

In this part, we are mainly interested in verification problems on such systems, although we keep in mind strategy synthesis as a further goal. Since interaction between agents is encoded with a particular structure, we usually refer to such systems as *protocols* (instead of games) composed of *processes* (instead of agents).

Parametrized verification: a global picture. When dealing with several identical processes, the naive approach based on building, by composition, the concrete system for a fixed number of processes then applying classical analysis techniques can be replaced by more clever techniques based on symmetry reduction, as shown in [ES96], which focuses on verification of symmetric systems.

However, as pointed out by the fourth requirement, the number of processes is not fixed *a priori*, so these techniques only provide partial answers for our systems, where the number of processes can be a *parameter*. Therefore, we switch to another setting where we want to solve verification for parametrized systems. In such a *parametrized* setting, the natural question is to characterize the set of parameter values for which the system is correct.

Although the problem may seem to be harder in general, it allows us to consider large parameter values for which we may reach *limit behaviours*. When such a limit is reached,

answering verification and synthesis questions becomes easy as it suffices to compute the answers for a finite number of parameter values up to some limit parameter.

Moreover, not only the latter approach is more general, but it might also turn out to be easier and more efficient, since large parameters may involve regularity properties in our model. For example, adding more and more agents in a system may result in a kind of saturation mechanism ensuring that each action can always be performed by at least one process. Such concepts are usually related to upward closure, as described in Section 2.4 of Chapter 2.

More formally, we are interested in determining the existence of a *cut-off* for a property φ, i.e., an integer N such that a system composed with more than N processes, ensures property φ.

Network of protocols. The presence of a leader often makes computation harder as it can ensure some additional synchronization mechanisms, so models that avoid its election may allow easier computations. Moreover, when the election mechanism can be generalized, we can imagine each agent to get its *own identity*, which usually leads to undecidable problems. For example, we can think of a system that allows each agent to encode one cell of the tape of a Turing machine, and be able to communicate (*addressing*) to its next and previous cell tape.

Processes may also be restricted to communicate within some communication graph. When topology does not correspond to a complete graph, broadcast communication with neighbours does not allow every process to receive a message. Therefore, some topologies may allow processes to earn *identities*, hence implying undecidability. However, we can usually recover decidability by allowing the communication topology to be unreliable, or to change over time as in [BFS14]. In this context, allowing reconfiguration of a message-passing communication network will usually be equivalent to considering shared register with non-atomic operations. This dual view of our protocols explain why such systems will be later abusively named *networks*.

Different means of communication. Literature on parametrized verification introduces several models that usually vary by the choice of interaction scheme between processes. In his nice survey on parametrized models [Esp14], Esparza shows that minor changes in the setting, such as the presence of a controller in the system, might drastically change the complexity of the verification problems. Nonetheless, he summarizes the communication classes in the following four groups with decreasing expressivity:

- Broadcast communication;
- Shared register with atomic operations (locking);
- *Rendez-vous* communication;
- Shared register with (only) non-atomic operations.

Intuitively, two main models exist either based on message passing, or on shared variable accesses. Differences between these classes occur when looking at their ability to ensure that an information has been transmitted (last class cannot ensure such property) and whether the system can elect a *leader* (only two first classes can do so), which means some agent is eventually differentiated from the others.

Further analysis of the relative expressiveness of some of those models can be found in [ARZ15], yielding several reductions of the verification problems for some of those classes of models.

A seminal paper on parametrized verification of such distributed systems is the work of German and Sistla [GS92]. In this work, the authors consider networks of processes all following the same finite-state automaton; the communication between processes is performed thanks to *rendez-vous* communication. Various related settings have been proposed and studied since then, which mainly differ by the way the processes communicate. Among those, let us mention broadcast communication [EFM99, DSZ10], token-passing [CTTV04, AJKR14], shared register with ring topologies [ABG15], or shared memory [EGM13, DEGM15].

Asynchronous shared-memory systems. We consider a communication model where the processes asynchronously access a shared register, and where read and write operations on this register are performed non-atomically, *i.e.* one transition can only perform a read or write operation at a time. A similar model has been proposed by Hague in [Hag11], where the behaviour of processes is defined by a pushdown automaton. The complexity of some reachability and liveness problems for shared-memory models have then been established in [EGM13] and [DEGM15], respectively. These works also consider networks with a specific distinguished process, called the *leader*, which runs a different program, from the beginning, and address the problem whether, for some number of processes, the leader can satisfy a given reachability or liveness property. In the case where there is no leader, and where processes are finite-state machines, the parametrized control-state reachability problem (asking whether one of the processes can reach a given control state) can be solved in polynomial time, by adapting the approach of [DSTZ12] for lossy broadcast protocols.

Example: network of sensors. As discussed before, such model is the weakest on the scale of communication mechanisms. For example, lack of atomicity does not allow our processes to implement mutual exclusion. However, we argue that this model still contains interesting questions.

As an illustration, let us consider a system composed of several small devices, for example sensors in a room, that report their data to a central receiver, through a wireless protocol. The wireless technology allows several communication channels (11 for example for 2.4Ghz WiFi). Because of short distance considerations, we can assume that radio resources are totally shared among sensors. The situation is depicted as a small C multi-threaded program, presented in Figure 6.1. When a sensor starts emitting on a channel i, it sends a particular message saying that channel i is in use, then the data, then another message saying that channel i is now free. A sensor usually tries to avoid sending data on a channel that is already in use, however, it has very low computation capacity and between the instant where it checks that a channel is not in use, and the instant where it effectively decides to emit, some other process may decide to transmit too, which results in data collision.

Sensors are not capable of avoiding such collisions, but can always try to change their channel in order to use all the radio resources. In this example, we may consider the following properties:

- *Coherence*: at any time, for all $i \in [0, 10]$, if $t[i] = 1$, there exists at least one thread-/process transmitting on channel i.

```
1  // An array of currently used channels
2  int t[11] = {0};
3
4  void thread() {
5    // Selected channel for current thread
6    int i = 0;
7
8    while(1) {
9      if(t[i] == 0) {
10       t[i] = 1;
11       // Actual transmission
12       t[i] = 0;
13     } else if(rand() < P) { // Branching with probability P
14       i = (i+1) % 11;
15     }
16   }
17 }
```

Figure 6.1 – Simple specification of wireless sensors (seen as threads) communicating through eleven radio channels. P is a constant between 0 and 1 describing the probability to switch to next channel if current one is already used.

- *Optimality*: all channels are eventually used.
- *Repeated optimality*: all channels are used infinitely often.

The resulting system is of course stochastic as there is some constant probability to switch to next channel if the current one is in use. Moreover, the composition of the multiple sensors is also non-deterministic, and may induce some stochastic behaviours. More formally, we consider the transcription of the program into a finite automaton, as depicted in Figure 6.2, with input alphabet $\Sigma = \{\mathbf{R}_i(d), \mathbf{W}_i(d) \mid i \in [0,10], d \in \{0,1\}\}$. A non-deterministic semantics is usually given, as in [EGM13], on a language theoretical approach, by considering the shuffling of an arbitrary number of words accepted by the process automaton, then intersected with the language of words over Σ^* that are consistent with the shared register (for example, $\mathbf{W}_i(0) \cdot \mathbf{R}_i(1)$ is not consistent).

We will adopt in the next chapter a more operational approach suited to the probabilistic setting.

Non-determinacy, fairness and qualitative stochasticity In this second part of the thesis, we further insert fairness assumptions in the model of parametrized networks with asynchronous shared memory, and consider again terminal reachability problems in this setting. There are different ways to include fairness in parametrized models: One approach is to enforce fairness expressed as a temporal-logic properties on the executions (e.g., any action that is available infinitely often must be performed infinitely often); this is the option chosen for parametrized networks with rendez-vous [GS92] and for systems with disjunctive guards (where processes can query the states of other processes) in [AJK16]. Another equivalent

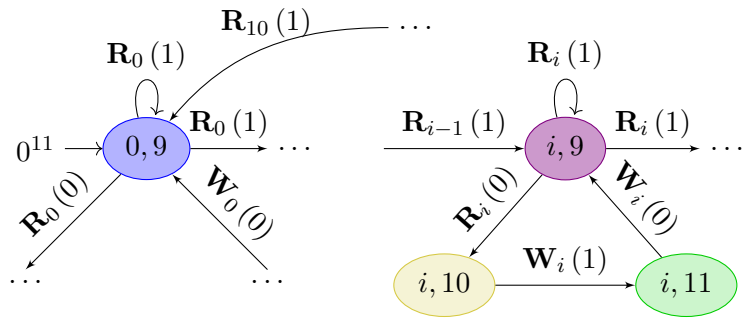


Figure 6.2 – Underlying (simplified) automaton for one thread of program of Figure 6.1. Each possible state of the thread is modeled by a node containing the current value of local variable i and current line number.

view consists in equipping our networks with a stochastic scheduler that, at each step of the execution, assigns the same probability to the available actions of all the processes. From a high-level perspective, both forms of fairness are similar.

However, notice that expressing fairness via temporal logic has to be done with care, as some formulae may allow very regular patterns (e.g., round-robin execution of the processes), whereas the stochastic approach always implies considering all possible interleaved runs with probability 1.

Chapter 7

Parametrized register protocols

In this section, we define different categories of parametrized register protocols and provide an operational semantics based on a transition system. This first semantics will allow us to consider the reachability/safety problem, as studied in [EGM13] or [DEGM15], namely reachability under a non-deterministic scheduler. We will then introduce a stochastic scheduler by introducing probabilities on distribution and introduce stochastic properties to be studied.

First of all, we define the operations that can be operated on the shared register.

Definition 7.1 (Operations over a set)**.** Let D be a set. We denote by $\mathrm{Op}\,(D)$ the set of *register operations*, which are partial functions from D to D.
The exact definition domain of an operation $\mathbf{f} \in \mathrm{Op}\,(D)$ is denoted by $\mathrm{dom}\,(\mathbf{f}) \subseteq D$.
Given two operations $\mathbf{f}, \mathbf{f}' \in \mathrm{Op}\,(D)$, the sequential composition $\mathbf{f}\cdot\mathbf{f}'$ is defined on $\mathrm{dom}\,(\mathbf{f} \cdot \mathbf{f}') = \mathrm{dom}\,(\mathbf{f}) \cap \mathbf{f}^{-1}(\mathrm{dom}\,(\mathbf{f}'))$ by $d \mapsto \mathbf{f}'(\mathbf{f}(d))$.
For $d \in D$, $\mathbf{W}\,(d)$ and $\mathbf{R}\,(d)$ represent the following particular partial functions:

$$\mathbf{W}\,(d) : \begin{cases} \mathrm{dom}\,(\mathbf{W}\,(d)) = D \\ d' \mapsto d \end{cases} \qquad \mathbf{R}\,(d) : \begin{cases} \mathrm{dom}\,(\mathbf{R}\,(d)) = \{d\} \\ d \mapsto d \end{cases}$$

We denote by $\mathrm{Op}_{R,W}\,(D)$ the set of such register operations and $\mathrm{Op}_{R,W}\,(D)^*$ its closure by sequential composition.

Here $\mathbf{W}$ means *write* in the register a new value, no matter the current content of the register, while $\mathbf{R}$ means *read* the content of the shared register. This last operation can only be triggered if the correct value is already stored in the register.

Intuitively, we will define finite state automata over the alphabet $\mathrm{Op}\,(D)$. A protocol will be given as two automata, the leader and the contributor, the last one being possibly duplicated several times:

Definition 7.2. A *register protocol* is given by $\mathcal{P} = (Q_l, Q_c, D, d_0, q_l, q_c, T_l, T_c)$, where

- Q_l is a finite set of control locations for the leader;
- Q_c is a finite set of control locations for the contributors, disjoint from Q_l;
- D is a finite alphabet of data for the shared register, with initial data d_0;
- $q_l \in Q_l$ (resp $q_c \in Q$) is an initial state for the leader (resp contributors);
- $T_l \subseteq Q_l \times \mathrm{Op}\,(D) \times Q_l$ is a finite set of transitions of the leader;

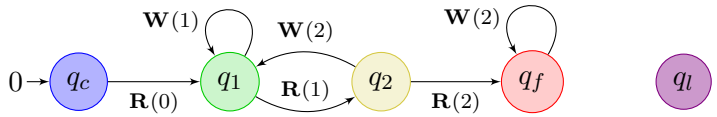


Figure 7.1 – Example of a leaderless non-atomic register protocol with $D = \{0, 1, 2\}$ and initial value 0. q_l will usually be ommited for leaderless register protocols.

- $T_c \subseteq Q_c \times \mathrm{Op}\,(D) \times Q_c$ is a finite set of transitions of the contributors.

Without further explicit precision, we will consider in the rest of the chapter a fixed register protocol $\mathcal{P} = (Q_l, Q_c, D, d_0, q_l, q_c, T_l, T_c)$.

Definition 7.3. We say that $\mathcal{P}$ is

- *atomic* if $T_c \subseteq Q_c \times \mathrm{Op}_{R,W}\,(D)^* \times Q_c$ and $T_l \subseteq Q_l \times \mathrm{Op}_{R,W}\,(D)^* \times Q_l$;
- *non-atomic* if moreover $T_c \subseteq Q_c \times \mathrm{Op}_{R,W}\,(D) \times Q_c$;
- *leaderless* if $T_l = \varnothing$.

Notice first that the leaderless concept is only relevant when dealing with non-atomic protocols, as atomicity allows the election of a leader (start with extra data value "not elected" then first contributor to play jumps to q_l by writing the actual initial value). Then, a non-atomic register protocol is a particular case of atomic protocol, which is also a particular case of register protocol.

Because of the monotonous properties we will describe later, we will be mainly interested in non-atomic protocols. Hardness results will be developed in the leaderless non-atomic case.

The number of involved data values in any atomic register protocol is bounded by $\sharp\,[T]$. This allows us to legitimately define the size of any atomic register protocol $\mathcal{P}$ by:

$$|\mathcal{P}| = \sharp\,[Q_c] + \sharp\,[Q_l] + \sharp\,[T_c] + \sharp\,[T_l]$$

A graphical representation of a leaderless non-atomic protocol is depicted in Figure 7.1.

Remark 7.4. *Notice we do not require our protocol to be complete. We say a protocol is* complete *if for each state and each register value, there exists an operation defined for this value, namely* $\forall q \in Q_c \ \bigcup_{(q,\mathbf{f},q')\in T_c} \mathrm{dom}\,(\mathbf{f}) = D$ *and* $\forall q \in Q_l \ \bigcup_{(q,\mathbf{f},q')\in T_l} \mathrm{dom}\,(\mathbf{f}) = D$.

7.1 Non-deterministic transition system

We now present the non-deterministic semantics of distributed systems associated with our register protocols. We consider the *asynchronous* composition of the automaton of the leader, with several copies of the contributor automaton, the number of copies is not fixed *a priori* and can be seen as a parameter.

We immediately consider the following abstraction: for each contributor state q, we only keep track of the number of processes in this particular state q. Namely, a configuration will store a multiset of Q_c, alongside the current state of the leader and the current memory data. This abstraction will reveal to be conceptually sound, as the copies will be indistinguishable, and will allow us to later consider easier comparison relations over configurations.

Definition 7.5 (Transition system associated with $\mathcal{P}$)**.** We define the transition system $(\Gamma, \rightarrow)$ in the following way:

- $\Gamma = Q_l \times \mathbb{N}^{Q_c} \times D$ is the set of configurations;
- $(q, \mu, d) \rightarrow (q', \mu', d')$ if there exists $\mathbf{f} \in \mathrm{Op}\,(D)$ with $d \in \mathrm{dom}\,(\mathbf{f})$, $d' = \mathbf{f}(d)$ and
  - either $(q, \mathbf{f}, q') \in T_l$, $\mu = \mu'$;
  - or $q = q'$ and there exist $q_1 \in \overline{\mu}$ and $q_2 \in \overline{\mu'}$ such that $(q_1, \mathbf{f}, q_2) \in T_c$ and $\mu' \oplus \{q_1\} = \mu \oplus \{q_2\}$.

We write respectively $(q, \mu, d) \overset{(q,\mathbf{f},q')}{\longrightarrow} (q', \mu', d')$ and $(q, \mu, d) \overset{(q_1,\mathbf{f},q_2)}{\longrightarrow} (q', \mu', d')$ in the first and second cases. This allows us to characterize relation $\rightarrow$ as the set-union of all individual transition relations:

$$[\rightarrow] = \bigcup_{t \in T_c \cup T_l} [\overset{t}{\rightarrow}]$$

Notice that some configurations may not have any successor through transition relation $\rightarrow$ since completeness is not required (Remark 7.4).

Furthermore, we introduce the following notations.

Definition 7.6. The size of a configuration $(q, \mu, d) = \gamma \in \Gamma$, or number of processes, is denoted $|\gamma| = |\mu| = \sum_{q'} \mu(q')$. We write Γ_n for the set of configurations of size n, which is finite. Data and multiset of configuration γ will be denoted respectively by $\nu\,(\gamma) = d$ and $\mathrm{st}(\gamma) = \mu$. For $q' \in Q_c$, we write $\gamma(q')$ as a shorthand for $\mathrm{st}(\gamma)(q')$. If $\mu' \in \mathbb{N}^{Q_c}$ is another multiset, we will write $\gamma \oplus \mu'$ as a shorthand for $(q, \mu \oplus \mu', d)$. Moreover, we identify q' and the multiset composed of one copy of q' allowing us to write configuration $\gamma \oplus q' = (q, \mu \oplus q', d) = (q, \mu \oplus \mathbb{1}_{\{q'\}}, d)$.

Remark 7.7. *One first important remark is that the transition relation $\rightarrow$ never introduces or removes processes: the number of processes along a path is fixed from the beginning. Although the transition system is infinite, the number of configurations of a given size is bounded.*

Another natural abstraction of our system consists in losing track of the exact number of processes in each state, and only keep information of the set of *available* states. By analogy with distribution and multisets, this abstraction will be called the support of a configuration.

Definition 7.8. Let $\gamma = (q, \mu, d) \in \Gamma$. We define the *support* of γ by $\overline{\gamma} = (q, \{q' \mid \mu(q') > 0\}, d)$. We write $q' \in \overline{\gamma}$ when $\mu(q') > 0$.

One main question that arises from the transition system definition is the choice, at each step, of the transition to be taken. Remark first, that because of our multiset construction, our choice is already limited to picking an available state from the support instead of choosing exactly which process will be involved. Once this state is chosen, it remains to determine which transition involving this state has to be taken. We argue in this chapter that both these decisions can be taken by a single scheduler, that we will assume first to be non-deterministic, then stochastic. Intuitively, we are interested in the *reachability* or *safety* from a given configuration. In the first case, this means checking whether one can manage to reach a particular configuration when all processes cooperate together with full awareness of the

state of the system, that is with a *cooperative* scheduler. In the safety case, we are interested in knowing whether the system is well designed to avoid a bad configuration, no matter the local specification of each process and the concrete interleaving, which are both chosen by a *non-cooperative* scheduler.

7.2 Parametrized reachability: a global picture

Let $\gamma_0, \gamma_f \in \Gamma$ respectively be an initial and a target configuration. We are interested in checking whether $\gamma_0 \to^* \gamma_f$ holds. Note first that this implies $|\gamma_0| = |\gamma_f|$ as $\to$ relation preserves configuration sizes. This problem has some drawbacks:

- It may be computationally hard. Intuitively, the problem can be encoded as a reachability problem in a $|\gamma_0|$-safe Petri Net. Conversely, reachability in 1-safe Petri Net can be coded back to reachability problem in a particular network protocol with fixed origin and target. Due to the constant number of processes, we can enforce atomicity and presence of a leader can be enforced even in the non-atomic and leaderless case. However, [CEP95] showed that reachability in 1-safe Petri Net is PSPACE-complete.
- On the other hand, solving reachability question for a particular initial configuration does not solve reachability for other instances. Moreover, the number of processes can be seen as an undetermined variable, and we may be interested in reachability no matter the exact value of this parameter, except the possible assumption for this parameter to be large.

We consider now a more general reachability problem. As our number of processes is not fixed *a priori*, a reasonable goal is to analyze the behaviour of our system for an arbitrary number of processes. This number will be left as a parameter and we are interested in detecting changing behaviour of our system depending on this parameter.

First of all, we consider the whole set of initial configurations:

Definition 7.9. Let $U_0 = \{(q_l, n \cdot q_c, d_0) \mid n \geq 1\}$ be the set of initial configurations.

As previously mentioned, the size of a configuration does not change over time so we have to consider a set of target configurations U_f, and the problem is now whether there exists $\gamma_0 \in U_0, \gamma_f \in U_f$ such that $\gamma_0 \to^* \gamma_f$.

7.3 Monotonicity

7.3.1 Upward closed reachability objectives

Elements of U_0 share the following property: they are all composed of the same leader state, register value, and set of contributor states (mainly $\{q_c\}$). They can be expressed easily using the following ordering relation:

Definition 7.10 (Ordering relation)**.** We consider the ordering relation $\preceq$ over Γ defined by $\gamma \preceq \gamma'$ if

- $\overline{\gamma} = \overline{\gamma'}$,
- $\forall q \in \overline{\gamma}\ \gamma(q) \leq \gamma'(q)$.

Definition 7.11. Let $A \subseteq \Gamma$, the upward closure $\uparrow A$ is defined by $\uparrow A = \{\gamma \mid \exists \gamma' \in A \ \gamma' \preceq \gamma\}$. A is called upward closed when $A = \uparrow A$.

Given the previous definitions, the set of initial configurations is easily expressed as the following upward closure.

$$U_0 = \uparrow\{(q_l, q_c, d_0\}$$

In the rest of the chapter, we will assume that U_f enjoy similar property, that is U_f is an upward closed set, and we will now give several reasons to justify this decision.

Note first that this assumption allows various kinds of reachability objectives, such as the following:

- *Coverability* of a given contributor state q_f by *some* process: $U_f = \{\gamma \mid \gamma(q_f) > 0\}$.
- *Consensus* to a given contributor state q_f by *all* processes: $U_f = \{\gamma \mid \gamma(q_f) = |\gamma|\}$.

Another important feature of these objectives is their finite unique representation, as an upward closure of a finite set of incomparable elements. This will allow us to consider the computational complexity of a reachability problem from U_0 to U_f, as we will be able to finitely describe any instance of our problem. This finite unique representation is due to results from Chapter 2 and the following result on well quasi-orders.

Theorem 7.12 (Dickson's lemma)**.** $(\Gamma, \preceq)$ *is a* well quasi-order *(wqo).*

A natural quantity for complexity study would be the size of a representation of an upward closed set U which will be likely a sum of the sizes of minimal elements $\sum_{\gamma \in \min U} |\gamma|$. However, this quantity reveals to be large for some simple objectives, as *coverability.* Indeed, for coverability objective, $\min\{\gamma \mid \gamma(q_f) > 0\} = 2^{\sharp[Q_c]-1}$, so the naive representation of such objectives is at least exponential. As this quantity is large, even for small minimal elements, we may lose some information about the difficulty of the reachability problem. We introduce below another notion of size, which will be more relevant for our later results, based on the maximal size of minimal elements appearing in an upward closed set.

Definition 7.13. Let $U \subseteq \Gamma$ an upward closed set. We define the size of U by

$$|U| = \max_{\gamma \in \min U} |\gamma|$$

As announced before the definition, this new notion of size captures the simplicity of both previous objectives consensus and coverability. Indeed, we check that in both cases, we have $|U_f| \leq \sharp\,[Q_c] \leq |\mathcal{P}|$.

7.3.2 Non-atomicity

We have seen that reachability is in general a computationally hard problem. A way to tackle this issue presented for example in [EGM13] is to consider non-atomic register protocols, which enjoy monotonicity properties, as we expect the result of the parametrized reachability analysis to provide additional regularity properties.

Definition 7.14 (Monotonous register protocol)**.** We say that $\mathcal{P}$ is *monotonous* when for any $\gamma_1, \gamma_2 \in \Gamma$ with $\gamma_1 \to \gamma_2$, the two following properties hold:

- For any $\gamma_1' \succeq \gamma_1$, there exists γ_2' such that $\gamma_1' \to^* \gamma_2'$ and $\gamma_2 \preceq \gamma_2'$.
- For any $\gamma_2' \succeq \gamma_2$, there exists γ_1' such that $\gamma_1' \to^* \gamma_2'$ and $\gamma_1 \preceq \gamma_1'$.

Our definition of monotonicity is very close to the concept of *well-structured transition system* (WSTS), defined in [FS01]. We can indeed remark that a protocol is monotonous if, and only if, both $(\Gamma, \to, \preceq)$ (first condition) and $(\Gamma, \to^{-1}, \preceq)$ (second condition) are WSTS.

Definition 7.15. We define, for any $A \subseteq \Gamma$:

$$\mathrm{Pre}(A) = \{\gamma \in \Gamma \mid \exists \gamma' \in A \ \gamma \to \gamma'\} \quad \mathrm{Post}(A) = \{\gamma' \in \Gamma \mid \exists \gamma \in A \ \gamma \to \gamma'\}$$

WSTS are particularly well suited for backward reachability analysis, since predecessor operator maps upward closed sets to upward closed sets. Here, both the initial system and the reversed one are WSTS, so both predecessor and successor operators preserves upward closed sets.

Lemma 7.16. *Assume $\mathcal{P}$ is monotonous, then* Pre*,* Post*,* Pre**,* Post* *preserve upward closed sets.*

Our systems also enjoy an additional property: they preserve size of the configurations over time, which gives us an equivalent simpler definition based on incrementation of states:

Lemma 7.17 (Monotonous register protocol)**.** *$\mathcal{P}$ is monotonous if, and only if, for any $\gamma_1, \gamma_2 \in \Gamma$ with $\gamma_1 \to \gamma_2$, the two following properties hold:*

- *For $q_1 \in \overline{\gamma_1}$, there exists $q_2 \in \overline{\gamma_2}$ such that $\gamma_1 \oplus q_1 \to^* \gamma_2 \oplus q_2$.*
- *For $q_2 \in \overline{\gamma_2}$, there exists $q_1 \in \overline{\gamma_1}$ such that $\gamma_1 \oplus q_1 \to^* \gamma_2 \oplus q_2$.*

Proof. We describe the proof of the first point, the second property is similar: Notice first that we can assume $\gamma_1 \to^* \gamma_2$ instead of $\gamma_1 \to \gamma_2$ in the hypothesis as the result extends immediately to arbitrary paths by induction on their lengths.

$\Rightarrow$ Assume $\mathcal{P}$ is monotonous, and let $q_1 \in \overline{\gamma_1}$, then $\gamma_1 \preceq \gamma_1' = \gamma_1 \oplus q_1$, so there exists $\gamma_2' \succeq \gamma_2$ such that $\gamma_1' \to^* \gamma_2'$, hence $|\gamma_2'| = |\gamma_1'| = |\gamma_1| + 1 = |\gamma_2| + 1$ so there exists $q_2 \in \overline{\gamma_2'} = \overline{\gamma_2}$ such that $\gamma_2' = \gamma_2 \oplus q_2$.

$\Leftarrow$ Conversely, assume that $\gamma_1 \to \gamma_2$ and consider γ_1' such that $\gamma_1 \preceq \gamma_1'$. We show by induction on $|\gamma_1'| - |\gamma_1|$ that there exists γ_2' such that $\gamma_1' \to^* \gamma_2'$ and $\gamma_2 \preceq \gamma_2'$.

- If $|\gamma_1'| = |\gamma_1|$ and $\gamma_1 \preceq \gamma_1'$, then $\gamma_1 = \gamma_1'$.
- If $|\gamma_1'| \geq |\gamma_1| + 1$ and $\gamma_1 \preceq \gamma_1'$, then there exists $q_1 \in Q_c$ and $\gamma_1'' \in \Gamma$, such that $\gamma_1' = \gamma_1'' \oplus q_1$ and $\gamma \preceq \gamma_1''$. By induction hypothesis, there exists γ_2'' such that $\gamma_1'' \to^* \gamma_2''$ and $\gamma_2 \preceq \gamma_2''$. By hypothesis, there also exists q_2 such that $\gamma_1'' \oplus q_1 \to \gamma_2'' \oplus q_2 = \gamma_2'$. Hence, γ_2' satisfies $\gamma_1' \to^* \gamma_2'$ and $\gamma_2 \preceq \gamma_2'$.

□

In the rest of the chapter, we will be mainly interested in monotonous protocols. In particular, any non-atomic operations **f** can be repeated (stuttered) an arbitrary amount of time. This simple remark applies to any non-atomic protocol which therefore are monotonous. As depicted in Figure 7.2, each process can be mimicked by another one (the copycat) performing the same transitions immediately after each transition of the original process.

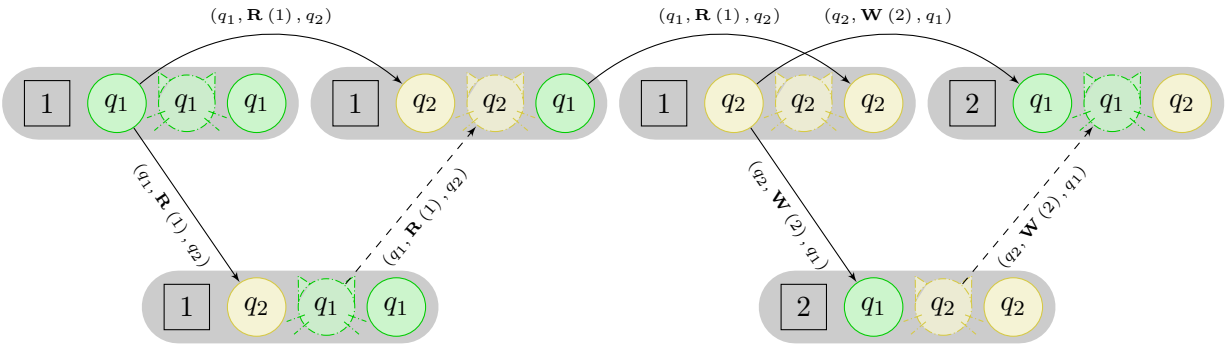


Figure 7.2 – Copycat lemma applied to path $(1, 2 \cdot q_1) \to^3 (2, q_1 \oplus q_2)$ of register protocol from Figure 7.1 by copying first process. Each of its transitions is copied (dashed) immediately after the original one, thanks to non-atomicity.

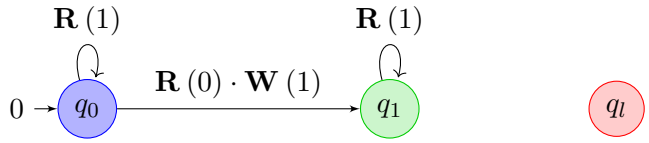


Figure 7.3 – Example of a simple atomic protocol which is not monotonous. Indeed, $(q_l, q_0, 0) \to (q_l, q_1, 1)$ but for any $n \geq 2$, we cannot have $(q_l, n \cdot q_0, 0) \to^* (q_l, n \cdot q_1, 1)$.

Lemma 7.18 (Copycat Lemma [DEGM15])**.** *Non-atomic register protocols are monotonous.*

Proof. Let $\gamma_1, \gamma_2 \in \Gamma$ such that $\gamma_1 \to \gamma_2$ and $q_1 \in \overline{\gamma_1}$. There exists $(q, \mathbf{f}, q') \in T$, such that $\gamma_1 \overset{(q,\mathbf{f},q')}{\to} \gamma_2$

- If $q \neq q_1$, then $q_1 \in \overline{\gamma_2}$ so $q_2 = q_1$ satisfies $\gamma_1 \oplus q_1 \to \gamma_2 \oplus q_2$.
- If $q = q_1$, then $q_2 = q'$ satisfies $\gamma_1 \oplus q_1 \to\to \gamma_2 \oplus q_2$.

