ENSC-201X-N^{o}YYY

THÈSE DE DOCTORAT
DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Présentée par

Monsieur Romain Brenguier

Pour obtenir le grade de

DOCTEUR DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Domaine :
Informatique

Titre :

Équilibres de Nash dans les Jeux Concurrents – Application aux Jeux Temporisés

Nash Equilibria in Concurrent Games – Application to Timed Games

Thèse présentée et soutenue à Cachan le 29 novembre 2012 devant le jury composé de :

Christof Löding	Rapporteur
Anca Muscholl	Rapporteur
Hugo Gimbert	Rapporteur
Kim G. Larsen	Examinateur
Jean-François Raskin	Examinateur
Patricia Bouyer-Decitre	Directrice de thèse
Nicolas Markey	Directeur de thèse

Laboratoire Spécification et Vérification
ENS Cachan, CNRS, UMR 8643
61, avenue du Président Wilson
94235 CACHAN Cedex, France

Contents

Acknowledgments

I met Patricia and Nicolas while I was in the Master Parisien de Recherche en Informatique (MPRI). They were giving the course on real-time systems. I was fascinated both by the subject and by the clear and rigorous way in which it was presented. I thank them for giving me the opportunity to work on this subject for my thesis and for all the things they taught me: write articles that are pleasant to read, make proofs that others can read, design nice presentations, review papers, and many other things.

My first encounter with game theory was in the MPRI course given by Olivier Serre and Wiesław Zielonka. I am thankful to them for introducing me to this field, it has since become a central part in my work.

I thank Christof Löding, Anca Muscholl, and Hugo Gimbert for reviewing this thesis, and for their comments. I also thank Kim G. Larsen and Jean-François Raskin who accepted to be part of the jury of my defense.

A fundamental base for this thesis has been the work of Michael Ummels. I had the chance of working him, he has pointed to me some really good ideas, and I thank him for this fruitful collaboration.

During this thesis, Ocan and I shared many interesting discussions and I often asked for his advices. I am grateful for all the suggestions he made, it has been of a great use while writing this thesis.

The administrative team of the LSV have always been helpful during these three years, I would like to thank particularly Virginie and Catherine.

My office is in a remote part of the LSV called the Iris building. It was never too lonely as I shared it with several other students along the years. For the friendly atmosphere which has always been there, I give my thanks to all them: Jules, Arnaud, Joe, Michael, Robert, Ocan, Steen, Rémy.

The mutual support is very strong among students of the LSV, and we shared great moments of conviviality. I cannot cite all of them but I will mention in particular Benjamin and Aiswarya for their great job organizing the PhD seminar, also known as the "goûter des doctorants".

During these two last years, I shared my flat with three friends: Antoine, Gaël and Robin. I am grateful to them for making it a place to relax and think about things outside of game theory and model checking. They reminded me that games are more than just theory: there is also Mario Kart, Scrabble and Poker.

My family, and my friends since high-school have followed my progress all along and they have always been a support when I come back to my home region of Dauphiné. They deserve all my gratitude.

Even when far from Cachan, and for every day of these three years, Öykü has been there for me. I thank her from all my heart.

Chapter 1

Introduction

1.1 Model Checking and Controller Synthesis

Verification Information and communication technology are part of our daily life. We carry hardware systems everywhere, in the form of mobile phones, laptop computers or personal navigation devices. We use softwares to read books, listen to music and chat with friends. Errors in the conception of these systems occur routinely. This is a particularly serious issue in the case of embedded systems, which are designed for specific tasks and are part of complex and critical systems, such as cars, trains, rockets. Flaws can injure people and cost a lot of money. Planes, nuclear power plants, life support equipments are examples of safety critical systems. For them, mistakes can cause human, environmental or economical disaster. Some design errors have become famous by their dramatic consequences, such as the bug in the Ariane-5 rocket in 1996. More recently, on the sixth of July 2012, because of a software bug, the Orange France mobile network remained out of order for twelve hours. It is necessary to ensure that the design of a safety critical system is correct, in that it possesses the desired properties. The approach of formal methods is to apply the formalism of mathematics to model and analyze them, in order to establish their correctness.

The aim of formal verification is to eliminate design errors. In software engineering, the most used verification technique is testing. It consists in running the software on sample scenarios, and in checking that the output is what is expected. In hardware verification, one popular method is simulation. In simulation, in order to save time and money, tests are performed on a model of the hardware, given in a hardware description language such as Verilog or VHDL. As for testing, simulation can detect errors but can not prove their absence.

Model Checking As an alternative, model checking proceeds by an exhaustive exploration of the possible state space of the system. It can reveal subtle errors that might be undiscovered by testing or simulation. It takes as input a model of the system, and a property to verify. The *model* of the system describes how the system works. In general, the system is modeled using finite-state automata or an extension of this model. For example, for software, the automaton represents the graph of the possible configurations of the program. Finite-state automata can be generated from languages similar to `C` for example. The *property* given to a model checker states what the system should do and what it should not. It is in general obtained from an informal specification written in a natural language, which should be formally translated in a specification language or logic. For instance, temporal logics are standard formalisms for this task. The aim of research in model checking is to design *algorithms* to check that the model of the system conforms to the formal specification. In the case the specification is not met, tools can provide counter examples, which helps in correcting the design, the model or the specification. To be usable, the method needs to be powerful, easy to use and efficient. The efficiency of the algorithm is generally in opposition with the two other points. A more powerful model requires more computational time to verify. Of course, as model checking is a model-based method, confidence in the analysis depends on the accuracy of the models.

Improvement in algorithms, data structures and computational power of modern computers, have made verification techniques quickly applicable to realistic designs, starting with the work of Burch, Clarke, McMillan and Dill for circuits [11] and Holzmann for protocols [31]. Efficient tools are available, for instance NuSMV [15] which is well suited for hardware verification and SPIN [32] that can verify communicating asynchronous processes.

Real-Time Systems Embedded systems are often subject to real-time constraints. They are time critical: correctness depends not only on the functional result but also on the time at which it is produced. For instance, if the undercarriage of a plane is lowered too late, it can have the same catastrophic consequences than if it were not opened at all. To express these constraints, time has to be integrated to the model. For these tasks the model of timed automata has been developed. It corresponds to the program graph equipped with real valued clocks, that can only be tested and reset to 0. Model checking has been shown decidable for this model [1]. Efficient tools such as Uppaal [5], Kronos [55], and HyTech [29] have been implemented, making use of clever data structures.

Controller Synthesis Reactive systems like device drivers and communication protocols, have to respond to external events. They are influenced by their environment, which is unpredictable or would be difficult to model. It is preferable not to specify precisely the environment but to give it some freedom. Such systems are managed using controllers, which monitor and regulate the activity of the system. The problem of controller synthesis was formulated by Ramadge and Wonham [45]. This problem is related to program synthesis, in which a program which satisfies the given specification should be automatically generated. Instead of looking for bugs by model checking, we want to synthesize a model of a controller with no flaw. Here the generated controller has to ensure that the system under control satisfies its specification whatever happens in the environment. It is very convenient to see the problem as a two-player game, where a player plays as a controller and the other one plays the environment. The system is controllable when the first player has a strategy that is winning for the condition given by the specification. Several algorithms have been developed, for discrete systems [54] and timed systems [22, 9]. Algorithms for timed systems have been implemented in the tool UPPAAL TIGA [4].

In the case of multiple systems, controlled by rational entities and interacting with each other, the approach of controller synthesis is no longer satisfactory. Each system has its own requirements and objectives, and considering worst-case behaviors of the environment is not satisfactory. Determining the good solutions in this context, is a classical problem in game theory. We will thus take inspiration from the solution concepts that have been proposed in this field. Let us first look at a brief history of game theory.

1.2 Games and Equilibria

Cournot Competition Game theory aims at understanding the decisions made by interacting *agents*. The study of game theory can be traced back to the work of Cournot, on *duopolies* in a book first published in 1838 [17]. This mathematician was the first to apply mathematics to economic analysis. He was studying the competition between two companies selling spring water. These companies had to decide on the quantity of bottles to produce. Their intention was to maximize their own profit. The solution Cournot proposed was that each company uses a *strategy* that is a best response to the strategy played by the opponent. This defines an equilibrium behavior for the whole system which actually coincides with what would later be known as *Nash equilibria*.

Normal-Form Games The real development of the field of game theory started with the work of Von Neumann and Morgenstern and their 1944 book [42]. In a game, players have to choose among a number of possible strategies, a combination of a strategy for each player gives an outcome, each player has her own preference concerning the possible outcomes. For Von Neumann and Morgenstern, the preferences are described in a matrix which for each combination of strategies, gives the integer *payoff* of all players. This representation of the game is said to be in *normal-form*. Consider for example the *payoff matrix* of the rock-paper-scissors game represented in Fig. 1.1. The actions of the first player are identified with the rows and the second player's with the columns. The first component in row r and column c corresponds to the payoff of the first player if she plays r and her opponent c, and the second component to the payoff of the second player. In rock-paper-scissors the payoffs of the players always sum up to 0, this is an instance of a two-player *zero-sum game*, which was the object of the work of Von Neumann and Morgenstern. Such a game is purely antagonistic, since players' *preferences* are opposed. A player tries to have the highest payoff, considering that the opponent is going to play the best *counter-strategy*. The strategy that ensures the best outcome in the worst case is called the *optimal strategy*. In zero-sum games, Von Neumann showed the existence of a pair of strategies, that is optimal, in the sense that each player minimizes her maximum loss [53]. In general, this requires *mixed strategies*, which allow randomization between several different actions. Therefore, the number of possible strategies is in fact infinite. For example in the rock-paper-scissors game, there are three pure strategies available, but the equilibrium is obtained by randomizing uniformly between these pure strategies.

Table 1.1: The game of rock-paper-scissors in normal-form.

	Rock	Paper	Scissors
Rock	0 , 0	-1 , 1	1 , -1
Paper	1 , -1	0 , 0	-1 , 1
Scissors	-1 , 1	1 , -1	0 , 0

Nash equilibrium When games are not zero-sum, in particular when there are more than two players, winning strategies are no longer suitable to describe rational behaviors. In particular when the objectives of the players are not opposite, cooperation should be possible. Then, instead of considering that the opponent can play any strategy, we will assume that they are, also, rational. The notion of equilibria aims at describing rational behaviors. If we are expecting some strategy from the adversaries then it is rational to play the best response, that is the strategy that maximizes the payoff if the strategies of the opponents are fixed. The solution for non zero-sum games is a strategy for each player, such that knowing what the others are going to play, none of them is interested in changing her own. In other terms, each strategy is a best response to the other strategies.

For example consider the Hawk-Dove game, first presented by the biologists Smith and Price [48]. One such game is given in matrix form in Table 1.2. Two animals are fighting over some prey and can choose to either act as a hawk or as a dove. If a player chooses hawk then for the opponent the best payoff is obtained by playing dove. We say that dove is the *best response* to hawk. Reciprocally the best response to dove is to play hawk. There are two "stable" situations $(\mathrm{Hawk}, \mathrm{Dove})$ and $(\mathrm{Dove}, \mathrm{Hawk})$, in the sense that no player has an interest in changing her strategy.

Nash showed the existence of such equilibria in any normal-form game [44], which again requires mixed strategies. This result has revolutionized the field of economics, where it is used to analyze competitions between firms or government economic policies for example. Game theory and the concept of Nash equilibrium are now applied to very diverse fields: in finance to analyze the evolution of market prices, in biology to understand the evolution of some species, in political sciences to explain public choices made by parties.

Table 1.2: The Hawk-Dove game.

	Hawk	Dove
Hawk	0 , 0	1 , 4
Dove	4 , 1	3 , 3

Games for Synthesis We aim at using the theory of non-zero-sum games for synthesizing complex systems in which several agents interact. Think for instance of several users behind their computers on a shared network. When designing a protocol, maximizing the overall performance of the system is desirable, but if a *deviation* can be profitable to the users, it should be expected that one of them takes advantage of this weakness. This happened for example to the bit-torrent protocol where selfish clients became more popular. Such deviations can harm the global performance of the protocol. The concept of Nash equilibrium is particularly relevant, for one's implementation to be used.

Unlike the ones we presented so far, in the context of controller synthesis, games are generally not presented in normal-form. Instead of being represented explicitly, it is more convenient to use games played on graphs. The graphs represent the possible configurations of the system. The controller can take actions to guide the system, and the system can also be influenced by the environment. Among games played on graph we can distinguish several classes. The simplest one are *turn-based games*. For these games, in each state, one player decides alone on the next state. In *concurrent games*, in each state, the players choose their actions independently and the joint move formed by these choices determines the next state. *Timed games* are examples of concurrent games, which are played on timed automata. In a given state, several players can have actions to play, but only the player that moves first has influence on the next state. Concurrent games and timed games are a valuable framework for the synthesis problem, we therefore chose to study the computation of Nash equilibria in this kind of games.

1.3 Examples

The examples we present now are going to be reused later, to illustrate the different concepts that will be introduced. We describe the general ideas of the problems.

1.3.1 Peer to Peer Networks

In a peer-to-peer network clients share files that might interest other clients. Clients want to download files from other clients but sending the files uses bandwidth and prefer to limit this. The interaction in this situation could be modeled by a normal-form game, as is represented in Table 1.3 for two players. To play this game, players have to choose independently if they are going to send or receive a file. An agent can only receive a file if another client is sending it. In this matrix game, the best response is never to send a file. However, this model is not accurate since in reality the situation is repeated. Considering that the game is repeated an infinite number of steps, players can take into account the previous actions of others to adapt their strategies. For instance, they can choose to cooperate and send the file in alternation. The *tit-for-tat* strategy, suggests that if one defects to send at his turn, the other stops as well. The best response is then to cooperate, since it gives a better accumulated payoff for both players.

Table 1.3: A game of sharing on a peer to peer network.

	Receive	Send
Receive	0 , 0	2, -1
Send	-1, 2	0, 0

1.3.2 Medium Access Control

This example was first formalized from the point of view of game theory in [41] Several users share access to a wireless channel. During each slot, they can choose to either transmit or wait for the next slot. The probability that a user is successful in its transmission decreases with the number of users emitting in the same slot. Furthermore each attempt at transmitting has its cost. The payoff thus increases with the number of successful transmissions but decreases with the number of attempts. The expected reward for one slot and two players, is represented in Table 1.4, assuming a cost of 2 for each transmission, a reward of 4 for a successful transmission, a probability 1 to be successful if only one player emit, and of $\frac{1}{4}$ if they both transmit at the same time.

Table 1.4: A game of medium access.

	Emit	Wait
Emit	-1,-1	2, 0
Wait	0 , 2	0, 0

Once again, for a real situation, one step is not enough and there would be a succession of slots and the payoff is then the accumulation of the payoff for each slot. A better model will be presented in Section 2.1, Example 2.

Table 1.5: Power control as a normal-form game

	$p_2 = 0$	$p_2 = 1$	$p_2 = 2$
$p_1 = 0$	0 , 0	0 , 0.6	0 , 0.4
$p_1 = 1$	0.6 , 0	0.1 , 0.1	0.03 , 0.13
$p_1 = 2$	0.4 , 0	0.13 , 0.03	0.05 , 0.05

1.3.3 Power Control in Cellular Networks

This game is inspired by the problem of power control in cellular networks. Game theoretical concepts are relevant for this problem and Nash equilibria are actually used to describe rational behaviors of the agents [39, 40].

Consider the situation where a number of cellular telephones are emitting over a cellular network. Each agent A_i, can choose the emitting power p_i of his phone. From the point of view of agent A_i, using a stronger power results in a better transmission, but it is costly since it uses energy, and it lowers the quality of the transmission for the others, because of interferences. The payoff for player i can be modeled by this expression from [47]:

$$\frac{R}{p_i}\left(1 - e^{-0.5\gamma_i}\right)^L \tag{1.1}$$

where γ_i is the *signal-to-interference-and-noise ratio* for player A_i, R is the rate at which the wireless system transmits the information in bits per seconds and L is the size of the packets in bits.

The interaction in this situation could be modeled by a normal-form game, as is represented in Table 1.5 for two players, three possible levels of emission and some arbitrary parameters. To play this game, players have to choose independently what power they will use, and the corresponding cell in the table gives the payoff for each of them. What would seem the best choice to maximize the total payoff of the player would be $p_1 = p_2 = 1$. However, knowing that player A_1 is going to choose p_1, player A_2's *best response* to maximize its own payoff is to choose $p_2 = 2$. A better model for this problem, can be given as a repeated game, where at each step, each agent A_i can choose to increase or not its emitting power p_i. We will develop on this model in Section 2.1, Example 1.