□

Non-atomicity is a crucial property for monotonicity, as illustrated in Figure 7.3 with a simple atomic protocol which is not monotonous.

7.4 Probabilistic transition system

As discussed earlier, we are interested in stochastic behaviours of our systems, which may lead to breaking symmetries. We will now introduce probabilities over our transition system and characterize the qualitative properties we will study. In order to do so, we will equip our transition system with probability over transitions between configurations. This implies replacing our non-deterministic scheduler by a *stochastic* scheduler. As opposed to reachability and safety properties, our new stochastic scheduler will neither play with nor against our protocols as transitions will be picked at random.

Definition 7.19 (Probabilistic transition system)**.** Let $p : (\gamma, \gamma') \in [\to] \mapsto p(\gamma, \gamma') \in \mathbb{R}$ such that for all $\gamma \in \Gamma$, $p(\gamma, -)$ is a probability distribution with support $\mathrm{Post}(\{\gamma\})$. We say that (Γ, p) is a probabilistic transition system of $\mathcal{P}$.

Due to the lack of completeness (see Remark 7.4), there might exist configurations γ such that $\mathrm{Post}(\{\gamma\}) = \varnothing$, which is not a possible probability distribution support. We address this issue by defining p only over $[\rightarrow]$ seen a subset of Γ^2.

Definition 7.20 (Induced $\mathbb{P}$ by p)**.** We consider a probabilistic transition system (Γ, p). For any sequence of configurations $\pi \in \Gamma^+$, we define $\mathbb{P}(\pi)$ by induction on π:

- For any $\gamma \in \Gamma$, $\mathbb{P}(\gamma) = 1$;
- For any $\pi \in \Gamma^*, \gamma, \gamma' \in \Gamma$, $\mathbb{P}(\pi \cdot \gamma \cdot \gamma') = \begin{cases} \mathbb{P}(\pi \cdot \gamma) \cdot p(\gamma, \gamma') \text{ if } \gamma \rightarrow \gamma' \\ 0 \text{ otherwise} \end{cases}$.

Intuitively, $\mathbb{P}(\pi)$ denotes the probability that any maximal run starting from $\pi[1]$ has a finite prefix π. Notice that because of possible deadlocks, a configuration γ may not have any successor, which means $\mathrm{Post}(\{\gamma\}) = \varnothing$ and any run visiting γ is of finite length. Thanks to Carathéodory's criterion and in particular Theorem 2.1, we show that for any $\gamma \in \Gamma$, $\mathbb{P}(\gamma\ -)$ is a probability measure over maximal runs of $(\Gamma, \rightarrow)$.

7.4.1 Qualitative analysis

From the definition of p and $\mathbb{P}$, we conclude that $\mathbb{P}(\pi) > 0$ if, and only if, $\pi \in \mathrm{paths}\,(\rightarrow)$, which means the exact probability values don't matter for positive probability property. We will continue the study of our network protocols by giving several properties that do not depend on the exact probability values, *i.e.* qualitative properties.

Remember first of all we are interested in reachability properties, which we define now in our stochastic context:

Definition 7.21. Let $n \in \mathbb{N}$ and $\Pi \subseteq \mathrm{paths}\,(\rightarrow)$ a measurable set of paths. We define $\mathbb{P}_n(\Pi) = \mathbb{P}(\Pi \cap U_0 \cdot \Gamma_n^*)$ the probability of achieving a path from Π with n contributors.
Let us consider $A \subseteq \Gamma$, we denote by $\Diamond A = (\Gamma \backslash A)^* \cdot A$ the set of finite paths *eventually* visiting A.

Notice that this definition is sound: $\Diamond A = (\Gamma \backslash A)^* \cdot A$ is always measurable, as a countable union of finite prefixes. For a given $n \in \mathbb{N}$, the paths in $U_0 \cdot \Gamma_n^*$ are in fact of the form $\{\gamma_0^n\} \cdot \Gamma_n^*$ with $\gamma_0^n = (q_l, n \cdot q_c, d_0)$, which means $\mathbb{P}_n$ actually defines a probability measure.

We now remark that deciding eventual reachability with probability 0, positive, or 1 are all three qualitative properties that can be expressed without mentioning the exact probabilistic transition system.

Lemma 7.22. *Let $A \subseteq \Gamma$. Both properties $\lceil \mathbb{P}_n(\Diamond A) \rceil = 1$ and $\lfloor \mathbb{P}_n(\Diamond A) \rfloor = 1$ do not depend on the actual values of p. Moreover, we have the following characterizations.*

$$\lceil \mathbb{P}_n(\Diamond A) \rceil = 1 \Leftrightarrow \mathbb{P}_n(\Diamond A) > 0 \Leftrightarrow (\Gamma_n \cap U_0) \cap \mathrm{Pre}^*(A) \neq \varnothing$$
$$\lfloor \mathbb{P}_n(\Diamond A) \rfloor = 1 \Leftrightarrow \mathbb{P}_n(\Diamond A) = 1 \Leftrightarrow \mathrm{Post}_A^*(\Gamma_n \cap U_0) \subseteq \mathrm{Pre}^*(A)$$

where

$$\mathrm{Post}_A(X) = \{\gamma \mid \exists \gamma' \in X \backslash A\ \gamma' \rightarrow \gamma\} = \mathrm{Post}(X \backslash A)$$

Proof. We will describe here the proof for the second property $\lfloor \mathbb{P}_n(\Diamond A) \rfloor = 1$, the other case is similar. First of all, notice that $\lfloor \mathbb{P}_n(\Diamond A) \rfloor = 1 \Leftrightarrow \mathbb{P}_n(\Diamond A) = 1$. We then prove the equivalence:

$\Leftarrow$ Assume $\mathrm{Post}^*_A(\Gamma_n \cap U_0) \subseteq \mathrm{Pre}^*(A)$, then $(\mathrm{Post}^*_A(\Gamma_n \cap U_0) \cup \mathrm{Pre}^*(A), p)$ forms a finite Markov chain with a set of transient states in $\mathrm{Post}^*_A(\Gamma_n \cap U_0)$.

$\Rightarrow$ Let $\gamma \in \mathrm{Post}^*(\Gamma_n \cap U_0)$, which means there exists a path $\pi : \gamma_0^n \in U_0 \to^* \gamma$, with $\pi \in (\Gamma \backslash A)^+$. We initially have $\mathbb{P}_n(\pi[1] \cdot \Gamma^* \cap \Diamond A) = 1$ and we show by induction that for any i, $\mathbb{P}_n((\pi[i] \cdot \Gamma^*) \cap \Diamond A) = 1$. Indeed,

$$\mathbb{P}_n((\pi[i] \cdot \Gamma^*) \cap \Diamond A) = \sum_{\gamma'} \mathbb{P}_n(\pi[i] \cdot \gamma') \cdot \mathbb{P}_n((\gamma' \cdot \Gamma^*) \cap \Diamond A)$$

with $\sum_{\gamma'} \mathbb{P}_n(\pi[i] \cdot \gamma') = 1$. In particular, for $\gamma' = \pi[i+1]$, $\mathbb{P}_n(\pi[i] \cdot \gamma') > 0$ so $\mathbb{P}_n((\pi[i] \cdot \Gamma^*) \cap \Diamond A) = 1$ implies $\mathbb{P}_n((\pi[i+1] \cdot \Gamma^*) \cap \Diamond A) = 1$.

In particular, $\mathbb{P}_n((\gamma \cdot \Gamma^*) \cap \Diamond A) = 1$, so $\gamma \in \mathrm{Pre}^*(A)$.

□

As opposed to Post operator, Post_A and even Post^*_A do not preserve upward closed sets, even when A is upward closed. For example, assume $\mathrm{Post}(\{(q, q_1, d)\}) = \{(q, q_2, d)\}$ and $A = \uparrow\{(q, 2 \cdot q_1, d)\}$ then $\mathrm{Post}_A(\uparrow\{(q, q_1, d)\}) = \mathrm{Post}(\{((q, q_1, d)\}) = \{(q, q_2, d)\}$ and $\mathrm{Post}^*_A(\uparrow\{(q, q_1, d)\}) = \uparrow\{(q, q_1, d)\} \cup \mathrm{Post}^1_A(\uparrow\{(q, q_1, d)\}) \cup \mathrm{Post}^2_A(\uparrow\{(q, q_1, d)\}) \cup \ldots = \uparrow\{(q, q_1, d)\} \cup \{(q, q_2, d)\}$ which are not upward closed sets. However, when minimal elements of an upward closed set A are minimal among elements of Γ, set subtraction by A preserves upward closed sets.

Definition 7.23. A reachability set U_f is called *simple* whenever it is upward closed and

$$\min U_f \subseteq \min \Gamma$$

We can check that simple reachability sets are generated by minimal elements of the form γ with $\forall q\ \gamma(q) \leq 1$. In particular, both reachability and consensus objectives are simple reachability objectives.

When U_f is a simple set reachability objective, Post_{U_f} and $\mathrm{Post}^*_{U_f}$ preserve upward closed sets, thanks to the following lemma:

Lemma 7.24. *For any upward closed set U and a simple set U_f, $U \backslash U_f$ is upward closed.*

Proof. Let $\gamma \preceq \gamma'$ with $\gamma \in U \backslash U_f$. Assume $\gamma' \in U_f$, then there exists $\gamma'_m \in \min U_f \subseteq \min \Gamma$ such that $\gamma'_m \preceq \gamma'$. This means that $\overline{\gamma} = \overline{\gamma'} = \overline{\gamma'_m}$ and we have the following multiset inequalities: $\mathrm{st}(\gamma) \leq \mathrm{st}(\gamma')$ and $\mathrm{st}(\gamma'_m) \leq \mathrm{st}(\gamma')$. This implies $\mathrm{st}(\gamma'_m) \leq \mathrm{st}(\gamma)$ which is absurd since $\gamma \notin U_f$. Hence, $\gamma' \in U \backslash U_f$. □

7.4.2 Cut-off property

We will study some limit behaviours of our systems according to the probabilistic reachability defined before. We introduce a notion of cut-off, similar to [AJK16].

Definition 7.25. Let I be a sub-interval of $[0, 1]$ and A a set of configurations. We say that $N \in \mathbb{N}$ is a I-cut-off for reaching A, if either

- for all parameter $n \geq N$, $\mathbb{P}_n(\Diamond A) \in I$,

- for all parameter $n \geq N$, $\mathbb{P}_n(\Diamond A) \notin I$.

The cut-off is said *positive* in the first case, and *negative* otherwise.

A cut-off is a parameter value that permits restricting the reachability analysis to a finite number of parameters: beyond this parameter the answer to the reachability question has a fixed answer either positive or negative.

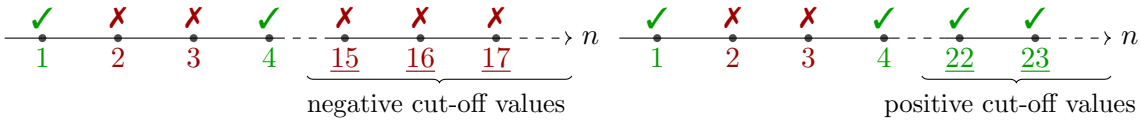


Figure 7.4 – Cut-off illustration for a property depending on parameter n: a cut-off value ensures that the property is always true or always false for bigger parameters.

Let us first remark that a cut-off needs not exist, however for a given interval I, existence of positive and negative cut-off are mutually exclusive properties.

When the target set A will be omitted, we will consider a generic upward closed set U_f. As the previous Lemma 7.22 concerning qualitative probabilistic properties applies, we will distinguish the following particular cut-off properties:

- $\{0\}$-cut-off is a *safety* property;
- $(0, 1]$-cut-off is a *reachability* (with positive probability) property;
- $\{1\}$-cut-off is *almost-sure reachability*.

The following chapters will be mainly dedicated to the two following questions:

- Existence: does there exists a I-cut-off for reaching U_f ?
- Decision problem: is this I-cut-off positive or negative ?

Chapter 8

Probabilistic reachability and safety

Before diving into the almost-sure reachability problem, we will build some tool box from the analysis of non-deterministic schedulers, thanks to the study of (positive) reachability and safety cut-offs.

In this section, we revisit results from [EGM13], where analysis of runs is based on two transition categories: useless write operations, that are immediately overridden, and first write operation for a given value $d \in D$. The safety problem from coverable objective boils down to the analysis for a small number, $\sharp\,[D]$, of contributors, where each contributor is in charge of writing exactly one of the value $d \in D$ for the first time. As a consequence, the safety (resp. reachability) cut-off problem is shown in co-NP (resp. NP), then proven to be co-NP-complete (resp. NP-complete) for non-atomic register protocols.

We adopt here a dual approach, where emphasis is put on the evolution of the set of contributor states in the support, that can therefore make the leader progress. The developed techniques, based on the symbolic graph, abstracting the system, will be refined later to analyze almost-sure reachability.

8.1 Existence

One first important remark consists in showing that safety and reachability are dual notions, which is not obvious from the definition, but is a direct consequence of the nature of our systems, that can ignore additional processes: let us denote $N \in \mathbb{N} \uplus \{\infty\}$ as the minimal value such that $\mathbb{P}_N(\Diamond U_f) > 0$ (with $N = \infty$ if it never holds). Then,

- If $N = \infty$, 0 is a negative cut-off for reachability, and a positive cut-off for safety.
- If $N < \infty$, by copycat lemma, N is a positive cut-off for reachability and a negative cut-off for safety, by minimality of N.

8.2 Symbolic graph

We introduce the main tool for the analysis of reachability under a non-deterministic scheduler, namely the abstraction of the number of states in a configuration.

Definition 8.1. The *symbolic graph* $G_{\mathcal{P}}$ of $\mathcal{P}$ is a directed graph $G_{\mathcal{P}} = (V, E)$ with

- $V = \{\overline{\gamma} \mid \gamma \in \Gamma\}$.

- $E = \{(\overline{\gamma}, \overline{\gamma'}) \mid \gamma, \gamma' \in \Gamma : \gamma \to \gamma'\}$.

Intuitively, the symbolic graph represents an abstraction through the support function, nodes are of the form $s = \overline{\gamma}$ with $\gamma \in \Gamma$ and there is an edge from s to s' if a transition is possible for some pair of concrete configurations, with respective support s and s'. From the definition, we can obviously state than any path $\pi \in \text{paths}(\to)$ can be converted into a path in the symbolic graph by abstracting each configuration in the path : $\overline{\pi} \in \text{paths}(E)$.

An example of (partial) symbolic graph is depicted in Figure 8.1.

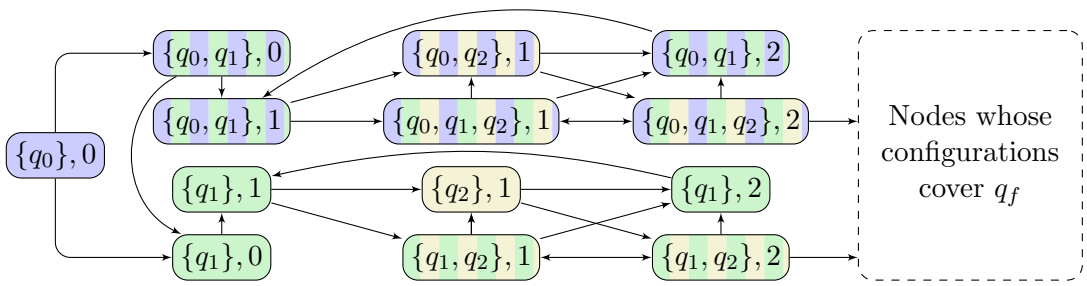


Figure 8.1 – Solving coverability of q_f for the protocol of Figure 7.1. Self-loops and target nodes are omitted.

The symbolic graph does not keep track of the number of contributor processes in each state: some process may have to be duplicated during transitions involving contributors. Hence, a natural notion to measure the length of a run is the number of transitions involving a contributor process.

Definition 8.2. Let $\pi \in \text{paths}(E)$ be a path. We denote by $|\pi|_l$ the *leader length* of π, defined as the quantity:

$$|\pi|_l = \sharp\left[\left\{i \;\middle|\; 1 \leq i < |\pi| \wedge \exists \gamma, \gamma' \in \Gamma \;\exists t \in T_l \;\gamma \xrightarrow{t} \gamma' \wedge \overline{\gamma} = \pi[i] \wedge \overline{\gamma'} = \pi[i+1]\right\}\right]$$

Then, we define the *contributor length* of π by :

$$|\pi|_c = |\pi| - |\pi|_l$$

A first important remark is that we can build again a *concrete* run from a *symbolic* run, involving no more contributor processes than the contributor length of the path.

Lemma 8.3. *Assume $\mathcal{P}$ is monotonous. If there exists a path from s to s' of contributor length n, then there exist a path from $\gamma \in \Gamma$ to $\gamma' \in \Gamma$ with $\overline{\gamma} = s$, $\overline{\gamma'} = s'$ and $|\gamma| \leq n + \sharp[Q_c]$.*

Proof. We proceed by induction on the length of the path $\pi \in \text{paths}(E)$ from s to s'.

- If $|\pi| = 1$, π is restricted to the single state $s = (q, X, d)$ so we define $\gamma = \gamma' = (q, \bigoplus_{q' \in X} q', d)$.
- If $sE^*s'Es''$, we apply induction hypothesis on sE^*s' with contributor length n, there exist $\gamma, \gamma' \in \Gamma$ with $\overline{\gamma} = s$, $\overline{\gamma'} = s'$, $|\gamma| \leq n + \sharp[Q_c]$ and $\gamma \to^* \gamma'$.
  If there exists $t = (q, \mathbf{f}, q') \in T_l$ such that $(q, X, d) = s'$ and $(q', X, \mathbf{f}(d)) = s''$ then the contributor length is still n and we let $\gamma'' = (q', \text{st}(\gamma'), \mathbf{f}(d))$ hence $\gamma' \xrightarrow{t} \gamma''$.

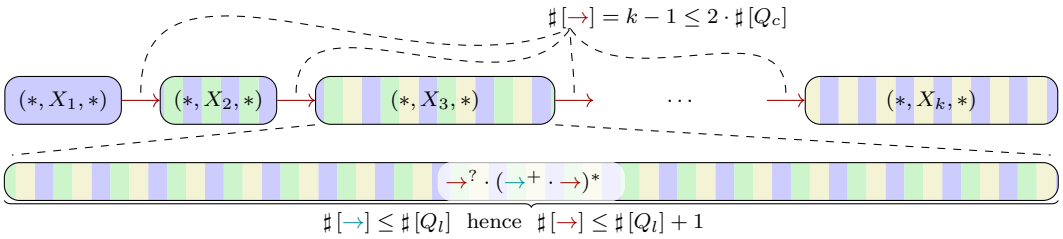


Figure 8.2 – Proof scheme of Lemma 8.4, red arrows $\rightarrow$ correspond to contributor transitions whereas blue arrows $\rightarrow$ correspond to leader transitions.

Otherwise, there exists $t = (q, \mathbf{f}, q') \in T_c$ such that $\gamma'(q) > 0$, $(\tilde{q}, X, d) = s'$ and $(\tilde{q}, X', \mathbf{f}(d)) = s''$ with $X \cup \{q'\} = X' \cup \{q\}$. We can again rebuild γ'' by firing transition t from γ', except if we still have at the end $q \in X'$ and $\gamma'(q) = 1$. In this last case, we apply Lemma 7.17 to path $\gamma \rightarrow^* \gamma'$ (monotonicity with state incrementation) to assume, without loss of generality, that $\gamma'(q) = 2$ (hence $\gamma''(q) = 1$) and $|\gamma| = |\gamma'| \leq n + \sharp[Q_c] + \mathbf{1}$.

□

Note that a similar argument can be found in [DEGM15]. The authors are considering coverability objectives. A path covering q_f can be abstracted into a symbolic path, that can be compressed into another symbolic path, which is non-decreasing on the contributors set coordinate. By the previous lemma, this new symbolic path can still be realized, and still covers q_f. This property ensures that the number of portions is bounded by $\sharp[Q_c]$, hence a polynomial witness for coverability. Although this non-decreasing property is sufficient for the study of coverability objectives, it cannot be used anymore for our arbitrary upward closed objectives.

On the other hand, the particular structure of non-atomic protocols allows us to give another bound on the symbolic path, which is still polynomial in the size of the network; more formally, we are able to perform the following path compression:

Lemma 8.4 (Diameter of the symbolic graph)**.** *Assume $\mathcal{P}$ is non-atomic (hence monotonous). If two nodes s and s' are connected in $G_{\mathcal{P}}$, then there exists a path from s to s' of contributor length smaller than*

$$(2 \cdot \sharp[Q_c] + 1) \cdot (\sharp[Q_l] + 1) + 2 \cdot \sharp[Q_c]$$

An important remark on this bound is its independence with the number of involved data values: the quantity neither depends on $\sharp[D]$ nor $\sharp[T_c]$ nor $\sharp[T_l]$. Intuitively, density of the graph $(\Gamma, \rightarrow)$ has no influence on its diameter. We depict the two steps of the path reduction in Figure 8.2 and in the following proof:

Proof. Let $\pi : s \rightarrow^* s'$. Thanks to the monotonicity property, we transform π into π' satisfying the following properties:

1. $\mathit{first}(\pi) = \mathit{first}(\pi')$, $\mathit{last}(\pi) = \mathit{last}(\pi')$,

2. For each $q \in Q_c$, there is at most one position i such that $q \notin \pi'[i]$ and $q \in \pi'[i+1]$,

3. For each $q \in Q_c$, there is at most one position i such that $q \in \pi'[i]$ and $q \notin \pi'[i+1]$,

With the previous requirements, it is easy to check that π' is composed of at most $2\sharp[Q_c]$ transitions of the contributors that changes the set of contributors. As a consequence, there are at most $2\sharp[Q_c]+1$ contiguous portions of π' where the set of contributors is fixed. For such a portion, we can assume, without loss of generality that the leader length is at most equal to $\sharp[Q_l]$, otherwise, the leader process would have visited the same state twice. Since the set of possible operations performed by the contributor is fixed during a portion, there is at most one operation performed before and between each leader transition, hence at most $\sharp[Q_l]+1$ transitions for the contributors. □

We conclude this chapter with the following result:

Theorem 8.5. *The two following problems are* NP*-complete:*

<u>ReachCoverCutOff</u>

INPUT: *A non-atomic register protocol $\mathcal{P}$ and a final state $q_f \in Q_c$.*
QUESTION: *Whether there exists a cut-off for positive probability reachability of $U_f = \{\gamma \mid \gamma(q_f) > 0\}$.*

<u>ReachConsensusCutOff</u>

INPUT: *A non-atomic register protocol $\mathcal{P}$ and a final state $q_f \in Q_c$.*
QUESTION: *Whether there exists a cut-off for positive probability reachability of $U_f = \{\gamma \mid \gamma(q_f) = |\gamma|\}$.*

Moreover, cut-off value for reachability of any upward closed target U_f is always polynomial in $|\mathcal{P}| + |U_f|$.

Proof. The proof structure is the same for any upward closed objective U_f. If a (concrete) path manages to visit U_f, we can compress this path thanks to lemmas 8.4 and 8.3 to involve only a polynomial number of processes. Moreover, thanks to monotonicity, this path is still valid for bigger parameters. This allows us to state that previous symbolic run provides a cut-off for reachability, and one can non-deterministically guess such a polynomial size run to decide whether U_f is reached in the symbolic graph, which concludes membership of these problems to NP.

NP-hardness follows from reduction from 3SAT, as described in [EGM13]. □

Chapter 9

Almost-sure reachability

We consider now the main problem introduced for probabilistic systems, namely the almost-sure reachability cut-off properties. As we will see, this new property is not comparable with reachability and safety properties as scheduler is neither bad nor good in helping reaching a final configuration. This section will first give some intuitions about the difficulty of the almost-sure reachability problem and why the previous tools are not suited anymore. Then, we will prove that non-atomic protocols always have a cut-off for almost-sure reachability for any simple objective. Since the proof relies on well quasi-orders, the cut-off value may have an arbitrary size, so before diving into a decision procedure, we will study hardness results. Lower bounds for minimal (tight) cut-off values will be given, then the decision will be proven to be PSPACE-hard. Finally, we will provide a refinement of our symbolic graph that will allows us to develop a EXPSPACE decision procedure.

Unless otherwise specified, we will mainly consider a non-atomic register protocol $\mathcal{P}$ with a simple reachability objective U_f (see Definition 7.23). Our results will involve an arbitrary leader, but hardness results will be given in the stronger case of a leaderless protocol.

9.1 First examples

9.1.1 Atomicity prevents cut-off existence

Let us first consider the example depicted in Figure 9.1, with coverability of q_f as simple target set U_f. We argue that due to the atomic operations involved, the protocol has no cut-off. Intuitively, two processes from state q_0 can cooperate to cancel each other by going to sink state q_2. If the number of initial processes is odd, there should always remain one process which always have the ability to trigger the transition to q_f. In the long run, such event will happen with probability one.

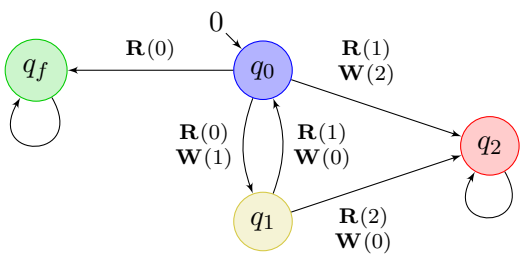


Figure 9.1 – Example of a register protocol with atomic read/write operations.

First of all, if n is even, we exhibit a run which sends all processes to state q_2: $(n \cdot q_0, 0) \to ((n-1) \cdot q_0 \oplus q_1, 1) \to ((n-2) \cdot q_0 \oplus q_1 \oplus q_2, 2) \to ((n-2) \cdot q_0 \oplus 2 \cdot q_2, 0) \to^{3\frac{n-2}{2}} (n \cdot q_2, 0)$. In the other case, we can easily check that from any initial state in $U_0 = \uparrow \{(q_0, 0)\}$, the system can only reach configurations γ where $\gamma(q_1) = \mathbb{1}_{\nu(\gamma) \neq 0}$ (one q_1 when register differs from 0, none otherwise). We derive then:

- If $\nu(\gamma) = 2$, then $\gamma(q_2)$ is odd;
- If $\nu(\gamma) \neq 2$, then $\gamma(q_2)$ is even;
- If $|\gamma|$ is odd, then for any successor γ', $\gamma'(q_2) < |\gamma'|$.

We conclude that for any odd parameter n, $\mathbb{P}_n(\Diamond U_f) = 1$.

This example gives us another reason to focus only on non-atomic registers protocols, for which such atomicity of processes cannot occur.

9.1.2 Symbolic graph is powerless

We have seen in the previous section that reachability in the symbolic graph is equivalent to reachability in $(\Gamma, \to)$.

A first intuition consists is checking almost-sure reachability in the symbolic graph to infer almost-sure reachability cut-off in the concrete system. Indeed, we can remark that the second property implies the existence of a positive cut-off.

Theorem 9.1. *Assume that $\mathcal{P}$ is monotonous and has a positive cut-off for almost-sure reachability of U_f. Then, any reachable node s in $G_{\mathcal{P}}$ from $\overline{U_0}$ can reach $\overline{U_f}$.*

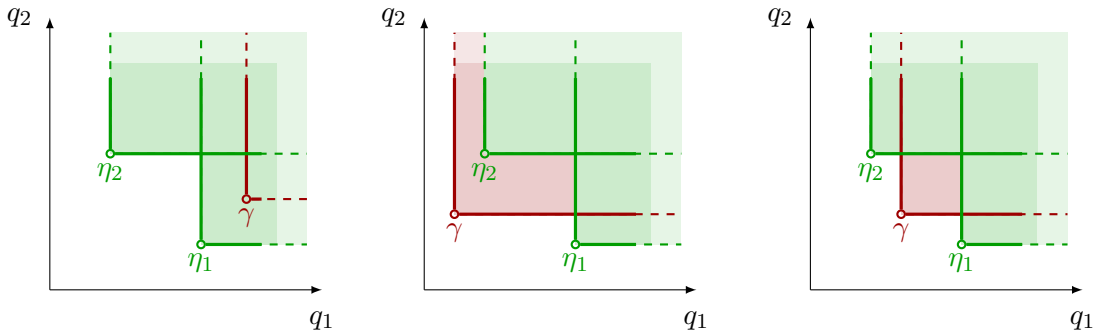
Proof. Assume $\mathcal{P}$ has a positive cut-off N. For any reachable node s in $G_{\mathcal{P}}$ from $\overline{U_0}$, we can find by Lemma 8.3 $\gamma_0 \in U_0$ and γ such that $\gamma_0 \to^* \gamma$ and $\overline{\gamma} = s$. We apply monotonicity to assume that $|\gamma| \geq N$ (otherwise replace γ by $\gamma \oplus (N - |\gamma|) \cdot q$ with same support, for some $q \in \overline{\gamma}$). By qualitative almost-sure reachability, there exists $\gamma_f \in U_f$ such that $\gamma \to^* \gamma_f$, thus $s = \overline{\gamma} E^* \overline{\gamma_f} \in \overline{U_f}$. □

However, we explain now that the converse property cannot be extended to the probabilistic transition system for almost-sure reachability. As a matter of fact, we consider the leaderless non-atomic protocol of Figure 7.1 and its symbolic graph depicted in Figure 8.1.

Nodes covering q_f in the symbolic graph are reachable from any intermediate abstract configuration in the symbolic graph. This means that equipped with any relevant probability transitions, the symbolic graph satisfies almost-sure reachability for covering q_f. However, some transitions require the presence of at least two processes in particular given states. This is in general not ensured by our abstraction. Consider for example, initial run with parameter n:

$$(n \cdot q_c, 0) \to ((n-1) \cdot q_c \oplus q_1, 0) \to ((n-1) \cdot q_c \oplus q_1, 1)$$

We can check that such configuration cannot reach $U_f = \{\gamma \mid \gamma(q_f) > 0\}$ anymore, as the only process in state q_1 has to read register value 2 from q_2, which can only be written by itself, when going back to state q_1. This transition exists in the symbolic graph, as $(\{q_2\}, 1) \to (\{q_1, q_2\}, 2)$ though it is not concretely feasible.



(a) $U \subseteq U'$ therefore $U \sqsubseteq U'$

(b) $U \not\sqsubseteq U'$ as γ can be arbitrarily increased on q_2 coordinate and stay outside U'.

(c) $U \sqsubseteq U'$ as eventually γ will be replaced by configurations with more q_1 or q_2 states enforcing inclusion in U'.

Figure 9.2 – Simple cases of ultimate inclusions analysis with $Q_c = \{q_1, q_2\}$, $U = \uparrow\{\gamma\}$ and $U' = \uparrow\{\eta_1, \eta_2\}$.

9.2 Existence

Due to the monotonicity property and the wqo structure of the transition system, both $\mathrm{Post}^*_{U_f}(U_0)$ and $\mathrm{Pre}^*(U_f)$ are upward closed sets that are finitely generated. By analyzing further how the reachable states evolve with bigger parameters, we now prove that any monotonous register protocol has a cut-off for almost-sure reachability.