1.3.4 Shared File System

We now take the example of a network file system. The problem occurs when several users have to share a resource. Several users on client computers can access the same files over a network, on a file server. The protocol for this file system can integrate file locking, like for instance NFS version 4. This prevents two clients accessing the same file at the same time. Such a protocol is said to be stateful, this is illustrated in Figure 1.1 for one lock and two clients. From the initial configuration, one of the client can lock a file, and until it is not unlocked the file can not be accessed by others.

To perform some task, the clients access files on the system, and try to minimize the time until their task is completed. By maintaining a lock on a file they need later, they might lower the time necessary for their task, but this can have the opposite effect for other clients. We have to organize the accesses, so that clients are not tempted to act in an unpredicted way.

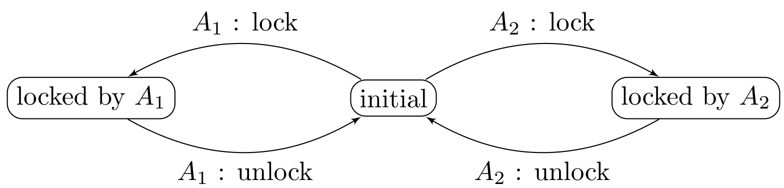


Figure 1.1: A simple model of a shared file with one lock.

We will come back on this example in Section 2.1, Example 3.

1.4 Contribution

In this thesis we are interested in the existence and computation of Nash equilibria in games played on graphs. As several Nash equilibria may coexist, it is relevant to look for particular ones. It is interesting to look for Nash equilibria in pure strategies, since they can be implemented using deterministic programs. It is also important to put constraints on the outcome of the equilibrium or on its payoff. We can for instance ask for an equilibrium where all players get their best payoff. Another constraint we allow, is on the actions used in the equilibria. We will see that the complexity of these different decision problems are closely related and lie most of the time in the same complexity classes.

Our approach consists in defining a transformation from multiplayer games to two-player zero-sum turn-based games. The search of Nash equilibria can now be seen as a antagonistic game in itself. Two-player zero-sum turn-based games have been much studied in computer science and efficient algorithms have been developed for synthesizing strategies, so that we can make use of them. With this transformation in mind, we study the complexity of finding Nash equilibria in finite concurrent games and treat classical objectives such as reachability, safety and other regular objectives. We describe the precise complexity class in which the problem lies for most of them. For instance, for Büchi objectives we show a polynomial algorithm to find equilibria, whereas for reachability objectives we show that the decision problems are NP-hard.

These classical objectives are only qualitative, since players can either win or lose. To be more quantitative, we combine several objectives. We propose several ways to order the possible outcomes according to these multiple objectives. In general we use preorders, allowing to model some uncertainty about the preferences of the players. To describe the preorders we use Boolean circuit. We give an algorithm for the general case and analyze the complexity of some preorders of interest.

Finally, to allow for a more faithful representation of embedded systems we aim at analyzing timed games. We propose a transformation, based on regions, from timed to finite concurrent games, preserving Nash equilibria under some restriction on the allowed actions. We apply the techniques we developed so far to the obtained game. Our most general decision problem is EXPTIME-complete.

1.5 Related Works

Algorithmic game theory has dealt early with the problem of finding Nash equilibria in normal-form games. As they always exist if mixed strategies are allowed, this cannot be formulated as a decision problem, instead it is also possible to look at the problem as a *search problem*, Daskalakis, Goldberg and Papadimitriou showed that the total search problem of Nash equilibrium is PPAD-complete [19]. One interesting question is whether there exists an equilibrium with a particular payoff, Gilboa and Zemel showed in 1989 that this question is NP-hard for normal-form games [25]. Another interesting problem is whether there exists a pure Nash equilibrium, Gottlob, Greco and Scarcello showed that this is NP-hard [27]. However these results consider a particular representation of the game in normal-form that is more succinct than the standard one. By contrast, in this work, we assume that the transition function of the game is given explicitly. For normal-form games, this means that the game is given as a matrix, in that case the existence of a pure Nash equilibrium is polynomial. Moreover, unlike normal-form games which are played in one shot, the games we consider are repeated, and are played on graphs.

A *repeated game* is basically a normal-form game, that is repeated for an infinite number of steps, each player receiving an instant reward at each step. An important result in the theory of repeated games, known as a folk theorem, is that any possible outcome can be an equilibrium given that all the players receive a payoff above the minimal one they can ensure. This does not apply to the context we study since our games have internal states and players do not get instant reward. Instead their preferences depend on the sequence that is obtained in the underlying graph.

In games played on graphs, the number of pure strategies is infinite, and Nash theorem no longer applies. The study of equilibria in graph games started by proving that in any turn-based game with Borel objectives, there exists a pure Nash equilibrium [14, 50]. In *stochastic games*, for some states, the successor is chosen randomly, according to a given probability distribution. In stochastic games where players take their actions simultaneously and independently at each step, Nash equilibria might not always exist. Instead, Chatterjee, Majumdar and Jurdziński showed that there are ε-Nash equilibrium for reachability objectives [14] and any strictly positive ε. For quantitative objectives, in turn-based games, Brihaye, Bruyère and De Pril showed the existence of equilibria, in games where players aim at minimizing the number of steps before reaching their objectives [10].

From the point of view of computer science, the existence of an equilibrium is not enough, we are interested in the complexity of finding a particular one. Ummels studied the complexity of Nash equilibria for classical objectives in turn-based games. He showed that for Streett objectives the existence of Nash equilibria with constraints on the payoff is NP-complete while it is polynomial for Büchi objectives [49]. With Wojtczak, they showed that adding stochastic vertices makes the problem of finding a pure Nash equilibrium undecidable [51]. In a more quantitative setting, they also studied limit-average objectives in concurrent games. For these games, the existence of a mixed Nash equilibrium with a constraint on the payoff is undecidable, while it is NP-complete when looking for pure Nash equilibria [52]. Klimos, Larsen, Stefanak and Thaarup, studied the complexity of Nash equilibria in concurrent games where the objective is to reach a state while minimizing some price. They show that the existence of a Nash equilibrium is NP-complete [36]. It is to note that the model of concurrent games we study is slightly more general in that players do not observe actions. This is sometimes called imperfect monitoring and has been discussed in other works [46, 18].

1.6 Outline

In Chapter 2, we define concurrent games which is the central model of games we are going to study. Together with this, we define the decision problems we are interested in: deciding if a player can ensure a given value, the existence of a Nash equilibrium, and the existence of a Nash equilibrium satisfying some constraints on its outcome and some constraints on the moves it uses. We also prove some basic properties, in particular undecidability in the general case, and possible translations from one decision problem to another.

In Chapter 3, we show how to transform a multiplayer concurrent game into a turn-based two-player game called the suspect game. This fundamental transformation relies on the central notion of *suspect*. We show that it is correct in the sense that finding a Nash equilibrium in the original game can be done by finding a winning strategy with some particular outcome in the *suspect game*. We also define a notion of *game-simulation* based on suspects, which preserves Nash equilibria.

In Chapter 4, we apply this transformation to finite games in order to describe the complexity of the different decisions problem, for classical objectives.

In Chapter 5, we extend this result to a more quantitative setting, where each player has several objectives. We propose several ways to order the possible outcomes from these multiple objectives and analyze the complexity of the different decision problems in each case.

In Chapter 6, we focus on timed games. We transform timed games into finite concurrent games, making use of a refinement of the classical region abstraction.

We show a correspondence between Nash equilibria in the timed game and in the region game, and then analyze the complexity of our decision problems.

Chapter 2

Concurrent Games

In this chapter, we define the model of concurrent games formally, we present the concept of Nash equilibrium and the most relevant decision problems that arise from this notion. Finally, We give some generic properties of these different problems in the context of concurrent games.

2.1 Definitions

A concurrent game corresponds to a transition system, over which the decision to go from one state to another is made jointly by players of the game. It is first necessary to recall the essential vocabulary of transition systems and introduce the notations we will use.

Transition systems. A *transition system* is a couple $\mathcal{S} = \langle \text{States}, \text{Edg} \rangle$ where States is a set of states and $\text{Edg} \subseteq \text{States} \times \text{States}$ is the set of *transitions*. A *path* π in $\mathcal{S}$ is a sequence $(s_i)_{0 \leq i < n}$ (where $n \in \mathbb{N}^+ \cup \{\infty\}$) of states such that $(s_i, s_{i+1}) \in \text{Edg}$ for all $i < n-1$. The *length* of π, denoted by $|\pi|$, is $n-1$. The set of finite paths (also called *histories*) of $\mathcal{S}$ is denoted by $\text{Hist}_{\mathcal{S}}$, the set of infinite paths (also called *plays*) of $\mathcal{S}$ is denoted by $\text{Play}_{\mathcal{S}}$, and $\text{Path}_{\mathcal{S}} = \text{Hist}_{\mathcal{S}} \cup \text{Play}_{\mathcal{S}}$ is the set of all paths of $\mathcal{S}$. Given a path $\pi = (s_i)_{0 \leq i < n}$ and an integer $j < n$: the *j-th prefix* of π, denoted by $\pi_{\leq j}$, is the finite path $(s_i)_{0 \leq i < j+1}$; the *j-th suffix*, denoted by $\pi_{\geq j}$, is the path $(s_{j+i})_{0 \leq i < n-j}$; the *$j$-th state*, denoted by $\pi_{=j}$, is the state s_j. If $\pi = (s_i)_{0 \leq i < n}$ is a history, we write $\text{last}(\pi) = s_{|\pi|}$ for the last state of π. If π' as a path such that $(\text{last}(\pi), \pi'_{=0}) \in \text{Edg}$, then the *concatenation* $\pi \cdot \pi'$ is the path ρ s.t. $\rho_{=i} = \pi_{=i}$ for $i \leq |\pi|$ and $\rho_{=i} = \pi'_{=(i-1-|\pi|)}$ for $i > |\pi|$. In the sequel, we write $\text{Hist}_{\mathcal{S}}(s)$, $\text{Play}_{\mathcal{S}}(s)$ and $\text{Path}_{\mathcal{S}}(s)$ for the respective subsets of paths starting in state s, i.e. the paths π such that $\pi_{=0} = s$. If π is a play, $\text{Occ}(\pi) = \{s \mid \exists j.\ \pi_{=j} = s\}$ is the sets of states that appears at least once along π and $\text{Inf}(\pi) = \{s \mid \forall i.\ \exists j \geq i.\ \pi_{=j} = s\}$ is the set of states that appears infinitely often along π.

Players have preferences over the possible plays, and in a multiplayer games, this preferences are in general different for each player. In order to be as general as possible, we describe these preferences as preorders over the possible plays. We recall here, generic definitions concerning preorders.

Preorders. We fix a non-empty set P. A *preorder* over P is a binary relation $\lesssim \subseteq P \times P$ that is reflexive and transitive. With a preorder $\lesssim$, we associate an *equivalence relation* $\sim$ defined so that $a \sim b$ if, and only if, $a \lesssim b$ and $b \lesssim a$. The *equivalence class* of a, written $[a]_{\lesssim}$, is the set $\{b \in P \mid a \sim b\}$. We also associate with $\lesssim$ a *strict partial order* $\prec$ defined so that $a \prec b$ if, and only if, $a \lesssim b$ and $b \not\lesssim a$. A preorder $\lesssim$ is said *total* if, for all elements $a, b \in P$, either $a \lesssim b$, or $b \lesssim a$. An element a in a subset $P' \subseteq P$ is said *maximal in* P' if there is no $b \in P'$ such that $a \prec b$; it is said *minimal in* P' if there is no $b \in P'$ such that $b \prec a$. A preorder is said *Noetherian* (or *upwards well-founded*) if any subset $P' \subseteq P$ has at least one maximal element. It is said *almost-well-founded* if any lower-bounded subset $P' \subseteq P$ has a minimal element.

We can define formally the games we are going to study.

Concurrent games [3]. A *concurrent game* is a tuple $\mathcal{G} = \langle \mathrm{States}, \mathrm{Agt}, \mathrm{Act}, \mathrm{Mov}, \mathrm{Tab}, (\precsim_A)_{A \in \mathrm{Agt}} \rangle$, where:

- States is a non-empty set of *states*;
- Agt is a finite non-empty set of *players*;
- Act is a non-empty set of *actions*, an element of $\mathrm{Act}^{\mathrm{Agt}}$ is called a *move*;
- $\mathrm{Mov}\colon \mathrm{States} \times \mathrm{Agt} \to 2^{\mathrm{Act}} \setminus \{\varnothing\}$ is a mapping indicating the *available actions* to a given player in a given state, we say that a move $m_{\mathrm{Agt}} = (m_A)_{A \in \mathrm{Agt}}$ is *legal* at s if $m_A \in \mathrm{Mov}(s, A)$ for all $A \in \mathrm{Agt}$;
- $\mathrm{Tab}\colon \mathrm{States} \times \mathrm{Act}^{\mathrm{Agt}} \to \mathrm{States}$ is the *transition table*, it associates, with a given state and a given move of the players, the state resulting from that move;
- for each $A \in \mathrm{Agt}$, $\precsim_A$ is a preorder over States^{ω}, called the *preference relation* of player A, when $\pi \precsim_A \pi'$, we say that π' is *at least as good* as π for A and when it is not the case (i.e., $\pi \not\precsim_A \pi'$), we say that A *prefers* π over π'.

A *finite concurrent game* is a concurrent game whose set of state and actions are finite.

In a concurrent game $\mathcal{G}$, whenever we arrive at a state s, the players simultaneously select an available action, which results in a legal move m_{Agt}; the next state of the game is then $\mathrm{Tab}(s, m_{\mathrm{Agt}})$. The same process repeats *ad infinitum* to form an infinite sequence of states.

In the sequel, as no ambiguity will arise, we may abusively write $\mathcal{G}$ for its underlying transition system $\langle \mathrm{States}, \mathrm{Edg} \rangle$ where $\mathrm{Edg} = \{(s, s') \in \mathrm{States} \times$

States $\mid \exists m_{\mathrm{Agt}} \in \prod_{A\in\mathrm{Agt}} \mathrm{Mov}(A).\ \mathrm{Tab}(s, m_{\mathrm{Agt}}) = s'\}$. The notions of paths and related concepts in concurrent games follow from this identification.

Example 1. As a first example, we aim at modeling the power control problem described in Section 1.3.3. The arena on which the game is played is represented in Fig. 2.1 for a simple instance with two players: $\mathrm{Agt} = \{A_1, A_2\}$, and three possible levels of emission. The set of states is $\mathrm{States} = [\![0, 2]\!] \times [\![0, 2]\!]$, it corresponds to the current level of emission of the respective players: state $(1, 0)$ corresponds to player A_1 using 1 unit of power and player A_2 not emitting. The two actions are to either increase one's emitting power or to stick with the current one: $\mathrm{Act} = \{\mathsf{+}, \mathsf{=}\}$. Transitions are labeled with the moves that trigger them, for instance there is a transition from $(0, 0)$ to $(1, 0)$ labeled by $\langle \mathsf{+}, \mathsf{=} \rangle$, meaning that if player A_1 choose action $\mathsf{+}$ and player A_2 action $\mathsf{=}$, the game goes from $(0, 0)$ to $(1, 0)$: formally $\mathrm{Tab}((0, 0), (\mathsf{+}, \mathsf{=})) = (1, 0)$. Allowed actions from a state are deduced from the label on outgoing edges, for instance in $(2, 2)$ the only outgoing edge is labeled by $\langle \mathsf{=}, \mathsf{=} \rangle$, so $\mathrm{Mov}((2, 2), A_1) = \mathrm{Mov}((2, 2), A_2) = \{\mathsf{=}\}$. We do not define a preference relation yet, but in the following we will see many practical ways to define these.