Thanks to Lemma 7.22, we know that for a given parameter n, $\mathbb{P}_n(\Diamond U_f) = 1$ if, and only if, $\mathrm{Post}^*_{U_f}(\Gamma_n \cap U_0) \subseteq \mathrm{Pre}^*(U_f)$, that is if, from any reachable configuration, there exists a finite suffix that can still reach U_f. A positive cut-off for almost-sure reachability corresponds to a parameter N above which the preceding inclusion always holds. Moreover, since the number of state is preserved over time, $\mathrm{Post}^*_{U_f}(\Gamma_n \cap U_0) = \mathrm{Post}^*_{U_f}(U_0) \cap \Gamma_n$, which means we can start computing $\mathrm{Post}^*_{U_f}(U_0)$ as an upward closed set, and later compute, for any n, the set $\mathrm{Post}{U_f}^*(\Gamma_n \cap U_0)$.

Definition 9.2. We let $U, U' \subseteq \Gamma$ be two upward closed sets (for $\preceq$). We say that U is *ultimately included* in U', noted $U \sqsubseteq U'$, whenever there exists $N \in \mathbb{N}$ such that

$$\forall k \geq N.\ U \cap \Gamma_k \subseteq U'$$

From the previous remarks, $\mathrm{Post}^*_{U_f}(U_0) \sqsubseteq \mathrm{Pre}^*(U_f)$ is equivalent to the existence of a positive almost-sure reachability cut-off for U_f. However, it is not obvious from the given definition that ultimate inclusion is decidable. We give in the following lemma a more concrete characterization, based on the structure of upward closed sets. Indeed, in order to check inclusion of two upward closed sets U and U', we can reason on their minimal elements: $U \subseteq U'$ if, and only if, for all $\gamma \in \min U$, there exists $\gamma' \in \min U'$ such that $\gamma' \preceq \gamma$. Graphically, several cases of ultimate inclusion can occur, as depicted in Figure 9.2. For example, inclusion may not hold for small parameters, but can still hold after increasing the parameter value. Let us denote $\Gamma_{\geq N} = \cup_{n \geq N} \Gamma_n$ the upward closed set of configurations of size larger than N. Then, for any N, $U \cap \Gamma_{\geq N}$ is upward closed and we can compute its minimal elements recursively, starting from $U = U \cap \Gamma_{\geq 0}$: for any $\gamma \in \min U \cap \Gamma_{\geq N}$, γ has size at least N, and

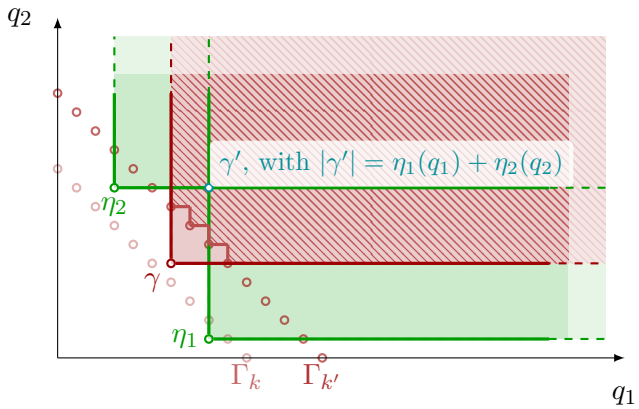


Figure 9.3 – Illustration of Lemma 9.3 on $U \sqsubseteq U'$ in the case where $U = \uparrow \{\gamma\}$ with $\overline{\gamma} = \{q_1, q_2\}$ and γ has two uncomparable minimal elements in U', η_1 and η_2. When k increases, the frontier of $\Gamma_{\geq k}$ is the diagonal Γ_k. Eventually, this frontier will overpass γ (for example for k'). If ultimate inclusion occurs, it occurs at most at the point of coordinate $(\eta_1(q_1), \eta_2(q_2))$ which corresponds to a configuration γ' of size/diagonal $\eta_1(q_1) + \eta_2(q_2) \leq \sharp\,[\overline{\gamma}] \cdot \max(|\eta_1|, |\eta_2|) \leq \sharp\,[Q_c] \cdot |U'|$.

- If $|\gamma| > N$, then γ is still a minimal element of $U \cap \Gamma_{\geq N+1}$,
- If $|\gamma| = N$, then γ cannot be a minimal element of $U \cap \Gamma_{\geq N+1}$ as $\gamma \notin \Gamma_{\geq N+1}$, however any configuration $\gamma \oplus q$ for any $q \in \overline{\gamma}$ is still part of $U \cap \Gamma_{\geq N+1}$. If there exists another element γ' in this set such that $\gamma' \preceq \gamma \oplus q$, then we should have $|\gamma'| = |\gamma \oplus q| = N + 1$ hence $\gamma' = \gamma \oplus q$ is minimal of size $N + 1$.

As the parameter N increases, it catches up the size of all minimal elements which eventually all fall down into the second category of minimal elements of size exactly N. These elements are updated by duplicating at each step any appearing state. The question remains then to determine whether these successive duplications of arbitrary appearing states can eventually lead to inclusion, or not. The next lemma states that one can consider one single state of the support to be duplicated. Moreover, we can bound the parameter to consider when looking at ultimate inclusion $U \sqsubseteq U'$, by a polynomial in the size of U', as graphically shown in Figure 9.3.

Lemma 9.3. *$U \sqsubseteq U'$ if, and only if, for any $\gamma \in \min U$ and $q \in \overline{\gamma}$, there exists $k \in \mathbb{N}$ such that $\gamma \oplus k \cdot q \in U'$. Moreover, when this is the case, we have $\forall k \geq \sharp\,[Q_c] \cdot |U'| \quad U \cap \Gamma_k \subseteq U'$*

Proof.
- ($\Rightarrow$) parameter $k = N$ from the definition is sufficient.
- ($\Leftarrow$) Let $\gamma \in \min U$, for each $q \in \overline{\gamma}$, we define $k(\gamma, q) \in \mathbb{N}$ such that $\gamma \oplus k(\gamma, q) \cdot q \in U'$. This means there exists $\eta \in \min U'$ such that $\eta \preceq \gamma \oplus k(\gamma, q) \cdot q$. Without loss of generality we can assume $k(\gamma, q)$ minimal, that is to say $k(\gamma, q) = \eta(q) - \gamma(q) \leq |U'| - \gamma(q)$. Let us define $k(\gamma) = \sum_{q \in \overline{\gamma}} k(\gamma, q) \leq \sharp\,[Q_c] \cdot |U'| - |\gamma|$.

  Consider now $\gamma \in U \cap \Gamma_k$ for $k \geq \sharp\,[Q_c] \cdot |U'|$. There exists $\gamma' \in \min U$ such that $\gamma' \preceq \gamma$.

Hence,

$$\sum_{q\in\overline{\gamma'}} \big(\gamma(q) - \gamma'(q)\big) = |\gamma| - |\gamma'| \geq \sharp\,[Q_c] \cdot |U'| - |\gamma'| \geq \sum_{q\in\overline{\gamma'}} \big(k(\gamma', q)\big)$$

so there necessarily exists $q \in \overline{\gamma'}$ such that $\gamma(q) - \gamma'(q) \geq k(\gamma', q)$. It follows $\gamma' \preceq \underbrace{\gamma' \oplus k(\gamma', q) \cdot q}_{\in U'} \preceq \gamma$.

□

Corollary 9.4. *If* $\mathrm{Post}^*_{U_f}(U_0) \sqsubseteq \mathrm{Pre}^*(U_f)$*, then* $\sharp\,[Q_c] \cdot |\mathrm{Pre}^*(U_f)|$ *is a positive cut-off for almost-sure reachability for* U_f*. Otherwise,* $|\mathrm{Post}^*_{U_f}(U_0)|$ *is a negative cut-off for almost-sure reachability for* U_f*.*

Proof. Using previous Lemma 9.3, we derive the correct bound in the positive case.

Otherwise, there exists $\gamma \in \min \mathrm{Post}^*_{U_f}(U_0)$ and $q \in \overline{\gamma}$ such that for any $k \in \mathbb{N}$, $\gamma \oplus k \cdot q \notin \mathrm{Pre}^*(U_f)$. Hence, $|\gamma| \leq |\mathrm{Post}^*_{U_f}(U_0)|$ is a negative cut-off since for all $k \geq |\gamma|$, $\gamma_k = \gamma \oplus (k - |\gamma|) \cdot q \in \mathrm{Post}^*_{U_f}(U_0) \cap \Gamma_k \backslash \mathrm{Pre}^*(U_f)$. □

9.3 Bound examples

The previous proof of almost-sure reachability cut-off existence is giving bounds on a possible cut-off, which are polynomial in $|\mathrm{Post}^*_{U_f}(U_0)|$ and $|\mathrm{Pre}^*(U_f)|$. Such values are interesting to compute as they allow us to solve the decision problem: it is enough to check almost-sure reachability for a fixed parameter equal to the cut-off. However, the given bounds can be *a priori* large. For example, when U_f is a coverable target, one can apply Petri Net bounds on coverability due to Rackoff [Rac78] and expect $|\mathrm{Pre}^*(U_f)|$ to be bounded by a double exponential in $|\mathcal{P}|$.

However, no general bound on an arbitrary upward closed set, especially for $\mathrm{Post}^*_{U_f}(U_0)$, can be given directly.

Before diving into the decision problem, we can wonder what optimal bounds on the cut-off we can expect. In order to do so, we will consider some instances and compute the minimal parameter (lower bound) which is still a cut-off, namely a *tight* cut-off.

9.3.1 Linear cut-off

We start our study with a family of leaderless non-atomic register protocols $(\mathcal{F}_n)_{n>0}$, depicted in Figure 9.4. For a fixed n, protocol $\mathcal{F}_n$ has $n+1$ states and n different data; intuitively, in order to move from s_i to s_{i+1}, two processes are needed: one writes i in the register and goes back to s_0, and the second process can proceed to s_{i+1} by reading i. Since backward transitions to s_0 are always possible and since states can always exit s_0 by writing a 0 and reading it afterwards, no deadlock can ever occur so the main question remains to determine if s_n is reachable by one of the processes as we increase the number of initial processes. As shown in Lemma 9.5, the answer is positive: $\mathcal{F}_n$ has a tight linear positive cut-off for coverability of s_n; it actually behaves like a "filter", that can test if at least n processes are running together. We exploit this property later in the next section.

Lemma 9.5. *Fix $n \in \mathbb{N}$. The "filter" protocol $\mathcal{F}_n$, depicted in Figure 9.4, has a tight positive cut-off for almost-sure reachability of $U_f = \{\gamma \mid \gamma(s_n) > 0\}$, equal to n.*

Proof. Let us consider the reachable configurations for m contributor processes in $\mathcal{F}_n$. We first prove that any reachable configuration $\gamma \in \mathrm{Post}^*_{U_f}(U_0) \cap \Gamma_m$ satisfies:

$$\forall j \leq m.\ \sum_{k=0}^{j} \gamma(s_k) \geq j + \mathbb{1}_{\nu(\gamma)=j+1}$$

The proof is by induction: the invariant is satisfied by the initial configuration $\gamma_0 = (s_0^m, 0)$. Let us now consider the run $\gamma_0 \to^* \gamma \to \gamma'$, in which γ satisfies the invariant, and with last transition $(q, \mathbf{f}, q') \in T_c$.

- If $\mathbf{f} = \mathbf{R}(0)$, then $q = s_0$ and $q' = s_1$. Along that transition, the right-hand-side term is unchanged; so is the left-hand-side term as soon as $j > 0$, so that the inequality is preserved for those cases. The case $j = 0$ is trivial.

- If $\mathbf{f} = \mathbf{R}(i)$ with $i > 0$, then $q = s_i$ and $q' = s_{i+1}$. We have $\mathrm{st}(\gamma') = \mathrm{st}(\gamma) \ominus s_i \oplus s_{i+1}$ and $\nu(\gamma') = \nu(\gamma) = i$. Again, along this read-transition, the right-hand side term is unchanged, while the left-hand-side term is unchanged for all $j \neq i$.

  It remains to prove the inequality for $j = i$. We apply the induction hypothesis in γ for $j = i-1$: since the transition $(q, \mathbf{R}(i), q')$ is available, it must hold that $\mathrm{st}(\gamma)(s_i) \geq 1$ and $\nu(\gamma) = i = j + 1$. Hence $\sum_{k=0}^{i-1} \mathrm{st}(\gamma)(s_k) \geq i-1+1 = i$, and $\sum_{k=0}^{i} \mathrm{st}(\gamma)(s_k) \geq i + 1$. This implies $\sum_{k=0}^{i} \mathrm{st}(\gamma')(s_k) \geq i$.

- If $\mathbf{f} = \mathbf{W}(i)$, then $q = s_i$ and $q' = s_0$. Thus $\mathrm{st}(\gamma') = \mathrm{st}(\gamma) \ominus s_i \oplus s_0$. For $j = i - 1$, the left-hand-side term of the inequality is increased by 1, while the right-hand-side one is either unchanged or also increased by 1. The property is preserved in both cases. For $j \neq i - 1$, the left-hand-side term cannot decrease, while the right-hand-side term cannot increase. Hence the invariant is preserved.

As a consequence, if $m < n$, we have (for $j = m$) $\sum_{k=0}^{m} \mathrm{st}(\gamma)(s_k) = m$ for any reachable configuration γ, so that $\mathrm{st}(\gamma)(s_n) = 0$.

Conversely, if $m \geq n$, from any configuration γ and for any $0 \leq i < n$, it is possible to reach $\gamma_i = (i \cdot s_0 \oplus (m - i) \cdot s_{i+1}, i)$:

- for $i = 0$: all processes can go to s_0, then write 0 in the register, and all move to s_1: $\gamma \to^* (m \cdot s_0, d) \xrightarrow{\mathbf{W}(0)} (m \cdot s_0, 0) \xrightarrow{\mathbf{R}(0)}{}^m (m \cdot s_1, 0)$;

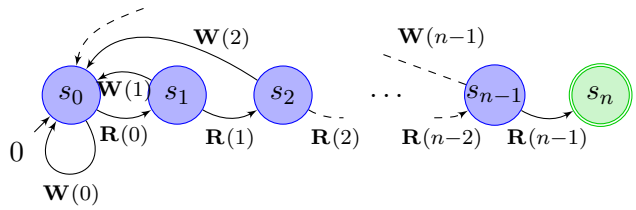


Figure 9.4 – A "filter" protocol $\mathcal{F}_n$ for $n > 0$.

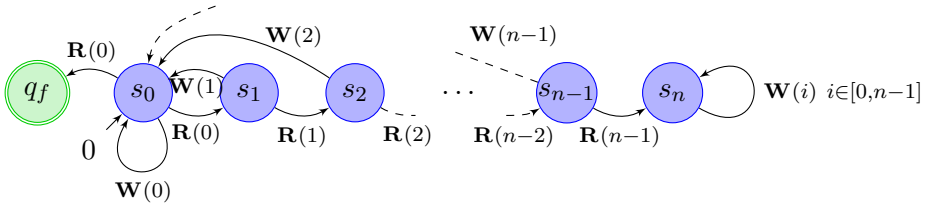


Figure 9.5 – The "reversed filter" protocol $\overline{\mathcal{F}}_n$ for $n > 0$, with negative cut-off for almost-sure reachability n.

- for $1 < i < n-1$, assuming $(i \cdot s_0 \oplus (m-i) \cdot s_{i+1}, i)$ can be reached, one of the processes in s_{i+1} can write $i+1$ (going back to s_0), and the remaining $m-i-1$ processes in s_{i+1} can go to s_{i+2}:

$$\begin{aligned}(i \cdot s_0 \oplus (m-i) \cdot s_{i+1}, i) &\xrightarrow{\mathbf{W}(i+1)} && ((i+1) \cdot s_0 \oplus (m-i-1) \cdot s_{i+1}, i+1) \\ &\xrightarrow{\mathbf{R}(i+1)}{}^{m-i-1} && ((i+1) \cdot s_0 \oplus (m-i-1) \cdot s_{i+2}, i+1)\end{aligned}$$

Thus from any $\gamma \in \Gamma$, configuration $\gamma_{n-1} = ((n-1) \cdot s_0 \oplus (m-n+1) \cdot s_n, n-1)$ is reachable. Furthermore, γ_{n-1} contains the final state s_n since $m \geq n$.

Hence, we deduce that there is a unique bottom strongly-connected component in Γ_m, and that γ_{n-1} belongs to it: this configuration is reached with probability 1 from $(m \cdot s_0, 0)$. It follows that $\mathbb{P}_m(\Diamond U_f) = 1$. □

9.3.2 Counter machine

We are now interested in finding non-trivial tight negative cut-offs. One way to build one such corresponding register protocol is to encode a mechanism able to count processes, and proceed to *deadlock* the system, avoiding reaching a target state q_f, if enough processes are present. For example, the previous filter protocol can count in unary up to n with $n+1$ states. We now modify the protocol as depicted in Figure 9.5: the previous target state s_n for coverability is modified in order to allow the other processes to join in s_n. The target state for coverability, denoted q_f is only reachable from s_0. Thus, as soon as one process has reached s_n, there is positive probability that all processes join s_n, and stay "trapped" in this location. This results in a linear negative cut-off ($n+2$ states for cut-off value n).

A natural idea we develop below is to count in binary, namely encoding a binary counter by several processes, each of them (en)coding one bit of the current value. One main problem that arises with such encoding is the coherence of our counter, as we need to avoid several states to encode the same bit with different values at the same time. Intuitively, such property cannot be enforced *a priori*, as stated by the following lemma, related to the monotonicity property.

Lemma 9.6. *Assume $\mathcal{P}$ is leaderless non-atomic. If $\pi_1 : (q, \mu_1, d) \to^* (q, \mu_1', d_1)$ and $\pi_2 : (q, \mu_2, d) \to^* (q, \mu_2', d_2)$ then $\forall d' \in \{d_1, d_2\}$, $(q, \mu_1 \oplus \mu_2, d) \to^* (q, \mu_1' \oplus \mu_2', d')$.*

Proof. We prove the result by induction on $|\pi_1| + |\pi_2|$.

- If $\pi_1 = \varepsilon$, the result is immediate. From now on, we assume $|\pi_1| \geq 1$ and (by symmetry) $|\pi_2| \geq 1$.

- If π_1 or π_2 starts with a read operation, we can apply this first transition and apply induction hypothesis on the resulting configuration directly (shared register value hasn't changed).
- If both π_1 and π_2 start with a write operation, then by monotonicity, $(q, \mu_1 \oplus \mu_2, d) \to^* (q, \mu_1 \oplus \mu'_2, d_2)$ and $(q, \mu_1 \oplus \mu'_2, d) \to^* (q, \mu'_1 \oplus'_2, d_1)$. This last path starts with a write operation, so value register d can be replaced by d_2 hence the result.

□

One way to tackle this issue consists in detecting, *after* the binary count, that effectively no more that one process encoded each counter bit, thanks to our previous filter protocol for positive cut-off. A general pattern for encoding a counting mechanism can be described by three phases, also depicted in Figure 9.6.

- *Initialization*: the initial state allows processes to choose randomly which bit position they want to encode, or if they want to be part of the counted processes (*tokens*). After the first write, the initial register value is erased and transitions from this phase cannot occur anymore. Denote with k_i (resp. k) the number of processes encoding the i-th bit (resp. the tokens).
- *Simulation*: Processes of the binary counter exchange the addition remainder thanks to the shared register. In order to enable the counting mechanism, one of the processes (tokens) to be counted perform a write transition, writing initial remainder 1, allowing the counter to progress. During the whole simulation phase, each state has positive probability to take an extra transition to the final target state.
- *Check*: Instead of writing a remainder for bit $n+1$, the process encoding bit n writes a final value `halt`, meaning the counter managed to count, with possible errors, at least 2^n tokens, allowing counter processes to reach the filter module of parameter $n+1$. This last filter has no more probabilistic transition to the target state so the only way to reach it is to play according to the filter gadget rules. When each bit of the counter is encoded by one single process ($\forall i\ k_i = 1$), the total amount of processes encoding bits is n so the final state cannot be reached and the counting is correct so $k \geq 2^n$.

Notice that the whole pattern is meant to build a negative cut-off, as any possible error in the pattern will lead to reaching the target state: not enough encoding processes ($\exists i\ k_i = 0$), not enough tokens ($k < 2^n$) or too many encoding processes ($\sum_i k_i \geq n+1$) ensure reaching q_f almost-surely; the only run that avoids q_f is the one that respects the three described phases.

Theorem 9.7. *There exists a family of leaderless non-atomic register protocols, which admit negative tight cut-offs for almost-sure reachability with coverability objective whose value is exponential in the size of the protocol.*

Proof. We first focus on the first part of the protocol of Figure 9.6, containing nodes named a_i, b_i, c_i, d_i and s_i. This part can be divided into three phases: the initialization phase lasts as long as the register contains #; the counting phase starts when the register contains `halt` for the first time; the simulation phase is the intermediate phase.

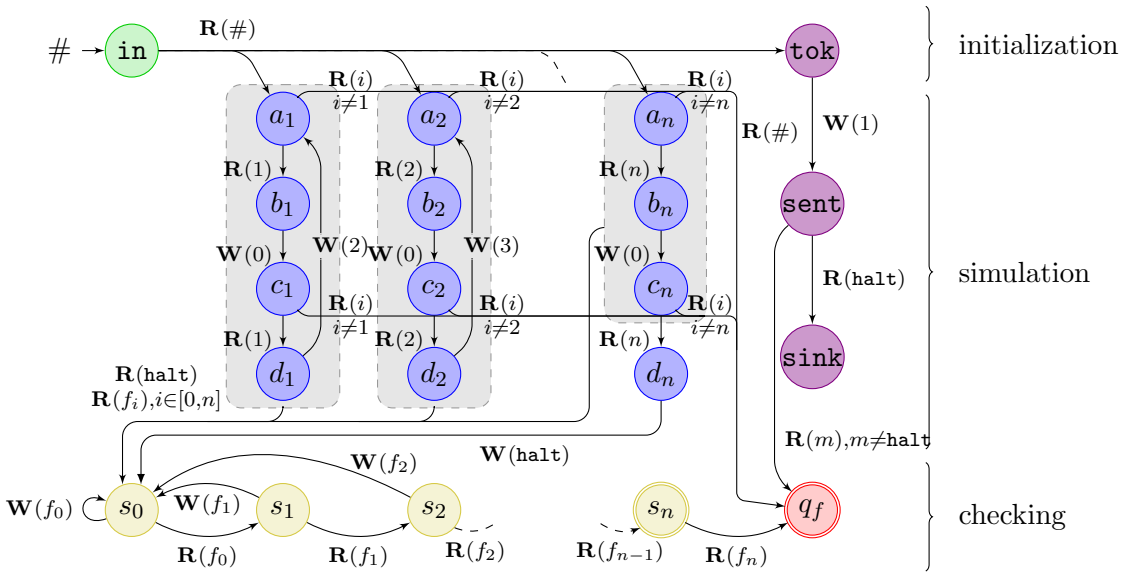


Figure 9.6 – Simulating an exponential counter: grey boxes contain the nodes used to encode the bits of the counter; yellow nodes at the bottom correspond to the filter module from Figure 9.4; purple nodes `tok`, `sent`and `sink`correspond to the second part of the protocol, and are used to produce tokens. Missing *read* edges are assumed to be self-loops.

During the initialization phase, processes move to locations a_i and `tok`, until some process in `tok`writes 1 in the register (or until some process reaches q_f, using a transition from a_i to q_f while reading #). Write γ_0 for the configuration reached when entering the simulation phase (i.e., when 1 is written in the register for the first time). We assume that $\mathrm{st}(\gamma_0)(a_i) > 0$ for some i, as otherwise all the processes are in `tok`, and they all will eventually reach q_f. Now, we notice that if $\mathrm{st}(\gamma_0)(a_i) = 0$ for some i, then location d_n cannot be reached, so that no process can reach the counting phase. In that case, some process (and actually all of them) will eventually reach q_f. We now consider the case where $\mathrm{st}(\gamma_0)(a_i) \geq 1$ for all i. One can prove (inductively) that d_i is reachable when $\mathrm{st}(\gamma_0)(\texttt{tok}) \geq 2^i$. Hence d_n, and thus also s_0, can be reached when $\mathrm{st}(\gamma_0)(\texttt{tok}) \geq 2^n$. Assuming q_f is not reached, the counting phase must never contain more than n processes, hence we actually have that $\mathrm{st}(\gamma_0)(a_i) = 1$. With this new condition, s_0 is reached if, and only if, $\mathrm{st}(\gamma_0)(\texttt{tok}) \geq 2^n$. When the latter condition is not true, q_f will be reached almost-surely, which proves the second part of our claim: the final location is reached almost-surely in systems with strictly less than $n + 2^n$ copies of the protocol.

We now consider the case of systems with at least $n + 2^n$ processes. We exhibit a finite execution of those systems from which no continuation can reach q_f, thus proving that q_f is reached with probability strictly less than 1 in those systems. The execution is as follows: during initialization, for each i, one process enters a_i; all other processes move to `tok`, and one of them write 1 in the register. The n processes in the simulation phase then simulate the consecutive incrementations of the counter, consuming one token at each step, until reaching d_n. At that time, all the processes in `tok` move to `sent`, and the process in d_n writes `halt` in the register and enters s_0. The processes in the simulation phase can then enter s_0, and those in `sent` can move to `sink`. We now have n processes in s_0, and the other

ones in `sink`. According to Lemma 9.5, location q_f cannot be reached from this configuration, which concludes our proof. □

Remark 9.8. *The question whether there exists protocols with exponential* positive *cut-offs remains open. The family of* filter *protocols described previously is (only) an example of protocols with a linear tight positive cut-off.*

9.3.3 PSPACE-hardness

The previous idea of checking afterwards whether a simulation was correct can also be exploited to encode a linear-bounded Turing machine [Sip97] instead of a simple binary counter. This technique is developed below to establish PSPACE-hardness for the cut-off problem.

We build a register protocol for which there is a negative cut-off for almost-sure coverability, if, and only if, the machine reaches its final state q_{halt} with the tape head reading the last cell of the tape.

Write n for the size of the tape of the Turing machine. We assume (without loss of generality) that the machine is deterministic, and that it accepts only if it ends in its halting state $q_{\mathtt{halt}}$ while reading the last cell of the tape. Our reduction works as follows (see Fig. 9.7): some processes of our network will first be assigned an index i in $[1; n]$ indicating the cell of the tape they shall encode during the simulation. The other processes are stuck in the initial location, and will play no role. The state q and position j of the head of the Turing machine are stored in the register. During the simulation phase, when a process is scheduled to play, it checks in the register whether the tape head is on the cell it encodes, and in that case it performs the transition of the Turing machine. If the tape head is not on the cell it encodes, the process moves to the target location (which we consider as the target for the almost-sure reachability problem). Finally, upon seeing $(q_{\mathtt{halt}}, n)$ in the register, all processes move to a $(n+1)$-filter protocol $\mathcal{F}_{n+1}$ (similar to that of Fig. 9.4) whose last location s_{n+1} is the aforementioned target location.

If the Turing machine halts, then the corresponding run can be mimicked with exactly one process per cell, thus giving rise to a finite run of the distributed system where n processes end up in the $(n+1)$-filter (and the other processes are stuck in the initial location); from there s_{n+1} cannot be reached. If the Turing machine does not halt, then assume that there is an infinite run of the distributed system never reaching the target location. This run cannot get stuck in the simulation phase forever, because it would end up in a strongly connected component from which the target location is reachable. Thus this run eventually reaches the $(n+1)$-filter, which requires that at least $n+1$ processes participate in the simulation (because with n processes it would simulate the exact run of the machine, and would not reach $q_{\mathtt{halt}}$, while with fewer processes the tape head could not go over cells that are not handled by a process). Thus at least $n+1$ processes would end up in the $(n+1)$-filter, and with probability 1 the target location should be reached.

We now formalize this construction, by describing the states and transitions of the protocol within these three phases. We fix a linear-bounded Turing machine $\mathcal{M} = (Q, q_0, q_{\mathsf{halt}}, \Sigma, \delta)$, where Q is the set of states, $q_0, q_{\mathsf{halt}} \in Q$ are the initial and halting states, Σ is the alphabet, and $\delta \subseteq Q \times \Sigma \times Q \times \Sigma \times \{-1, +1\}$ is the set of transitions. We define the data alphabet $D = \{\#\} \uplus Q \times \Sigma \uplus \{f_i \mid 0 \leq i \leq n\}$, and the set of locations $P = \{p_{\mathsf{init}}, p'_{\mathsf{init}}, p_{\mathsf{sink}}\} \uplus \big([1; n] \times \Sigma \times (Q \cup \{\varepsilon\})\big) \uplus \{s_i \mid 0 \leq i \leq n+1\}$. The set of locations corresponds to three phases (see Fig. 9.7):

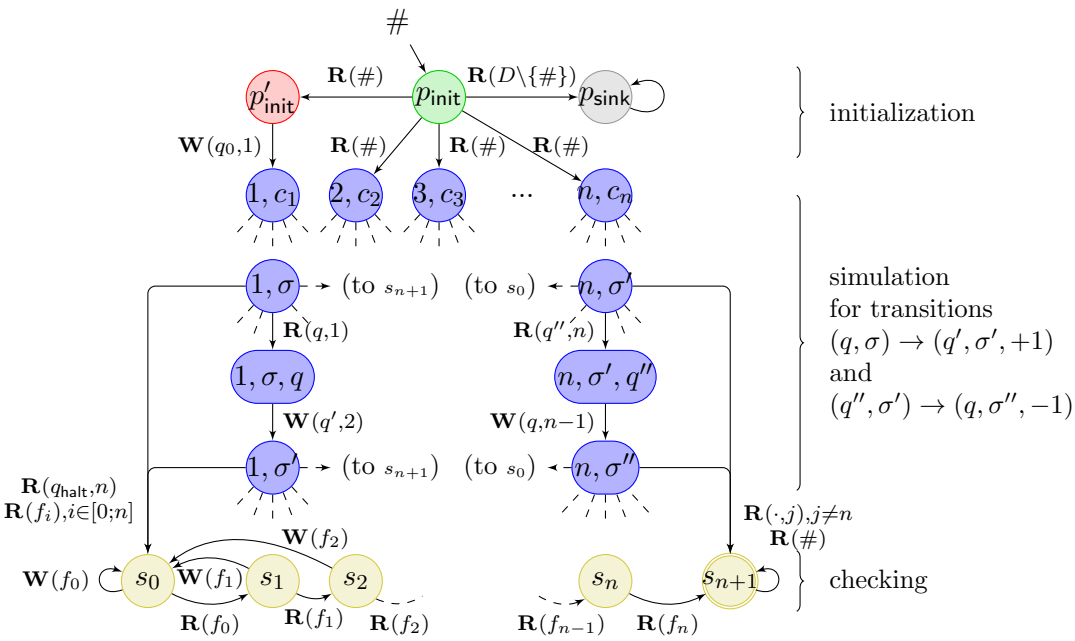


Figure 9.7 – Distributed protocol $\mathcal{P}_{\mathcal{M}}$ encoding the linear-bounded Turing machine $\mathcal{M}$.

- The initialization phase contains p_{init}, p'_{init} and p_{sink}. From the initial state p_{init}, upon reading # (the initial content of the register), the protocol has transitions to each state (i, σ_i) for all $2 \leq i \leq n$, where σ_i is the i-th letter of the initial content of the tape. If reading anything different from #, the protocol moves to the sink state p_{sink}. Finally, there are transitions $(p_{\mathsf{init}}, \mathbf{R}(\#), p'_{\mathsf{init}})$ and $(p'_{\mathsf{init}}, \mathbf{W}(q_0, 1), (1, c_1))$, where q_0 is the initial state of the Turing machine: this pair of transitions is used to initialize the computation, by setting the content of the first cell and modifying the register, so that the initialization phase is over (there are no transitions writing # in the register).