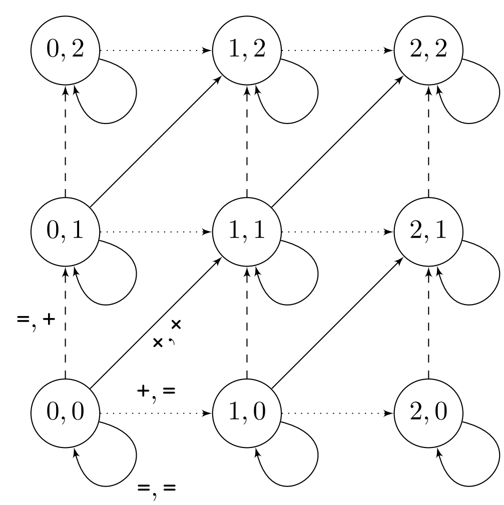


Figure 2.1: A simple game-model for the power control in a cellular network

Example 2. As a second example, we model the problem of medium access control described in Section 1.3.2. The arena on which the game is played is represented in Fig. 2.2 for a simplified instance with two players: $\mathrm{Agt} = \{A_1, A_2\}$. The set of states $\mathrm{States} = [\![0, 2]\!] \times [\![0, 2]\!]$ corresponds to the number of successful frames that players have transmitted so far, and assuming they have a total of 2 frames to transmit. The two actions are to either transmit or wait:

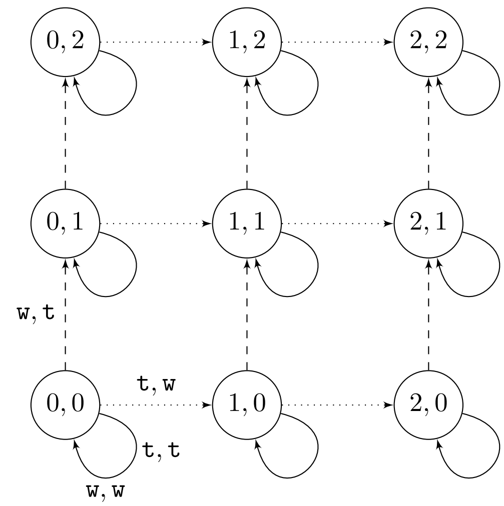


Figure 2.2: A simple game-model for the medium access control

Act = {t, w}. If the two players attempt to transmit at the same time then no frame is received.

Example 3. As a third example, we model the problem of shared file system described in Section 1.3.4. The arena on which the game is played is represented in Fig. 2.3 for an instance with two players: Agt $= \{A_1, A_2\}$ and two files F_1 and F_2 that are shared.

Strategies. Let $\mathcal{G}$ be a concurrent game, and $A \in \mathrm{Agt}$. A *strategy* for A maps histories to available actions, formally it is a function $\sigma_A \colon \mathrm{Hist}_{\mathcal{G}} \to \mathrm{Act}$ such that $\sigma_A(\pi) \in \mathrm{Mov}(\mathrm{last}(\pi), A)$ for all $\pi \in \mathrm{Hist}_{\mathcal{G}}$. A strategy σ_P for a coalition $P \subseteq \mathrm{Agt}$ is a tuple of strategies, one for each player in P. We write $\sigma_P = (\sigma_A)_{A\in P}$ for such a strategy. A *strategy profile* is a strategy for Agt. We write $\mathrm{Strat}^P_{\mathcal{G}}$ for the set of strategies of coalition P, and $\mathrm{Prof}_{\mathcal{G}} = \mathrm{Strat}^{\mathrm{Agt}}_{\mathcal{G}}$.

Outcomes. Let $\mathcal{G}$ be a game, P a coalition, and σ_P a strategy for P. A path π is *compatible* with the strategy σ_P if, for all $k < |\pi|$, there exists a move m_{Agt} such that

1. m_{Agt} is legal at $\pi_{=k}$,
2. $m_A = \sigma_A(\pi_{\leq k})$ for all $A \in P$, and
3. $\mathrm{Tab}(\pi_{=k}, m_{\mathrm{Agt}}) = \pi_{=k+1}$.

We write $\mathrm{Out}_{\mathcal{G}}(\sigma_P)$ for the set of paths in $\mathcal{G}$ that are compatible with strategy σ_P of P, these paths are called *outcomes* of σ_P. We write $\mathrm{Out}^{\mathrm{f}}_{\mathcal{G}}(\sigma_P)$ (resp.

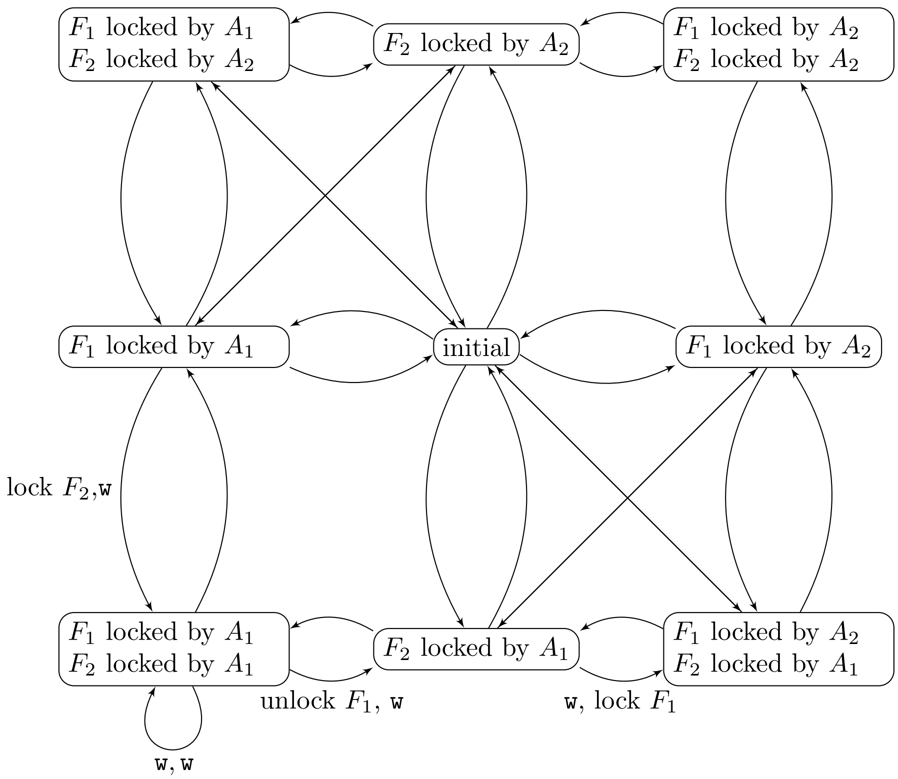


Figure 2.3: A game-model for a network file system.

$\mathrm{Out}^{\infty}_{\mathcal{G}}(\sigma_P)$) for the finite (resp. infinite) outcomes, and $\mathrm{Out}_{\mathcal{G}}(s, \sigma_P)$, $\mathrm{Out}^{\mathrm{f}}_{\mathcal{G}}(s, \sigma_P)$ and $\mathrm{Out}^{\infty}_{\mathcal{G}}(s, \sigma_P)$ for the respective sets of outcomes of σ_P with initial state s. Notice that any strategy profile has a single infinite outcome from a given state, thus when given a strategy profile σ_{Agt}, we identify $\mathrm{Out}_{\mathcal{G}}(s, \sigma_{\mathrm{Agt}})$ with the unique play it contains. When $\mathcal{G}$ is clear from the context we simply write $\mathrm{Out}(s, \sigma_{\mathrm{Agt}})$ for $\mathrm{Out}_{\mathcal{G}}(s, \sigma_{\mathrm{Agt}})$.

Note that, unless explicitly mentioned, we only consider *pure* (i.e., non-randomized) strategies. Notice also that strategies are based on the sequences of visited states, and not on the sequences of actions played by the players. This is realistic when considering multi-agent systems, where only the global effect of the actions of the agents is assumed to be observable. Being able to see (or deduce) the actions of the players, would make the various computations easier. When we can infer from the transition that was taken, the actions of all players, we call the game *action-visible*, formally this is when $\mathrm{Tab}(s, m_{\mathrm{Agt}}) = \mathrm{Tab}(s, m'_{\mathrm{Agt}})$ if, and only if, $m_{\mathrm{Agt}} = m'_{\mathrm{Agt}}$. A particular case of an action-visible game is the classical notion of *turn-based game*, a game is said to be *turn-based* if for each state s the set of allowed moves is a singleton for all but at most one player A, this player A is said to *control* that state, and she chooses a possible successor for the current state, i.e. $\text{Act} = \text{States}$ and $\text{Tab}(s, m_{\text{Agt}}) = s'$ if, and only if, $m_A = s'$.

Example 4. For example, the game we used in Example 2 to model medium access control, represented in Fig. 2.2, is not action-visible: if the game stays in the same state, it is not possible to know if it is because both players have waited or if it is because there was a collision when they tried to transmit. On the other hand, the game of Example 2.1 modeling power control, is action-visible since two different legal moves will always lead to two different states. We give an example of a turn-based game in Fig. 2.4.

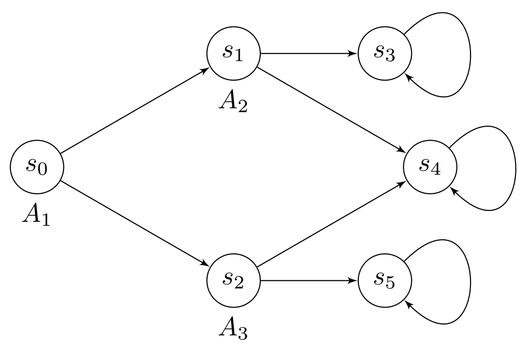


Figure 2.4: A simple turn-based game. It is convenient to use a different representation for this class of game: instead of labeling the edges with actions, we denote below each state, which player is controlling that state, so in that case player A_1 controls state s_0, player A_2 controls state s_1 and player A_3 controls state s_2.

2.2 Value and Nash Equilibria

A concurrent game involving only two players (A and B, say) is *zero-sum* if, for any two plays π and π', it holds $\pi \precsim_A \pi'$ if, and only if, $\pi' \precsim_B \pi$. Such a setting is purely antagonistic, as both players have opposite preference relations. The most relevant concept in such a setting is that of *optimal strategies* where one player has to consider the strategy of the opponent that is the worst for her, while trying to ensure the maximum possible with respect to her preference. The minimum outcome she can get if she plays optimally is called her *value*. This is a central notion in two-player games since the minimax theorem of Von Neumann [53], but it is also applicable in the context of multi-player games.

Value. Give a game $\mathcal{G}$, we say that a strategy σ_A for A *ensures* π from s if every outcome of σ_A from s is at least as good as π for A, i.e. $\forall \pi' \in \mathrm{Out}^{\infty}_{\mathcal{G}}(s, \sigma_A).\ \pi \precsim_A \pi'$. We also say that A *can ensure* π when such a strategy σ_A exists. A *value* of game $\mathcal{G}$ for player A from a state s is a maximal element of the set of paths that player A can ensure, i.e. it is a path π such that there is σ_A, $\forall \pi' \in \mathrm{Out}(s, \sigma_A).\ \pi \precsim_A \pi'$.

Remark. When preference relations are described by preorders, their might be incomparable values. Consider for example the turn-based game whose arena is represented in Fig. 2.4, and with a preference relation for player A_1 given by the preorder represented in Fig. 2.5. By choosing s_1, A_1 can ensure $s_0 \cdot s_1 \cdot s_4$, and by choosing s_2 she can ensure $s_0 \cdot s_2 \cdot s_4$, but she cannot ensure a path that is at least as good as both: these are two incomparable values, which are $s_0 \cdot s_1 \cdot s_4$ and $s_0 \cdot s_2 \cdot s_4$.

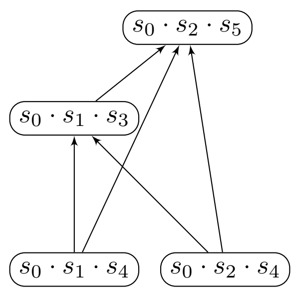


Figure 2.5: An example of a preference relation for the arena of Fig. 2.4. There is an edge $\pi_1 \to \pi_2$ when $\pi_1 \prec \pi_2$. With this preference relation, player A_1 has two incomparable values.

The decision problem associated to the notion of value is called the value problem and is defined as follow:

Value problem: Given a game $\mathcal{G}$, a state s of $\mathcal{G}$, a player A and a play π, can A ensure π from s (i.e. is there a strategy σ_A for player A such that for any outcome $\rho \in \mathrm{Out}^{\infty}_{\mathcal{G}}(s, \sigma_A)$, it holds $\pi \precsim_A \rho$)?

In *non-zero-sum* games, optimal strategies are usually too restricted since they consider that one player always plays against all the others. More relevant concepts are *equilibria*, which correspond to strategies on which the players can agree. One of the most studied notion of equilibria is *Nash equilibria*, in which the strategy of any player is optimal assuming the strategy of the others are fixed. We now introduce this concept formally.

Nash equilibria. Let $\mathcal{G}$ be a concurrent game and let s be a state of $\mathcal{G}$. Given a move m_{Agt} and an action m' for some player A, we write $m_{\mathrm{Agt}}[A \mapsto m']$ for the move n_{Agt} with $n_B = m_B$ when $B \neq A$ and $n_A = m'$. This is extended to strategies in the natural way. A *Nash equilibrium* [44] of $\mathcal{G}$ from s is a strategy profile $\sigma_{\mathrm{Agt}} \in \mathrm{Prof}_{\mathcal{G}}$ such that for all players $A \in \mathrm{Agt}$ and all strategies $\sigma' \in \mathrm{Strat}^A$:

$$\mathrm{Out}(s, \sigma_{\mathrm{Agt}}[A \mapsto \sigma']) \precsim_A \mathrm{Out}(s, \sigma_{\mathrm{Agt}})$$

In that context σ'_A is called a *deviation* due to A, and A is called a *deviator*.

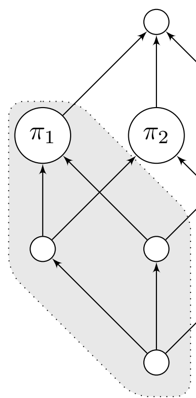


Figure 2.6: The notion of *improvement* for a non-total order.

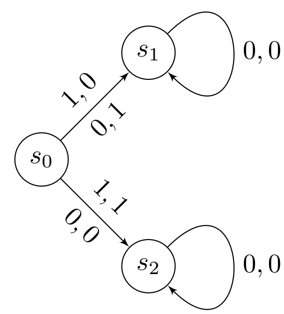


Figure 2.7: Example of a concurrent game with no pure Nash equilibrium

Hence, Nash equilibria are strategy profiles where no single player has an incentive to unilaterally deviate from her strategy.

Remark. The definition of Nash equilibria we have chosen, allows to model uncertainty about the preferences of players, by giving a partial order for the preferences. For instance for the preorder of Fig. 2.6, the fact that it is not known whether the player prefers π_1 or π_2, is modeled by these two paths being incomparable. We have to consider that π_2 is a possible improvement of π_1, and also that π_1 possibly improve π_2. If we want this player to play a Nash equilibrium whose outcome is π_1 we have to ensure that if she changes her strategy, the outcome will still be in the gray area. We can in fact notice that a Nash equilibrium for a preference relations $\precsim_{\mathrm{Agt}}$, is also a Nash equilibrium for all preference relations which refine $\precsim_{\mathrm{Agt}}$: formally $\precsim'$ refines $\precsim$ if for all paths π and π', $\pi \precsim \pi'$ implies $\pi \precsim' \pi'$; then if σ_{Agt} is a Nash equilibrium, then for all player A and strategy σ'_A: $\mathrm{Out}(s, \sigma_{\mathrm{Agt}}[A \mapsto \sigma']) \precsim_A \mathrm{Out}(s, \sigma_{\mathrm{Agt}})$, therefore if $\precsim'_A$ refines $\precsim_A$, $\mathrm{Out}(s, \sigma_{\mathrm{Agt}}[A \mapsto \sigma']) \precsim'_A \mathrm{Out}(s, \sigma_{\mathrm{Agt}})$, and σ_{Agt} is also a Nash equilibrium if we replace the preference relation $\precsim_{\mathrm{Agt}}$ by $\precsim'_{\mathrm{Agt}}$.

Remark. Although we restrict our strategy to pure strategies, a pure Nash equilibrium is resistant to mixed strategies.

Remark. In concurrent games, when we restrict strategies to pure one there might not always be a Nash equilibrium. For example, in the game represented in Fig. 2.7 and called the matching pennies, we consider that player A_1 prefers to reach state s_1 while player A_2 prefers to reach state s_2. If the strategies are fixed, then one of the two players can change her strategy in order to reach the state she prefers. Hence, starting from state s_0, there is no Nash equilibrium with pure strategies in that game.

Since they do not always exist, the basic question about Nash equilibria, is whether there exist one in from given game. We will formulate this question as a decision problem.

Existence problem: Given a game $\mathcal{G}$ and a state s in $\mathcal{G}$, does there exist a Nash equilibrium in $\mathcal{G}$ from s?

Deciding if there exists a Nash equilibrium, is often not enough, as several can coexist and there might exist one that is better for everyone than some other. Thus we refine the existence problem by adding constraints on the outcome. For instance we can ask if there is a Nash equilibrium whose outcome is the best for every player. The constraint on outcomes will be given by two plays π_A^- and π_A^+ for each player $A \in \mathrm{Agt}$, giving respectively a lower and an upper bound for the desired outcome. When the outcome π of a strategy profile σ_{Agt} satisfies $\pi_A^- \precsim_A \pi \precsim_A \pi_A^+$ for all $A \in \mathrm{Agt}$, we say that σ_{Agt} *satisfies the outcome constraint* $(\pi_A^-, \pi_A^+)_{A\in\mathrm{Agt}}$.

Existence with constrained outcomes: Given a game $\mathcal{G}$, a state s in $\mathcal{G}$, and two plays π_A^- and π_A^+ for each player A, does there exist a Nash equilibrium σ_{Agt} in $\mathcal{G}$ from s satisfying the outcome constraint $(\pi_A^-, \pi_A^+)_{A\in\mathrm{Agt}}$ (i.e. $\pi_A^- \precsim_A \mathrm{Out}(s, \sigma_{\mathrm{Agt}}) \precsim_A \pi_A^+$ for all $A \in \mathrm{Agt}$)?

In some situations, we also want to restrict the moves that are used by the strategies of the equilibria. For design reason, we might want to restrict the number of actions we are going to use, in order for the strategy to be simpler to implement. But we still want to be resistant to actions outside of the one we chose. The constraint on the move will be formally given by a function $\mathrm{Allow}\colon (\mathrm{States} \times \mathrm{Act}^{\mathrm{Agt}}) \to \{\mathsf{true}, \mathsf{false}\}$. When $\mathrm{Allow}(s, m_{\mathrm{Agt}}) = \mathsf{true}$ we say that the move m_{Agt} is *allowed* in s. When a strategy profile σ_{Agt} is such that for any history h, $\mathrm{Allow}(\mathrm{last}(h), \sigma_{\mathrm{Agt}}(h)) = \mathsf{true}$, we say that σ_{Agt} *satisfy the move constraint* Allow. This leads to the decision problem with constrained moves:

Existence with constrained moves: Given a game $\mathcal{G} = \langle \mathrm{States}, \mathrm{Agt}, \mathrm{Act}, \mathrm{Mov}, \mathrm{Tab}, (\precsim_A)_{A\in\mathrm{Agt}}\rangle$, a state $s \in \mathrm{States}$, two plays π_A^- and π_A^+ for each player A, a function $\mathrm{Allow}\colon (\mathrm{States}\times\mathrm{Act}^{\mathrm{Agt}}) \to \{\mathsf{true}, \mathsf{false}\}$, does there exist a Nash equilibrium σ_{Agt} in $\mathcal{G}$ from s satisfying the outcome constraint $(\pi_A^-, \pi_A^+)_{A\in\mathrm{Agt}}$ and the move constraint Allow (i.e. for any history h from s, $\mathrm{Allow}(\mathrm{last}(h), \sigma_{\mathrm{Agt}}(h)) = \mathsf{true}$) ?

In the following when talking about the constrained existence problem, we will refer to the existence with constrained moves which is the most general problem. It is clear that if we can solve the existence with constrained move we can also solve the existence with constrained outcome, and if we can solve the existence with existence with constrained outcome we can solve the existence problem.

When studying the complexity of this problem, we will restrict the functions Allow given as input to the ones that are computable in polynomial time. To ensure this we could for example ask for a function given by a Boolean circuit. We now see an example of such a constraint Allow, we will also see in Chapter 6, that this constraint is useful when interpreting timed games as concurrent games, to decide the existence problem.

Example 5. In a real situation, involving several similar devices, it would be more practical if all devices run the same program. When the situation is modeled as a game, this means that the strategy should to be the same for all players. Assuming they are synchronous, we can make use of the constraint on moves. We enforce the restriction with the function defined by $\mathrm{Allow}(s, m_{\mathrm{Agt}})$ if and only if $m_A = m_B$ for any players $A, B \in \mathrm{Agt}$.

The complexity of all these decision problems heavily depends on what preorders we allow for the preference relation and how they are represented. Even for finite games, in their most general form, all these problems are undecidable, as we prove in the next section.

2.3 Undecidability in Weighted Games

A weighted game is a standard concurrent game where preference for each player A is given by a *weight function* $\mathtt{cost}_A : \mathrm{States} \mapsto \mathbb{Z}$. The *accumulated cost* of play ρ is given by the sum of the weights: $\mathtt{cost}_A(\rho) = \sum_{i \geq 0} \mathtt{cost}_A(\rho_i)$. The goal of the player is then to minimize the accumulated cost. Formally, for ρ and ρ' two plays, $\rho \precsim_A \rho'$ if, and only if, $\mathtt{cost}_A(\rho)$ is finite and $\mathtt{cost}_A(\rho) \leq \mathtt{cost}_A(\rho')$ or $\mathtt{cost}_A(\rho')$ is infinite.

Theorem 2.1. *The existence and constrained-existence problems are undecidable for weighted games.*

Proof. We first prove the result for the constrained existence problem. We encode the halting problem for two-counter machines into a turn-based weighted game with 5 players. This problem is known to be undecidable. Without loss of generality, we will assume that the two counters are reset to zero before the machine halts. The value of counter c_1 is encoded in the following way: if its value is c_1 for a given history of the two-counter machine, then the accumulated cost for player A_1 of the corresponding history is $c_1 - 1$ and the accumulated cost for B_1 is $-c_1$. Having two players for one counter will make it easy to test whether it equals 0. Similarly to code the value of c_2, we have one player A_2 whose accumulated cost is equal to $c_2 - 1$ and B_2 whose accumulated cost is equal to $-c_2$. To initialize the value of the counter, we visit a state whose weight is -1 for A_1 and A_2 before going to the initial state. And we do the opposite in the

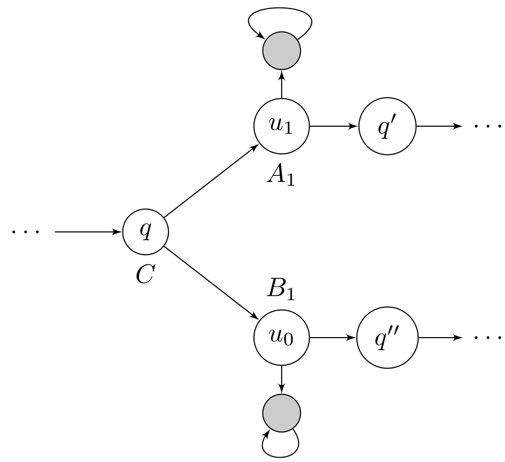


Figure 2.8: Testing whether $c_1 = 0$.

halting state so that in a normal execution the accumulated weight for all A_i and B_i is 0. Incrementing counter c_i, consists in visiting a state whose weight is 1 for A_i and -1 for B_i, and vice versa for decrementing this counter. More precisely, if instruction q_k of the two-counter machine consists in incrementing c_i and jumping to $q_{k'}$, then the game will have a transition from state q_k to a state p_k whose weight is given by $\texttt{cost}_{A_i}(p_k) = 1$, $\texttt{cost}_{B_i}(p_k) = -1$, and $\texttt{cost}_A(p_k) = 0$ for a player $A \in \text{Agt} \setminus \{A_i, B_i\}$, and another transition from there to $q_{k'}$.

It remains to encode the zero-test, for this the game will involve an additional player C, the aim of this player will be to reach the state corresponding to the final state of the two-counter machine, this is encoded by giving a negative cost for C to the state of the game corresponding to the final state of the two-counter machine. The equilibrium we will ask for, is one where C reaches her goal, and A_1, A_2, B_1 and B_2 gets an accumulated cost of 0. Now, a zero-test is encoded by a module shown in Fig. 2.8. In this module, player C will try to avoid the two sink states (marked in grey), since this would prevent her from reaching her goal.

When entering the module, player C has to choose one of the available branches: if she decides to go to u_1, then A_1 could take the play into the self-loop, which is an improvement for her if her accumulated cost in the history is below 0, which corresponds to having $c_i = 0$; hence player C should play to u_1 only if $c_1 \neq 0$, so that A_1 will have no interest in going to this self-loop.

Similarly, if player C decides to go to u_0, player B_1 has the opportunity to "leave" the main stream of the game, and go to the sink state. If the accumulated cost for B_1 is below 0 up to that point, corresponding to a value of c_1 strictly positive, then B_1 has the opportunity to play in the self-loop, and to win. Conversely, when $c_1 = 0$, B_1 has no interest in playing in the self-loop since her accumulated cost would be 0. Hence, if $c_i = 0$ when entering the module,

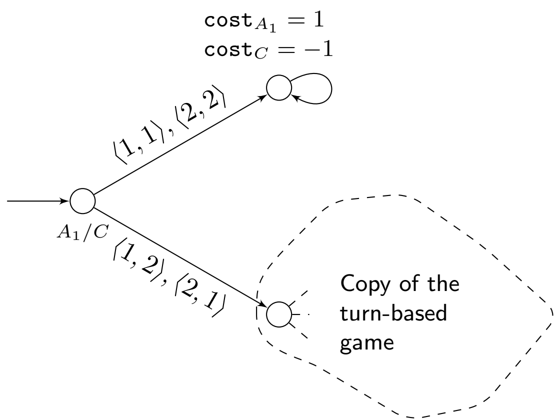


Figure 2.9: Extending the game with an initial concurrent module

then player C should go to u_0.

One can then easily show that the 2-counter machine stops if, and only if, there is a Nash equilibrium in the resulting game, in which player C reach her goal and players A_1, B_1, A_2 and B_2 have an accumulated cost of 0. Indeed, assume that the machine stops, and consider the strategies where player C plays (in the first state of the test modules) according to the value of the corresponding counter, and where players A_1, B_1, A_2 and B_2 always keep the play in the main stream of the game. Since the machine stops, player C wins, while players A_1, B_1, A_2 and B_2 get an accumulated cost of 0. Moreover, none of them has a way to improve their payoff: since player C plays according to the values of the counters, players A_1 and A_2 would not benefit from deviating from their above strategies. Conversely, if there is such a Nash equilibrium, then in any visited test module, player C always plays according to the values of the counter c_i: otherwise, player A_i (or B_i) would have the opportunity to win the game. By construction, this means that the outcome of the Nash equilibrium corresponds to the execution of the two-counter machine. As player C wins, this execution reaches the final state.

We can prove hardness for the existence problem as well, by adding a module as represented in Fig. 2.9. There exists a Nash equilibrium in this game if and only if there is one where C reaches her goal in the turn-based game. We will use the idea of this module several time in the following. In the next section, we will provide a generic lemma that generalizes this idea.

□

2.4 General Properties

This section contains generic lemmas that we reuse several times later.

2.4.1 Nash Equilibria as Lasso Runs

We first characterize outcomes of Nash equilibria as ultimately periodic runs in finite games.

Lemma 2.2. *Let s be a state of a finite game $\mathcal{G}$. Assume that every player has a preference relation which only depends on the set of states that are visited and on the set of states that are visited infinitely often (in other terms, if* $\text{Inf}(\rho) = \text{Inf}(\rho')$ *and* $\text{Occ}(\rho) = \text{Occ}(\rho')$*, then $\rho \sim_A \rho'$ for every player $A \in \text{Agt}$).*

If there is a Nash equilibrium with outcome ρ, then there is a Nash equilibrium with outcome ρ' of the form $\pi \cdot \tau^\omega$ such that $\rho \sim_A \rho'$, and where $|\pi|$ and $|\tau|$ are bounded by $|\text{States}|^2$.

Proof. Let σ_{Agt} be a Nash equilibrium, and ρ be its outcome from s. We define a new strategy profile σ'_{Agt}, whose outcome from s is ultimately periodic, and then show that σ'_{Agt} is a Nash equilibrium from s.

To begin with, we inductively construct a history $\pi = \pi_0\pi_1 \dots \pi_n$ that is not too long and visits precisely those states that are visited by ρ.

The initial state is $\pi_0 = \rho_0 = s$. Then we assume we have constructed $\pi_{\leq k} = \pi_0 \dots \pi_k$ which visits exactly the same states as $\rho_{\leq k'}$ for some k'. If all the states of ρ have been visited in $\pi_{\leq k}$ then the construction is over. Otherwise there is an index i such that ρ_i does not appear in $\pi_{\leq k}$. We therefore define our next target as the smallest such i: we let $t(\pi_{\leq k}) = \min\{i \mid \forall j \leq k.\ \pi_j \neq \rho_i\}$. We then look at the occurrence of the current state π_k that is the closest to the target in ρ: we let $c(\pi_{\leq k}) = \max\{i < t(\pi_{\leq k}) \mid \pi_k = \rho_i\}$. Then we emulate what happens at that position by choosing $\pi_{i+1} = \rho_{c(\pi_{\leq i})+1}$. Then π_{i+1} is either the target, or a state that has already been seen before in $\pi_{\leq k}$, in which case the resulting $\pi_{\leq k+1}$ visits exactly the same states as $\rho_{\leq c(\pi_{\leq i})+1}$.

At each step, either the number of remaining targets strictly decreases, or the number of remaining targets is constant but the distance to the next target strictly decreases. Therefore the construction terminates. Moreover, notice that between two targets we do not visit the same state twice, and we visit only states that have already been visited, plus the target. As the number of targets is bounded by $|\text{States}|$, we get that the length of the path π constructed thus far is bounded by $1 + |\text{States}| \cdot (|\text{States}| - 1)/2$.

Using similar ideas, we now inductively construct $\tau = \tau_0\tau_1 \dots \tau_m$, which visits precisely those states which are seen infinitely often along ρ, and which is not too long. Let l be the least index after which the states visited by ρ are visited infinitely often: $l = \min\{i \mid \forall j \geq i.\ \rho_j \in \text{Inf}(\rho)\}$. The run $\rho_{\geq l}$ is such that its set of visited states and its set of states visited infinitely often coincide. We therefore define τ in the same way we have defined π above, but for play $\rho_{\geq l}$. As a by-product, we also get $c(\tau_{\leq k})$, for $k < m$.

We now need to glue π and τ together, and to ensure that τ can be glued to itself, so that $\pi \cdot \tau^\omega$ is a real run. We therefore need to link the last state of π with the first state of τ (and similarly the last state of τ with its first state). This possibly requires appending some more states to π and τ: we fix the target of π and τ to be τ_0, and apply the same construction as previously. The total length of the resulting paths π and τ is bounded by $1 + (|\text{States}| - 1) \cdot (|\text{States}| + 2)/2$ which less than $|\text{States}|^2$.

We let $\rho' = \pi \cdot \tau^\omega$, and abusively write $c(\rho'_{\leq k})$ for $c(\pi_{\leq k})$ if $k \leq |\pi|$ and $c(\tau_{\leq k'})$ with $k' = (k - 1 - |\pi|) \mod |\tau|$ otherwise. We now define our new strategy profile, having ρ' as outcome from s. Given a history h:

- if h followed the expected path, i.e., $h = \rho'_{\leq k}$ for some k, we mimic the strategy at $c(h)$: $\sigma'_{\text{Agt}}(h) = \sigma_{\text{Agt}}(\rho_{c(h)})$. This way, ρ' is the outcome of σ'_{Agt} from s.
- otherwise we take the longest prefix $h_{\leq k}$ that is a prefix of ρ', and define $\sigma'_{\text{Agt}}(h) = \sigma_{\text{Agt}}(\rho_{c(h_{\leq k})} \cdot h_{\geq k+1})$.