- The second phase, called *simulation phase*, uses register alphabet $Q \times [1, n]$, in order to encode the state and position of the head of the Turing machine. The state space for the simulation phase is $[1; n] \times \Sigma \times (Q \cup \{\varepsilon\})$: state (i, σ, ε) (written (i, σ) in the sequel) encodes the fact that the content of the i-th cell is σ; the states of the form (i, σ, q) are intermediary states used during the simulation of one transition: when in state (i, σ) and reading (q, i) in the register, the protocol moves to (i, σ, q), from which it moves to (i, σ') and writes (q', j) in the register, provided that the machine has a transition $(q, \sigma) \to (q', \sigma', j - i)$. If the active process does not encode the position that the tape head is reading (i.e., the process is in state (i, σ) and reads (q, j) with $j \neq i$) then it moves to the final state s_{n+1} of the third phase.

- The role of the *counting phase* is to count the number of processes participating in the simulation. When seeing the halting state in the register, each protocol moves to a module whose role is to check whether at least $n+1$ protocols are still "running". This uses data $\{f_i \mid 0 \leq i \leq n\}$ and states $\{s_i \mid i \in [0, n+1]\}$, with transitions from any state of the simulation phase to s_0 if the register contains (q_{halt}, n) or any of $\{f_i \mid 0 \leq i \leq n\}$.

We now prove that our construction is correct:

Lemma 9.9. *The register protocol $\mathcal{P}_{\mathcal{M}}$, with coverability objective s_{n+1}, has a negative cut-off if, and only if, the Turing machine $\mathcal{M}$ reaches q_{halt} in the last cell of the tape.*

Proof. First assume that there is a negative cut-off: there exists N_0 such that for any $N \geq N_0$, starting from the initial configuration $(p_{\mathsf{init}}^N, \#)$ of the system $(\Gamma_N, \rightarrow)$ made of N copies of $\mathcal{P}_{\mathcal{M}}$, the probability that at least one process reaches s_{n+1} is strictly less than 1. Since $(\Gamma_N, \rightarrow, p)$ is a finite Markov chain, this implies that there is a cone of executions never visiting s_{n+1}, i.e., a finite execution ρ whose continuations never visit s_{n+1}. Since the register initially contains #, this finite execution (or a finite continuation of it) must contain at least one configuration where some process has entered the simulation part.

Now, in the simulation phase, we notice that, right after taking a transition $((i, \sigma, q), \mathbf{W}(q', i \pm 1), (i, \sigma'))$, the transition $((i, \sigma'), \mathbf{R}(\cdot, j), s_{n+1})$ is always enabled. It follows that at the end of the finite run ρ, no simulation transition should be enabled; hence all processes that had entered the simulation part should have left it. Hence some process must have visited s_0 along ρ (because we assume that ρ does not involve s_{n+1}). Moreover, by Lemma 9.5, for s_{n+1} not to be reachable along any continuation of ρ, no more than n processes must be able to reach s_0 along any continuation of ρ, hence at most n processes may have entered the simulation phase. On the other hand, for s_0 to be visited, some process has to first write (q_{halt}, n) in the register; since the register initially contains $(q_0, 1)$, and no process can write $(\cdot, i+1)$ without first reading $(\cdot, i)$, then for each $i \in [1, n]$ there must be at least one process visiting some state (i, σ_i), for some σ_i; It follows that at least n processes must have entered the simulation phase.

In the end, along ρ, exactly one process visits (i, c_i), for each $i \in [1, n]$, and encode the content of the i-th cell. As a consequence, along ρ, each cell of the tape of the Turing machine is encoded by exactly one process, and the execution mimics the exact computation of the Turing machine. Since the configuration (q_{halt}, n) is eventually reached, the Turing machine halts with the tape head on the last cell of the tape.

Conversely, assume the Turing machine halts, and consider the execution of $N \geq n$ processes where exactly one process goes in each of the (i, c_i) and mimics the run of the Turing machine (the other processes going to p_{sink}). We get a finite execution ending up in a configuration where all processes are either in p_{init} or in p_{sink}, except for n processes that are in the counting phase. No continuation of this prefix ever reaches s_{n+1}, so that the probability that some process reaches s_{n+1} is strictly less than 1. □

We conclude by stating the following complexity results:

Theorem 9.10. *The following problems are* PSPACE*-hard.*

<u>ASCoverCutOff</u>

INPUT: *A non-atomic register protocol $\mathcal{P}$ and a final state $q_f \in Q_c$.*
QUESTION: *Whether there exists a cut-off for almost-sure reachability of $U_f = \{\gamma \mid \gamma(q_f) > 0\}$.*

<u>ASConsensusCutOff</u>

INPUT: *A non-atomic register protocol $\mathcal{P}$ and a final state $q_f \in Q_c$.*
QUESTION: *Whether there exists a cut-off for almost-sure reachability of $U_f = \{\gamma \mid \gamma(q_f) = |\gamma|\}$.*

9.4 Decision procedure

As previous Section 9.2 proved that the existence of a cut-off is ensured, the question of the nature of cut-off (positive or negative) arises. Answering this question boils down to deciding the asymptotic behaviour of our system, when the parameter is large enough.

One naive approach to this problem consists in computing a cut-off value as given by Corollary. 9.4, then simulating the system to decide whether almost-sure reachability holds for this particular parameter. However the corresponding value is polynomial in both the size of the $\mathrm{Pre}^*(U_f)$ and $\mathrm{Post}^*_{U_f}(U_0)$ minimal elements, for which we have no bound *a priori*. Notice however that the theory of well-quasi-orders ensures that these elements can be computed in finite time. For example, $\mathrm{Post}^*_{U_f}(U_0)$ can be computed by saturation, by considering the sequence $(X_n)_n$, with $X_0 = U_0$ and $\forall n\ X_{n+1} = X_n \cup \mathrm{Post}_{U_f}(X_n)$. For each n, X_n is an upward closed set, and its minimal elements can be computed iteratively. Moreover, this sequence is non-decreasing for inclusion, and is eventually stationary; otherwise, we could extract an infinite sequence of minimal elements which are incomparable, which would be contradictory with the wqo structure.

On the other hand, our previous analysis gives us some hints about the complexity of a decision procedure. First of all, Theorem 9.10 tells us the problem is PSPACE-hard. Then, our different constructions show examples of tight linear and exponential cut-off, so we can expect the minimal elements to be at least of the same size (up to a polynomial).

The following sections show how to deal with coverability objectives, which allow the usage of already known bounds; then we proceed to develop general techniques for any objectives. In particular, we will be able to derive upper bounds on tight cut-off values.

9.4.1 Refined symbolic graph

As the symbolic graph revealed to be insufficient to analyze almost-sure reachability, we consider now a refinement where it can keep track of up to a fixed portion of configurations.

In order to do so, we first introduce a new protocol that encodes part of contributors' configuration inside the leader structure. We will then derive results from the analysis of its resulting symbolic graph.

Definition 9.11 (Fixed protocol)**.** Let $\mathcal{P} = (Q_l, Q_c, D, d_0, q_l, q_c, T_l, T_c)$, a register protocol, and $k \in \mathbb{N}$, we define $\widetilde{\mathcal{P}}^k$ by $\widetilde{\mathcal{P}}^k = (Q'_l, Q_c, D, d_0, q'_l, q_c, T'_l, T_c)$, where

- $Q'_l = Q_l \times \mathbb{N}_k^{Q_c}$,
- $q'_l = (q_l, q_c^k) \in Q_l$
- $T'_l = \left\{((q,\mu), \mathbf{f}, (q',\mu)) \mid (q, \mathbf{f}, q') \in T_l,\ \mu \in \mathbb{N}_k^{Q_c}\right\} \cup$
  $\left\{((q,\mu), \mathbf{f}, (q,\mu')) \mid \exists q_t \in \overline{\mu}, q'_t \in \overline{\mu'}\ (q_t, \mathbf{f}, q'_t) \in T_c \wedge \mu' \oplus q_t = \mu \oplus q'_t\right\}$

Intuitively, fixed protocols of index k encodes an initial protocol with at least k processes, that are taken apart and considered as part of the leader process. This isolation technique will reveal to be convenient to develop decision procedures and bounding techniques that couldn't be achieved directly by the symbolic graph, which cannot keep track of any quantitative number of processes. As one may notice, non-atomic, monotonous, atomic properties on register protocols are preserved, but not the leaderless property. We can also easily check the following translation lemma:

Lemma 9.12 (Translation)**.** *Let $q, q' \in Q_l$, $\mu_1, \mu_1' \in \mathbb{N}_k^{Q_c}$ and $\mu_2, \mu_2' \in \mathbb{N}_n^{Q_c}$. Then,*

$$((q, \mu_1), \mu_2, d) \to ((q', \mu_1'), \mu_2', d') \quad \text{in } \widetilde{\mathcal{P}}^k$$

if, and only if,

$$(q, \mu_1 \oplus \mu_2, d) \to (q', \mu_1' \oplus \mu_2', d') \quad \text{in } \mathcal{P}$$

Definition 9.13 (Fixed subset)**.** Let $A \subseteq \Gamma$, we define

$$\widetilde{A}^k = \left\{ ((q, \mu_1), \mu_2, d) \;\middle|\; \mu_1 \in \mathbb{N}_k^{Q_c} \wedge (q, \mu_1 \oplus \mu_2, d) \in A \right\}$$

As we can expect, for any upward closed set of configurations U of $\mathcal{P}$, $\widetilde{U}^k$ is still upward closed, although the number of minimal elements undergo an exponential blow up. Note however that minimal elements have similar sizes: $|U| = |\widetilde{U}^k|$. Thanks to Lemma 9.12, we also infer that $\widetilde{\mathrm{Pre}^*(U)}^k = \mathrm{Pre}^*(\widetilde{U}^k)$.

9.4.2 Symbolic based algorithm

The main result of this section consists in showing that for a parameter large enough, the almost-sure reachability cut-off is preserved in the symbolic graph of the fixed protocol. More precisely, the parameter value $K = \sharp\,[Q_c] \cdot |\mathrm{Pre}^*(U_f)|$ will preserve almost-sure reachability. Notice that this quantity is polynomial in the size of the network and in the size of the predecessor set. Surprisingly, this quantity is independant of the sucessor set itself.

Lemma 9.14. *Assume $\mathcal{P}$ is monotonous. There is a negative cut-off for almost-sure reachability of U_f in $\mathcal{P}$, if, and only if, there is a node in the symbolic graph $G_{\widetilde{\mathcal{P}}^K}$ that is reachable from $((q_l, K \cdot q_c), \{q_c\}, d_0)$ but from which $\overline{\widetilde{U_f}^K}$ is not reachable.*

Proof. $\Leftarrow$ We apply Theorem 9.1 to show that $\widetilde{\mathcal{P}}^K$ has no positive cut-off. Thanks to Corollary. 9.4, a negative cut-off should then exist for $\widetilde{\mathcal{P}}^K$. We infer by translation Lemma 9.12 that $\mathcal{P}$ has also a negative cut-off.

$\Rightarrow$ Let N be a negative cut-off for almost-sure reachability of U_f in $\mathcal{P}$. We can assume without loss of generality that $N \geq K$, and consider $\gamma = (q, \mu, d) \in \Gamma_N \cap \left(\mathrm{Post}^*_{U_f}(U_0) \backslash \mathrm{Pre}^*(U_f)\right) = \mathrm{Post}^*_{U_f}(\Gamma_N \cap U_0) \backslash \mathrm{Pre}^*(U_f)$, which is non-empty thanks to Lemma 7.22.
Write $\{\eta_i \mid 1 \leq i \leq m\} = \{\eta \in \min \mathrm{Pre}^*(U_f) \mid \overline{\eta} = \overline{\gamma}\}$, for the finite subset of minimal elements of $\mathrm{Pre}^*(U_f)$ that have the same support as γ. Since $\gamma \notin \mathrm{Pre}^*(U_f)$, for each $1 \leq i \leq m$, $\eta_i \not\preceq \gamma$ and because of support equality, there exists $q^i \in \overline{\gamma}$ such that $\mu(q^i) = \gamma(q^i) < \eta_i(q^i) \leq |\mathrm{Pre}^*(U_f)|$. We conclude we can write $\mu = \mu_1 \oplus \mu_2'$ with $\overline{\mu_1} = \{q^i \mid 1 \leq i \leq m\}$ and $\overline{\mu_2'} = \overline{\mu} \backslash \overline{\mu_1}$. As $|\mu_1| < \sharp\,[Q_c] \cdot |\mathrm{Pre}^*(U_f)| = K \leq N = |\mu_1| + |\mu_2'|$, we can write $\mu_2' = \mu_2 \oplus \mu_3$ with $|\mu_2| = K - |\mu_1|$ and μ_3 contains the remaining $N - K \geq 1$ states neither taken in μ_1 nor in μ_2. Consider the configuration $((q, \mu_1 \oplus \mu_2), \mu_3, d)$ in $\widetilde{\mathcal{P}}^K$, which is reachable from $\widetilde{U_0}^K$ by translation Lemma 9.12, then its support $s = ((q, \mu_1 \oplus \mu_2), \overline{\mu_3}, d)$ is reachable in $G_{\widetilde{\mathcal{P}}^K}$ from $((q_l, K \cdot q_c), \{q_c\}, d_0)$. Assume that $sE^*\overline{\widetilde{U_f}^K}$ in the symbolic graph to expose a contradiction. Then, there exists μ_3' such that $\overline{\mu_3'} = \overline{\mu_3}$ and $((q, \mu_1 \oplus \mu_2), \mu_3', d) \to^* \gamma_f \in \widetilde{U_f}^K$ by symbolic correspondance (Lemma 8.3).

Then, $\gamma' = (q, \mu_1 \oplus \mu_2 \oplus \mu'_3, d) \to^* \gamma'_f \in U_f$, however, for each $1 \leq i \leq m$, $q^i \notin \overline{\mu_2 \oplus \mu'_3}$ so $\gamma'(q^i) = \mu_1(q^i) = \gamma(q^i) < \eta_i(q^i)$ hence $\eta_i \not\preceq \gamma'$ which is contradictory to $\gamma' \in \mathrm{Pre}^*(U_f)$. Hence, the hypothesis $sE^*\widetilde{U_f}^K$ was flawed.

□

This lemma gives a general approach to decide almost-sure reachability cut-off for any monotonous protocol, based on the exploration of a finite graph. The remaining challenge consists in analyzing the resulting complexity, which will depend on the exact system and objective considered.

9.4.3 Complexity bounds on covering

We consider first the complexity of the algorithm that can be deduced from Lemma 9.14, in the case of almost-sure reachability of a coverability target $U_f = \{\gamma \mid \gamma(q_f) > 0\}$. Using results by Rackoff on the coverability problem in Vector Addition Systems [Rac78], we can bound $|\mathrm{Pre}^*(U_f)|$ (then $K = \sharp[Q_c] \cdot |\mathrm{Pre}^*(U_f)|$) by a *double-exponential* in the size of the protocol.

Due to the symmetry abstraction that leads us to consider only multiset of states as configurations, the resulting symbolic graph of index K does not have another exponential blow up, as stated by Lemma 9.15. Therefore, it suffices to solve a reachability problem in NLOGSPACE [Sip97] on this doubly-exponential graph, that can be constructed on the fly: this boils down to NEXPSPACE complexity with regard to the protocol's size, hence EXPSPACE complexity, by Savitch's theorem [Sip97].

Lemma 9.15. *Let Q be a finite set. Then*

$$\left|\mathbb{N}_n^Q\right| = \binom{|Q| + n - 1}{n} \leq (2n)^{|Q|-1}$$

Proof. Note $Q = \{q_1 \dots q_k\}$, then the following function is a bijection

$$\begin{cases} \mathbb{N}_k^Q \to \left\{(i_1, \dots i_{k-1}) \in \mathbb{N}^{k-1} \;\middle|\; 1 \leq i_1 < i_2 < \dots < i_{k-1} \leq n + k - 1\right\} \\ \gamma \mapsto \left(j + \sum_{j' \leq j} \gamma(q_{j'})\right)_{1 \leq j < k} \end{cases}$$

Hence, $|\mathbb{N}_n^Q| = \binom{k+n-1}{k-1} = \frac{1}{(k-1)!} \prod_{i=1}^{k-1}(n+i) = \prod_{i=1}^{k-1} \frac{n+i}{i} = \prod_{i=1}^{k-1}\left(1 + \frac{n}{i}\right) \leq (2n)^{k-1}$ □

Theorem 9.16. *The following problem is in* EXPSPACE*:*
ASCoverCutOff

INPUT: *A non-atomic register protocol $\mathcal{P}$ and a final state $q_f \in Q_c$.*
QUESTION: *Whether there exists a cut-off for almost-sure reachability of $U_f = \{\gamma \mid \gamma(q_f) > 0\}$.*

Proof. $\mathrm{Pre}^*(U_f)$ is exactly the set of configurations that can cover q_f, i.e., configurations γ from which there exists a path $\gamma \to^* \gamma'$ with $\gamma'(q_f) > 0$. Recall also that it can be written as an upward closure of minimal elements: $\mathrm{Pre}^*(U_f) = \uparrow\{\eta_1, \dots, \eta_m\}$. Now, consider the

value K in Lemma 9.15: it is defined as $K = \sharp\,[Q_c] \cdot |\mathrm{Pre}^*(U_f)|$, with $|\mathrm{Pre}^*(U_f)| = \max_i |\eta_i|$ the maximal size of minimal configurations. The value of K can be bounded using classical results on the coverability problem in Vector Addition Systems (VAS) [Rac78].

Intuitively, a b-dimensional VAS is a system composed of an initial b-dimensional vector $\mathbf{v_0}$ of naturals (the *axiom*), and a finite set of b-dimensional integer vectors (the *rules*). An *execution* is built as follows: it starts from the axiom and, at each step, the next vector is derived from the current one by adding a rule, provided that this derivation is *admissible*, i.e., that the resulting vector only contains non-negative integers. An execution ends if no derivation is admissible. The *coverability problem* asks if a given target vector $\mathbf{v} = (v_1, \ldots, v_b)$ can be covered, i.e., does there exists a (possibly extendable) execution $\mathbf{v_0} \rightsquigarrow \mathbf{v_1} \rightsquigarrow \ldots \rightsquigarrow \mathbf{v_n} = \mathbf{v}'$ such that, for all $1 \leq i \leq b$, it holds that $v_i \leq v'_i$.

Our distributed system $\mathcal{P}$ can be seen as a $\sharp\,[Q_c]$-dimensional VAS where each transition of the contributor is modeled by a rule vector modifying the multiset of the current configuration. Formally, one has to take into account that available rules depend on the data stored in the shared register and that leader process can also perform transitions atomically. This can be achieved by either considering the expressively equivalent model of VAS with states (VASS, see e.g., [RY86]) or by adding $\mathcal{O}(\sharp\,[D])$ dimensions to enforce this restriction. Over such a VAS(S), we are interested in the coverability of the vector corresponding to the multiset q_f (i.e., containing only one copy of q_f and no other state). In particular, we want to bound the size of vectors needed to cover q_f, as it will lead to a bound on minimal elements η_i of $\mathrm{Pre}^*(U_f)$, hence a bound on the value K.

Results by Rackoff (hereby as reformulated by Demri *et al.* [DJLL13, lemma 3]) state that if a covering execution exists from an initial vector $\mathbf{v_0}$, then there is one whose length may be doubly-exponential in the size of the input: singly-exponential in the size of the rule set and the target vector, and doubly-exponential in the dimension of the VAS. Hence, for our distributed system $\mathcal{P}$, seen as a VAS, this implies that if q_f can be covered from a configuration γ, there is a covering execution whose length is bounded by some L in $2^{\mathcal{O}(\sharp[Q_c]\cdot\sharp[D])^{\mathcal{O}(\sharp[Q_c]+\sharp[D])}}$. This bound on the *length* of the execution obviously also implies a bound on the *number of processes* actively involved in the execution (because at each transition, only one process is active). Hence, we can deduce that if a configuration $\gamma = (q, \mu, d)$ can cover q_f (i.e., there exists a path $\gamma \to^* \gamma'$ with $\gamma'(q_f) > 0$), then it is also the case of configuration $\gamma'' = (q, \mu'', d)$, which we build as follows: $\forall\, q' \in Q,\ \mu''(q') = \min\{\mu(q'), L\}$. That is, it also holds that there exists a path $\gamma'' \to^* \gamma'''$ with $\gamma'''(q_f) > 0$.

By Lemma 9.14, for our algorithm to be correct, it suffices to consider the symbolic graph of fixed protocol of index $\sharp\,[Q_c] \cdot L$ and to solve sequentially two reachability problems over this graph. Let us study the size of this graph. Its state space is $V = Q_c \times \mathbb{N}^{Q_c}_{L\cdot\sharp[Q_c]} \times 2^{Q_c} \times D$. Hence, we have that: $\sharp\,[V] \leq \sharp\,[Q_c] \cdot (2 \cdot L \cdot \sharp\,[Q_c])^{\sharp[Q_c]} \cdot 2^{\sharp[Q_c]} \cdot \sharp\,[D]$, which is doubly-exponential in both the state space of the protocol and the size of the data alphabet (because L is). Since reachability over directed graphs lies in NLOGSPACE [Sip97] with regard to the size of the graph, we obtain NEXPSPACE-membership for reachability queries with regard to the size of the protocol. Finally, by Savitch's theorem [Sip97], we know that NEXPSPACE = EXPSPACE, which allows us to define the following non-deterministic EXPSPACE algorithm:

1. For all nodes $s_1 \in \overline{\widetilde{U_0}^K}$ and s_2.

2. Guess a node $s_3 \in \overline{\widetilde{U_f}^K}$.

3. Check that s_2 is reachable from s_1.
4. Check that s_3 is reachable from s_2.

□

9.4.4 General bounding scheme

Our previous EXPSPACE algorithm solves the almost-sure reachability cut-off problem for a coverability objectives thanks to a result from Rackoff on coverability in vector addition systems. We will now develop a general bounding technique, adapted to our systems, to give general complexity for several classes of protocols and objectives.

The first natural idea to bound the size of $\mathrm{Pre}^*(U_f)$ consists in considering the abstraction of any run $\gamma \to^* U_f$ in the symbolic graph, applying symbolic path reduction from Lemma 8.4, then constructing again a new path thanks to Lemma 8.3. We get this way a new run of the form $\gamma' \to^* U_f$ with $\overline{\gamma} = \overline{\gamma'}$ and γ' now of polynomial size. Assuming γ was minimal in $\mathrm{Pre}^*(U_f)$ we can deduce some information on its size, since $|\gamma'|$ is polynomial. First of all, if $\sharp\left[\overline{\mathrm{st}(\gamma)}\right] = 1$, then necessarily $\gamma \preceq \gamma'$ and we conclude $|\gamma|$ is also polynomial. Otherwise, we cannot conclude since $\mathrm{st}(\gamma)$ and $\mathrm{st}(\gamma')$ may be component-wise incomparable. In this last case, we can at least derive that there exists $q \in \overline{\gamma}$ such that $\gamma(q) < \gamma'(q)$ which is polynomial. The other components remains though unbounded but we can iterate this reasoning, by fixing already the size of the previously bounded component on q. We show in the next lemma that the fixed protocol defined in previous Section 9.4.1 is well suited for this purpose.

Lemma 9.17 (Bounding base)**.** *Assume $\mathcal{P}$ is non-atomic, U_0 and U_f are simple sets. Let $\gamma \in \min \mathrm{Post}^*_{U_f}(U_0) \cup \min \mathrm{Post}^*(U_0) \cup \min \mathrm{Pre}^*(U_f)$, then*

$$|\gamma| \leq (1/2) \cdot (18 \cdot \sharp\,[Q_c] \cdot \sharp\,[Q_l])^{\sharp[Q_c]^{\sharp[\overline{\gamma}]}}$$

In particular,

$$|\gamma| \in \mathcal{O}\left(\sharp\,[Q_l]\right)^{\mathcal{O}(\sharp[Q_c])^{\mathcal{O}(\sharp[Q_c])}} \subset 2^{\mathcal{O}(|\mathcal{P}|)^{\mathcal{O}(|\mathcal{P}|)}}$$

Before diving into the technical proof of this lemma, let us give some remarks about this result:

- For short, this lemma proves that not only the minimal predecessor configurations for a coverability objective (Rackoff bound), but any predecessor or successor minimal element, for any simple objective, have at most doubly exponential size.
- As for Lemma 8.4, the actual bound does not depend on the number of involved values in D, as opposed to the proof of previous Theorem 9.16 for the special case of coverability objectives.
- The contributors and the leader have different roles: for a fixed set of contributor control states, the bound is polynomial in the number of leader control states.

Proof. We first introduce an auxiliary function B, that will be useful to bound our configuration sizes. For all $l \geq 1$, we define

$$B(0, l) = 0$$
$$\forall n \geq 0\ B(n+1, l) = P(l) + B\left(n, l(2P(l))^{\sharp[Q_c]-1}\right)$$
$$\text{with } P(l) = (2 \cdot \sharp\,[Q_c] + 1)(l+1) + 3 \cdot \sharp\,[Q_c]$$

Notice that $P(l)$ only depends on Q_c and not on Q_l. Moreover, it is monotone. B is also monotone:

- Monotone in l: we prove by induction on $n \geq 0$, that for any $l \leq l'$, $B(n,l) \leq B(n,l')$. Result is true for $n = 0$, for $n = n'+1$, we let $k = l(2P(l))^{\sharp[Q_c]-1} \leq l'(2P(l'))^{\sharp[Q_c]-1} = k'$. We apply induction hypothesis on n' with $k \leq k'$ to obtain $B(n',k) \leq B(n',k')$ then $B(n,l) \leq B(n,l')$ since we also have $P(l) \leq P(l')$.

- Monotone in n: for any $n \geq 0$, and any $l \geq 0$,

$$B(n+1,l) = \underbrace{P(l)}_{\geq 0} + B\big(n, \underbrace{l(2P(l))^{\sharp[Q_c]-1}}_{\geq l}\big) \geq B(n,l)$$

We prove now a stronger result by induction on $\sharp[\overline{\gamma}]$,

$$|\gamma| \leq B(\sharp[\overline{\gamma}], \sharp[Q_l])$$

- If $\sharp[\overline{\gamma}] = 0$, we have $|\gamma| = 0$, hence the result for the base case.

- Let us now consider $\gamma \in \min \mathrm{Pre}^*(U_f)$ with $\sharp[\overline{\gamma}] \geq 1$ (reasoning on $\mathrm{Post}^*(U_0)$ or $\mathrm{Post}^*_{U_f}(U_0)$ is similar), which means there exists a path of the form $\gamma \to^* \gamma_f \in U_f$.

  We abstract this path into the symbolic graph, then reduce it (Lemma 8.4), and finally reconstruct it again (Lemma 8.3). Contributor length is at most equal to $(2 \cdot \sharp[Q_c] + 1)(\sharp[Q_l] + 1) + 2\sharp[Q_c]$ then reconstruction can add up to $\sharp[Q_c]$ processes.

  The new resulting path is of the form $\gamma' \to^* \gamma'_f$ with $|\gamma| \leq (2 \cdot \sharp[Q_c] + 1)(\sharp[Q_l] + 1) + 3\sharp[Q_c] = P(\sharp[Q_l])$, $\overline{\gamma} = \overline{\gamma'}$ and $\overline{\gamma_f} = \overline{\gamma'_f}$. This last property implies that $\gamma' \in \mathrm{Pre}^*(U_f)$, since U_f is a simple reachability objective.

  There exists at least one $q \in \overline{\gamma}$ such that $0 < \gamma(q) \leq P(\sharp[Q_l])$ (otherwise γ' would have been smaller than γ which is minimal). Write $k = \gamma(q)$, we have $\gamma = (q_l, \mu \oplus k \cdot q, d)$ and $\gamma_q = ((q_l, k \cdot q), \mu, d)$ as a configuration of $\widetilde{\mathcal{P}}^k$. Thanks to Lemma 9.12, we notice that $\gamma_q \in \min \mathrm{Pre}^*(\widetilde{U_f}^k)$ in $\widetilde{\mathcal{P}}^k$, and we can apply the induction hypothesis since $\sharp[\overline{\gamma_q}] = \sharp[\overline{\gamma}] - 1$. Note that induction hypothesis is applied on another protocol, with the same control states for contributors, but with a refined set of control states for the leader, which is now $Q'_l = Q_l \times \mathbb{N}_k^{Q_c}$. This set is of cardinality $\sharp[Q_l] \cdot \sharp\left[\mathbb{N}_k^{Q_c}\right]$, hence

$$|\gamma| = k + |\gamma_q| \leq k + B(\sharp[\overline{\gamma_q}], \sharp\left[Q'_l\right]) = k + B(\sharp[\overline{\gamma}] - 1, \sharp[Q_l] \cdot \sharp\left[\mathbb{N}_k^{Q_c}\right]) \leq B(\sharp[\overline{\gamma}], \sharp[Q_l])$$

  since $k \leq P(l)$, $\sharp\left[\mathbb{N}_k^{Q_c}\right] \leq (2k)^{\sharp[Q_c]-1}$ and B is monotone.

We now proceed to bound $B(n,l)$ in order to get the announced result. First, remark that for $l \geq 1$, $P(l) \leq 9 \cdot \sharp[Q_c] \cdot l$. Then $l(2P(l))^{\sharp[Q_c]-1} \leq (18 \cdot \sharp[Q_c] \cdot l)^{\sharp[Q_c]}$. We show by induction on n that:

$$\forall n, l \; B(n,l) \leq \underbrace{\left(\sum_{i=0}^{n-1} (1/2)^{i+2}\right)}_{\leq 1/2} (18 \cdot \sharp[Q_c] \cdot l)^{\sharp[Q_c]^n}$$

- For $n = 0$, this sums up to $B(0, l) \leq 0$.
- For $n \geq 0$,

$$\begin{aligned} B(n+1, l) &\leq 9 \cdot \sharp\,[Q_c] \cdot l + B(n, l(18 \cdot \sharp\,[Q_c] \cdot l)^{\sharp[Q_c]-1}) \\ &\leq 9 \cdot \sharp\,[Q_c] \cdot l + \left(\sum_{i=0}^{n-1} (1/2)^{i+2}\right) \left(18 \cdot \sharp\,[Q_c] \cdot l(18 \cdot \sharp\,[Q_c] \cdot l)^{\sharp[Q_c]-1}\right)^{\sharp[Q_c]^n} \\ &\leq 9 \cdot \sharp\,[Q_c] \cdot l + \left(\sum_{i=0}^{n-1} (1/2)^{i+2}\right) (18 \cdot \sharp\,[Q_c] \cdot l)^{\sharp[Q_c]^{n+1}} \end{aligned}$$

We conclude the proof by noticing that $2^{2^{n+1}}/2^{n+2} \geq 1$ so for $\sharp\,[Q_c] \geq 2$, $9 \cdot \sharp\,[Q_c] \cdot l \leq (9 \cdot \sharp\,[Q_c] \cdot l)^{\sharp[Q_c]^{n+1}} \leq (1/2)^{n+2}(18 \cdot \sharp\,[Q_c] \cdot l)^{\sharp[Q_c]^{n+1}}$.