We now show that σ'_{Agt} is a Nash equilibrium. Assume that one of the players changes her strategy while playing according to σ'_{Agt}: either the resulting outcome does not deviate from $\pi \cdot \tau^\omega$, in which case the payoff of that player is not improved; or it deviates at some point, and from that point on, σ'_{Agt} follows the same strategies as in σ_{Agt}. Assume that the resulting outcome is an improvement over ρ' for the player who deviated. The suffix of the play after the deviation is the suffix of a play of σ_{Agt} after a deviation by the same player. By construction, both plays have the same visited and infinitely-visited sets. Hence we have found an advantageous deviation from σ_{Agt} for one player, contradicting that σ_{Agt} is a Nash equilibrium. □

2.4.2 Encoding Value as an Existence Problem with Constrained Outcomes

In the rest of the paper, we prove several hardness results for the constrained-existence problem. Several of them can be inferred from the hardness of the corresponding value problem, using the following lemma:

Lemma 2.3. *Let $\mathcal{G}$ be a two-player zero-sum game. Assume that the two players are A and B, and that the preference relation $\precsim_A$ for player A is total, Noetherian and almost-well-founded. Assume furthermore that $\mathcal{G}$ is determined, i.e., for all play π:*

$$[\exists \sigma_A.\ \forall \sigma_B.\ \pi \precsim_A \text{Out}(\sigma_A, \sigma_B)] \quad \Leftrightarrow \quad [\forall \sigma_B.\ \exists \sigma_A.\ \pi \precsim_A \text{Out}(\sigma_A, \sigma_B)]$$

Let $\mathcal{G}'$ be the (non-zero-sum) game obtained from $\mathcal{G}$ by replacing the preference relation of player B by the one where all plays are equivalent. Then, for every state s, for every play π from s, the two following properties are equivalent:

(*i*) *there is a Nash equilibrium in $\mathcal{G}'$ from s with outcome ρ such that $\pi \not\precsim_A \rho$;*

(*ii*) *player A cannot ensure π from s in $\mathcal{G}$.*

Under the hypotheses of this lemma, the constrained-existence problem is then at least as hard as the complement of the value problem.

Proof. In this proof, σ_A and σ'_A (resp. σ_B and σ'_B) refer to player-A (resp. player-B) strategies. Furthermore we will write $\mathrm{Out}(\sigma_A, \sigma_B)$ instead of $\mathrm{Out}_{\mathcal{G}}(s, (\sigma_A, \sigma_B))$.

We first assume there is a Nash equilibrium (σ_A, σ_B) in $\mathcal{G}'$ from s such that $\pi \not\precsim_A \mathrm{Out}(\sigma_A, \sigma_B)$. Since $\precsim_A$ is total, $\mathrm{Out}(\sigma_A, \sigma_B) \prec_A \pi$. Consider a strategy σ'_A of player A in $\mathcal{G}$. As (σ_A, σ_B) is a Nash equilibrium, it holds that $\mathrm{Out}(\sigma'_A, \sigma_B) \precsim_A \mathrm{Out}(\sigma_A, \sigma_B)$, which implies $\mathrm{Out}(\sigma'_A, \sigma_B) \prec_A \pi$. We conclude that condition (ii) holds.

Assume now property (ii). As the preference relation is Noetherian, we can select π^+ which is the largest element for $\precsim_A$ which can be ensured by player A. Let σ_A be a corresponding strategy: for every strategy σ_B, $\pi^+ \precsim_A \mathrm{Out}(\sigma_A, \sigma_B)$. Towards a contradiction, assume now that for every strategy σ'_B, there exists a strategy σ'_A such that $\pi^+ \prec_A \mathrm{Out}(\sigma'_A, \sigma'_B)$. Consider the set S of such outcomes, and define π' as its minimal element (this is possible since the order $\precsim_A$ is almost-well-founded). Notice then that $\pi^+ \prec_A \pi'$, and also that for every strategy σ'_B, there exists a strategy σ'_A such that $\pi' \precsim_A \mathrm{Out}(\sigma'_A, \sigma'_B)$. Then, as the game is determined, we get that there exists some strategy σ'_A such that for all strategy σ'_B, it holds that $\pi' \precsim_A \mathrm{Out}(\sigma'_A, \sigma'_B)$. In particular, strategy σ'_A ensures π', which contradicts the maximality of π^+. Therefore, there is some strategy σ'_B for which for every strategy σ'_A, $\pi^+ \not\prec_A \mathrm{Out}(\sigma'_A, \sigma'_B)$, which means $\mathrm{Out}(\sigma'_A, \sigma'_B) \precsim_A \pi^+$. We show now that (σ_A, σ'_B) is a witness for property (i). We have seen on the one hand that $\pi^+ \precsim_A \mathrm{Out}(\sigma_A, \sigma'_B)$, and on the other hand that $\mathrm{Out}(\sigma_A, \sigma'_B) \precsim_A \pi^+$. By hypothesis, $\pi^+ \prec_A \pi$, which yields $\mathrm{Out}(\sigma_A, \sigma'_B) \prec_A \pi$ Pick another strategy σ'_A for player A. We have seen that $\mathrm{Out}(\sigma'_A, \sigma'_B) \precsim_A \pi^+$, which implies $\mathrm{Out}(\sigma'_A, \sigma'_B) \precsim_A \mathrm{Out}(\sigma_A, \sigma'_B)$. This concludes the proof of (i). ☐

Remark. Note that any finite total preorder is also Noetherian and almost-well-founded. Any total preorder isomorphic to the set of non-positive integers is also Noetherian and almost-well-founded.

2.4.3 Encoding Value as an Existence Problem

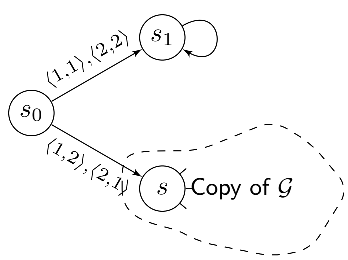


Figure 2.10: Extending game $\mathcal{G}$ with an initial concurrent module. The label A_1/C below the initial state, means that only the choices of these two players matters for determining the next state. Therefore only the actions on A_1 and C are shown on the outgoing transitions.

We prove a similar result for the existence problem. In this reduction however, we have to modify the game by introducing a truly concurrent move at the beginning of the game. This is necessary since for turn-based games with ω-regular winning conditions, there always exists a Nash equilibrium [14], hence the existence problem would be trivial.

From a zero-sum game $\mathcal{G}$, given a state s and a play π from s, we define a game $\mathcal{G}_\pi$ by adding two states s_0 and s_1. From s_0, A and B play a matching-penny game to either go to the sink state s_1, or to the state s in the game $\mathcal{G}$, as shown in Fig. 2.10. We assume the same hypotheses than in Lem. 2.3 for the preference relation $\precsim_A$. Let π^+ be in the highest equivalence class for $\precsim_A$ smaller than π (it exists since $\precsim_A$ is Noetherian). In $\mathcal{G}_\pi$, player B prefers runs that end in s_1: formally, the preference relation $\precsim_B^\pi$ of player B is given by $\pi' \precsim_B^\pi \pi'' \Leftrightarrow \pi'' = s_0 \cdot s_1^\omega \vee \pi' \neq s_0 \cdot s_1^\omega$. On the other hand, player A prefers a path of $\mathcal{G}$ over going to s_1, if and only if, it is at least as good as π. Formally, the preference relation $\precsim_A^\pi$ for player A is given by $s_0 \cdot \pi' \precsim_A^\pi s_0 \cdot \pi'' \Leftrightarrow \pi' \precsim_A \pi''$, and $s_0 \cdot s_1^\omega \sim_A'' s_0 \cdot \pi^+$.

Lemma 2.4. *Let $\mathcal{G}$ be a two-player zero-sum game (with players A and B); we require moreover that $\mathcal{G}$ is determined, and that the preference relation $\precsim_A$ for player A is total, Noetherian and almost-well-founded. Pick a state s and a play π in $\mathcal{G}$ from s, and consider the game $\mathcal{G}_\pi$ defined above. Then the following two properties are equivalent:*

1. *there is a Nash equilibrium in $\mathcal{G}_\pi$ from s_0;*
2. *player A cannot ensure π from s in $\mathcal{G}$.*

If the new preference relations are definable in the class of games that is considered, the existence problem is then at least as hard as the complement of the value problem.

Proof. Assume that player A cannot ensure at least π from s in $\mathcal{G}$, then according to Lemma 2.3, there is a Nash equilibrium (σ_A, σ_B) in the game $\mathcal{G}'$ of Lemma 2.3 with outcome ρ such that $\pi \not\precsim_A \rho$. Consider the strategy profile $(\sigma_A^\pi, \sigma_B^\pi)$ in $\mathcal{G}_\pi$ that consists in playing the same action for both players in s_0, and then if the path goes to s, to play according to (σ_A, σ_B). Player B gets her best possible payoff under that strategy profile. If A could change her strategy to get a payoff better than $s_0 \cdot \pi^+$, then it would induce a strategy in $\mathcal{G}'$ giving her a payoff better than ρ (when played with strategy σ_B), which contradicts the fact that (σ_A, σ_B) is a Nash equilibrium in $\mathcal{G}'$. Therefore, $(\sigma_A^\pi, \sigma_B^\pi)$ is a Nash equilibrium in $\mathcal{G}_\pi$.

Conversely, assume that A can ensure π from s in $\mathcal{G}$, and assume towards a contradiction that there is a Nash equilibrium $(\sigma_A^\pi, \sigma_B^\pi)$ in $\mathcal{G}_\pi$ from s_0. Then $\mathrm{Out}(\sigma_A^\pi, \sigma_B^\pi)$ does not end in s_1, otherwise player A could improve by switching to s and then playing according to a strategy which ensures π. Also, $\mathrm{Out}(\sigma_A^\pi, \sigma_B^\pi)$ cannot end in $\mathcal{G}$ either, otherwise player B would improve by switching to s_1. We get that there is no Nash equilibrium in $\mathcal{G}_\pi$ from s_0, which concludes the proof. □

2.4.4 Encoding the Existence Problem with Constrained Outcome as an Existence Problem

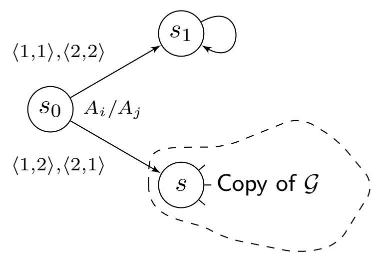


Figure 2.11: Extending game $\mathcal{G}$ with an initial concurrent module, to obtain game $E(\mathcal{G}, A_i, A_j, \rho)$.

The next lemma makes a link between the existence of a Nash equilibrium where a player A_i gets a payoff above some bound and the (unconstrained) existence of a Nash equilibrium in a new game $E(\mathcal{G}, A_i, A_j, \rho)$, where ρ is a play in $\mathcal{G}$.

The construction is similar to the previous one, for two selected players: given a concurrent game $\mathcal{G}$, a state s, a play ρ from s, and two distinct players A_i and A_j, we define the game $E(\mathcal{G}, A_i, A_j, \rho)$ by adding two states s_0 and s_1 as in Fig. 2.11. In s_0, the two players A_i and A_j play a matching-penny game to either go to the sink state s_1, or to state s in the game $\mathcal{G}$.

For player A_j, the preference relation is given by $\precsim'_{A_j}$ such that $s_0 \cdot s_1^\omega \prec'_{A_j} s_0 \cdot \pi$ and $s_0 \cdot \pi \precsim'_{A_j} s_0 \cdot \pi' \Leftrightarrow \pi \precsim_{A_j} \pi'$ for any path π and π' of $\mathcal{G}$. For player A_i the preference relation is $s_0 \cdot \pi \precsim'_{A_i} s_0 \cdot \pi' \Leftrightarrow \pi \precsim_{A_i} \pi'$, for any path π and π' of $\mathcal{G}$, and $s_0 \cdot s_1^\omega \sim_{A_i} s_0 \cdot \rho$. For any other player A_k, the preference relation is given by $s_0 \cdot \pi \precsim'_{A_k} s_0 \cdot \pi' \Leftrightarrow \pi \precsim_{A_k} \pi'$ for any path π and π' in $\mathcal{G}$, and $s_0 \cdot s_1^\omega \sim_{A_k} s_0 \cdot \rho$.

Lemma 2.5. *For any Nash equilibrium in $\mathcal{G}$ whose outcome π is such that $\rho \precsim_{A_i} \pi$, there is a Nash equilibrium in $E(\mathcal{G}, A_i, A_j, \rho)$ whose outcome is $s_0 \cdot \pi$. Reciprocally, for any Nash equilibrium in $E(\mathcal{G}, A_i, A_j, \rho)$, its outcome $s_0 \cdot \pi$ is such that $\rho \precsim_{A_i} \pi$ and there is a Nash equilibrium in $\mathcal{G}$ whose outcome is π.*

Proof. Assume that there is a Nash equilibrium $(\sigma_A)_{A\in\mathrm{Agt}}$ in $\mathcal{G}$ with outcome π such that $\rho \precsim_{A_i} \pi$. Then $s_0 \cdot s_1^\omega \precsim'_{A_i} s_0 \cdot \pi$. Consider the strategy profile in $E(\mathcal{G}, A_i, A_j, \rho)$ that consists for A_i and A_j in playing different actions in s_0 and when the path goes to s, to play according to $(\sigma_A)_{A\in\mathrm{Agt}}$. Players A_i and A_j have no interest in changing their strategies in s_0, since for A_j all plays of $\mathcal{G}$ are better than $s_0 \cdot s_1^\omega$, and for A_i the play $s_0 \cdot \pi$ is better than $s_0 \cdot s_1^\omega$. Hence, this is a Nash equilibrium in game $E(\mathcal{G}, A_i, A_j, \rho)$.

Reciprocally, if there is a Nash equilibrium in $E(\mathcal{G}, A_i, A_j, \rho)$, its outcome cannot end in s_1, since A_j would have an interest in changing her strategy in

s_0 (all plays of $\mathcal{G}$ are then better for her). The strategies followed from s thus defines a Nash equilibrium in $\mathcal{G}$. $\square$

If the new preference relations are definable in the class of games that is considered, the existence problem is then at least as hard as the constrained existence problem. Note however that the reduction assumes lower bounds on the payoffs, and we do not have a similar result for upper bounds on the payoffs. For instance, as we will see in Part II of the paper, for a conjunction of Büchi objectives, we do not know whether the existence problem is in P (as the value problem) or NP-hard (as is the existence of an equilibrium where all the players are losing).

Chapter 3

The Suspect Game

In this chapter, we show how, from a multiplayer game $\mathcal{G}$, we can construct a two-player zero-sum game $\mathcal{H}$, such that there is a correspondence between Nash equilibria in $\mathcal{G}$ and winning strategies in $\mathcal{H}$. This transformation is conceptually much deeper than the reductions given in Section 2.4. It will allow us to use algorithmic techniques from zero-sum games to solve our problems.

For this chapter, we fix a game $\mathcal{G} = \langle \mathrm{States}, \mathrm{Agt}, \mathrm{Act}, \mathrm{Mov}, \mathrm{Tab}, (\precsim_A)_{A\in\mathrm{Agt}}\rangle$.

3.1 The Suspect Game Construction

We begin with introducing a few extra definitions.

Trigger strategy. A strategy profile σ_{Agt} is a *trigger strategy* for an infinite path π from some state s if, for any strategy σ'_A of any player $A \in \mathrm{Agt}$, the path π is at least as good as the outcome of $\sigma_{\mathrm{Agt}}[A \mapsto \sigma'_A]$ from s (that is, $\mathrm{Out}(s, \sigma_{\mathrm{Agt}}[A \mapsto \sigma'_A]) \precsim_A \pi$).

The following result is a direct consequence of the definition:

Lemma 3.1. *A Nash equilibrium is a trigger strategy for its outcome. Reciprocally, if σ_{Agt} is a trigger strategy for its outcome, then it is a Nash equilibrium.*

We now define the central notion of suspect player. In Nash equilibria, players have to prevent deviators from improving their outcome. As our game are not necessarily action-visible, by observing the sequence of states, it is not always possible to know which player is responsible for the deviation. The suspect set represent the possible identity of the deviator.

Suspects. Given two states s and s', and a move m_{Agt}, the set of *suspect players* for (s, s') and m_{Agt} is the set

$$\mathrm{Susp}((s, s'), m_{\mathrm{Agt}}) = \{A \in \mathrm{Agt} \mid \exists\, m' \in \mathrm{Mov}(s, A).\ \mathrm{Tab}(s, m_{\mathrm{Agt}}[A \mapsto m']) = s'\}.$$

Given a play ρ and a strategy profile σ_{Agt}, the set of suspect players for ρ and σ_{Agt} is the set of players that are suspect along each transition of ρ, i.e., it is the set

$$\mathrm{Susp}(\rho, \sigma_{\text{Agt}}) = \left\{ A \in \text{Agt} \;\middle|\; \forall i < |\rho| .\ A \in \mathrm{Susp}\big((\rho_{=i}, \rho_{=i+1}), \sigma_{\text{Agt}}(\rho_{\leq i})\big) \right\}.$$

Intuitively, player $A \in \text{Agt}$ is a suspect for transition (s, s') and move m_{Agt} if she can unilaterally change her action to activate the transition from s to s'. Obviously, if $\mathrm{Tab}(s, m_{\text{Agt}}) = s'$, then $\mathrm{Susp}((s, s'), m_{\text{Agt}}) = \text{Agt}$. Similarly, we easily infer that player A is in $\mathrm{Susp}(\rho, \sigma_{\text{Agt}})$ if, and only if, there is a strategy σ'_A such that $\mathrm{Out}(s, \sigma_{\text{Agt}}[A \mapsto \sigma'_A]) = \rho$.

Example 6. As an example, consider the simple game model for medium access control that we gave in Example 2. From state $s = (0, 0)$, we consider the move $(m_{A_1}, m_{A_2}) = (\mathtt{w}, \mathtt{w})$. For state $s' = (1, 0)$, A_1 is suspect since she can change her action to $m'_{A_1} = \mathtt{t}$ so that the next state is s'; A_2 is not suspect because if she changes her action the new state would be $(0, 1)$. Hence $\mathrm{Susp}((s, s'), (\mathtt{w}, \mathtt{w})) = \{A_1\}$. Notice that if $s' = (0, 0)$ then both player are suspect because they can stick to the same action: $m'_A = m_A$ and the transition will be triggered since it is the natural outcome for (m_{A_1}, m_{A_2}). On the contrary if $s' = (1, 1)$ then nobody is suspect since there is no transition from s to s'. Now consider another move $(m_{A_1}, m_{A_2}) = (\mathtt{t}, \mathtt{w})$, the natural next state for this move from s is $(1, 0)$ but if A_1 changes her action to $\mathtt{w}$ or if A_2 changes her action to $\mathtt{t}$ the next state would be $(0, 0)$, therefore both player are suspect for the transition $(0, 0) \to (0, 0)$, hence $\mathrm{Susp}(((0, 0), (0, 0)), (\mathtt{t}, \mathtt{w})) = \{A_1, A_2\}$.

Suspect game. With a game $\mathcal{G}$, an infinite path π and a constraint Allow on the moves, we associate a two-player turn-based game $\mathcal{H}(\mathcal{G}, \pi, \text{Allow})$. We simply write $\mathcal{H}$ when $\mathcal{G}$, π and Allow are clear from the context. The players in $\mathcal{H}$ are named `Eve` and `Adam`. Since $\mathcal{H}$ is turn-based, its state space can be written as the disjoint union of the set $V_\exists$ controlled by `Eve`, which is (a subset of) $\text{States} \times 2^{\text{Agt}}$, and the set $V_\forall$ controlled by `Adam`, which is (a subset of) $\text{States} \times 2^{\text{Agt}} \times \text{Act}^{\text{Agt}}$. The game is played in the following way:

1. from a configuration (s, P) in $V_\exists$, `Eve` chooses a legal move m_{Agt} from s such that $\text{Allow}(s, m_{\text{Agt}}) = \mathsf{true}$;
2. the next state is (s, P, m_{Agt});
3. then `Adam` chooses some state s' in States;
4. the new state is $(s', P \cap \mathrm{Susp}((s, s'), m_{\text{Agt}}))$.

These four steps are repeated to form an infinite path in $\mathcal{H}$. When the state s' chosen by `Adam` in step 3, is such that $s' = \mathrm{Tab}(s, m_{\text{Agt}})$, we say that `Adam` *obeys* `Eve`. When this is the case, the new configuration in step 4 is (s', P).

We define projections π_1 and π_2 from $V_\exists$ on States and 2^{Agt}, resp., by $\pi_1(s, P) = s$ and $\pi_2(s, P) = P$. We extend these projections to plays in a natural way (but only using `Eve`'s states in order to avoid stuttering), letting $\pi_1((s_0, P_0) \cdot (s_0, P_0, m_0) \cdot (s_1, P_1) \cdots) = s_0 \cdot s_1 \cdots$. For any play ρ, $\pi_2(\rho)$ (seen as a sequence of sets of players of $\mathcal{G}$) is non-increasing, therefore its limit $L(\rho)$ is well defined. We notice that if $L(\rho) \neq \varnothing$, then $\pi_1(\rho)$ is a real infinite path in $\mathcal{G}$. An outcome ρ is *winning for* ***Eve*** , if for all $A \in L(\rho)$, it holds $\pi_1(\rho) \precsim_A \pi$. The *winning region* $W(\mathcal{G}, \pi, \text{Allow})$ (later simply denoted by W when $\mathcal{G}$, π and Allow are clear from the context) is the set of configurations of $\mathcal{H}(\mathcal{G}, \pi, \text{Allow})$ from which `Eve` has a winning strategy.

Intuitively `Eve` tries to have the players play a Nash equilibrium constrained by Allow, and `Adam` tries to disprove that it is a Nash equilibrium, by finding a possible deviation that improves the payoff of one of the players.

At first sight, the number of states in $\mathcal{H}$ is exponential (in the number of players of $\mathcal{G}$). However, there are two cases for which we easily see that the number of states of $\mathcal{H}$ is actually only polynomial:

- if there is a state in which all the players have several possible moves, then the transition table (which is part of the input [38]) is also exponential in the number of players;
- if the game is turn-based, then the transition table is "small", but either all the players are suspect or there is at most one suspect player, so that the number of reachable states in $\mathcal{H}$ is also small.

We now prove that this can be generalized:

Lemma 3.2. *The number of reachable configurations from* States $\times$ {Agt} *in* $\mathcal{H}$ *is polynomial in the size of* $\mathcal{G}$*.*

Proof. The game $\mathcal{H}$ contains the state (s, Agt) and the states $(s, \text{Agt}, m_{\text{Agt}})$, where m_{Agt} is a legal and allowed move from s; the number of these states is bounded by $|\text{States}| + |\text{Tab}|$. The successors of those states that are not of the same form, are the $(t, \text{Susp}((s,t), m_{\text{Agt}}))$ with $t \neq \text{Tab}(s, m_{\text{Agt}})$. If some player $A \in \text{Agt}$ is a suspect for transition (s,t), then besides m_A, she must have at least a second action m', for which $\text{Tab}(s, m_{\text{Agt}}[A \mapsto m']) = t$. Thus the transition table from state s has size at least $2^{|\text{Susp}((s,t), m_{\text{Agt}})|}$. The successors of $(t, \text{Susp}((s,t), m_{\text{Agt}}))$ are of the form (t', P) or (t', P, m_{Agt}) where P is a subset of $\text{Susp}((s,t), m_{\text{Agt}})$; there can be no more than $(|\text{States}| + |\text{Tab}|) \cdot 2^{|\text{Susp}((s,t), m_{\text{Agt}})|}$ of them, which is bounded by $(|\text{States}| + |\text{Tab}|) \cdot |\text{Tab}|$. The total number of reachable states is then bounded by $(|\text{States}| + |\text{Tab}|) \cdot (1 + (|\text{States}| + |\text{Tab}|) \cdot |\text{Tab}|)$. □

3.2 Relation Between Trigger Strategies and Winning Strategies of the Suspect Game

The next two lemmas state the correctness of our construction, establishing a correspondence between winning strategies in $\mathcal{H}$ and Nash equilibria in $\mathcal{G}$.

Lemma 3.3. *Let π and ρ be two infinite paths in $\mathcal{G}$ and* Allow *a constraint on the moves. The following two conditions are equivalent:*

- `Eve` *has a winning strategy in $\mathcal{H}(\mathcal{G}, \pi, \mathrm{Allow})$ from (s, Agt), and its outcome ρ' from s when* `Adam` *obeys* `Eve` *is such that $\pi_1(\rho') = \rho$;*
- *there is a trigger strategy σ_{Agt} for π in $\mathcal{G}$ from state s whose outcome from s is ρ, and that satisfies the move constraint* Allow.

Proof. Assume there is a winning strategy $\sigma^\exists$ for `Eve` in $\mathcal{H}$ from (s, Agt), whose outcome from s when `Adam` obeys `Eve` is ρ' with $\pi_1(\rho') = \rho$. We define the strategy profile σ_{Agt} according to the actions played by `Eve`. Pick a history $g = s_1 s_2 \cdots s_{k+1}$, with $s_1 = s$. Let h be the outcome of $\sigma^\exists$ from s ending in a state of $V_\exists$ and such that $\pi_1(h) = s_1 \cdots s_k$. This history is uniquely defined as follows: the first state of h is (s_1, Agt), and if its $(2i+1)$-st state is (s_i, P_i), then its $(2i+2)$-nd state is $(s_i, P_i, \sigma^\exists(h_{\leq 2i+1}))$ and its $(2i+3)$-rd state is $(s_{i+1}, P_i \cap \mathrm{Susp}((s_i, s_{i+1}), \sigma^\exists(h_{\leq 2i+1})))$. Now, write (s_k, P_k) for the last state of h, and let $h' = h \cdot (s_k, P_k, \sigma^\exists(h)) \cdot (s_{k+1}, P_k \cap \mathrm{Susp}((s_k, s_{k+1}), \sigma^\exists(h)))$. Then we define $\sigma_{\mathrm{Agt}}(g) = \sigma^\exists(h')$. Notice that when $g \cdot s$ is a prefix of $\pi_1(\rho')$, then $g \cdot s \cdot \sigma_{\mathrm{Agt}}(g \cdot s)$ is also a prefix of $\pi_1(\rho')$. In particular, $\mathrm{Out}(s, \sigma_{\mathrm{Agt}}) = \pi_1(\rho') = \rho$. Notice also, that in every case $\mathrm{Allow}(\mathrm{last}(h), \sigma_{\mathrm{Agt}}(h)) = \mathtt{true}$, so that σ_{Agt} satisfies the move constraint.

We now prove that σ_{Agt} is a trigger strategy for π. Pick a player $A \in \mathrm{Agt}$, a strategy σ'_A for player A, and let $g = \mathrm{Out}(s, \sigma_{\mathrm{Agt}}[A \mapsto \sigma'_A])$. With an infinite play g, we associate an infinite play h in $\mathcal{H}$ in the same way as above. Then player A is a suspect along all the transitions of g, so that she belongs to $L(h)$. Now, as $\sigma^\exists$ is winning, $\pi_1(h) \precsim_A \pi$, which proves that σ_{Agt} is a trigger strategy.

Conversely, assume that σ_{Agt} is a trigger strategy for π whose outcome is ρ, and define the strategy $\sigma^\exists$ by $\sigma^\exists(h) = \sigma_{\mathrm{Agt}}(\pi_1(h))$, this is a correct strategy for `Eve`, since we assume that σ_{Agt} satisfies the move constraint Allow. Notice that the outcome ρ' of $\sigma^\exists$ when `Adam` obeys `Eve` satisfies $\pi_1(\rho') = \rho$.

Let η be an outcome of $\sigma^\exists$ from s, and $A \in L(\eta)$. Then A is a suspect for each transition along $\pi_1(\eta)$, which means that for all i, there is a move m_i^A such that

$$\pi_1(\eta_{=i+1}) = \mathrm{Tab}(\pi_1(\eta_{=i}), \sigma_{\mathrm{Agt}}(\pi_1(\eta_{\leq i}))[A \mapsto m_i^A]).$$

Therefore there is a strategy σ'_A such that $\pi_1(\eta) = \mathrm{Out}(s, \sigma_{\mathrm{Agt}}[A \mapsto \sigma'_A])$. Since σ_{Agt} is a trigger strategy for π, it holds that $\pi_1(\eta) \precsim_A \pi$. As this holds for any $A \in L(\eta)$, $\sigma^\exists$ is winning. □

Theorem 3.4. *Let ρ be an infinite path in $\mathcal{G}$. The following two conditions are equivalent:*

- *there is a path ρ' from (s, Agt) in $\mathcal{H}(\mathcal{G}, \rho, \mathrm{Allow})$,*
  1. *along which* `Adam` *always obeys* `Eve`*;*
  2. *such that $\pi_1(\rho') = \rho$; and*

3. *such that for all index i, there is a strategy $\sigma_\exists^i$ for* `Eve`*, for which any play in $\rho'_{\le i} \cdot \mathrm{Out}(\rho'_{=i}, \sigma_\exists^i)$ is winning for* `Eve`*;*

- *there is a Nash equilibrium σ_{Agt} from s in $\mathcal{G}$ whose outcome is ρ and that satisfies the move constraint* Allow*.*

Proof. Let ρ' be a path in $\mathcal{H}(\mathcal{G}, \pi, \mathrm{Allow})$ and assume it satisfies all three conditions. We define a strategy $\lambda_\exists$ that follows ρ' when `Adam` obeys. Along ρ', this strategy is defined as follows: $\lambda_\exists(\rho'_{\le 2i}) = m_{\mathrm{Agt}}$ such that $\mathrm{Tab}(\pi_1(\rho'_{=i}), m_{\mathrm{Agt}}) = \pi_1(\rho'_{=i+1})$. Such a legal and allowed move must exist since `Adam` obeys `Eve` along ρ' by condition 1 and `Eve` only plays allowed moves. Now, if `Adam` deviates from the obeying strategy (at step i), we make $\lambda_\exists$ follow the strategy $\sigma_\exists^i$ (given by condition 3), which will ensure that the outcome is winning for `Eve`.

The outcomes of $\sigma_\exists$ are then either the path ρ', or a path ρ'' obtained by following a winning strategy after a prefix of ρ'. The path ρ'' is losing for `Adam`, hence for all $A \in L(\rho')$, $\rho'' \precsim_A \rho'$. This proves that $\sigma_\exists$ is a winning strategy. Applying Lemma 3.3, we obtain a strategy profile σ_{Agt} in $\mathcal{G}$ that is a trigger strategy for π, and which satisfies the move constraint Allow. Moreover, the outcome of σ_{Agt} from s is $\pi_1(\rho')$ (using condition 2), so that σ_{Agt} is a Nash equilibrium.

Conversely, the Nash equilibrium is a trigger strategy, and as it satisfies the move constraint Allow, from Lemma 3.3 we get a winning strategy $\sigma_\exists$ in $\mathcal{H}$. The outcome ρ' of $\sigma_\exists$ from s when `Adam` obeys `Eve` is such that $\rho = \pi_1(\rho')$ is the outcome of the Nash equilibrium. Now for all prefix $\rho'_{\le i}$, the strategy $\sigma_\exists^i \colon h \mapsto \sigma_\exists(\rho'_{\le i} \cdot h)$ is such that any play in $\rho'_{\le i} \cdot \mathrm{Out}(\rho'_{=i}, \sigma_i^1)$ is winning for A_1. □

Remark. Assume the preference relations of each player A in $\mathcal{G}$ are prefix-independent, i.e., for all plays ρ and ρ', $\rho \precsim_A \rho'$ if, and only if, for all indices i and j, $\rho_{\ge i} \precsim_A \rho'_{\ge j}$. Then the winning condition of `Eve` is also prefix-independent, and condition 3 just states that ρ' has to stay within the winning region of `Eve`.

Example 7. We depict part of the suspect game for the game of Figure 2.1 with a constraint on move that impose that the player plays the same actions: $\mathrm{Allow}(s, m_{\mathrm{Agt}}) = \mathsf{true}$ if $m_{A_1} = m_{A_2}$. Note that the structure of $\mathcal{H}(\mathcal{G}, \rho, \mathrm{Allow})$ does not depend on ρ. Only the winning condition is affected by the choice of ρ.

3.3 Game Simulation

The notion of suspect is central for our study of Nash equilibria. Based on this concept, we introduce the notion of *game simulation*. We now define this concept of game simulation and prove that it has the expected properties. We will then show that is has the property that when $\mathcal{G}'$ game-simulates $\mathcal{G}$, then

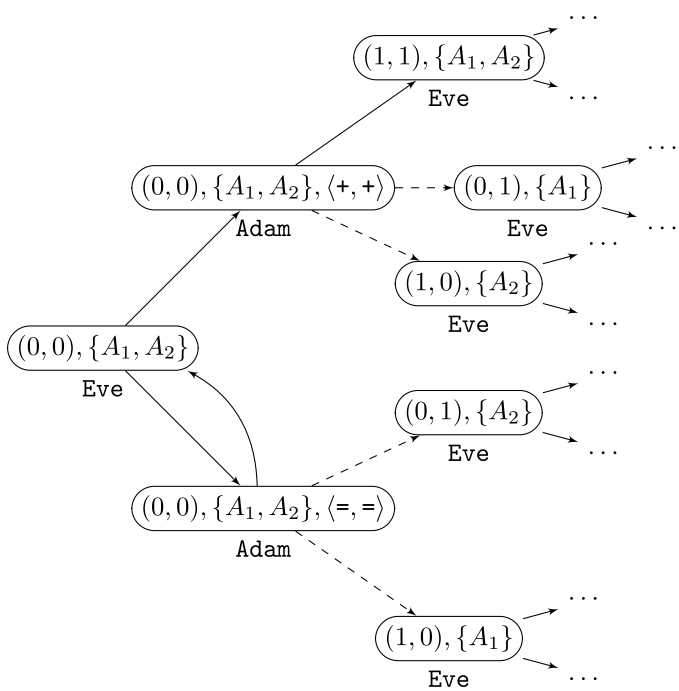


Figure 3.1: A part of a suspect game for our simple model of power control. Dashed transitions correspond to `Adam` not obeying `Eve`. States where the set of suspects is empty, i.e. $P = \varnothing$, are not represented, since these states are always winning for `Eve`, it is never interesting for `Adam` to choose such a state.

a Nash equilibrium in the latter game gives rise to a Nash equilibrium in the former one.