□

A first immediate consequence of this doubly exponential bounding, consists in giving an explicit bound on tight cut-off values, for any simple objective, thanks to Corollary. 9.4 on the cut-off existence.

Corollary 9.18. *Assume $\mathcal{P}$ is a non-atomic protocol. Then $\mathcal{P}$ has a tight cut-off for almost-sure reachability of any simple objective, of at most doubly exponential size.*

Simulating the system for this particular value, or applying again the same technique as for coverability objective, lead us to a general EXPSPACE procedure:

Theorem 9.19. *The following problem is in* EXPSPACE*:*
ASConsensusCutOff

INPUT: *A non-atomic register protocol $\mathcal{P}$ and a final state $q_f \in Q_c$.*
QUESTION: *Whether there exists a cut-off for almost-sure reachability of $U_f = \{\gamma \mid \gamma(q_f) = |\gamma|\}$.*

Chapter 10

Extensions and discussions

As an illustration to the previous chapters, we describe several extensions of our framework for which previously developed techniques can be applied. We conclude our study in the last section, by a summary of complexity results, in the extended cases.

10.1 Model checking

In [DEGM15], techniques based on the symbolic graph are developed to address the question of the liveness property in a non-atomic protocol under a non-deterministic scheduler, with coverability objective, and an extra Büchi automaton, encoding an LTL formula. This allows the authors to consider the model checking of network protocols, in various settings, where leader and contributor can be represented by finite state machines or pushdown systems. Intuitively, given an LTL formula φ and a network protocol $\mathcal{P}$ with transition system $(\Gamma, \rightarrow)$, we want to check whether there exists an infinite run $\pi \in \Gamma^\omega$ satisfying φ. Existence of such a run is then proved to be equivalent to the existence of a particular lasso run $\pi = \pi_1 \cdot \pi_2^\omega$ with π_1 and π_2 both being finite paths of $\mathcal{P}$. Analyzing the existence of the symbolic graph in this context is sufficient, as we can consider enough processes in each state to keep the system unblocked, as the extra processes can be considered as "extra noise" and used over time to unlock the run periodically. Additionally, we can state that the existence of such an infinite run implies a cut-off property, as we can always add new processes, which will mimic (copycat) an existing process in the infinite run.

However we may argue that such infinite run, that visits infinitely often some target state, does not provide a cut-off for positive probability. As a matter of fact, we check only the existence of one single infinite run. As opposed to a finite prefix that can witness a positive probability reachability, a single infinite run may have probability 0 to occur. From a non-probabilistic perspective, processes and scheduler have to cooperate in the long run, which may seem unlikely to happen.

Therefore, it may be interesting to consider almost-sure repeated reachability and repeated reachability with positive probability. In the first case, the analysis is similar to the almost-sure reachability:

- First of all, let us denote by $\Box\Diamond A$ the set of infinite paths that visit the set of configurations A infinitely often. Basically, $\Diamond A$ represents the set of infinite runs that eventually reach A, $\Diamond A = \Gamma^* \cdot A \cdot (\Gamma^* \uplus \Gamma^\omega)$, and $\Box\Diamond A = (\Gamma^* \cdot A)^\omega$.

- As in Lemma 7.22, we consider (Γ, p) a probabilistic transition system of $\mathcal{P}$, and show that as for $\mathbb{P}_n(\Diamond A)$, qualitative values of $\mathbb{P}_n(\Box\Diamond A)$ do not depend on p. Indeed, such property depends only on the support of p which is determined by the specification of $\mathcal{P}$. In particular, we can characterize almost-sure repeated reachability by $\lfloor \mathbb{P}_n(\Box\Diamond A) \rfloor = 1$ which is equivalent to the following property on Pre and Post:

  $$\mathrm{Post}^*(U_0 \cap \Gamma_n) \subseteq \mathrm{Pre}^*(A)$$

  Intuitively, we replaced operator $X \mapsto \mathrm{Post}(X \backslash A)$ by the more simple mapping $\mathrm{Post}(\cdot)$, which means reachability should still hold after reaching A. Since Γ_n is finite, there exists a positive probability to reach A again from any reachable configure γ. Indeed, this value is bounded from below by $\min\{\underbrace{\mathbb{P}(\{\gamma\}\Gamma^* \cap \Diamond A)}_{>0} \mid \underbrace{\gamma \in \mathrm{Post}^*(A)}_{\sharp[\cdot]<\infty}\} > 0$

- Let $A = U_f$ be a simple objective. Then, $\mathrm{Pre}^*(U_f)$ is upward-closed. We adapt the proof of Corollary. 9.4 to show that either $\sharp\,[Q_c] \cdot |\mathrm{Pre}^*(U_f)|$ is a positive cut-off, or $|\mathrm{Post}^*(U_f)|$ is a negative cut-off for almost-sure repeated reachability. Thanks to Lemma 9.17, we can also see that any tight cut-off is at most doubly exponential, and obtain an equivalent EXPSPACE decision procedure.

The second question, about the existence of a cut-off for positive probability of repeated reachability is surprisingly harder, and remains currently open: a positive probability indeed corresponds to finding a finite path from U_0 to some configuration γ from which repeated reachability holds almost-surely. If we denote by B the set of such configurations, we can remark that B is not upward closed: consider for example filter protocol $\mathcal{F}_n$ example of Figure 9.4 with repeated reachability objective $U_f = \{\gamma \mid \gamma(s_n) = 0\}$. Because of this lack of regularity, we cannot directly answer this last question, which could be possibly solved by additional work based on ideals, or through the use of the refined symbolic graph to keep some coordinates low.

10.2 r-register protocol

We have mainly developed tools for the analysis of non-atomic protocols, where processes communicate through a single shared variable. On the other side, the analysis of atomic protocols seems to be much harder, as one can expect that atomicity allows us to encode arbitrary Petri Nets. One may argue that there still remains a big gap between theoretical results on the cut-off existence, that are stated in the monotonous case, and algorithms and bounds that are only valid for non-atomic protocols. One may ask, if some monotonous systems, but not necessarily non-atomic still enjoy good combinatorial properties and if previous algorithms can be lifted to more complex frameworks. This section explores one possible way of improving our model by the use of several registers.

10.2.1 Tools enhancement

First of all, let us remark that our previous copycat Lemma 7.18 can be more generally stated this way:

Lemma 10.1 (Copycat lemma improved)**.** *Assume $\mathcal{P}$ involves only* stuttering *operations for contributor, namely, for all $(q, \mathbf{f}, q') \in T_c$, $(\mathbf{f} \cdot \mathbf{f}) = \mathbf{f}$. Then $\mathcal{P}$ is monotonous.*

The proof is the same as for the original lemma, as the copycat can mimic an already existing process thanks to this stuttering property.

On the other hand, bounds and algorithms were later given thanks to a path reduction in the symbolic graph, by Lemma 8.4. Recall the proof of this lemma, the crucial property is the ability for a fixed set of contributor states (in the support) to produce in one step the desired value on the shared register, in order to make the leader or the support progress. We relax this notion now, to allow a fixed number of steps.

Definition 10.2. Let $k \in \mathbb{N}_{>0}$ and $X \subseteq \mathrm{Op}\,(D)$. X is *k-valued* if for all $d \in D$, all $Y \subseteq X$ and $\mathbf{f} \in Y^+$, there exists $\mathbf{f}' \in \cup_{i \leq k} Y^i$ such that

$$\mathbf{f}(d) = \mathbf{f}'(d)$$

$\mathcal{P}$ is *k-valued* if $T_c \subseteq Q_c \times X \times Q_c$ for some k-valued set of operations X.

Intuitively, for any current value d and sequence of operations applied on d, we can extract a (possibly re-ordered) sequence of operations of length at most k, that has the same effect on d. As expected the set of non-atomic operations $\mathrm{Op}_{R,W}\,(D)$ is 1-valued: proof is done by picking the last write operation, if any, or any read operation otherwise.

10.2.2 Operations over r registers

Before explaining how previous results can be generalized to k-valued monotonous protocols, we give below an example of concrete k-valued set of operations, that is still stuttering.

Definition 10.3 (Operations over a r registers)**.** Let D a set and $r \geq 1$. For $i, j \in [1, r]$ and $d \in D$, we define the following operations over D^r.

$$\mathbf{W}_i\,(d) : \begin{cases} \mathrm{dom}\,(\mathbf{W}_i\,(d)) = D^r \\ \qquad\qquad d' \mapsto d'[i/d] = d'[1] \cdots d'[i-1] \cdot d \cdot d'[i+1] \cdots d'[r] \end{cases}$$

$$\mathbf{R}_i\,(d) : \begin{cases} \mathrm{dom}\,(\mathbf{R}_i\,(d)) = D^{i-1} \times \{d\} \times D^{r-1-i} \\ \qquad\qquad d' \mapsto d' \end{cases}$$

$$\mathbf{M}_{i \to j} : \begin{cases} \mathrm{dom}\,(\mathbf{M}_{i \to j}) = D^r \\ \qquad\qquad d' \mapsto d'[j/d'[i]] = d'[1] \cdots d'[j-1] \cdot d[i] \cdot d'[j+1] \cdots d'[r] \end{cases}$$

We denote by $\mathrm{Op}_{R,W,M}\,(D^r)$ the set of such register operations.

Our initial shared register is now split into r coordinates, each of them being called (abusively) a register. Non-atomic read and write operations are now done on one coordinate at a time, leaving the other registers unchanged.

As one can expect, when restricting to read and write operations, which are in some sense still non-atomic, we achieve r-valued monotonicity. This next result generalizes the case $r = 1$ studied in the previous chapters.

Lemma 10.4. *For any r, and $\mathcal{P}$ such that $T_c \subseteq Q_c \times Op_{R,W}\,(D^r) \times Q_c$, $\mathcal{P}$ is an r-valued monotonous protocol.*

Proof. First of all, $\mathcal{P}$ is monotonous, as $\mathrm{Op}_{R,W}(D^r)$ is stuttering. Let us consider $\mathbf{f} \in Y^*$ with $Y \subseteq \mathrm{Op}_{R,W}(D^r)$ and $d \in D^r$. For each $i \in [1,r]$, if $d[i] = (\mathbf{f}(d))[i])$ we define $\mathbf{f}_i = \mathrm{Id}$. Otherwise, there exists $\mathbf{f}_i = \mathbf{W}_i(d[i]) \in Y$.

Hence, $\mathbf{f} = \mathbf{f}_1 \cdots \mathbf{f}_r \in Y^k$ with $k = \sharp[\{i \mid d[i] \neq (\mathbf{f}(d))[i]\}] \leq r$. □

The previous proof handles the read and write case. However, the newly introduced move (or copy) operation can also be stuttered, so monotonicity is preserved. We prove in the next lemma that this new operation also keeps our protocol k-valued, where k is a polynomial in r.

Lemma 10.5. *For any r, and $\mathcal{P}$ such that $T_c \subseteq Q_c \times Op_{R,W,M}(D^r) \times Q_c$, $\mathcal{P}$ is an r^3-valued monotonous protocol.*

Proof. First of all, $\mathcal{P}$ is monotonous, as $\mathrm{Op}_{R,W,M}(D^r)$ is stuttering. We consider $\mathbf{f} = \mathbf{f}_1 \cdots \mathbf{f}_n$ with for all i, $\mathbf{f}_i \in Y \subseteq \mathrm{Op}_{R,W,M}(D^r)$. We will construct another sequence of $\mathbf{f}'_{i_1}, \ldots \mathbf{f}'_{i_k}$ with $k \leq r^2$, such that $\mathbf{f}(d) = (\mathbf{f}'_{i_1} \cdots \mathbf{f}'_{i_k})(d)$.

Without loss of generality, we make the two following assumptions:

- There are no read operations in $\mathbf{f}$, otherwise, we can remove them, since they have no side effect.
- The number l of write operations is smaller than the number r of registers, by pigeonhole principle (otherwise, some written value was useless). The remaining write operations can be seen as extra registers indexed from $r+1$ to $r+l$ with a unique move operation. For example, the i-th ($1 \leq i \leq l$) write operation $\mathbf{W}_{j_i}(d'_i)$ is replaced by move operation $\mathbf{M}_{r+i \to j_i}$, assuming the initial value d satisfies $d[r+i] = d'_i$. We denote by $r' = r + l \leq 2r$ the total amount of registers.

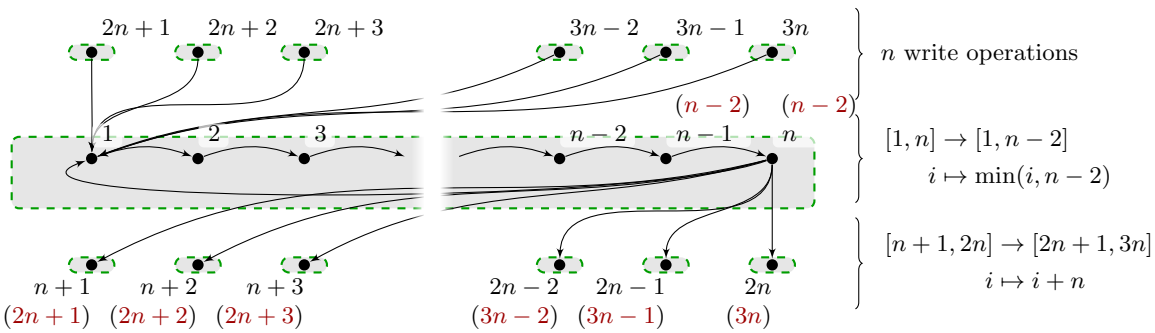
Thus, the rest of the analysis can be done with move operations only, whose sequence performs a mapping m from $[1,r]$ to $[1,r']$. Let us denote with $Z \subseteq \mathrm{Op}_{R,W,M}\left(D^{r'}\right)$ the set of available move operations. Z can be seen as a graph, with node set $[1,r']$ and edges $(i,j) \in [1,r'] \times [1,r]$ for any $\mathbf{M}_{i \to j} \in Z$.

We consider the strongly connected components of Z. Inside such a strongly connected component C, whenever an internal edge is used, it is erasing an internal value. Moreover, $\sharp[C] - 1$ values can be preserved inside C by moving them sequentially along a cycle of C. One sequential move (shift) of all values requires $\sharp[C] - 1$ operations, hence $\sharp[C](\sharp[C] - 1)$ to perform a complete cycle of the values.

For each $i \in [1,r]$, the final value $\mathbf{f}(d)[i] = d[m(i)]$ is obtained by a finite path in Z. We can assume this path is cycle-free, and whenever, the path uses internal nodes of a strongly connected component C, it preserves all values except 1, which can be achieved with at most $\sharp[C](\sharp[C] - 1)$ operations. With additional move operations between two components, we can assign register i its final value in at most r^2, hence a total number of r^3 for the whole mapping m. □

As shown on examples of Figure 10.1, there exists a sequence of write and move operations, that cannot be expressed with less than a cubic number of operations. We conclude that up to a multiplicative constant, the r^3 bound given in the previous lemma is optimal.

(a) $r = n$ registers, $Z = \{(i, (i \mod n) + 1)\}$, and desired mapping $m : i \mapsto \min(n, i + 1)$. Only value 1 can be erased. We have to shift all values to the right $n - 1$ times. Each shift requires $n - 1$ move operations, hence the whole mapping requires $(n - 1)^2$ move operations.



(b) $r = 2n$ and n additional write operations (pseudo registers $2n + 1$ to $3n$). Because of the middle component cycle, that has to keep $n - 2$ of its initial values, each assignment $i \in [n + 1, 2n] \mapsto i + n$ will require a first write inside the cycle, then $n - 1$ shifts (with $n - 1$ operation each), hence a global cost of at least $n(n - 1)^2 \sim r^3/8 \in \Omega(r^3)$ operations.

Figure 10.1 – Examples of directed graphs associated with Z, as a set of move operations $Y = \{\mathbf{M}_{i \to j} \mid (i, j) \in Z\}$, with a possible mapping $m : [1, r] \to [1, r]$. For a node $i \in [1, r]$, we label the node with the desired value for the mapping by $(m(i))$. We omit the label when $m(i) = i$.

10.2.3 Discussion on the r-register extension

We revisit below previous results when considering k-valued monotonous protocols, instead of non-atomic ones.

Lemma 10.6 (Diameter of the symbolic graph)**.** *Assume $\mathcal{P}$ is k-valued monotonous. If two nodes s and s' are connected in $G_{\mathcal{P}}$, then there exists a path from s to s' of contributor length smaller than*

$$k \cdot (2 \cdot \sharp [Q_c] + 1) \cdot (\sharp [T_l] + 1) + 2 \cdot \sharp [Q_c]$$

The proof is similar to original Lemma 8.4 but between two transitions of the leader, we may need up to k contributor transitions, to achieve the next data value required by the leader. Since contributors may not be able to produce the same register values between two leader steps (even with the same support), we can now only bound the number of leader steps by $\sharp [T_l]$ instead of simply $\sharp [Q_l]$. Note that in the leaderless case, this quantity remains small, otherwise, it can be of the same magnitude as $\sharp [D]^k$.

Lemma 10.7 (Bounding base)**.** *Assume $\mathcal{P}$ is k-valued and monotonous, U_0 and U_f simple sets. Let $\gamma \in \min \mathrm{Post}^*_{U_f}(U_0) \cup \min \mathrm{Post}^*(U_0) \cup \min \mathrm{Pre}^*(U_f)$, then*

$$|\gamma| \in 2^{\mathcal{O}(k \cdot |\mathcal{P}|)^{\mathcal{O}(k \cdot |\mathcal{P}|)}}$$

In the leaderless case, since $\sharp [T_l] = 0$, we can even give a bound that is polynomial in k:

$$|\gamma| \leq (1/2) \cdot (6 \cdot (2k+1) \cdot \sharp [Q_c] \cdot \sharp [Q_l])^{\sharp [Q_c]^{\sharp [\overline{\gamma}]}}$$

As a consequence, tight cut-off upper bounds and EXPSPACE procedure can be lifted to k-valued monotonous protocols, and even to protocols over r registers with read, write and move operations:

Corollary 10.8. *The following problem is in* EXPSPACE*:*
<u>ASCoverCutOff</u>

INPUT: *An integer $r \in \mathbb{N}$, a register protocol $\mathcal{P}$ with operation set $Op_{R,W,M}(D^r)$ and a final state $q_f \in Q_c$.*

QUESTION: *Whether there exists a cut-off for almost-sure reachability of $U_f = \{\gamma \mid \gamma(q_f) > 0\}$.*

Moreover, such a tight cut-off is doubly exponential.

Proof. Even if r is provided in binary form, one can first ensure that any of the r registers is used so that[1] $r \leq 2 \max(\sharp [T_l], \sharp [T_c]) \leq 2|\mathcal{P}|$ (otherwise, we can rewrite $\mathcal{P}$ in polynomial time to make use of $r' < r$ registers). Moreover, $\mathcal{P}$ is r^3-valued, so

$$\max(|U_0|, |U_f|) \in 2^{\mathcal{O}(8 \cdot |\mathcal{P}|^4)^{\mathcal{O}(8 \cdot |\mathcal{P}|^4)}}$$

gives a doubly exponential bound for tight cut-off, hence an EXPSPACE decision procedure. □

[1] Remember that a move operation may involve two different registers, hence a factor 2.

10.2.4 Comparison with non-atomic protocols

One may notice that multiple registers can be emulated through the use of the leader in the following way. We encode the current value $d \in D^r$ in the state of the leader, and allow him to write messages on the shared register, of the form "$d[i]$ equals v" for any $i \in [1, r]$ and $v = d[i]$. When reading such message, the leader can also move to a state encoding the updated value. From the point of view of contributors, any operation $\mathbf{R}_i(d')$ (resp $\mathbf{W}_i(d')$) can be converted into reading (resp writing) the message "$d[i]$ equals d'". Any path in the r-register protocol can be converted in a path in the described 1-register protocol and conversely, therefore, positive probability and almost-sure reachability properties are preserved.

However, this method has several drawbacks:

- First of all, this encoding can only deal with read and write operations.
- From a combinatorial point of view, the new state space of the leader Q'_l is now of cardinality $\sharp[Q_l] \times \sharp[D]^r$ which can be huge. In the general case, the previous bound is also exponential in r, but leaderless case or other fragments may feature lower complexity.
- Last but not least, this encoding does not work for leaderless protocols, where we can extract a polynomial bound in r, and independent of $\sharp[D]$.

In the leaderless case, such reduction seems impossible, and we conjecture that r-register protocols are indeed more expressive than non-atomic (or 1-register) protocols.

10.3 Process identifiers

As noticed before, non-atomic protocols allow us to bound the diameter of the symbolic graph by a quantity that only depends on the state space of the protocol, and not on the domain size $\sharp[D]$, nor the number of transitions $\sharp[T_c] + \sharp[T_l]$. First of all, this means we can encode in only one transition a read operation that only checks that the current register value is different from a given value d, by adding several transitions $(q_1, R(d'), q_2)$ for any $d' \in D \backslash \{d\}$. This also means that D can virtually encode an infinite number of values, without any impact on the complexity results.

We argue that our techniques can be adapted to encode processes with unique identifiers (pid). We can indeed equip our protocols with a new write operation, writing the identifier of the current process to the shared register. If we add another operation checking that the process id corresponds to the one written in the shared register, monotonicity will be broken. However, we can still allow each process to execute an operation only if the shared register value *differs* from the process identifier.

Intuitively, such systems will allow the election of a leader process, but this single process will not be able to check that it is indeed the leader. We do not study further this model as it requires to redefine our whole transition system to incorporate unique process identifiers, then to redefine the well quasi order to compare configurations.

10.4 Conclusions

As a partial conclusion to the four last chapters, we present in Table 10.1 a summary of the bounds stated for positive reachability and almost-sure reachability of simple objectives in r-register protocols with read, write and move operations.

<table>
<tr><td></td><td>Positive probability</td><td colspan="2">Almost-sure</td></tr>
<tr><td>Characterization</td><td>$(\Gamma_n \cap U_0) \cap \mathrm{Pre}^*(U_f) \neq \varnothing$</td><td colspan="2">$\mathrm{Post}^*_{U_f}(U_0 \cap \Gamma_n) \subseteq \mathrm{Pre}^*(U_f)$</td></tr>
<tr><td>Tight positive worst case</td><td>Polynomial</td><td>at least linear
(even when L1)</td><td rowspan="2">at most
2EXP</td></tr>
<tr><td>Tight negative worst case</td><td>Constant 1</td><td>at least EXP
(even when L1)</td></tr>
<tr><td>Decision problem</td><td>PTIME
(for L1-cover [EGM13]),
NP-complete</td><td colspan="2">PSPACE-hard (even when L1),
EXPSPACE</td></tr>
</table>

Table 10.1 – Summary of the presented results for reachability cut-offs for arbitrary r-register protocols. Hardness results are still valid in the fragment (L1) of non-atomic leaderless protocols with 1 register.

As discussed in Section 10.1, these results can easily be transposed to the almost-sure repeated reachability case in order to explore protocol model checking. As discussed in Section 10.3, similar results can be obtained when adding the ability to manipulate process identifiers inside registers, though the concrete study of this model is left as further work. We can also notice a complexity gap between our EXPSPACE procedure and the PSPACE-hardness of the decision problem. In terms of tight cut-offs, this corresponds to a gap between the worst known case of an exponential (negative) cut-off and the general doubly exponential bounds on cut-off extracted from the general bounding scheme from Lemma 9.17.

As testified by our several constructions and experiments, we haven't managed to build a cut-off, or minimal elements (in the $\mathrm{Pre}^*(U_f)$) bigger than a simple exponential. A reasonable conjecture of a simple exponential bound on $|\mathrm{Pre}^*(U_f)|$ would close this complexity gap. As a matter of fact, we saw in Theorem 9.16 that our procedure is based on the refined symbolic graph of index $\sharp[Q_c] \times |\mathrm{Pre}^*(U_f)|$ whose exploration can be done in NLOGSPACE in its size. With such a conjecture, this graph would be of exponential size (instead of doubly exponential), thus giving a PSPACE procedure.

Another related further work is the study of protocols composed of pushdown systems, as considered in [EGM13]. In their work, the authors show that in the non-deterministic scheduler setting (positive reachability probability), the reachability problem is NP-complete if either the leader or the contributor is a pushdown machine, and becomes PSPACE-complete if both leader and contributor are pushdown machine.

Chapter 11

Toward Strategy Synthesis

In this chapter, we try to address the local strategy synthesis problem. More precisely, we are interested in the (partial) determinization of under-specified protocols, in order to satisfy a given objective. Strategy synthesis is often considered from a global point of view, where power is given to the scheduler which can observe the whole system and choose the next transition accordingly.

Here, the scheduler is considered as a source of uncertainty, that cannot be controlled. We adopt a more distributed approach, where strategies are given to all processes (or agents) with their restricted observation power. Since processes are running the same automaton, it is reasonable to consider that the same local strategy is distributed among all agents. Once a strategy is defined over an under-specified network, we can consider the resulting system, which would be another protocol on which we can apply previously seen techniques.

We introduce concepts similar to the first part of the thesis, namely actions, then allowed actions, and finally strategy classes. Since we are looking at qualitative reachability questions, the exact probability distributions involved in a mixed strategy are not relevant, so our randomization only consists in picking a support of actions. In a non-stochastic context, this model would correspond to non-determinism, that is solved by the scheduler. Here, the scheduler has a stochastic behaviour, and will be pick an available transition at random. Several strategy classes will be considered which can require more or less memory and randomization.

The chapter is divided into three sections: we first introduce the local strategy semantics in the framework of non-atomic protocols, and justify why our analysis can be done without taking exact probability values into account. Then, we address the characterization of local strategies ensuring a positive cut-off, in the two simple cases of reachability and safety in a leaderless protocol with a coverability objective.

11.1 Definitions

11.1.1 Allowed actions and randomization

In the rest of the chapter, $\mathcal{P} = (Q_l, Q_c, D, d_0, q_l, q_c, T_l, T_c)$ is a given shared register protocol. We denote for simplicity $Q = Q_l \uplus Q_c$ and $T = T_l \uplus T_c$.

As in Definition 3.1, we define a set of allowed actions, and build strategies over these allowed actions. Here actions are subsets of transitions of $\mathcal{P}$. When an agent chooses an action, the scheduler will be in charge of picking the exact transition to be played, non-

deterministically or randomly. We additionally require that the subset of transitions are defined on the same domain as the initial transitions, thus no agent can block a run (to a finite prefix) that was initially live (infinite run).

Definition 11.1 (Actions and allowed transitions)**.** Let $q \in Q$ be a state of $\mathcal{P}$. A subset $A \subseteq T$ is called an *action from* q if the two following conditions hold:

- All transitions start from q: $A \subseteq \{(q, \mathbf{f}, q') \in T \mid \mathbf{f} \in \mathrm{Op}\,(D)\,, q' \in Q\}$;
- Register operations of A are globally defined on the same domain:

$$\bigcup_{\substack{(\mathbf{f},q')/\\(q,\mathbf{f},q')\in A}} \mathrm{dom}\,(\mathbf{f}) = \bigcup_{\substack{(\mathbf{f},q')/\\(q,\mathbf{f},q')\in T}} \mathrm{dom}\,(\mathbf{f})$$

We denote by $\mathsf{Act}(q)$ the set of actions from q.

A set of *allowed actions* for protocol $\mathcal{P}$, from state q, is defined as

$$\mathsf{Allow}(q) \in 2^{\mathsf{Act}(q)} \backslash \{\varnothing\}$$

We have seen in the previous chapters that having more than one transition from a given state with the same register operation corresponds to some non-deterministic move that is resolved by the scheduler. In a probabilistic setting, this means that if action A contains two transitions that both accept the same register value d, the transition with value d will be randomized.

Graphically, when more than one action is allowed from a state, each action A is represented by an outgoing edge to an intermediary node, with several outgoing edges for each transition $t \in A$. This representation is similar to the stochastic nodes of our stochastic games, as the choice of the exact transition to be taken is left to the environment, or scheduler. For the sake of simplicity, when $\sharp\,[\mathsf{Allow}(q)] = 1$, that is to say when only one action A is allowed, we omit the representation of action A and its intermediary node, and directly represent outgoing transitions from q. However, we have to keep in mind that the choice of the exact transition taken from A is left to the scheduler, which will choose at random.

For example, let us consider the simple protocol of Figure 11.1. When current register value is 0 and action A_1 is played, two transitions are available: $(q_0, \mathbf{R}\,(0)\,, q_0)$ and $(q_0, \mathbf{R}\,(0)\,, q_1)$. In this case, the scheduler will choose (at random) which transition effectively takes place.

11.1.2 Local strategies

We proceed now to define local strategies. Here, the word local means that strategies can only see a history for a given player, that is to say a word $h \in Q^+$.

Intuitively, a strategy assigns to each history a decision, which is a distribution over the allowed actions. Since we are considering qualitative questions, from a state q only the support $\delta \subseteq \mathsf{Allow}(q) \subseteq \mathsf{Act}$ of the distribution is relevant. The exact transition to be taken is resolved in the following way:

- A process is chosen, which provides (thanks to its local strategy) a distribution support δ.

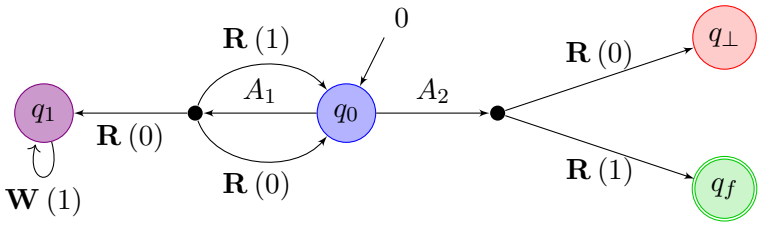


Figure 11.1 – Example of a leaderless non-atomic protocol, with two allowed actions from state q_0: $A_1 = \{(q_0, \mathbf{R}(0), q_0), (q_0, \mathbf{R}(0), q_1), (q_0, \mathbf{R}(1), q_0)\}$ and $A_2 = \{(q_0, \mathbf{R}(0), q_\perp), (q_0, \mathbf{R}(1), q_f)\}$

- An action $A \in \delta$ is chosen.
- A transition $t \in A$ is chosen among valid transitions[1].

Each of these three choices is made by the scheduler, at random. Again, we argue that exact probability values are irrelevant and that we can focus directly on the set of transitions $\{t \in A \mid A \in \delta\}$ a strategy can provide to the scheduler:

Definition 11.2 (Local strategies for a protocol)**.** A mapping $\sigma : Q^+ \to 2^T$ is a *local strategy for* $\mathcal{P}$ *and* Allow if for all $h \in Q^+$, $\sigma(h) = \bigcup_{A \in \delta} A$ for some non-empty set $\delta \subseteq \mathsf{Allow}(last(q))$. We denote by $\mathbb{S}$ the set of local strategies for $\mathcal{P}$ and Allow, or *strategies* for short when $\mathcal{P}$ and Allow are clear from the context.

We can then define sub-classes of strategies as in Definition 3.3.

Definition 11.3. σ is pure, when for all $h \in Q^+$, $\sigma(h) \in \mathsf{Allow}(last(q))$. We define the following classes:

- $S \subseteq \mathbb{S}$ the class of pure strategies;
- $\mathbb{M}(k)$ for any $k \in \mathbb{N}$ the class of strategies requiring a *memory of size* k;
- $\mathbb{M} = \mathbb{M}(0)$ the class of memoryless strategies;
- $\mathbb{F} = \cup_{k' \geq 0} \mathbb{M}(k)$;
- We also define their pure counterpart: $M(k) = \mathbb{M}(k) \cap S$, $M = \mathbb{M} \cap S$ and $F = \mathbb{F} \cap S$.