Game simulation. Consider two games $\mathcal{G} = \langle \text{States}, \text{Agt}, \text{Act}, \text{Mov}, \text{Tab}, (\precsim_A)_{A\in\text{Agt}}\rangle$ and $\mathcal{G}' = \langle \text{States}', \text{Agt}, \text{Act}', \text{Mov}', \text{Tab}', (\precsim'_A)_{A\in\text{Agt}}\rangle$ with the same set Agt of players, and a constraint on moves in each game: $\text{Allow}\colon (\text{States}\times\text{Act}^{\text{Agt}}) \to \{\mathsf{true},\mathsf{false}\}$, $\text{Allow}'\colon (\text{States}'\times{\text{Act}'}^{\text{Agt}}) \to \{\mathsf{true},\mathsf{false}\}$. A relation $\lhd \subseteq \text{States}\times\text{States}'$ is a *game simulation* between $\mathcal{G}$ and $\mathcal{G}'$ with respect to Allow and Allow$'$, if $s \lhd s'$ implies that for each allowed move m_{Agt} in $\mathcal{G}$ there exists an allowed move m'_{Agt} in $\mathcal{G}'$ such that

1. $\text{Tab}(s, m_{\text{Agt}}) \lhd \text{Tab}'(s', m'_{\text{Agt}})$, and
2. for each $t' \in \text{States}'$ there exists $t \in \text{States}$ with $t \lhd t'$ and $\text{Susp}((s',t'), m'_{\text{Agt}}) \subseteq \text{Susp}((s,t), m_{\text{Agt}})$.

If $\lhd$ is a game simulation and $s_0 \lhd s'_0$, we say that $\mathcal{G}'$ *game-simulates* (or simply *simulates*) $\mathcal{G}$ with respect to the constraints Allow and Allow$'$. When there are two paths ρ and ρ' such that $\rho_{=i} \lhd \rho'_{=i}$ for all $i \in \mathbb{N}$, we will simply write $\rho \lhd \rho'$.

A game simulation $\lhd$ is *preference-preserving* from $(s_0, s'_0) \in \text{States} \times \text{States}'$ if for all $\rho^1, \rho^2 \in s_0 \cdot \text{States}^\omega$ and $\rho^3, \rho^4 \in s'_0 \cdot \text{States}^\omega$ with $\rho^1 \lhd \rho^3$ and $\rho^2 \lhd \rho^4$, for all $A \in \text{Agt}$ it holds that $\rho^1 \precsim_A \rho^2$ if, and only if, $\rho^3 \precsim_A \rho^4$.

As we show now, Nash equilibria are preserved by game simulation, in the following sense:

Proposition 3.5. *Let $\mathcal{G}$ and $\mathcal{G}'$ be two games involving the same players. Fix two states s_0 and s'_0 in $\mathcal{G}$ and $\mathcal{G}'$ respectively and a constraint on moves in each game:* $\text{Allow} \colon (\text{States} \times \text{Act}^{\text{Agt}}) \to \{\textsf{true}, \textsf{false}\}$*,* $\text{Allow}' \colon (\text{States}' \times \text{Act}'^{\text{Agt}}) \to \{\textsf{true}, \textsf{false}\}$*. Assume that $\lhd$ is a preference-preserving game simulation from (s_0, s'_0) with respect to the constraints* Allow *and* Allow′*. If there exists a Nash equilibrium σ_{Agt} in $\mathcal{G}$ from s_0 which respects the move constraint* Allow*, then there exists a Nash equilibrium σ'_{Agt} in $\mathcal{G}'$ from s'_0 with $\text{Out}_{\mathcal{G}}(s_0, \sigma_{\text{Agt}}) \lhd \text{Out}_{\mathcal{G}'}(s'_0, \sigma'_{\text{Agt}})$, which respects the move constraint* Allow′*.*

Proof. We fix a strategy profile σ_{Agt} in $\mathcal{G}$ which respect constraint Allow and ρ the outcome of σ_{Agt} from s_0. We derive a strategy profile σ'_{Agt} in $\mathcal{G}'$ which respect constraint Allow and its outcome ρ' from s'_0, is such that:

(a) for every $\overline{\rho}' \in \text{Play}_{\mathcal{G}'}(s'_0)$, there exists $\overline{\rho} \in \text{Play}_{\mathcal{G}}(s_0)$ s.t. $\overline{\rho} \lhd \overline{\rho}'$ and $\text{Susp}(\overline{\rho}', \sigma'_{\text{Agt}}) \subseteq \text{Susp}(\overline{\rho}, \sigma_{\text{Agt}})$;

(b) $\rho \lhd \rho'$.

Assume we have done the construction, and that σ_{Agt} is a Nash equilibrium in $\mathcal{G}$. We prove that σ'_{Agt} is a Nash equilibrium in $\mathcal{G}'$. Towards a contradiction, assume that some player A has a strategy $\overline{\sigma}'_A$ in $\mathcal{G}'$ for which she prefers $\overline{\rho}' = \text{Out}_{\mathcal{G}'}(s', \sigma'_{\text{Agt}}[A \mapsto \overline{\sigma}'_A])$ over ρ'. Note that $A \in \text{Susp}(\overline{\rho}', \sigma'_{\text{Agt}})$. Applying (a) above, there exists $\overline{\rho} \in \text{Play}_{\mathcal{G}}(s_0)$ such that $\overline{\rho} \lhd \overline{\rho}'$ and $\text{Susp}(\overline{\rho}', \sigma'_{\text{Agt}}) \subseteq \text{Susp}(\overline{\rho}, \sigma_{\text{Agt}})$. In particular, $A \in \text{Susp}(\overline{\rho}, \sigma_{\text{Agt}})$, and there exists a strategy $\overline{\sigma}_A$ for A such that $\overline{\rho} = \text{Out}_{\mathcal{G}}(s_0, \sigma_{\text{Agt}}[A \mapsto \overline{\sigma}])$. As $\rho \lhd \rho'$ (by (b)) and $\lhd$ is preference-preserving from (s_0, s'_0), $\overline{\rho}$ is preferred by player A over ρ, which contradicts the fact that σ_{Agt} is a Nash equilibrium. Hence, σ'_{Agt} is a Nash equilibrium in $\mathcal{G}'$ from s'_0.

It remains to show how we construct σ'_{Agt} (and ρ'). We first build ρ' inductively, and define σ'_{Agt} along that path.

- initially, we let $\rho'_{=0} = s'_0$. Since $\lhd$ is a game simulation containing (s_0, s'_0), we have $s_0 \lhd s'_0$, and there is an allowed move m'_{Agt} associated with $\sigma_{\text{Agt}}(s_0)$ compels with the definition of a game simulation. Then $\rho_{=0} \lhd \rho'_{=0}$, and $\text{Susp}(\rho'_{=0}, \sigma'_{\text{Agt}}(\rho'_{=0})) \subseteq \text{Susp}(\rho_{=0}, \sigma_{\text{Agt}}(\rho_{=0}))$.
- assume we have built $\rho'_{\leq i}$ and σ'_{Agt} on all the prefixes of $\rho'_{\leq i}$, and that they are such that $\rho_{\leq i} \lhd \rho'_{\leq i}$ and $\text{Susp}(\rho'_{\leq i}, \sigma'_{\text{Agt}}) \subseteq \text{Susp}(\rho_{\leq i}, \sigma_{\text{Agt}})$ (notice that $\text{Susp}(\rho'_{\leq i}, \sigma'_{\text{Agt}})$ only depends on the value of σ'_{Agt} on all the prefixes of $\rho_{\leq i}$). In particular, we have $\rho_{=i} \lhd \rho'_{=i}$, so that with the move $\sigma_{\text{Agt}}(\rho_{\leq i})$, we can associate an allowed move m'_{Agt} (to which we

set $\sigma'_{\mathrm{Agt}}(\rho'_{\leq i})$) satisfying both conditions of the definition of a game simulation. This defines $\rho'_{=i+1}$ in such a way that $\rho_{\leq i+1} \triangleleft \rho'_{\leq i+1}$; moreover, $\mathrm{Susp}(\rho'_{\leq i+1}, \sigma'_{\mathrm{Agt}}) = \mathrm{Susp}(\rho'_{\leq i}, \sigma'_{\mathrm{Agt}}) \cap \mathrm{Susp}((\rho'_{=i}, \rho'_{=i+1}), m'_{\mathrm{Agt}})$ is indeed a subset of $\mathrm{Susp}(\rho_{\leq i+1}, \sigma_{\mathrm{Agt}})$.

It remains to define σ'_{Agt} outside its outcome ρ'. Notice that, for our purposes, it suffices to define σ'_{Agt} on histories starting from s'_0. We again proceed by induction on the length of the histories, defining σ'_{Agt} in order to satisfy (a) on prefixes of plays of $\mathcal{G}'$ from s'_0. At each step, we also make sure that for every $h' \in \mathrm{Hist}_{\mathcal{G}'}(s'_0)$, there exists $h \in \mathrm{Hist}_{\mathcal{G}}(s)$ such that $h \triangleleft h'$, $\mathrm{Susp}(h', \sigma'_{\mathrm{Agt}}) \subseteq \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})$, and $\sigma_{\mathrm{Agt}}(h)$ and $\sigma'_{\mathrm{Agt}}(h')$ satisfy the conditions of the definition of a game simulation in the last states of h and h', resp.

As we only consider histories from s'_0, the case of histories of length zero was already handled. Assume we have defined σ'_{Agt} for histories h' of length i, and fix a new history $h' \cdot t' \in \mathrm{Hist}_{\mathcal{G}'}(s'_0)$ of length $i+1$ (that is not a prefix of ρ). By induction hypothesis, there is $h \in \mathrm{Hist}_{\mathcal{G}}(s_0)$ such that $h \triangleleft h'$, and $\mathrm{Susp}(h', \sigma'_{\mathrm{Agt}}) \subseteq \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})$, and $\sigma_{\mathrm{Agt}}(h)$ and $\sigma_{\mathrm{Agt}}(h')$ satisfy the required properties. In particular, with t', we can associate t s.t. $t \triangleleft t'$ and $\mathrm{Susp}((\mathrm{last}(h'), t'), \sigma'_{\mathrm{Agt}}(h')) \subseteq \mathrm{Susp}((\mathrm{last}(h), t), \sigma_{\mathrm{Agt}}(h))$. Then $(h \cdot t) \triangleleft (h' \cdot t')$. Since $t \triangleleft t'$, there is an allowed move m'_{Agt} associated with $\sigma_{\mathrm{Agt}}(h \cdot t)$ and satisfying the conditions of the definition of a game simulation. Letting $\sigma'_{\mathrm{Agt}}(h' \cdot t') = m'_{\mathrm{Agt}}$, we fulfill all the requirements of our induction hypothesis.

We now need to lift the property from histories to infinite paths. Consider a play $\overline{\rho}' \in \mathrm{Play}_{\mathcal{G}'}(s'_0)$, we will construct a corresponding play $\overline{\rho}$ in $\mathcal{G}$. Set $\overline{\rho}_0 = s_0$. If $\overline{\rho}$ has been defined up to index i and $\overline{\rho}_i \triangleleft \overline{\rho}'_i$ (this is true for $i = 0$), thanks to the way σ'_{Agt} is constructed, $\sigma_{\mathrm{Agt}}(\overline{\rho}_{\leq i})$ and $\sigma'_{\mathrm{Agt}}(\overline{\rho}'_{\leq i})$ satisfy the conditions of the definition of a game simulation in $\overline{\rho}_{\leq i}$ and $\overline{\rho}'_i$, respectively. We then pick $\overline{\rho}_{i+1}$ such that $\overline{\rho}_{i+1} \triangleleft \overline{\rho}'_{i+1}$ and $\mathrm{Susp}((\overline{\rho}_i, \overline{\rho}_{i+1}), \sigma_{\mathrm{Agt}}(\overline{\rho}_i)) \subseteq \mathrm{Susp}((\overline{\rho}'_i, \overline{\rho}'_{i+1}), \sigma'_{\mathrm{Agt}}(\overline{\rho}'_i))$. This being true at each step, the path $\overline{\rho}$ that is obtained, is such that $\overline{\rho} \triangleleft \overline{\rho}'$ and $\mathrm{Susp}(\overline{\rho}', \sigma'_{\mathrm{Agt}}) \subseteq \mathrm{Susp}(\overline{\rho}, \sigma_{\mathrm{Agt}})$. Which is the desired property. $\square$

Chapter 4

Single objectives

In this chapter, we restrict our study to finite concurrent game. We aim at precisely describing the complexity of the Nash equilibria problems for simple preference relations, defined by a single (ω-regular) objectives. These preferences are purely qualitative since for one player, a play is either winning or losing. We will see how to use the suspect game construction in order to solve the existence problem with constrained moves in this context.

4.1 Specification of the Objectives

We fix a game $\mathcal{G} = \langle \mathrm{States}, \mathrm{Agt}, \mathrm{Act}, \mathrm{Mov}, \mathrm{Tab}, (\precsim_A)_{A \in \mathrm{Agt}} \rangle$ for the rest of the section. Each preference relation $\precsim_A$ will be given as a single objective. An *objective* (or *winning condition*) is an arbitrary set of plays. If Ω_A is the objective for player A, the preference relation $\precsim_A$ is defined by: $\rho \precsim_A \rho'$ if and only if $\rho' \in \Omega_A$ (we say that ρ' is winning for A) or $\rho \notin \Omega_A$ (we say that ρ is losing for A). An objective Ω can be specified in various ways. We focus on the following standard ones:

- A *reachability objective* is given by a target set $T \subseteq \mathrm{States}$, the corresponding set of plays is $\Omega = \{\rho \in \mathrm{Play} \mid \mathrm{Occ}(\rho) \cap T \neq \varnothing\}$;
- A *safety objective* is given by a target set $T \subseteq \mathrm{States}$, the corresponding set of plays is $\Omega = \{\rho \in \mathrm{Play} \mid \mathrm{Occ}(\rho) \cap T = \varnothing\}$;
- A *Büchi objective* is given by a target set $T \subseteq \mathrm{States}$, the corresponding set of plays is $\Omega = \{\rho \in \mathrm{Play} \mid \mathrm{Inf}(\rho) \cap T \neq \varnothing\}$;
- A *co-Büchi objective* is given by a target set $T \subseteq \mathrm{States}$, the corresponding set of plays is $\Omega = \{\rho \in \mathrm{Play} \mid \mathrm{Inf}(\rho) \cap T = \varnothing\}$;
- A *parity objective* is given by a priority function $p\colon \mathrm{States} \mapsto [\![0, d]\!]$ with $d \in \mathbb{N}$, the corresponding set of plays is $\Omega = \{\rho \in \mathrm{Play} \mid \min(\mathrm{Inf}(p(\rho))) \text{ is even}\}$;
- A *Streett objective* is given by a tuple $(Q_i, R_i)_{i \in [\![1,k]\!]}$, the corresponding set of plays is $\Omega = \{\rho \in \mathrm{Play} \mid \forall i.\ \mathrm{Inf}(\rho) \cap Q_i \neq \varnothing \Rightarrow \mathrm{Inf}(\rho) \cap R_i \neq \varnothing\}$;

- A *Rabin objective* is given by a tuple $(Q_i, R_i)_{i \in [\![1,k]\!]}$, the corresponding set of plays is $\Omega = \{\rho \in \text{Play} \mid \exists i.\ \text{Inf}(\rho) \cap Q_i \neq \varnothing \wedge \text{Inf}(\rho) \cap R_i = \varnothing\}$;
- A *Muller objective* is given by a coloring function $c\colon \text{States} \mapsto C$, and a set $\mathcal{F} \subseteq 2^C$, the corresponding set of plays is $\Omega = \{\rho \in \text{Play} \mid \text{Inf}(c(\rho)) \in \mathcal{F}\}$;
- A *circuit objective* is given by a Boolean circuit C with the set States as input nodes and one output node. A play ρ is winning if and only if C evaluates to true when states in $\text{Inf}(\rho)$ are set to true and all other states are set to false. Figure 4.1 displays an example of a circuit for the game of Figure 2.4;

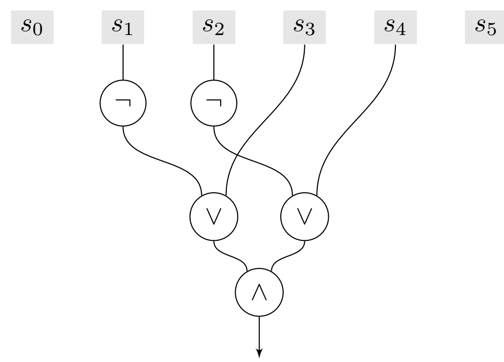


Figure 4.1: Example of a Boolean circuit defining a winning condition for the arena presented in Example 4. The winning condition defined is that if s_1 appears infinitely often then s_3 also appears infinitely often, and if s_2 appears infinitely often then s_4 also does.