Remark 11.4. *For any $q \in Q$, $\mathsf{Act}(q)$ is stable by union, so $\mathsf{Act}(q)$ also contains all randomized actions. However, they may not be allowed, depending on the actual choice of* Allow *function and the strategy class we are considering. When for all state $q \in Q$, $\mathsf{Allow}(q)$ is stable by union, pure and randomized classes coincide. Indeed, in the qualitative context, a randomized strategy only picks the support of action distributions.*

Remark 11.5. *As for our stochastic games, agents do not see played actions inside the history, nonetheless the protocol $\mathcal{P}$ can be modified to encode this information inside the history. Moreover, agents do not see the register value before playing, as it would break monotonicity of the resulting protocol.*

[1] Remember that for a transition $t = (q, \mathbf{f}, q') \in A$ to be valid, the current register value d must be in $\mathrm{dom}(\mathbf{f})$.

Agents neither see past register values, though this information can be encoded in the current state, as the past transitions. Indeed, we can transform each transition $t = (q, \mathbf{R}(d), q')$ *(resp.* $t = (q, \mathbf{W}(d), q')$ *) in several transitions* $t' = ((q, d'), \mathbf{R}(d), (q', d))$ *(resp.* $t' = ((q, d'), \mathbf{W}(d), (q', d))$*) for all possible* $d' \in D$*.*

11.1.3 Semantics

Once we are given a local strategy σ, we are able to build runs of our protocol that satisfy the local strategy. Notice that agents have partial observation, since they are given only the history about their own visited states along the run. A formal definition of such runs is developed below.

Definition 11.6. Let σ a strategy, and $\pi \in \text{paths}(\rightarrow)$, we say that π *agrees with* σ if there exists $\rho \in (\mathbb{N}^{Q^+})^{|\pi|}$, a sequence of multisets of histories, such that

- $\mathit{first}(\rho) = \text{st}(\mathit{first}(\pi))$
- For all $i \in [1, |\pi|]$, there exists $t = (q, \mathbf{f}, q') \in T$, $h \in Q^*$ such that
  - $t \in \sigma(h \cdot q)$,
  - $\pi[i] \xrightarrow{t} \pi[i+1]$
  - $\rho[i] \oplus (h \cdot q \cdot q') = \rho[i+1] \oplus (h \cdot q)$.

We will denote with $\text{agree}(\sigma)$ the set of paths that agree with σ, and ρ will be called the corresponding sequence of π and σ.

Intuitively, we attach to the finite run π a sequence ρ of the same length that stores at any time, the growing histories for each process. Note that these histories have various lengths, depending on how much a single process has already moved. At a given time $i \in [1, |\pi|]$, transition $t = (q, \mathbf{f}, q')$ is taken if one of the processes in state q has an history $h \cdot q$ for which strategy σ allows t. If this condition holds, the transition is triggered, so $\rho[i+1]$ is defined from $\rho[i]$ by replacing one occurrence of $h \cdot q$ by the new history $h \cdot q \cdot q'$.

In particular, we can check that, at any time, when removing historical information from ρ, we get back the multi-set of states of processes:

$$\forall i \in [1, |\pi|].\ \forall q \in Q. \quad \pi[i](q) = \sum_{h \in Q^*} \rho[i](h \cdot q)$$

Under the prefix relation $\sqsubseteq$, we can consider the set of maximal runs

$$\max_{\sqsubseteq} \text{agree}(\sigma) = \left\{ \pi \in \text{agree}(\sigma) \;\middle|\; \forall \pi' \in Q^* \uplus Q^\omega\ \pi \sqsubseteq \pi' \Rightarrow \pi = \pi' \right\}$$

As a denumerable intersection of measurable sets, $\max \text{agree}(\sigma)$ is proven to be measurable for $\mathbb{P}$. This allows us to define a probability measure over this strategy:

Definition 11.7. Let $n \in \mathbb{N}$, $\Pi \subseteq \text{paths}(\rightarrow)$ a measurable set of paths, and σ a local strategy. If $\mathbb{P}_n(\max \text{agree}(\sigma)) > 0$, we define

$$\mathbb{P}_n^\sigma(\Pi) = \mathbb{P}_n\left(\Pi \;\middle|\; \max_{\sqsubseteq} \text{agree}(\sigma)\right) = \frac{\mathbb{P}_n(\Pi \cap \max \text{agree}(\sigma))}{\mathbb{P}_n(\max \text{agree}(\sigma))}$$

Otherwise, we let $\mathbb{P}_n^\sigma(\Pi) = 0$.

Remark 11.8. *Another possible way of defining $\mathbb{P}_n^\sigma$ would consist in defining a new protocol $\mathcal{P}\langle\sigma\rangle$ from $\mathcal{P}$, with a new state space $Q'_c = Q_c^+$, and $T'_c = \{(h \cdot q, \mathbf{f}, h \cdot q \cdot q') \mid (q, \mathbf{f}, q') \in T_c \cap \sigma(h) \wedge h \in Q^*\}$ and a similar definition for Q'_l and T'_l. Intuitively, we would store, for each process its whole history h as the current state, then allow only transitions from $\sigma(h)$. We can easily check that each valid path of this new protocol, can be projected by taking last state for each history in each configuration, to a path in $\mathcal{P}$ that agrees with σ. However, all our study is based on a formal model where protocols have a* finite state space*. In particular, technical difficulties may occur when trying to consider a well quasi order on configurations in infinite dimensions, or when trying to prove that qualitative properties do not depend on the exact transition probabilities (Lemma 7.22). In fact, these technical difficulties are avoided by directly considering the conditional probability. In particular, visiting the same state twice but with different memory states may infer different transition probabilities.*

However, we keep in mind the intuition that for a finite memory local strategy σ, we could build a protocol $\mathcal{P}\langle\sigma\rangle$ with a finite number of states, having the same qualitative properties as for $\mathbb{P}_n^\sigma(\cdot)$.

The previous definition makes implicit the choice of a probabilistic transition system (Γ, p) for $\mathcal{P}$ in order to define $\mathbb{P}_n^\sigma$. However, as in Lemma 7.22, we characterize qualitative properties without having to consider the exact probability values of p:

Lemma 11.9. *Let $A \subseteq \Gamma$ and $\sigma \in \mathbb{S}$. The property $\lceil \mathbb{P}_n^\sigma(\Diamond A) \rceil = 1$ does not depend on the actual values of p. Moreover, we have the following characterization:*

$$\lceil \mathbb{P}_n^\sigma(\Diamond A) \rceil = 1 \Leftrightarrow \mathbb{P}_n^\sigma(\Diamond A) > 0 \Leftrightarrow (U_0 \cdot \Gamma_n^*) \cap (\Gamma_n^* \cdot A) \cap \mathrm{agree}\,(\sigma) \neq \varnothing$$

Proof. The proof is immediate by applying the previous definition to state that

$$\mathbb{P}_n^\sigma(\Diamond A) > 0 \Leftrightarrow \mathbb{P}_n^\sigma(\Diamond A \cap \max \mathrm{agree}\,(\sigma)) > 0 \wedge \mathbb{P}_n^\sigma(\max \mathrm{agree}\,(\sigma)) > 0$$

□

11.1.4 Cut-off property

Our main goal is the synthesis of strategies that will ensure a *positive* cut-off. This leads to the following definition.

Definition 11.10. Let σ be a local strategy, I a sub-interval of $[0, 1]$ and A a set of configurations. We say that σ is a cut-off strategy for probability interval I and objective A, if there exists $N \in \mathbb{N}$ such that for all parameter $n \geq N$, $\mathbb{P}_n^\sigma(\Diamond A) \in I$.

As before, we define three qualitative cut-off properties, namely $\{0\}$, $(0, 1]$ and $\{1\}$ cut-offs for a simple reachability objective U_f, respectively for safety, positive reachability and almost-sure reachability.

The sections 11.2 and 11.3 respectively focus on the first two problems, in the restricted non-atomic, leaderless case, with coverability objective U_f.

As opposed to the cut-off decision problem, these two problems are not dual in the local strategy synthesis framework.

11.2 Reachability

We study first the local synthesis problem for positive reachability, that is to say, whether there exists a cut-off strategy σ for positive reachability. In this setting, the scheduler is cooperative and we are interested in the existence of a single path that agrees with σ.

11.2.1 Mixed strategies

As the local strategy will cooperate with the scheduler, a reasonable strategy is to play as many actions as possible, to enable as many paths as possible. This intuition is summarized below.

Lemma 11.11. *Let $\sigma, \sigma' \in \mathbb{S}$ two strategies such that*

$$\forall h \in Q^+ \ \sigma(h) \subseteq \sigma'(h)$$

Then $\text{agree}(\sigma) \subseteq \text{agree}(\sigma')$.

Proof. Immediate: for any $\pi \in \text{agree}(\sigma)$, we consider a corresponding sequence of multiset of histories $\rho \in (\mathbb{N}^{Q^+})^{|\pi|}$ from Definition 11.6 and check that is valid for path π'. □

Let us denote by $\text{Reach} \subseteq \mathbb{S}$ the set of cut-off strategies for positive probability of covering q_f. Then, the following theorem states that a memoryless randomized strategy suffices.

Theorem 11.12. *Let σ be the memoryless strategy playing all allowed actions at random, and assume that* $\text{Reach} \neq \varnothing$. *Then $\sigma \in \text{Reach}$. In particular,* $\text{Reach} \cap \mathbb{M} \neq \varnothing$.

Proof. Let $\sigma' \in \text{Reach}$, then there exist $\pi \in \text{paths}(\rightarrow) \cap \text{agree}(\sigma')$, $\gamma_0 \in U_0$ and $\gamma_f \in U_f$ such that $\pi : \gamma_0 \rightarrow^* \gamma_f$.

For all $h \in Q^+$, $\sigma'(h) \subseteq \sigma(h)$, so and $\pi \in \text{agree}(\sigma)$ by Lemma 11.11, thus $\sigma \in \text{Reach}$. □

The previous theorem basically states that the existence of a mixed cut-off strategy for positive probability can be reduced to the decision problem of a positive cut-off in the underlying protocol alone, which belongs to P for the particular case of non-atomic leaderless coverability.

11.2.2 Pure strategies

Another approach, studied in [BFS15], considers pure strategies only. In their context, they show that polynomial memory suffices for a pure strategy to cover q_f, if such strategy ever exists.

Theorem 11.13. *Assume* $\text{Reach} \cap S \neq \varnothing$. *Then* $\text{Reach} \cap M(P(|\mathcal{P}|, \sharp[\mathsf{Allow}])) \neq \varnothing$, *where P is a polynomial.*

The proof is very similar to the one of [BFS15, proposition 2], yet using another model, which only allows certain forms of actions[2] and where players can see the local actions.

We argue that these differences are not relevant for the announced result, and give a sketch of the proof.

[2]In the context of lossy broadcast messaging, each local strategy chooses which message to send, and which transition to take when receiving a message. This corresponds to actions of the form $\{(q, \mathbf{W}(d), q')\}$ and $\{(q, \mathbf{R}(d_1), q_1) \dots (q, \mathbf{R}(d_n), q_n)\}$ with d_i all different.

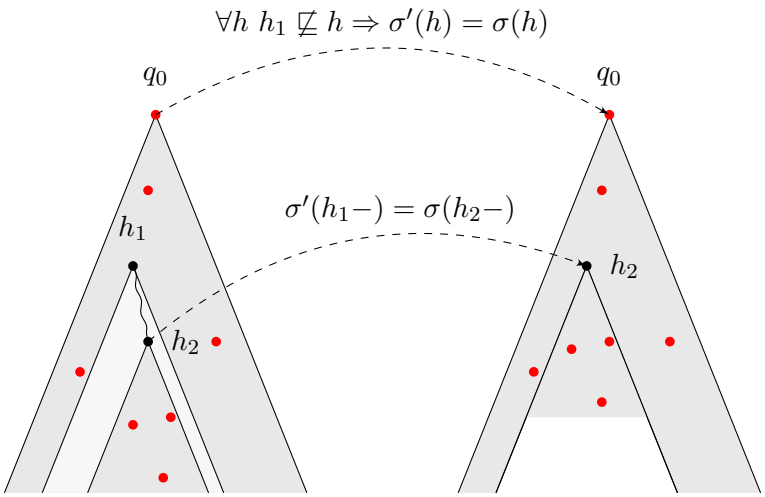


Figure 11.2 – Illustration of the proof of Theorem 11.13, when $h_1 \sqsubseteq h_2$, $\mathit{first}(h_1) = \mathit{first}(h_2)$ and subtrees rooted in h_1 and h_2 contain the same set of important nodes (in red). Since, h_1 and h_2 have the same final state, we can replace strategy from subtree rooted in h_1 by the one rooted in h_2 (same allowed actions). This is possible since all the important nodes are in the subtree rooted in h_2 or outside the subtree rooted in h_1, therefore all removed nodes and transitions can be generated from somewhere else.

Proof. Let $\sigma \in \text{Reach} \cap S$. We consider $\pi \in \text{agree}(\sigma) \cap U_0 \cdot \Gamma^* U_f$ and the sequence of multiset of histories ρ as given from Definition 11.6. At step i, we denote by $\hat{\rho}[i] \in Q^+$ the history that is picked to evolve, that is the unique history (the same as in Definition 11.6) word $h = \hat{\rho}[i]$ such that $\rho[i+1](h) = \rho[i](h) - 1$.

We define $T' \subseteq T$ as the set of transitions that can be seen:

$$T' = \bigcup_{1 \leq i \leq |\rho|} \sigma(\hat{\rho}[i])$$

In particular, there exists a transition $t_f = (q, \mathbf{f}, q_f) \in T'$.

We define the set H of seen histories as

$$H = \{\hat{\rho}[i] \mid 1 \leq i \leq |\rho|\}$$

H is closed by prefix, and can be seen as a tree, with root q_0 (the initial state of $\mathcal{P}$). For $h \in H$, the subtree rooted in h can be expressed as $H \cap (hQ^*)$.

We define $I \subseteq H$, the set of important nodes, as the histories that trigger a transition for the first time in the run, namely:

$$I = \{\hat{\rho}[i] \quad | \quad 1 \leq i \leq |\rho| \quad \wedge \quad \forall j \in [1, i-1]\ \sigma(\hat{\rho}[i]) \not\subseteq \sigma(\hat{\rho}[j])\}$$

We argue that we can convert σ, π and ρ simultaneously, such that T' is kept unchanged, $q_0 \in H$, and additionally, for any $h_1, h_2 \in H$, if $\mathit{last}(h_1) = \mathit{last}(h_2)$ and $I \cap (h_1 Q^*) = I \cap (h_2 Q^*)$, then $h_1 = h_2$.

Intuitively, this last property states that there are no two subtrees $H \cap (h_1 Q^*)$ and $H \cap (h_2 Q^*)$, with roots ending in the same state, containing the same important nodes.

Assume, that the property is not satisfied for some h_1 and h_2:

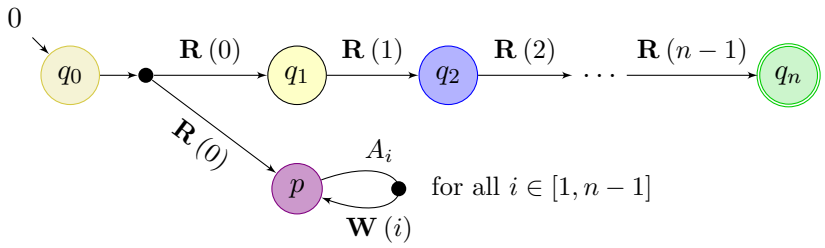


Figure 11.3 – Example of a protocol with allowed actions requiring linear memory for a pure strategy to be a cut-off strategy for positive coverability of q_n.

- If h_1 and h_2 are roots of disjoint subtrees, then $I \cap (h_1 Q^*) = I \cap (h_2 Q^*) = \varnothing$. We can "remove" h_1, h_2 and its successors from H.
- Otherwise, assume, without loss of generality, that h_1 is a parent of h_2 ($h_1 \sqsubseteq h_2$). Then, all important nodes lie in the subtree $H \cap (h_2 Q^*)$, that is to say $H \cap (h_1 Q^* \setminus h_2 Q^*)$ contains no important nodes. Thus, this last set can be "removed" and replaced by the subtree $H \cap (h_2 Q^*)$ now rooted from history h_1. This operation is depicted on Figure 11.2.

Such procedure removes at each step at least one node, so we are guaranteed to terminate. Each node h that is "removed" may have enabled some transition t, however, since $h \notin I$, we ensure that some other node $h' \in I$ is produced and can enable the same transition t. Thanks to the copycat lemma, we can produce as many processes with history h' as required, and before they are needed.

Once the announced property is achieved, we can bound the size $\sharp\,[H]$ of the tree by $\sharp\,[Q] \cdot (\sharp\,[\mathsf{Allow}] + 1)$ which is a polynomial in $|\mathcal{P}|$ and $\sharp\,[\mathsf{Allow}]$. In particular, the resulting strategy σ requires only polynomial memory. □

As illustration of Theorem 11.13, we provide in Figure 11.3 an example of protocol and allowed actions, which require linear memory when considering pure actions only.

11.2.3 Summary

It is not always possible to "trade randomness for memory". Consider for example the protocol and allowed actions depicted in Figure 11.4: there exists a mixed cut-off strategy for positive coverability, however no pure strategy is a cut-off strategy.

We conclude our study of positive probability coverability by the following theorem:

Theorem 11.14. *The following problem can be solved in polynomial time:*
ReachCoverStrat

INPUT: *A leaderless non-atomic protocol $\mathcal{P}$, a set of allowed actions* Allow *and a final state $q_f \in Q$.*

QUESTION: *Whether there exists a cut-off strategy σ for positive probability reachability of $U_f = \{\gamma \mid \gamma(q_f) > 0\}$.*

The following problem is NP*-complete:*
pReachCoverStrat

INPUT: *A leaderless non-atomic protocol $\mathcal{P}$, a set of allowed actions* Allow *and a final state $q_f \in Q$.*

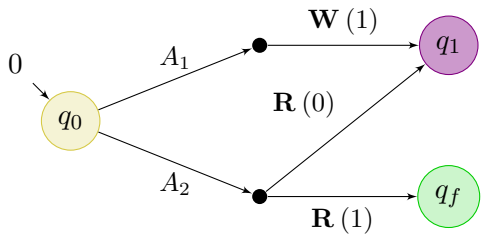


Figure 11.4 – Example of a protocol with allowed actions requiring randomization for a local strategy to be a cut-off strategy for positive coverability of q_f. In order to cover q_f, a local strategy has to play A_1 for one of the process, in order to write value 1, then play A_2 for another process. However, each process only sees its own history, which is always q_0.

QUESTION: *Whether there exists a cut-off strategy σ,* which is pure*, for positive probability reachability of $U_f = \{\gamma \mid \gamma(q_f) > 0\}$.*

Proof.
- As stated by Theorem 11.12, we can check that U_f is reached with positive probability in the original protocol $\mathcal{P}$, which can be done in polynomial time.
- NP-completeness of pReachCoverStrat is similar to the proof of [BFS15, theorem 2].

□

11.3 Safety

We address now the existence problem of a cut-off strategy for safety from a coverability objective. The main contribution of the section is the proof that pure memoryless strategies suffice. Intuitively, both pure and memoryless classes are restrictions of the strategies, that can only reduce the set of accessible configurations.

We define by Safe $\subseteq \mathbb{S}$ the set of cut-off strategies for safety. While a cut-off strategy for reachability requires only a finite run to prove reachability, a cut-off strategy for safety has to avoid U_f for any arbitrary run. Thus, under-approximation of the strategy by a finite tree is not a suitable technique anymore, as runs may explore arbitrary long histories. Our proof consists in transforming a cut-off strategy for safety σ into another strategy that is still safe, and memoryless "for small histories", then applying a limit/diagonal argument.

As for Lemma 9.6, we can sum two paths together in the leaderless case:

Lemma 11.15. *Assume $\mathcal{P}$ is leaderless non-atomic. Let σ be a strategy, $\pi_1 : (q, \mu_1, d) \to^* (q, \mu_1', d_1)$ and $\pi_2 : (q, \mu_2, d) \to^* (q, \mu_2', d_2)$, that both agree with σ. Then $(q, \mu_1 \oplus \mu_2, d) \to^* (q, \mu_1' \oplus \mu_2', d')$ with $d' = d_1$ or $d' = d_2$, and there exists such a path that agrees with σ.*

The proof is omitted, as it is similar to Lemma 9.6. In fact, this result can be derived from this previous lemma applied to the "meta protocol" $\mathcal{P}(\sigma)$ informally defined before.

For $h \in Q^+$, we say that h is *enabled by* σ if there exists a run $\pi \in \text{agree}(\sigma) \cap U_0 \cdot \Gamma^*$ with associated histories ρ from Definition 11.6 such that h appears in ρ, that is to say $\exists i\ h \in \overline{\rho[i]}$, where $\overline{\rho[i]}$ corresponds to the support of multiset of histories $\rho[i]$. We denote by $\text{en}(\sigma)$ the set of enabled histories in σ.

Lemma 11.16. *Let $\sigma \in \mathbb{S}$, $h_1, h_2 \in Q^+$ such that*

- *last*$(h_1) =$ *last*(h_2);
- $h_2 \in \mathrm{en}(\sigma)$.

We define σ' *by*

$$\sigma' : \begin{cases} h_1 \cdot h \in h_1 \cdot Q^* \mapsto \sigma(h_2 \cdot h) \\ h \notin h_1 \cdot Q^* \mapsto \sigma(h) \end{cases}$$

Then, $\mathrm{en}(\sigma') \subseteq \mathrm{en}(\sigma)$.

Proof. Consider $h_f \in \mathrm{en}(\sigma')$, so there exists $\pi \in \mathrm{agree}\,(\sigma') \cap U_0\Gamma^*$, with ρ the corresponding sequence of histories, such that $h_f \in \overline{last(\rho)}$. We will convert ρ in order to agree with σ.

h_2 is enabled by σ so there exists a run $\pi' \in \mathrm{agree}\,(\sigma)$, with sequence of histories ρ' such that $h_2 \in \overline{last(\rho')}$. For any i such that $\hat{\rho}[i] = h_1$, we add a copy of π' into π thanks to Lemma 11.15. Any further execution of process with history h_1 is replaced by one of the process with history h_2 added in π'.

At the end of the transformation, ρ is a run that agrees with σ and contains history h_f, hence $h_f \in \mathrm{en}(\sigma)$. □

Notice that $\mathrm{en}(\sigma)$ contains an history ending in U_f if, and only if, $\sigma \notin \mathrm{Safe}$. We conclude, that the property carried by the lemma is stronger than just keeping safety property. This extra property is relevant to converge when iteratively constructing a memoryless strategy.

The following theorem concludes that pure memoryless strategies suffice for safety cut-off.

Theorem 11.17. *Assume* $\mathrm{Safe} \neq \varnothing$*, then* $\mathrm{Safe} \cap M \neq \varnothing$.

Proof. Let $\sigma \in \mathrm{Safe}$ and let $\preceq$ be a total ordering over Q^+ compatible with the length function ($\forall h, h'\ h \preceq h' \Rightarrow |h| \leq |h'|$). We define

$$P = \{(h_1, h_2) \mid h_1, h_2 \in Q^+,\ last(h_1) = last(h_2),\ h_2 \preceq h_1, h_1 \neq h_2\}$$

and consider $f : \mathbb{N} \to P$ that enumerates P in the lexicographic order: this is possible since $\preceq$ is well-founded.

For any $n \in \mathbb{N}$, we construct $\sigma^n \in \mathrm{Safe}$ by induction:

- $\sigma^0 = \sigma$
- σ^{n+1} is defined by applying Lemma 11.16 to σ^n and the pair of histories $(h_1, h_2) = f(n)$ whenever $h_2 \in \mathrm{en}(\sigma^n)$, otherwise $\sigma^{n+1} = \sigma^n$. In both cases, we have $\mathrm{en}(\sigma^{n+1}) \subseteq \mathrm{en}(\sigma^n)$.

Let $k \in \mathbb{N}$, we define $N(k)$ such that:

$$\forall n \geq N(k)\ f(n) = (h_1, h_2) \Rightarrow |h_1| > k$$

Indeed, there is a finite number of pair of words of length smaller than k.

For a fixed history h and $n \geq N(|h|)$, if $f(n) = (h_1, h_2)$, then $|h_1| > |h|$ and $h \notin h_1 \cdot Q^*$, so whether Lemma 11.16 was applied or not, we have $\sigma^n(h) = \sigma^{n+1}(h)$. We conclude that

$$\forall h \in Q^+\ \forall n \geq N(|h|)\ \sigma^n(h) = \sigma^{N(|h|)}(h)$$

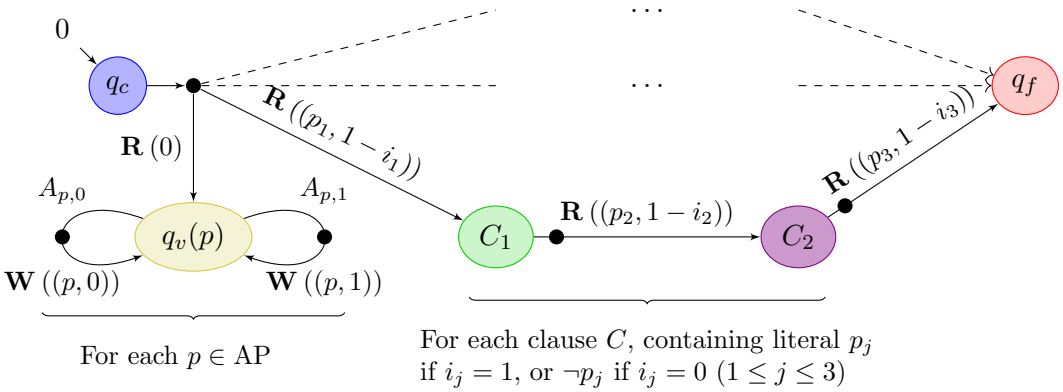


Figure 11.5 – Reduction of 3SAT to the existence of a cut-off strategy for safety.

This allows us to define the limit strategy, σ^∞ on $E^\infty = \cap_{n\geq 0}\mathrm{en}(\sigma^n)$:

$$\sigma^\infty : h \in E^\infty \mapsto \sigma^{N(|h|)}(h)$$

We complete the definition of σ^∞ over Q^+ in the following way: for each $q \in Q$, let $a_q = \sigma(hq)$ for some fixed $hq \in E^\infty$ (or any allowed action otherwise). Then for any $h'q \notin E^\infty$, we let $\sigma^\infty(h'q) = a_q$.

We note that any history enabled by σ^∞ is in[3] E^∞: for any run that agrees with σ^∞, consider its finite set of histories H. Then, for $n = \max_{h\in H} N(|h|)$ we have $\forall h \in H\ \sigma^n(h) = \sigma^\infty(h)$ so the same run also agrees with σ^n. This means that $\mathrm{en}(\sigma^\infty) \subseteq \mathrm{en}(\sigma^n) \subseteq E^\infty$. In particular, $\sigma^\infty \in$ Safe.

We also check that σ^∞ is in fact memoryless, by showing by induction on $h_1 \in Q^+$ (with respect to $\preceq$), that

$$\forall (h_1, h_2) \in P \cap (Q^+ \times E^\infty)\ \forall n \geq N(|h_1|) \quad \sigma^n(h_1) = \sigma^n(h_2)$$

If no such pair exists, the result is immediate. Otherwise, after $N(|h_1|)$, we have enumerated all pairs starting with h_1 as first coordinate. In particular, there exists some maximal h_1' for which Lemma 11.16 was applied for the last time, on (h_1, h_1'), hence, $h_2 \preceq h_1'$ by maximality (remember h_2 is always enabled). We conclude by induction hypothesis applied on pair (h_1', h_2): for all $n \geq N(|h_1|) \geq N(|h_1'|)$, $\sigma^n(h_1) = \sigma^n(h_1') = \sigma^n(h_2)$.

Finally, we extract, for each state q, an action $\sigma'(q) \in \mathsf{Allow}(q) \cap 2^{\sigma^\infty(q)}$ in order to define $\sigma' \in M$, with $\sigma' \subseteq \sigma^\infty$. By Lemma 11.11, we conclude that $\sigma' \in \mathrm{Safe} \cap M$. □

We conclude our study of safety coverability by the following theorem:

Theorem 11.18. *The two following problems are* NP*-complete:*
SafeCoverStrat

INPUT: *A leaderless non-atomic protocol $\mathcal{P}$, a set of allowed actions* Allow *and a final state $q_f \in Q$.*

QUESTION: *Whether there exists a cut-off strategy σ for safety from $U_f = \{\gamma \mid \gamma(q_f) > 0\}$.*

pSafeCoverStrat *defined as* SafeCoverStrat *for pure strategies only.*

[3]Note the inclusion may be strict, as we may have $\mathrm{en}(\sigma^\infty) \neq E^\infty$.

Proof.

- Membership in NP is due to the existence of memoryless strategies stated in Theorem 11.17. Such a strategy σ is of linear size in the original protocol, and we can check that the resulting composed protocol $\mathcal{P}\langle\sigma\rangle$ cannot reach q_f. Because we are considering coverability objective of q_f without leader, this check can be done in polynomial time (see [EGM13] or consider an increasing support path in the symbolic graph).

- NP-hardness is proven by a reduction of 3SAT ([Coo71]). Given an instance of this problem, with atomic proposition set AP and set of clauses X, we build the protocol depicted in Figure 11.5, where only non-trivial actions are allowed from states $q_v(p)$, for each $p \in \mathrm{AP}$. Intuitively, the strategy chooses a valuation for p in $q_v(p)$, while the scheduler challenges the whole assignment by picking a clause C and checking that each literal of C has the wrong truth value, which leads to unsafe state q_f.

  More formally, if there exists a cut-off strategy for safety from q_f, we can assume by Theorem 11.17 that such strategy is memoryless and pure, and consider the corresponding valuation ν. Since, q_f cannot be reached by the scheduler, we check that ν satisfies all the clauses in X. Conversely, if ν satisfies X, we consider the memoryless strategy that plays $A_{p,\nu(p)}$ from $q_v(p)$ for each $p \in \mathrm{AP}$ and check that such strategy is safe.

□

Surprisingly, the safety problem in non-atomic protocols with a leader, but without strategies, is proven by [EGM13] to be coNP-complete. Hardness is achieved thanks to the leader process which encodes the valuations of atomic propositions (in the dual reachability objective setting). This has to be compared with our proof of Theorem 11.18 which exploits strategies in order to encode valuations. On the other side, membership in NP is proved thanks to the lack of a leader (polynomial time check).

11.4 Conclusions

We have studied several techniques for the existence of cut-off strategies for positive reachability and safety objectives. Both problems are not dual anymore when shifting to synthesis of local strategies. On the one hand, allowing randomization or memory is crucial for positive reachability objectives:

- If we allow randomization, we have all incentive to consider the most randomized strategy.

- If we restrict to pure strategies, polynomial memory is required, and suffice.

We argue that randomizing all actions may be expensive in a practical setting, so even when randomization helps, we may be interested in synthesizing strategies that still record some information in order to minimize the amount of required entropy. Moreover, mixed strategies may drastically increase expected time to objective. Such quantitative questions were eluded on purpose and left as future work.

On the other hand, safety requires no memory. However, note that our techniques heavily rely on three assumptions: non-atomicity, lack of a leader and coverability objective. First property seems legitimate as the cut-off decision problem for atomic networks is already an open problem. However, Chapter 8 studied more general cases with a leader process and a

general simple objective. We conjecture that such generalizations will require memory, even in the safety case, but the existence of sufficient polynomial memory strategies is an open problem.

Finally, little is known about the existence of almost-sure reachability cut-off strategies. First of all, results from Chapter 9 states that the problem is at least PSPACE-hard, even without leader. Moreover, technical difficulties occur when defining a discrete characterization, in the flavour of lemmas 7.22 and 11.9. Indeed, a Pre/Post characterization would require carrying the intermediate memories of each process. Nonetheless, the low complexity classes of strategies for positive reachability and safety objectives raises hopes for the almost-sure reachability question, which somehow combines the previous properties.

Bibliography

[ABG15] C. Aiswarya, Benedikt Bollig, and Paul Gastin. An automata-theoretic approach to the verification of distributed algorithms. In Luca Aceto and David de Frutos-Escrig, editors, *Proceedings of the 26th International Conference on Concurrency Theory (CONCUR'15)*, volume 42 of *Leibniz International Proceedings in Informatics*, pages 340–353. Leibniz-Zentrum für Informatik, September 2015.

[AGC05] Bernard Aboba, Erik Guttman, and Stuart Cheshire. Dynamic configuration of IPv4 link-local addresses. 2005.

[AHK07] Luca de Alfaro, Thomas Henzinger, and Orna Kupferman. Concurrent reachability games. *Theoretical Computer Science*, 386(3):188–217, 2007.

[AJK16] Simon Außerlechner, Swen Jacobs, and Ayrat Khalimov. Tight cutoffs for guarded protocols with fairness. In Barbara Jobstmann and K. Rustan M. Leino, editors, *Proceedings of the 17th International Workshop on Verification, Model Checking, and Abstract Interpretation (VMCAI'16)*, volume 9583 of *Lecture Notes in Computer Science*, pages 476–494. Springer-Verlag, January 2016.

[AJKR14] Benjamin Aminof, Swen Jacobs, Ayrat Khalimov, and Sasha Rubin. Parametrized model checking of token-passing systems. In Kenneth L. McMillan and Xavier Rival, editors, *Proceedings of the 15th International Workshop on Verification, Model Checking, and Abstract Interpretation (VMCAI'14)*, volume 8318 of *Lecture Notes in Computer Science*, pages 262–281. Springer-Verlag, January 2014.

[ALW89] Martín Abadi, Leslie Lamport, and Pierre Wolper. Realizable and unrealizable specifications of reactive systems. In *Proceedings of the 16th International Colloquium on Automata, Languages and Programming*, ICALP '89, pages 1–17, London, UK, UK, 1989. Springer-Verlag.

[AM04] Luca de Alfaro and Rupak Majumdar. Quantitative solution of omega-regular games. *Journal of Computer and System Sciences*, 68(2):374 – 397, 2004.

[ARZ15] Benjamin Aminof, Sasha Rubin, and Florian Zuleger. On the expressive power of communication primitives in parameterised systems. In Martin Davis, Ansgar Fehnker, Annabelle K. McIver, and Andrei Voronkov, editors, *Proceedings of the 20th International Conference Logic Programming and Automated Reasoning (LPAR'15)*, volume 9450 of *Lecture Notes in Computer Science*, pages 313–328. Springer-Verlag, November 2015.

[BBG08] Christel Baier, Nathalie Bertrand, and Marcus Größer. On decision problems for probabilistic Büchi automata. In Roberto Amadio, editor, *Foundations of Software Science and Computational Structures: 11th International Conference, FOSSACS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29 - April 6, 2008. Proceedings*, pages 287–301, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[BBMU11] Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Nash equilibria in concurrent games with Büchi objectives. In *Proceedings of the 30th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'11)*, volume 13 of *LIPIcs*, pages 375–386. Leibniz-Zentrum für Informatik, 2011.

[BFS14] Nathalie Bertrand, Paulin Fournier, and Arnaud Sangnier. Playing with probabilities in reconfigurable broadcast networks. In *Proc. of FOSSACS*, LNCS 8412, pages 134–148. Springer, 2014.

[BFS15] Nathalie Bertrand, Paulin Fournier, and Arnaud Sangnier. Distributed Local Strategies in Broadcast Networks. In Luca Aceto and David de Frutos Escrig, editors, *26th International Conference on Concurrency Theory (CONCUR 2015)*, volume 42 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 44–57, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[BL69] Julius Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of American Mathematical Society*, 138:295–311, 1969.

[BMR+16] Patricia Bouyer, Nicolas Markey, Mickael Randour, Arnaud Sangnier, and Daniel Stan. Reachability in networks of register protocols under stochastic schedulers. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP'16) – Part II*, volume 55 of *Leibniz International Proceedings in Informatics*, pages 106:1–106:14, Rome, Italy, July 2016. Leibniz-Zentrum für Informatik.

[BMS14] Patricia Bouyer, Nicolas Markey, and Daniel Stan. Mixed Nash equilibria in concurrent games. In *Proceedings of the 34th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'14)*, volume 29 of *Leibniz International Proceedings in Informatics*, pages 351–363. Leibniz-Zentrum für Informatik, December 2014.

[BMS16] Patricia Bouyer, Nicolas Markey, and Daniel Stan. Stochastic equilibria under imprecise deviations in terminal-reward concurrent games. In Domenico Cantone and Giorgio Delzanno, editors, *Proceedings of the 7th International Symposium on Games, Automata, Logics, and Formal Verification (GandALF'16)*, volume 226 of *Electronic Proceedings in Theoretical Computer Science*, pages 61–75, Catania, Italy, September 2016.

[Bre12] Romain Brenguier. *Nash Equilibria in Concurrent Games – Application to Timed Games*. PhD thesis, ENS Cachan, France, 2012.

[BSHV03] Henrik C. Bohnenkamp, Peter van der Stok, Holger Hermanns, and Frits W. Vaandrager. Cost-optimization of the IPv4 zeroconf protocol. In *2003 International Conference on Dependable Systems and Networks (DSN 2003), 22-25 June 2003, San Francisco, CA, USA, Proceedings*, pages 531–540. IEEE Computer Society, 2003.

[CD14] Krishnendu Chatterjee and Laurent Doyen. Partial-observation stochastic games: How to win when belief fails. *ACM Transactions on Computational Logic*, 15(2:16), April 2014.

[CDT09] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player nash equilibria. *Journal of the ACM (JACM)*, 56(3):14, 2009.

[CEP95] Allan Cheng, Javier Esparza, and Jens Palsberg. Complexity results for 1-safe nets. *Theoretical Computer Science*, 147(1):117 – 136, 1995.

[CFGR16] Rodica Condurache, Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. The complexity of rational synthesis. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 121:1–121:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[CH12] Krishnendu Chatterjee and Thomas A Henzinger. A survey of stochastic ω-regular games. *Journal of Computer and System Sciences*, 78(2):394–413, 2012.

[Cha05] Krishnendu Chatterjee. Two-player nonzero-sum ω-regular games. In *Proc. 16th International Conference on Concurrency Theory (CONCUR'05)*, volume 3653 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2005.

[CJM04] Krishnendu Chatterjee, Marcin Jurdziński, and Rupak Majumdar. On Nash equilibria in stochastic games. In *Proc. 18th International Workshop on Computer Science Logic (CSL'04)*, volume 3210 of *Lecture Notes in Computer Science*, pages 26–40. Springer, 2004.

[Con92] Anne Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203 – 224, 1992.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

[CSM00] Part 3: Carrier sense multiple access with collision detect on (csma/cd) access method and physical layer specifications. *IEEE Std 802.3, 2000 Edition*, pages i–1515, 2000.

[CTTV04] Edmund M. Clarke, Muralidhar Talupur, Tayssir Touili, and Helmut Veith. Verification by network decomposition. In Philippa Gardner and Nobuko Yoshida, editors, *Proceedings of the 15th International Conference on Concurrency Theory (CONCUR'04)*, volume 3170 of *Lecture Notes in Computer Science*, pages 276–291. Springer-Verlag, August-September 2004.

[CW03] Scott A. Crosby and Dan S. Wallach. Denial of service via algorithmic complexity attacks. In *Proceedings of the 12th Conference on USENIX Security Symposium - Volume 12*, SSYM'03, pages 3–3, Berkeley, CA, USA, 2003. USENIX Association.

[DEGM15] Antoine Durand-Gasselin, Javier Esparza, Pierre Ganty, and Rupak Majumdar. Model checking parameterized asynchronous shared-memory systems. In Daniel Kroening and Corina S. Pasareanu, editors, *Proceedings of the 27th International Conference on Computer Aided Verification (CAV'15)*, volume 9206 of *Lecture Notes in Computer Science*, pages 67–84. Springer-Verlag, July 2015.

[DJLL13] Stéphane Demri, Marcin Jurdziński, Oded Lachish, and Ranko Lazić. The covering and boundedness problems for branching vector addition systems. *Journal of Computer and System Sciences*, 79(1):23–38, February 2013.

[DKM+15] Ankush Das, ShankaraNarayanan Krishna, Lakshmi Manasa, Ashutosh Trivedi, and Dominik Wojtczak. On pure nash equilibria in stochastic games. In *Theory and Applications of Models of Computation*, volume 9076 of *Lecture Notes in Computer Science*, pages 359–371. Springer International Publishing, 2015.

[DKN+12] Marie Duflot, Marta Kwiatkowska, Gethin Norman, David Parker, Sylvain Peyronnet, Claudine Picaronny, and Jeremy Sproston. Practical applications of probabilistic model checking to communication protocols. In S. Gnesi and T. Margaria, editors, *Formal Methods for Industrial Critical Systems: A Survey of Applications*, pages 133–150. Wiley, 2012.

[DSTZ12] Giorgio Delzanno, Arnaud Sangnier, Riccardo Traverso, and Gianluigi Zavattaro. On the complexity of parameterized reachability in reconfigurable broadcast networks. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *Proceedings of the 32nd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'12)*, volume 18 of *Leibniz International Proceedings in Informatics*, pages 289–300. Leibniz-Zentrum für Informatik, December 2012.

[DSZ10] Giorgio Delzanno, Arnaud Sangnier, and Gianluigi Zavattaro. Parameterized verification of ad hoc networks. In Paul Gastin and François Laroussinie, editors, *Proceedings of the 21st International Conference on Concurrency Theory (CONCUR'10)*, volume 6269 of *Lecture Notes in Computer Science*, pages 313–327. Springer-Verlag, September 2010.

[EFM99] Javier Esparza, Alain Finkel, and Richard Mayr. On the verification of broadcast protocols. In *Proceedings of the 14th Annual Symposium on Logic in Computer Science (LICS'99)*, pages 352–359. IEEE Comp. Soc. Press, July 1999.

[EGLM15] Javier Esparza, Pierre Ganty, Jérôme Leroux, and Rupak Majumdar. Verification of Population Protocols. In Luca Aceto and David de Frutos Escrig, editors, *26th International Conference on Concurrency Theory (CONCUR 2015)*, volume 42 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 470–482, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[EGM13] Javier Esparza, Pierre Ganty, and Rupak Majumdar. Parameterized verification of asynchronous shared-memory systems. In Natasha Sharygina and Helmut Veith, editors, *Proceedings of the 25th International Conference on Computer Aided Verification (CAV'13)*, volume 8044 of *Lecture Notes in Computer Science*, pages 124–140. Springer-Verlag, July 2013.

[ES96] E. Allen Emerson and A. Prasad Sistla. Symmetry and model checking. *Formal Methods in System Design*, 9(1-2):105–131, 12 1996.

[Esp14] Javier Esparza. Keeping a crowd safe: On the complexity of parameterized verification (invited talk). In Ernst W. Mayr and Natacha Portier, editors, *Proceedings of the 31st Symposium on Theoretical Aspects of Computer Science (STACS'14)*, volume 25 of *Leibniz International Proceedings in Informatics*, pages 1–10. Leibniz-Zentrum für Informatik, March 2014.

[Fin64] A. M. Fink. Equilibrium in a stochastic n-person game. *J. Sci. Hiroshima Univ. Ser. A-I Math.*, 28(1):89–93, 1964.

[FS01] Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1-2):63–92, April 2001.

[FV96] Jerzy Filar and Koos Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag New York, Inc., New York, NY, USA, 1996.

[GS53] D. Gale and F. Stewart. *Infinite games with perfect information*, volume 28, chapter Contributions to the theory of games, pages 245–266. Princeton University Press, 1953.

[GS92] Steven M. German and A. Prasad Sistla. Reasoning about systems with many processes. *Journal of the ACM*, 39(3):675–735, July 1992.

[Hag11] Matthew Hague. Parameterised pushdown systems with non-atomic writes. In Supratik Chakraborty and Amit Kumar, editors, *Proceedings of the 31st Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'11)*, volume 13 of *Leibniz International Proceedings in Informatics*, pages 457–468. Leibniz-Zentrum für Informatik, December 2011.

[Jur98] Marcin Jurdzinski. Deciding the winner in parity games is in UP $\cap$ co-UP. *Inf. Process. Lett.*, 68(3):119–124, 1998.

[Kak41] Shizuo Kakutani. A generalization of Brouwer's fixed point theorem. *Duke Math. J.*, 8(3):457–459, 09 1941.

[KPV15] Orna Kupferman, Giuseppe Perelli, and Moshe Y. Vardi. *Synthesis with Rational Environments*, pages 219–235. Springer International Publishing, Cham, 2015.

[KS81] P. R. Kumar and T. H. Shiau. Existence of value and randomized strategies in zero-sum discrete-time stochastic dynamic games. *SIAM Journal on Control and Optimization*, 19(5):617–634, 1981.

[Lip76] Richard J. Lipton. *The reachability problem requires exponential space.* Research report (Yale University. Department of Computer Science). Department of Computer Science, Yale University, 1976.

[LL69] Thomas M. Liggett and Steven A. Lippman. Short notes: Stochastic games with perfect information and time average payoff. *SIAM Review*, 11(4):604–607, 1969.

[Mar75] Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102:363–371, 1975.

[Mar98] Donald A. Martin. The determinacy of Blackwell games. *The Journal of Symbolic Logic*, 63(4):1565–1581, 1998.

[Min67] Marvin Minsky. *Computation: Finite and Infinite Machines.* Prentice Hall International, 1967.

[Nas50] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

[Neu28] John von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928.

[Put94] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[Rac78] Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6:223–231, 1978.

[RF10] Halsey L. Royden and Patrick M. Fitzpatrick. *Real Analysis.* Prentice Hall, 2010.

[RY86] Louis E. Rosier and Hsu-Chun Yen. A multiparameter analysis of the boundedness problem for vector addition systems. *Journal of Computer and System Sciences*, 32(1):105–135, February 1986.

[Sec97] Piercesare Secchi. Stationary strategies for recursive games. *Math. Oper. Res.*, 22(2):494–512, May 1997.

[Sel65] Reinhard Selten. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit. *Zeitschrift für die gesamte Staatswissenschaft*, 121:301–324 and 667–689, 1965.

[Sel75] Reinhard Selten. A reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4:25–55, 1975.

[Sha53] Lloyd S. Shapley. A value for n-person games. *Contributions to the theory of games*, 2:307–317, 1953.

[Sip97] Michael Sipser. *Introduction to the theory of computation.* PWS Publishing Company, 1997.

[SS01] Piercesare Secchi and William D. Sudderth. Stay-in-a-set games. *International Journal of Game Theory*, 30:479–490, 2001.

[Umm08] Michael Ummels. The complexity of Nash equilibria in infinite multiplayer games. In *Proc. 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'08)*, volume 4962 of *Lecture Notes in Computer Science*, pages 20–34. Springer, 2008.

[Umm10] Michael Ummels. *Stochastic Multiplayer Games: Theory and Algorithms*. Ph.D. Thesis, Department of Computer Science, RWTH Aachen, Germany, January 2010.

[UW09] Michael Ummels and Dominik Wojtczak. Decision problems for Nash equilibria in stochastic games. In *Proc. 23rd International Workshop on Computer Science Logic (CSL'09)*, volume 5771 of *Lecture Notes in Computer Science*, pages 515–530. Springer, 2009.

[UW11a] Michael Ummels and Dominik Wojtczak. The complexity of Nash equilibria in limit-average games. In *Proc. 22nd International Conference on Concurrency Theory (CONCUR'11)*, volume 6901 of *Lecture Notes in Computer Science*, pages 482–496. Springer, 2011.

[UW11b] Michael Ummels and Dominik Wojtczak. The complexity of Nash equilibria in limit-average games. Technical Report abs/1109.6220, CoRR, 2011.

[UW11c] Michael Ummels and Dominik Wojtczak. The complexity of Nash equilibria in stochastic multiplayer games. *Logical Methods in Computer Science*, 7(3), 2011.

[VNM47] J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1947.

[ZP96] Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1–2):343–359, 1996.

List of Figures

1.1 Simple games that require randomization 1
1.2 Hide-or-run game 3

3.1 Graphical representation of a $3 + 1/2$-player arena 16
3.2 Shifted hide-or-run game 29

4.1 Representations of a one-shot game 33
4.2 Matching-pennies game 33
4.3 Embedded game example 36
4.4 Counter example of Theorem 4.8 without action-visibility 39
4.5 Computation of an additional equilibrium payoff in game $\mathcal{C}'$. 39
4.6 Rescale module $\mathcal{R}_k$ 41
4.7 Testing module $\mathcal{T}$ 44
4.8 Counting modules C_k and D_k 44
4.9 Game $\mathcal{G}_{\mathcal{M}}$ for a given 2-counter machine $\mathcal{M}$ 47
4.10 Reduction of 1-maximal Nash equilibrium 51
4.11 Encoding of terminal rewards with qualitative objectives 52

5.1 Game where first player to quit loses 58
5.2 Example of an arena with some exiting actions 61
5.3 Encoding of imprecise deviations 66
5.4 Encoding of imprecise deviations: graphical representation 66
5.5 Comparison between ε-Nash equilibria and equilibria under imprecise deviations 68

6.1 Communication of wireless sensors as a threaded program 76
6.2 Communication of wireless sensors as a register protocol 77

7.1 Simple leaderless non-atomic protocol 80
7.2 Copycat lemma 85
7.3 Non-monotonous atomic protocol 85
7.4 Graphical representation of cut-off situations 88

8.1 Symbolic graph example 90
8.2 Path reduction in the symbolic graph 91

9.1 Example of a register protocol with atomic read/write operations. 93
9.2 Ultimate inclusion analysis 95
9.3 Polynomial bound for ultimate inclusion 96

9.4 Filter protocol $\mathcal{F}_n$ 98
9.5 Reversed filter protocol $\overline{\mathcal{F}}_n$ 99
9.6 Example of exponential negative cut-off 101
9.7 Distributed protocol $\mathcal{P}_{\mathcal{M}}$ for a linear-bounded Turing Machine $\mathcal{M}$ 103

10.1 Example of mapping operations over multiple register 117

11.1 Leaderless non-atomic protocol with allowed actions 123
11.2 Strategy reduction in Theorem 11.13 127
11.3 Protocol with pure allowed actions requiring linear memory 128
11.4 Protocol with allowed actions requiring randomization 129
11.5 Reduction of 3SAT to the existence of a cut-off strategy for safety. 131

List of Tables

4.1 Analysis of $\mathcal{R}_k$ from s_k with one fixed strategy 42
4.2 Analysis of $\mathcal{R}_k$ from r_0 with one fixed strategy 43
4.3 Decidability status of Nash Equilibria with qualitative objectives 55

10.1 Complexity results for cut-off decision problems 120

Le contenu du manuscrit étant rédigé en langue anglaise, notamment afin de pouvoir être évalué par un jury de thèse international, le présent chapitre expose un résumé substantiel en français.

Introduction

Les comportements stochastiques fournissent une modélisation d'évènements incertains qui permettent toutefois de fournir des mécanismes de désynchronisation voire pour sortir d'une situations symétriques. Imaginons par exemple que deux amis se retrouvent chaque jour pour jouer au jeu ≪ Pierre-Papier-Ciseaux ≫. Les deux joueurs choisissent un symbole parmi ⊕, ⊕ et ⊕, et la partie continue tant qu'un vainqueur n'est pas désigné.

Comme ces deux amis jouent fréquemment l'un contre l'autre, il leur est possible de retenir la stratégie de leur adversaire. Chaque jour, chaque joueur peut décider de changer d'avis et jouer une autre séquence de symboles pour contrer les actions supposées de son adversaire, ce qui conduit à un changement quotidien de sa stratégie, ou continuer de jouer de la même façon. D'autre part, il leur est nécessaire de se rappeler des coups joués précédemment, en nombre potentiellement élevé. Un jour, un des joueurs décide d'arrêter de tenter de prévoir les coups de son adversaire et lui annonce qu'à chaque tour, il jouera au hasard en lançant un dé : si sa valeur est 1 ou 2, son coup sera ⊕, ⊕ si le dé tombe sur 3 ou 4, et ⊕ sinon. Conscient de cette stratégie, son adversaire n'a cependant aucune méthode pour s'assurer une victoire avec probabilité supérieure à $\frac{1}{2}$. Jouer au hasard contraint d'une part les deux joueurs à gagner de façon équiprobable, mais assure également que la partie s'achève rapidement en moyenne : on peut calculer un nombre moyen de tours avant victoire égal à $\frac{3}{2}$.

Dans cette thèse, nous étudions la vérification formelle et la synthèse de systèmes aux comportements stochastiques. En effet, l'aléa en informatique est un élément clé dans la désynchronisation de processus, pour éviter les situations d'interblocage ou les collisions dans les protocoles de communications. Le concept fondamental dans ce domaine se rapproche ainsi de notre exemple de jeu ≪ Pierre-Papier-Ciseaux ≫, présenté en figure 1.1b. De nombreux protocoles s'appuient ainsi sur la randomisation, c'est le cas notamment de CSMA/CD [CSM00], pour lequel chaque participant choisit aléatoirement le délai avant une réémission pour cause de collision. Ainsi, l'aspect stochastique de ce dernier protocole est fondamental pour son étude [DKN+12]. Une autre mise en application de la randomisation à des fins de désynchronisation apparaît par exemple dans le protocole distribué Zeroconf, permettant le choix d'adresses sur un lien local [BSHV03, AGC05].

La randomisation permet également de se prémunir efficacement face à certaines formes d'attaques informatiques : du fait de spécifications ouvertes, de l'engouement pour l'open source, voire des progrès du désassemblage de programmes, il est plus que raisonnable de supposer que le fonctionnement interne de tout programme est publiquement connu, et que sa robustesse doit être assurée sous cette hypothèse. Cependant, certains algorithmes manipulant des structures de données ne possèdent une complexité raisonnable que lorsque les données en entrée suivent une distribution uniforme, et non dans le pire cas. Une absence d'aléa en entrée d'un système de traitement de données peut ainsi avoir des conséquences néfastes si un attaquant est en mesure de déclencher à dessein une exécution dans le pire cas. Pour illustrer ce propos, nous pouvons citer une attaque informatique consistant à exploiter le déterminisme des fonctions de hachage employées dans les langages de programmations modernes : avec peu de puissance de calcul et peu de bande passante, un attaquant a ainsi pu réaliser une attaque

par déni de service de serveurs web, en provoquant de nombreuses collisions [CW03].

Informatique et théorie des jeux

La théorie des jeux et le raisonnement stratégique constituent une discipline aux résultats prolifiques. Introduits par Von Neumann [Neu28] grâce à son théorème du min-max, puis davantage formalisé [VNM47], la modélisation systématique sous forme de jeux s'est révélée fructueuse dans de nombreux domaines, notamment l'économie grâce aux travaux de Nash ou Selten [Nas50, Sel65], car elle assure une bonne représentation des interactions et comportements entre personnes ou entités, appelés des *joueurs* ou *agents*. Ceux-ci prennent des décisions, c'est-à-dire des *actions* que l'on modélise par des *stratégies*, prenant en compte leurs *observations* de l'état du jeu et de l'environnement. Chaque joueur possède un *objectif*, généralement une fonction d'utilité voire une relation de préférence. L'utilité ou récompense fournie à chaque joueur est déterminée en fonction de la partie effectivement jouée, c'est-à-dire de la façon dont les actions des différents joueurs ont été jouées.

Lorsque l'utilité est fournie en tant que nombre, les jeux sont représentés par des matrices, représentant pour chaque joueur le gain final après le déroulement d'une partie en un coup. Le jeu est donc supposé ponctuel dans le temps. En informatique et comme nous l'avons vu dans les exemples précédents, une composante séquentielle apparaît, que nous modélisons en général grâce aux formalismes issus de la théorie des automates. Par exemple, le problème de Church s'intéresse à la réalisation d'une relation binaire définie par une formule logique sur les séquences infinies de mots. Plus précisément, ce problème soulève la question de l'implémentation d'une telle formule, sous la forme d'un circuit, prenant en entrée un mot infini, et fournissant en sortie un second mot infini en relation avec le premier. La réponse à cette question peut être obtenue par l'analyse d'un jeu à deux joueurs joués sur un graphe, où le premier joueur propose une sortie, en choisissant certaines transitions, tandis que le premier joueur essaye de l'en empêcher, en choisissant des transitions pour le mot d'entrée, mettant en défaut la relation binaire. Implémenter un tel circuit revient dans ce contexte à exhiber une stratégie gagnante pour le premier joueur [BL69], ce qui signifie habituellement *résoudre* le jeu.

Cette méthodologie est particulièrement adaptée à l'étude des *systèmes réactifs*, pour lesquels nous nous intéressons à la synthèse d'un contrôleur qui devra satisfaire des propriétés logiques tout en restant robuste face à un environnement, dont le comportement est imprévisible [ALW89].

Dans ces exemples, le premier joueur est dit *existentiel*, car nous nous intéressons à l'existence d'une seule stratégie gagnante, contre n'importe quelle stratégie du second joueur, que nous qualifions d'*universel*.

Non-déterminisme, aléa et problèmes quantitatifs

Lorsque l'on caractérise la victoire d'un joueur contre toutes les stratégies de son adversaire, nous adoptons une approche dans le *pire cas* puisque toutes les actions possibles sont considérées. Nous pouvons cependant remarquer que cette forme d'incertitude est très forte car elle considère des scenarios peu crédibles, qui ne se produisent pas forcément en pratique. À titre d'exemple, le modèle de transmission de données précédemment décrit peut vraisemblablement incorporer une composante d'équité entre les différents protagonistes, c'est-à-dire

supposer qu'il existe forcément un instant lors duquel il est possible d'émettre sans interférence par aucune autre partie.

Une notion intermédiaire d'incertitude est ainsi obtenue en considérant les comportements stochastiques, que l'on peut représenter comme les actions d'un joueur tierce, l'environnement, qui ne prend pas de décision à proprement parler, mais reste source d'aléa. Contrairement à un joueur universel, l'environnement n'a pas d'objectif propre, et ses actions peuvent contribuer ou s'opposer aux objectifs du joueur existentiel dont nous cherchons une stratégie. Introduire des probabilités dans ce modèle a deux conséquences majeures : premièrement, certaines exécutions envisagées dans le pire cas sont mises de côté. Par exemple, la succession infinie de matchs nuls dans la boucle du jeu de « Pierre-Feuille-Ciseaux » est impossible, dès lors qu'un des joueurs joue aléatoirement. Cette propriété se rapproche en ce sens de la notion d'équité. D'autre part, les phénomènes stochastiques fournissent un nouvel aspect quantitatif aux problèmes traités. Il devient par exemple possible de se demander si un joueur peut gagner avec une valeur de probabilité supérieure à un certain seuil.

Optimisation de stratégies

Il existe plusieurs concepts d'optimalité pour une stratégie, qui dépendent des conditions de gain considérées. Premièrement, une stratégie *optimale* conceptualise une situation où un joueur donné doit maximiser son gain, contre toutes les actions possibles de son adversaire. La meilleure valeur atteignable, contre la pire riposte est alors appelée valeur du jeu [Sha53]. Dans ce contexte, le jeu est supposé joué par deux joueurs aux objectifs antagonistes, c'est-à-dire à *somme nulle*. Bien que les valeurs du jeu pour le premier et deuxième joueurs ne soient pas nécessairement égales, de nombreux résultats de *déterminisation* établissent cette égalité pour de larges classes de jeux, notamment les jeux à tour déterministes, avec des objectifs réguliers [Mar75].

Lorsque des actions concurrentes et randomisées sont autorisées, des stratégies optimales peuvent ne pas exister bien que le jeu puisse toujours être déterminisé, au sens où la valeur n'est atteignable qu'asymptotiquement [Mar98]. Nous pouvons illustrer ce phénomène grâce au jeu à deux joueurs à somme nulle $\mathcal{H}$ (*hide-or-run* « courir ou fuir ») représenté par la figure 1.2. Le joueur 1 peut soit se cacher ($\hbar$) ou courir à l'abri (r) tandis que le second joueur peut choisir de jeter une boule de neige (s) ou attendre (w). Il a été démontré par [KS81] que ce jeu n'admet pas de stratégies optimales, bien que le premier joueur peut s'assurer une victoire avec probabilité arbitrairement proche de 1.

Lorsque le jeu implique plus de deux joueurs, ou qu'une partie des gains sont communs à plusieurs joueurs, nous remarquons que les notions de stratégies optimales et valeurs ne sont plus adaptées : les agents ne jouent plus nécessairement les uns contre les autres mais peuvent coopérer. Partant de cette observation, John Nash introduisit une nouvelle notion, dite d'équilibre, pour traiter les jeux à somme non nulle. Dans un tel équilibre, nous nous donnons une stratégie pour chaque joueur, que nous souhaitons optimale par rapport aux stratégies fixées des autres joueurs. Sur une structure de graphe, un équilibre décrit des coopérations entre les joueurs, pouvant s'accorder pour visiter certains états profitables. Lorsqu'un joueur ne respecte pas sa stratégie initialement prévue par l'équilibre, les autres joueurs peuvent réagir de plusieurs façon, en fonction de la notion d'équilibre choisie : ceux-ci peuvent riposter et abandonner leurs propres objectifs, ou décider de continuer à optimiser leurs propres objectifs malgré la déviation. Dans le premier cas, nous parlerons d'équilibres

de Nash, tandis que la seconde notion, plus restrictive, se nomme un équilibre parfait en sous-jeux [Sel65]. Cette dernière impose en effet en particulier l'existence d'un équilibre de Nash depuis tout état du jeu.

Lorsqu'un joueur choisit une stratégie dans un équilibre, il le fait en fonction du choix des autres joueurs qui cherchent à optimiser leurs propres objectifs, ce que l'on qualifie de choix *rationnels*. Cette hypothèse est particulièrement pertinentes dans le domaine de l'informatique où les joueurs représentent des programmes, des systèmes autonomes ou des périphériques, qui interagissent ensemble. Ainsi, une stratégie pour un joueur donné correspond à un programme ou un micrologiciel, que l'on recherche, suivant le contexte, à synthétiser ou que l'on suppose préexistant [KPV15].

Problématique

Dans cette thèse, nous nous concentrons sur la contribution des aspects stochastiques aux jeux sur graphes, munis d'objectifs simples d'accessibilité et sûreté.

Nous nous concentrons sur la décidabilité et calculabilité de problèmes sur les jeux stochastiques en fonction des paramètres suivants : (a) le nombre de joueurs, (b) leur mode d'interaction, (c) la mémoire utilisée par leurs stratégies, (d) l'aléa que peut produire une stratégie, (e) l'incertitude induite par l'environnement (stochastique et/ou non déterministe), (f) le relâchement possible des concepts étudiés.

Premièrement, nous nous intéressons à l'étude des jeux concurrents jouées sur un graphe par un nombre fixé d'agents, disposant de comportements stochastiques. Plus précisément, nous étudions le gain d'expressivité apporté par les actions concurrentes stochastiques dans un équilibre de Nash. Ainsi, nous montrerons qu'il est impossible de décider l'existence d'un équilibre de Nash dès lors que trois joueurs sont présents, ce qui justifiera l'étude d'une notion relâchée d'équilibre.

Malgré leur spécification très générale, les jeux concurrents ont une structure très rigide, qui n'est pas nécessairement adaptée à la modélisation informatique : une problématique importante que nous considérons ici est l'étude d'un modèle suffisamment flexible pour capturer les interactions stochastiques entre un nombre arbitraire de joueurs tout en conservant des propriétés de décidabilité et calculabilité.

Ainsi, nous introduisons un modèle où le nombre d'agents est un *paramètre* que l'on suppose de grande taille. Puisque le paramètre n'est *a priori* pas fixé, le graphe du jeu ne peut pas être explicitement décrit, et un modèle d'interaction doit être fourni à la place.

Travaux connexes

Les travaux fondateurs de Nash [Nas50] montrant l'existence d'un équilibre dans un jeu à un tour constituent le point de départ de notre étude. L'introduction de stratégies mixtes, c'est-à-dire randomisées, constitue l'argument clé permettant d'appliquer un théorème de point fixe sur une fonction bien choisie, que l'on peut montrer être continue. Cet argument peut être adapté aux jeux sur des graphes, par exemple dans le cas d'un *horizon fini*, c'est-à-dire quand la durée de la partie est bornée.

Le cas d'un *horizon infini* reste plus complexe à étudier bien que l'on puisse citer le cas des fonctions d'utilité décroissant exponentiellement au cours du temps. Les joueurs sont ainsi vivement encouragés à terminer la partie le plus vite possible et à engranger les gains en

début de partie. Cela a également pour effet d'assurer des propriétés de régularité (continuité) des fonctions précédemment évoquées et encore une fois assure l'existence d'un équilibre de Nash [Fin64].

Le cas d'un horizon infini sans facteur d'atténuation est en général plus complexe à résoudre, même dans le cas d'un seul joueur interagissant avec son environnement au comportement stochastique. Dans ce dernier cas, le jeu peut être vu comme un *processus de décision markovien*, pour lequel une famille de strategies asymptotiquement optimale existe [Put94]. Ce modèle est également étudié dans le cas général de plusieurs joueurs sous le nom de *processus de décision markovien compétitifs* [FV96].

Les fonctions d'utilité considérées dans cette thèse sont de simples objectifs d'accessibilité et de sureté de certains états, ce qui constitue un cas très particulier d'horizon infini. Ainsi, l'existence d'un équilibre est assurée pour les jeux munis d'objectifs de sûreté pour tous les joueurs, c'est-à-dire que chaque joueur a pour objectif de confiner le jeu à un certain sous-ensemble d'états marqués comme sûrs [SS01]. Dans le cas de l'accessibilité, l'existence d'un équilibre de Nash reste un problème ouvert, bien qu'une notion relâchée d'équilibre, où chaque joueur peut encore dévier en n'améliorant toutefois son gain que d'une petite valeur $\varepsilon > 0$, reste assurée [CJM04]. Des extensions considérant des classes de gains définies par des langages ω-réguliers sont étudiées, notamment dans le cas de deux joueurs [Cha05], tandis que le cas général reste ouvert. Pour une étude plus précise des résultats connus sur les jeux stochastiques et particulièrement les équilibres de Nash, nous pourrons nous référer à [CH12].

Lorsque nous retirons la composant stochastique, les objectifs ω-réguliers peuvent être étudiés plus simplement, notamment par l'encodage d'un équilibre de Nash comme une stratégie gagnante dans un nouveau jeu à deux joueurs [Bre12]. En ne retirant l'aspect concurrent, c'est-à-dire en considérant des jeux à tours, le problème de l'existence d'un équilibre dans le cas stochastique est déjà indécidable [Umm10]. Plus précisément, l'auteur montre qu'il est impossible de décider si un jeu impliquant 11 joueurs ou plus, avec des objectifs d'accessibilité terminaux, admet un équilibre où un joueur désigné gagne avec probabilité 1. Toutefois, si l'on se restreint à des stratégies sans mémoire, le problème peut être résolu en espace polynomial, même en imposant des conditions plus fortes sur les gains finaux des joueurs. Ceci fournit donc une méthode de calcul de la valeur moyenne d'un équilibre, pour chaque joueur. Nous pouvons aussi comparer ce résultat au problème du calcul d'un équilibre de Nash dans le cas d'un jeu à un tour, c'est-à-dire décrit par une matrice, qui est PPAD-complet même en présence de deux joueurs [CDT09]. Notons cependant que ces études de restrictions sans stochasticité ou sans actions concurrentes se concentrent sur des problèmes de décisions et de calcul, car l'existence d'un équilibre n'est pas assurée. On rappelle en effet que ces deux ingrédients sont cruciaux pour la preuve d'existence.

Nous avons vu que la complexité de l'étude d'un système dépend du nombre considéré de joueurs, ou agents, que l'on peut voir comme un paramètre variable. La vérification de systèmes paramétrés constitue ainsi un autre axe d'étude, introduit notamment par German et Sistla [GS92]. En général, les interactions entre agents peuvent être modélisées par des systèmes d'addition de vecteurs ou, de façon équivalente, par des réseaux de Petri, où chaque agent est représenté par un jeton. Accéder à un état de contrôle particulier par un agent signifie qu'un jeton a atteint une place particulière du réseau, ce qui correspond à une propriété de couverture. Le problème de décision correspondant à la couverture a été intensivement étudié par [Rac78, Lip76] qui l'établissent EXPSPACE-complet. Nous nous rendons ainsi compte que le problème de la vérification de systèmes paramétrés est un problème déjà difficile dans le cas non-stochastique. Cependant, des travaux sur des restrictions tels que les protocoles

de population [EGLM15], incluant des notions d'équité, nous montrent que l'ajout d'aspects stochastiques peut simplifier certains problèmes de décision. D'autre part, les réseaux de Petri et les protocoles de population restent des modèles très généraux, que l'on peut restreindre, en fonction des primitives de communications que l'on s'autorise. Une discussion sur le choix de ces primitives est présentées par [Esp14], et nous prenons ici un des modèles les plus faibles, consistant en la communication au travers d'un registre partagé entre les agents, via des écritures non-atomiques. Ce modèle, précédemment étudiés par [EGM13, DEGM15] dans le cas non-stochastique, se rapproche fortement des réseaux de diffusion avec reconfiguration [BFS14] pour lesquels des raisonnements stratégiques peuvent même être appliqués avec succès [BFS15].

Publications scientifiques

Les résultats présentés dans ce manuscrit complètent des publications auxquelles l'auteur de cette thèse a pris part : Le chapitre 4 décrit le problème d'indécidabilité de l'existence d'un équilibre, publié dans [BMS14], et discute le choix crucial d'un modèle de jeu où les joueurs n'observent pas les actions jouées. Le chapitre suivant 5 présente l'existence et le calcul d'une notion relâchée d'équilibre, tels que présentés dans [BMS16]. Enfin, la seconde partie de la thèse étend [BMR$^+$16], dont le théorème principal est repris en chapitre 9. Les principales extensions concernent l'étude d'objectifs plus généraux, mais aussi la présence de plusieurs registres en chapitre 10.

Partie 1 : Jeux stochastiques

La première partie du manuscrit s'intéresse à l'étude des jeux stochastiques concurrents dans lesquels les joueurs sont en nombre fixé et possèdent des objectifs d'accessibilité terminale ou de sûreté.

Cadre théorique

Le chapitre 3 fournit une définition formelle des jeux considérés : un *jeu concurrent* $\mathcal{G}$ fait intervenir un ensemble fini Agt de *joueurs*, ou *agents*. L'ensemble d'états du jeu est noté States, est supposé fini, et possède un élément distingué s_0 à partir duquel toute partie est initiée. La partie continue lorsqu'à chaque tour, les joueurs choisissent chacun une action parmi un ensemble fini Act, ce qui induit une transition vers un nouvel état. Chaque joueur agit conformément à une *stratégie*, notée σ_i pour le joueur $i \in \mathsf{Agt}$, correspondant à une fonction associant à tout mot $h \in \mathsf{States}^+$ une *décision*. Ainsi, h représente un *historique*, c'est-à-dire la séquence finie d'états visités depuis le début de la partie, en s_0, jusqu'à l'état courant depuis lequel une décision est prise par chaque joueur. Ici, les décisions des joueurs sont des *distributions de probabilité* sur l'ensemble des actions possibles, ce qui signifie que le prochain état est choisi de manière probabiliste. Lorsque σ_i est donné pour tout agent $i \in \mathsf{Agt}$, on note σ le *un profil de stratégies*, qui nous permet alors de définir une loi de probabilité notée $\mathbb{P}^\sigma$ sur l'ensemble des parties possibles. Le choix d'une stratégie σ_i par un joueur i se fait dans l'optique de l'optimisation de sa *fonction de gain*, qui lui est propre. Dans cette thèse, nous considérons principalement des fonctions de gain, ou *objectifs*, dits d'*accessibilité terminale*, c'est-à-dire que le gain de chaque joueur est un nombre réel déterminé par l'état final dans

lequel la partie s'est achevée. Lorsque la partie continue indéfiniment, le gain associé est toujours nul. De manière symétrique, on parle d'objectif de sûreté lorsque le joueur considéré a pour objectif de maintenir la partie dans un sous-ensemble d'état donné.

Formellement, une fonction de gain ϕ associe à tout mot fini ou infini dans $\mathsf{States}^+ \uplus \mathsf{States}^\omega$ un nombre réel. Dans le cas des deux types de fonctions (accessibilité terminale et sûreté), on démontre aisément qu'une telle fonction est mesurable, ce qui permet de définir son espérance $\mathbb{E}^\sigma(\phi)$ pour toute loi de probabilité $\mathbb{P}^\sigma$ où σ est une stratégie.

Un jeu possède un *profil d'objectifs* Φ, c'est-à-dire une fonction d'objectif Φ_i pour chaque joueur i. Celui-ci cherche à maximiser cette fonction, c'est-à-dire la quantité $\mathbb{E}^\sigma(\Phi_i)$ malgré les interactions avec les autres joueurs.

Deux cas de figures se dégagent :

- Lorsque seuls deux joueurs interviennent et que la somme $\Phi_1 + \Phi_2 = 0$ est nulle, tout gain acquis par un joueur constitue une perte pour l'autre joueur. Sous des hypothèses satisfaites ici, on démontre (voir [KS81]) que le jeu est déterminé, c'est-à-dire :

$$\overline{\nu}_1(s) = \inf_{\sigma_2 \in \mathbb{S}_2} \sup_{\sigma_1 \in \mathbb{S}_1} \mathbb{E}^{(\sigma_1,\sigma_2)}(\Phi_1 \mid s) = \underline{\nu}_1(s) = \sup_{\sigma_1 \in \mathbb{S}_1} \inf_{\sigma_2 \in \mathbb{S}_2} \mathbb{E}^{(\sigma_1,\sigma_2)}(\Phi_1 \mid s)$$

  De plus, ces valeurs sont atteintes asymptotiquement par des stratégies ne dépendant que de l'état courant, c'est-à-dire *stationnaires*. Elles ne sont pas nécessairement atteintes de manière exacte, comme le montre l'exemple de la figure 1.2.

- Lorsque plus de deux joueurs interviennent, ou que la somme de leur gain n'est plus constante, de nouveaux concepts d'optimalité ont été proposés notamment par Nash. Un profil de stratégie σ forme ainsi un équilibre de Nash (cf [Nas50]) lorsque :

$$\forall i \in \mathsf{Agt}\ \forall \sigma'_i\ \mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s) \leq \mathbb{E}^\sigma(\Phi_i \mid s)$$

  où $\sigma[i/\sigma'_i]$ désigne un nouveau profil de stratégie où le joueur i joue désormais la stratégie σ'_i.

Il est important de mentionner que la plupart des résultats établis dans la littérature considèrent des jeux où les actions sont observables, c'est-à-dire que la stratégie est une fonction définie sur des historiques de la forme $(s_0, A_0, s_1, A_1, \ldots s_k)$ où chaque A_i correspond au profil d'actions choisi conjointement par les joueurs depuis l'état s_i menant à l'état s_{i+1}. Dans cette thèse, les actions ne sont pas supposées observables (seule la séquence $(s_0, s_1 \ldots s_k) \in \mathsf{States}^+$ est fournie à la stratégie), ce qui nous mène à montrer dans un premier temps que les théorèmes de déterminisation précédents peuvent être étendus à ce nouveau cadre.

Indécidabilité de l'existence d'un équilibre de Nash

L'existence d'équilibre est assurée dans le cas de jeux se terminant en un tour, dès lors que l'on autorise des stratégies randomisées, grâce au théorème de Nash. Dans le cas de jeux sur des graphes, l'existence de stratégies optimales n'est pas assurée dans les jeux à somme nulle, ce qui implique a fortiori qu'un équilibre de Nash n'existe pas nécessairement.

Le problème de décision consiste ainsi à déterminer si, étant donné un jeu stochastique fourni en entrée, il existe une profil de stratégie formant un équilibre de Nash. Le chapitre 4 répond à cette question de manière négative : le problème est montré indécidable dès lors que

le jeu fait intervenir trois joueurs. La preuve s'appuie sur la réduction du problème de l'arrêt d'une machine à deux compteurs [Min67], de façon similaire à une précédente preuve à 14 joueurs [Umm08].

La preuve présentée ici s'appuie sur des caractéristiques supplémentaires, ce qui permet d'abaisser le nombre de joueurs nécessaires requis pour l'indécidabilité :

- Les jeux sont supposés concurrents, ce qui signifie que plusieurs joueurs choisissent simultanément la prochaine action, par opposition aux jeux à tour où au plus un joueur peut réaliser un choix non trivial depuis chaque état.
- Les actions sont inobservables, ce qui permet de réaliser des encodages plus complexes de la machine à deux compteurs.

De manière équivalente, la réduction nous permet de considérer le cas *qualitatif*, où les fonctions de gain ne prennent que des valeurs égales à 0 ou 1. Dans le cas de l'accessibilité terminale, cela revient à dire qu'un joueur gagne uniquement s'il arrive à atteindre un certain ensemble d'états finaux. Dans le cas d'un objectif de sûreté, le joueur ne gagne que s'il parvient à rester dans un ensemble d'états donné.

Nous établissons ainsi que l'existence d'un équilibre est indécidable dès lors que le jeu implique au moins trois joueurs aux objectifs qualitatifs, dont l'un au moins est un objectif d'accessibilité terminale et un autre un objectif de sûreté. Ceci est à mettre en regard d'autres configurations pour lesquelles l'existence d'un équilibre est assuré :

- Lorsqu'un seul joueur est présent, le jeu se résume à un processus de décision Markovien, pour lequel l'existence d'une stratégie optimale est connue.
- Lorsque tous les joueurs possèdent des objectifs de sûreté qualitatifs, [SS01] établit l'existence d'un équilibre de Nash.

Deux cas restent ainsi ouverts : l'existence d'un équilibre en présence de deux joueurs, ainsi que le cas où tous les objectifs considérés sont de sûreté.

Équilibres aux déviations imprécises

Le chapitre 5 conclut la première partie en introduisant une notion relâchée d'équilibres. Une première notion présentée par [CJM04] établit l'existence d'ε-équilibres de Nash, pour tout $\varepsilon > 0$. Formellement, un tel équilibre est caractérisé par un profil σ tel que :

$$\forall i \in \mathsf{Agt}\ \forall \sigma'_i\ \mathbb{E}^{\sigma[i/\sigma'_i]}(\Phi_i \mid s) \leq \mathbb{E}^{\sigma}(\Phi_i \mid s) + \varepsilon$$

Nous établissons ici l'existence d'une notion duale, que nous montrons incomparable à la notion précédente : un profil σ est un *équilibre sous déviation ε-imprécise* lorsque :

$$\forall i \in \mathsf{Agt}\ \forall \sigma'_i\ \exists \sigma''_i\ d(\sigma'_i, \sigma''_i) \leq \varepsilon\ \wedge\ \mathbb{E}^{\sigma[i/\sigma''_i]}(\Phi_i \mid s) \leq \mathbb{E}^{\sigma}(\Phi_i \mid s)$$

où $d(\sigma'_i, \sigma''_i)$ désigne la distance entre les stratégies σ'_i et σ''_i vue comme le supremum des distances entre distributions depuis tous les historiques possibles.

Ainsi, lorsqu'une déviation σ'_i est proposée par le joueur i, toute déviation à distance inférieure à ε doit également améliorer le gain du joueur i afin de mettre en échec l'équilibre.

Nous montrons qu'un tel profil de stratégies σ existe nécessairement, en adaptant des techniques de Nash ainsi que [SS01]. De plus, σ peut être supposé stationnaire, ce qui ouvre la voie au développement d'un algorithme de calcul en espace polynomial.

Partie 2 : Réseaux paramétrés

Dans la seconde partie du manuscrit, nous nous intéressons à un modèle plus réaliste où le nombre d'agents n'est pas déterminé à l'avance, et l'observation de chaque joueur est imparfaite. Plus précisément, le nombre d'agents est un paramètre, et nous souhaitons répondre à des problèmes de vérification indépendamment du paramètre choisi.

Différentes méthodes d'interaction

Puisque le paramètre n'est pas fixé, il n'est plus possible de considérer le graphe d'état du système complet, et il est nécessaire de spécifier séparément le comportement de chaque agent, puis les modalités d'interaction entre agents. Dans le chapitre 6, nous discutons des différentes méthodes d'interactions envisagées dans l'étude des systèmes paramétrés. Un bilan présenté par Esparza [Esp14] propose ainsi une classification des moyens de communication, par expressivité décroissante :

- Les diffusions de messages globaux (*broadcast*) ;
- Les variables partagées avec verrou ;
- La communication par rendez-vous ;
- Les variables partagées avec opérations *non atomiques* (sans verrou).

C'est ce dernier mode de communication qui est étudié ici, du fait de ses propriétés de monotonicité. En effet, lorsque plusieurs agents ou processus communiquent entre eux, il est toujours possible de dupliquer n'importe quel processus en lui faisant exécuter les mêmes transitions au même instant, celles-ci étant soit des lectures, soit des écritures. L'hypothèse de non atomicité se révèle cruciale car elle interdit la lecture puis écriture simultanée du registre, opération qui ne pourrait pas être dupliquée.

Réseaux paramétrés

Nous définissons ainsi dans le chapitre 7 le modèle formel de réseaux paramétrés avec variable partagée. Celui-ci est composé d'un domaine de définition D de la variable partagée, sa valeur initiale $d_0 \in D$, ainsi que de deux automates finis. Chaque automate (q, Q, T) décrit le comportement d'un processus, initialement dans l'état $q \in Q$, et dont les transitions, définies par l'ensemble T, sont étiquetées par des opérations de lecture $\mathbf{R}(d)$ ou d'écriture $\mathbf{W}(d)$, pour une certaine valeur de registre $d \in D$. Intuitivement, s'il existe une transition $(q, \mathbf{R}(d), q') \in T$ où $d \in D$ est la valeur courante du registre partagé, alors tout processus dans l'état q peut effectuer une transition vers l'état q', *sans altérer le registre*. S'il existe une transition $(q, \mathbf{W}(d), q') \in T$, alors tout processus dans l'état q peut effectuer une transition vers l'état q', en inscrivant la valeur d dans le registre partagé, et ceux peu importe la valeur courante du registre.

Nous appelons *configuration* d'un réseau paramétré $\mathcal{P}$ un triplet $\gamma = (q, \mu, d)$ où

- d est la valeur courante du registre partagé ;
- q est l'état courant du processus *leader*, décrit par le premier automate (q_l, Q_l, T_l) ;

- $\mu \in \mathbb{N}^{Q_c}$ est un multiensemble des états des processus *contributeurs*, décrits par le second automate (q_c, Q_c, T_c).

À chaque instant, l'ordonnanceur choisit un processus parmi le leader et les différents contributeurs et réalise une transition, mettant à jour l'état q ou le multiensemble μ, résultant en une nouvelle configuration γ'. On note généralement $\gamma \to \gamma'$ lorsqu'une telle transition est possible de la configuration γ vers γ', ou de manière équivalente $\gamma' \in \mathrm{Post}(\gamma)$ voire $\gamma \in \mathrm{Pre}(\gamma')$.

Objectifs et régularité

On remarque aisément que l'ajout d'un ou plusieurs processus contributeurs n'altère pas pas la validité d'un chemin : si $(q, \mu, d) \to^* (q', \mu', d')$ et $q_a \in Q_c$ est un état contributeur tel que $\mu(q_a) > 0$, alors il existe $q'_a \in Q_c$ tel que $\mu'(q'_a) > 0$ et $(q, \mu \oplus q_a, d) \to^* (q', \mu' \oplus q'_a, d')$. Ce fait est présenté dans le lemme de copie 7.18 et la figure 7.2 et justifie l'étude des opérations non atomiques.

La principale question adressée dans cette seconde partie est l'accessibilité d'une configuration *recouvrant* un état fixé $q_f \in Q_c$, c'est-à-dire que l'on cherche à savoir si q_f est atteint par *n'importe lequel* des processus contributeurs. L'ensemble de configurations est ainsi $U_f = \{(q, \mu, d) \mid \mu(q_f) > 0\}$. On remarque que l'ajout de processus contributeurs supplémentaires ne nuit pas à l'accessibilité, puisqu'on peut choisir de conserver ces processus supplémentaires inactifs, et toujours atteindre U_f.

Un autre objectif traité dans cette thèse est le *consensus*, où l'on requiert désormais que *tous* les processus atteignent q_f au même instant, c'est-à-dire $U'_f = \{(q, \mu, d) \mid \forall q' \neq q_f \ \mu(q') = 0\}$.

On introduit la relation d'ordre $\preceq$ définie pour tout couple de configuration (q, μ, d) et (q', μ', d') par $(q, \mu, d) \preceq (q', \mu', d')$ si, et seulement si, $q = q'$, $d = d'$ et pour tout q_a, soit $\mu(q_a) = \mu'(q_a) = 0$, soit $0 < \mu(q_a) \leq \mu'(q_a)$. Munis de cette relation d'ordre, les deux objectifs précédents sont exprimables aisément, car *clos par le haut* pour la relation $\preceq$, qui est un *bel ordre*. Cela signifie en particulier que les ensembles U_f considérés sont engendrés par la clôture d'un nombre fini d'éléments minimaux. Dans les deux cas de la couverture et du consensus, ces éléments minimaux sont même de petite taille : si $(q, \mu, d) \in \min U_f$, alors $\forall q' \ \mu(q') \in \{0, 1\}$. Ces objectifs seront nommés dans la suite *simple* et la plupart des résultats s'appliqueront ainsi

De la même manière, nous définissons l'ensemble des configurations initiales par $U_0 = \uparrow (q_l, q_c, d_0)$.

Ordonnanceur

Deux cas de figures d'ordonnanceurs sont considérés :

- Un ordonnanceur non déterministe : existe-t-il un chemin d'une configuration initiale $\gamma_0 \in U_0$ vers une configuration finale $\gamma_f \in U_f$.
- Un ordonnanceur probabiliste : étant donné une configuration initiale $\gamma_0 \in U_0$ et en considérant des transitions choisies de manière probabiliste, quelle est la probabilité d'atteindre U_f.

Nous retenons ici la seconde approche, d'un point de vue *qualitatif*, en évaluant uniquement si la probabilité d'accessibilité est nulle, entre 0 et 1, ou égale à 1. En effet, dans ce cadre plus

restreint, la question de l'accessibilité probabiliste ne dépend pas des valeurs de probabilités exactes choisies par l'ordonnanceur.

Analyse non déterministe

Le chapitre 8 s'intéresse à l'accessibilité avec probabilité strictement positive. On remarque premièrement qu'une telle propriété équivaut à l'existence d'un chemin avec ordonnanceur non déterministe. Nous revisitons ainsi des résultats présentés dans [EGM13].

En particulier, nous montrons que tout chemin entre U_0 et U_f peut être compressé en un nombre polynomial de transitions et impliquant donc un faible nombre de processus. Ceci est réalisé à l'aide d'un graphe *symbolique*, ne gardant pas en mémoire le nombre exact de processus contributeurs, mais uniquement l'ensemble des états activés par au moins un contributeur. Ce graphe a ainsi de bonnes propriétés de compatibilité avec la relation $\preceq$.

D'autre part, nous rappelons une précédente preuve de réduction du problème 3SAT vers l'accessibilité dans un réseau, ce qui démontre que le problème d'accessibilité avec probabilité strictement positive est NP-complet.

Analyse probabiliste

Le problème de l'accessibilité presque sûre, c'est-à-dire avec probabilité 1, est traité dans le chapitre 9. Contrairement au cas précédent, l'ajout de nouveaux processus peut débloquer de nouveaux comportements qui a leur tour, pourraient empêcher l'accessibilité avec probabilité 1.

Valeur seuil

Cette nouvelle analyse repose donc sur un nouveau concept de *valeur seuil*, à partir de laquelle le paramètre fixant le nombre de contributeurs impose *toujours* une accessibilité presque sûre ou à l'inverse, impose *toujours* une accessibilité avec probabilité strictement inférieure à 1.

Formellement, en notant $\mathbb{P}_n(\Diamond U_f)$ la probabilité d'atteindre U_f à partir d'une configuration initiale à n contributeurs, on montre que la suite $\lceil \mathbb{P}_n(\Diamond U_f) \rceil$ à valeur dans $\{0, 1\}$ admet une limite, c'est-à-dire qu'il existe un entier N tel que :

- ou bien pour tout $n \geq N$, $\mathbb{P}_n(\Diamond U_f) = 1$: N est alors une valeur seuil *positive* ;
- ou bien pour tout $n \geq N$, $\mathbb{P}_n(\Diamond U_f) < 1$: N est alors une valeur seuil *négative* ;

La preuve d'existence s'appuie sur l'analyse d'inclusion d'ensemble d'accessibilité et de co-accessibilité, qui est facilité par la structure de bel ordre de $\preceq$.

Bornes de complexité

Une fois l'existence d'une valeur seuil établie, il reste à déterminer s'il s'agit d'un seuil positif ou négatif, répondant ainsi à une analyse globale du problème. En effet, une valeur seuil fournit une indication sur le comportement limite d'un protocole. Afin de répondre à cette question, il suffit d'analyser notre système pour un paramètre étant une valeur seuil, que l'on veut donc la plus petite possible. Nous nous intéressons ainsi à des exemples de protocole dont la valeur seuil minimale est importante, ce qui fournit des bornes inférieures

- Nous démontrons premièrement qu'il existe des protocoles dont la valeur seuil minimale est *linéaire* en la taille du protocole.
- À l'aide de ce premier résultat, il est possible de générer un nouveau protocole dont la valeur seuil est négative, et exponentielle en la taille du protocole.
- Cet exemple est ensuite généralisé, pour réaliser l'encodage d'une machine de Turing linéairement bornée. Un seuil négatif dans le protocole obtenu devient alors équivalent à l'arrêt de la machine de Turing. Le problème de décision d'un seuil négatif, puis a fortiori, d'un seuil positif, est donc PSPACE-difficile.

De manière orthogonale, nous développons une technique générale pour borner la taille d'une valeur seuil par une quantité *doublement exponentielle* en la taille du protocole. Ce résultat généralise ainsi les bornes doublement exponentielles de Rackoff [Rac78] pour la couverture dans les réseaux de Petri. Ceci fournit d'autre part un algorithme de décision en espace (simplement) exponentiel, grâce au théorème de Savitch [Sip97].

Extensions

Des extensions sont présentées sont dans le chapitre 10 :

- Le *model checking* presque sûr d'un protocole, par rapport à une formule LTL ;
- La gestion d'un nombre fini de registres partagés, en introduisant des opérations de copie d'un registre à un autre ;
- La gestion d'identifiant de processus (pid), tant que la monotonicité est préservée.

Nous indiquons comment les techniques précédentes peuvent être adaptées, et dans les deux premiers cas, nous fournissons des bornes doublement exponentielles ainsi qu'un algorithme de décision en espace exponentiel.

Stratégies locales

Le dernier chapitre 11 introduit alors le concept de stratégies issues de la première partie de la thèse, et montre que certains problèmes de synthèse sont décidables, sous certaines hypothèses :

- Nous considérons uniquement l'accessibilité avec probabilité strictement positive, ou strictement inférieure à 1,
- pour des objectifs de couverture,
- avec des stratégies locales, c'est-à-dire n'ayant pour observation que la séquence d'état localement visités.

Dans les deux cas étudiés, accessibilité avec probabilité strictement supérieure à 0 ou strictement inférieure à 1, la mémoire requise par une stratégie locale peut être limitée à une quantité polynomiale, ce qui signifie encore une fois que les problèmes correspondant sont NP-complets.

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Stratégies randomisées dans les jeux concurrents

Mots clefs : Jeux, stratégies, stochastiques, Nash, équilibres, concurrent

Résumé : Ce travail se concentre sur l'étude de jeux joués sur des graphes finis, par un nombre arbitraire de joueurs, dont les objectifs ne sont pas antagonistes. Chaque joueur représente un agent, c'est-à-dire un programme, un processus, ou un périphérique, qui interagit avec les autres joueurs et leur environnement commun dans le but de satisfaire au mieux son objectif individuel. Des concepts tels que les équilibres de Nash, permettant d'exprimer l'optimalité des stratégies des joueurs, ont été étudiés dans un cadre déterministe, et l'existence de tels équilibres n'est pas assurée, même lorsque les objectifs des joueurs sont de simples conditions d'accessibilité ou de sûreté. En effet, lorsque les joueurs jouent de manière déterministe, le système évolue en conservant une certaine symétrie, ce qui nous motive à considérer un modèle stochastique où les joueurs et l'environnement sont sources d'aléa. Dans le premier cas, nous montrons que les concepts classiques d'équilibres de Nash ne peuvent être calculés, et introduisons des notions approchées d'équilibres calculables. Dans le deuxième cas, nous nous intéressons à l'analyse de systèmes composés d'un nombre arbitraires de processus, dont l'exécution est déterminée par un ordonnanceur, c'est-à-dire l'environnement, probabiliste.

Title : Randomized Strategies in Concurrent Games

Keywords : Games, Strategy, Stochastic, Nash, Equilibrium, Concurrent

Abstract : We study games played on graphs by an arbitrary number of players with non-zero sum objectives. The players represent agents (programs, processes or devices) that can interact to achieve their own objectives as much as possible. Solution concepts, as Nash Equilibrium, for such optimal plays, need not exist when restricting to pure deterministic strategies, even with simple reachability or safety objectives. As a matter of fact, symmetry is preserved when players behave deterministically, which motivates the studies where either the players or the environment can use randomization. In the first case, we show that classical concepts like Nash Equilibria, cannot be computed even with a fixed number of agents and propose computable approximations. In the second case, we study systems composed of several copies of the same process, communiting through a shared register, and whose executions are determined by a stochastic scheduler.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France