- A *deterministic Büchi automaton objective* is given by a *deterministic Büchi automaton* $\langle Q, \Sigma, \delta, q_0, R\rangle$, with $\Sigma = \text{States}$. Then $\Omega = \text{L}(\mathcal{A})$.
- A *deterministic Rabin automaton objective* is given by a deterministic Rabin automaton $\langle Q, \Sigma, \delta, q_0, (E_i, F_i)_{i \in [\![1,k]\!]}\rangle$, with $\Sigma = \text{States}$. Then $\Omega = \text{L}(\mathcal{A})$.

The value problem has standard solutions in game theory; they are given in Table 4.1. In this section we solve the existence problem with constrained moves in all the cases, and the results are summarized in the second column of Table 4.1.

Streett and Muller objectives are not explicitly mentioned in the rest of the section. The complexity of their respective (constrained) existence problems, which is given in Table 4.1, can easily be inferred from other ones. The $\mathsf{P}^{\mathsf{NP}}_{\parallel}$-hardness for the existence problem with Streett objectives follows from the corresponding hardness for parity objectives (parity objectives can be encoded efficiently as Streett objectives). Hardness for the existence problem in Muller games, is deduced from hardness of the value problem, applying Lemma 2.4. For both objectives, membership in PSPACE follows from PSPACE membership for objectives given as Boolean circuits, since they can efficiently be encoded as Boolean circuits.

Table 4.1: Summary of the complexities for single objectives

Objective	Value	(Constrained) Existence
Reachability	P-c [28]	NP-c (Sect. 4.2)
Safety	P-c [28]	NP-c (Sect. 4.4)
Büchi	P-c [28]	P-c (Sect. 4.3)
co-Büchi	P-c [28]	NP-c (Sect. 4.5)
Parity	UP$\cap$ co-UP[34]	$\mathsf{P}_{\parallel}^{\mathsf{NP}}$-c (Sect. 4.7)
Streett	co-NP-c [23]	$\mathsf{P}_{\parallel}^{\mathsf{NP}}$-h and in PSPACE
Rabin	NP-c [23]	$\mathsf{P}_{\parallel}^{\mathsf{NP}}$-c (Sect. 4.7)
Muller	PSPACE-c [20]	PSPACE-c
Circuit	PSPACE-c [20]	PSPACE-c (Sect. 4.6)
Det. Büchi Automata	P-c	PSPACE-h (Sect. 4.8) and in EXPTIME
Det. Rabin Automata	NP-c	PSPACE-h and in EXPTIME(Sect. 4.8)

An important simplification. We prove all those results using the suspect-game construction. It is first interesting to notice that given a constraint Allow and two plays π and π' the games $\mathcal{H}(\mathcal{G}, \pi, \text{Allow})$ and $\mathcal{H}(\mathcal{G}, \pi', \text{Allow})$ only differ in their winning conditions. In particular, the structure of the game only depends on $\mathcal{G}$ and Allow, and has polynomial size (see Lemma 3.2). We denote it with $\mathcal{H}(\mathcal{G}, \text{Allow})$. Moreover, as each relation $\precsim_A$ is given by a single objective Ω_A, the winning condition for Eve in $\mathcal{H}(\mathcal{G}, \pi, \text{Allow})$ rewrites as: for every $A \in L(\rho) \cap \text{Los}(\pi)$, $\pi_1(\rho)$ is losing (in $\mathcal{G}$) for player A, where $\text{Los}(\pi)$ is the set of players losing along π in $\mathcal{G}$. This winning condition only depends on $\text{Los}(\pi)$ (not on the precise value of play π). Therefore in this section, the suspect game is denoted with $\mathcal{H}(\mathcal{G}, L, \text{Allow})$, where $L \subseteq \text{Agt}$, and Eve wins play ρ if, for every $A \in L(\rho) \cap L$, A loses along $\pi_1(\rho)$ in $\mathcal{G}$. In many cases we will be able to simplify this winning condition, and to obtain simple algorithms for the corresponding problems.

4.2 Reachability Objectives

It is known that the value problem for a reachability winning condition is P-complete [28]. We will design an NP algorithm for solving the existence problem with constrained moves, and will end this section with the NP-hardness of the (constrained) existence problem.

Reduction to a safety game. We assume the preference relation of each player $A \in \text{Agt}$ is a single reachability objective which is given by the target set T_A. Given $L \subseteq \text{Agt}$, in the suspect game $\mathcal{H}(\mathcal{G}, L, \text{Allow})$, we show that the objective of Eve reduces to a safety objective. We define the safety objective Ω_L in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$ by the target set $T_L = \{(s, P) \mid \exists A \in P \cap L.\ s \in T_A\}$.

Lemma 4.1. *Eve has a winning strategy in game $\mathcal{H}(\mathcal{G}, L, \text{Allow})$ if, and only if, Eve has a winning strategy in game $\mathcal{H}(\mathcal{G}, \text{Allow})$ with safety objective Ω_L.*

Proof. We first show that any play in Ω_L is winning in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$. Let $\rho \in \Omega_L$, and let $A \in L(\rho) \cap L$. Towards a contradiction assume that $\text{Occ}(\pi_1(\rho)) \cap T_A \neq \varnothing$: there is a state (s, P) along ρ with $s \in T_A$. Obviously $L(\rho) \subseteq P$, which implies that $A \in P \cap L$. This contradicts the fact that $\rho \notin \Omega_L$. We have shown so far that any winning strategy for Eve in $\mathcal{H}(\mathcal{G}, \text{Allow})$ with safety objective Ω_L is a winning strategy for Eve in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$.

Now assume that Eve has no winning strategy in game $\mathcal{H}(\mathcal{G}, \text{Allow})$ with safety objective Ω_L. Turn-based games with safety objectives being determined, Adam has a strategy $\sigma_\forall$ which ensures that no outcome of $\sigma_\forall$ is in Ω_L. If $\rho \in \text{Out}(\sigma_\forall)$, there is a state (s, P) along ρ such that there is $A \in P \cap L$ with $s \in T_A$. We now modify the strategy of Adam such that as soon as such a state is reached we switch from $\sigma_\forall$ to the strategy that always obeys Eve. This ensures that in every outcome ρ' of the new strategy, we reach a state (s, P) such that there is $A \in P \cap L$ with $s \in T_A$, and $L(\rho') = P$. This Adam's strategy thus makes Eve lose the game $\mathcal{H}(\mathcal{G}, L, \text{Allow})$, and Eve has no winning strategy in game $\mathcal{H}(\mathcal{G}, L, \text{Allow})$. $\square$

Algorithm. The algorithm for solving the existence problem with constrained moves in a game where each player has a single reachability objective relies on Theorem 3.4 and Lemma 2.2, and on the above analysis:

- *(i)* guess a lasso-shaped play $\rho = \tau_1 \cdot \tau_2^\omega$ (with $|\tau_i| \leq 2|\text{States}|^2$) in $\mathcal{H}(\mathcal{G}, \text{Allow})$, such that Adam obeys Eve along ρ, and $\pi = \pi_1(\rho)$ satisfies the constraint on the payoff;
- *(ii)* compute $W(\mathcal{G}, \text{Los}(\pi), \text{Allow})$, the set of winning states for Eve in suspect game $\mathcal{H}(\mathcal{G}, \text{Los}(\pi), \text{Allow})$, where $\text{Los}(\pi)$ is the set of losing players along π;
- *(iii)* check that ρ stays in $W(\mathcal{G}, \text{Los}(\pi), \text{Allow})$.

First notice that this algorithm runs in NP: the witness ρ guessed in step i has size polynomial; the suspect game $\mathcal{H}(\mathcal{G}, \text{Los}(\pi), \text{Allow})$ has also polynomial size (Lemma 3.2); Step ii can be done in polynomial time using a standard attractor computation [28] as the game under analysis is equivalent to a safety game; finally step iii can obviously be performed in polynomial time.

Step i ensures that conditions 1 and 2 of Theorem 3.4 hold for ρ and step iii ensures condition 3. Correctness of the algorithm then follows from Theorem 3.4 and Lemma 2.2.

Hardness. We prove NP-hardness of the existence problem with constrained outcomes by encoding an instance of 3SAT as follows. We assume set of atomic propositions $\mathsf{AP} = \{p_1, \ldots, p_h\}$, and we let $\phi = \bigwedge_{i=1}^{k} c_i$ where $c_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$ where $\ell_{i,j} \in \{p, \neg p \mid p \in \mathsf{AP}\}$. We build the turn-based game $\mathcal{G}_\phi$ with $k+1$ players $\mathrm{Agt} = \{A, C_1, \ldots, C_k\}$ as follows: for every $1 \leq j \leq h$, player A chooses to visit either location p_j or location $\neg p_j$. Location p_j is winning for the clause players C_m if, and only if, p_j is one of the literals in c_m, and similarly location $\neg p_j$ is winning for C_m if, and only if, $\neg p_j$ is one of the literals of c_m. The construction is illustrated in Example 8. Now, it is easy to check that this game has a Nash equilibrium winning for all players $(C_i)_{1 \leq i \leq k}$ if, and only if, ϕ is satisfiable.

We prove hardness for the existence problem by using the transformation described in Section 2.4.4 once for each player. We define the game $\mathcal{G}_0$ similar to $\mathcal{G}$ but with an extra player C_{k+1} who does not control any state for now. For $i \in [\![1, k]\!]$, we define $\mathcal{G}_i = E(\mathcal{G}_{i-1}, C_i, C_{k+1}, \rho)$, where ρ is a winning path for C_i. The preference relation can be expressed in any $\mathcal{G}_i$ by a reachability condition, by giving to C_{k+1} a target which is the initial state of $\mathcal{G}$. According to Lemma 2.5 there is a Nash equilibrium in $\mathcal{G}_i$ if, and only if, there is one in $\mathcal{G}_{i-1}$ where A_i wins. Therefore there is a Nash equilibrium in $\mathcal{G}_k$ if, and only if, ϕ is satisfiable. This entails NP-hardness of the existence problem.

Example 8. As an example of the construction, consider the formula $\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_3)$. The arena of the game is represented in Figure 4.2. The target set for the reachability objectives are defined by $T_{C_1} = \{x_1, x_2, \neg x_3\}$ and $T_{C_2} = \{\neg x_1, \neg x_3\}$. The formula is satisfiable and therefore there is a Nash equilibrium that makes both C_1 and C_2 win. One such strategy consists in choosing $\neg x_1$ and then x_2.

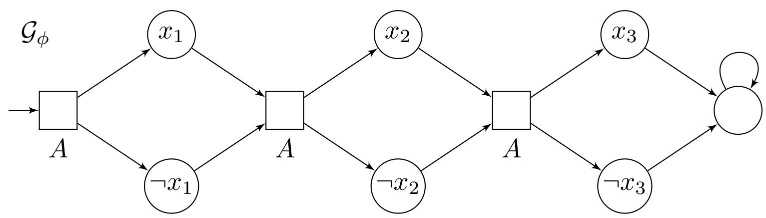


Figure 4.2: Example of a reachability game for the reduction of 3SAT

4.3 Büchi Objectives

The P-completeness of the value problem for Büchi objectives is folk result [28]. In this section we design a polynomial-time algorithm for solving the existence problem with constrained moves for Büchi objectives. The P-hardness of the (constrained) existence problem will then be inferred from the P-hardness of the value problem, applying Lemmas 2.3 and 2.4.

Reduction to a co-Büchi game. We assume the preference relation of each player $A \in \mathrm{Agt}$ is a single Büchi objective given by target set T_A. Given $L \subseteq \mathrm{Agt}$, in the suspect game $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$, we show that the objective of `Eve` is equivalent to a single co-Büchi objective. We define the co-Büchi objective Ω_L in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$ given by the target set $T_L = \{(s, P) \mid \exists A \in P \cap L.\ s \in T_A\}$. Notice that the target set is defined in the same way as for reachability objectives.

Lemma 4.2. *A play ρ is winning for* `Eve` *in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$ if, and only if, $\rho \in \Omega_L$.*

Proof. Assume ρ is winning for `Eve` in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$. For every $A \in L(\rho) \cap L$, $\mathrm{Inf}(\pi_1(\rho)) \cap T_A = \varnothing$. Towards a contradiction, assume that $\mathrm{Inf}(\rho) \cap T_L \neq \varnothing$. There exists (s, P) such that there is $A \in P \cap L$ with $s \in T_A$, which appears infinitely often along ρ. In particular, $P = L(\rho)$ (otherwise it would not appear infinitely often along ρ). Hence, we have found $A \in L(\rho) \cap L$ such that $\mathrm{Inf}(\pi_1(\rho)) \cap T_A \neq \varnothing$, which is a contradiction. Therefore, $\rho \in \Omega_L$.

Assume $\rho \in \Omega_L$: for every (s, P) such that there exists $A \in P \cap L$ with $s \in T_A$, (s, P) appears finitely often along ρ. Let $A \in L(\rho) \cap L$, and assume towards a contradiction that there is $s \in T_A$ such that s appears infinitely often along $\pi_1(\rho)$. This means that $(s, L(\rho))$ appears infinitely often along ρ, which contradicts the above condition. Therefore, ρ is winning for `Eve` in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$. $\square$

Algorithm. As for reachability objectives, the winning region for `Eve` in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$ can be computed in polynomial time. An `NP` algorithm similar to the one for reachability objectives can therefore be inferred. However we can do better than guessing a path π by looking at the strongly connected components of the game. This will yield a polynomial-time algorithm.

We first characterize the 'good' paths in $\mathcal{H}(\mathcal{G}, \mathrm{Allow})$ in terms of the strongly-connected components they define. This characterization can be made in a more general context than single Büchi objectives for each player. We therefore assume that the preference relation of each player only depends on the set of states that are visited infinitely often, and we fix constraints given as infinite paths u^A and w^A for each player A, and an initial state s in $\mathcal{G}$. For each $K \subseteq \mathrm{States}$, we write $v^A(K)$ for the equivalence class of all paths π that visits infinitely often exactly K, i.e. $\mathrm{Inf}(\pi) = K$. We also write $v(K) = (v^A(K))_{A \in \mathrm{Agt}}$. We look for a transition system $\langle K, E\rangle$, with $K \subseteq \mathrm{States}$ and $E \subseteq K \times K$, for which the following properties hold:

(1) $u^A \lesssim_A v^A(K) \lesssim_A w^A$ for all $A \in \mathrm{Agt}$;

(2) $\langle K, E\rangle$ is strongly connected;

(3) $\forall k \in K.\ (k, \mathrm{Agt}) \in W(\mathcal{G}, v(K), \mathrm{Allow})$;

(4) $\forall (k,k') \in E.\ \exists (k, \mathrm{Agt}, m_{\mathrm{Agt}}) \in W(\mathcal{G}, v(K), \mathrm{Allow}).\ \mathrm{Tab}(k, m_{\mathrm{Agt}}) = k'$;

(5) $(K \times \{\mathrm{Agt}\})$ is reachable from (s, Agt) in $W(\mathcal{G}, v(K), \mathrm{Allow})$;

where $W(\mathcal{G}, v(K), \mathrm{Allow})$ denotes the winning region of `Eve` in the suspect game $\mathcal{H}(\mathcal{G}, v(K), \mathrm{Allow})$.[1]

Lemma 4.3. *Under the assumption that the preference relation of each player only depends on the set of states that are visited infinitely often, there is a transition system $\langle K, E\rangle$ satisfying conditions 1–5 if, and only if, there is a path ρ from (s, Agt) in $\mathcal{H}(\mathcal{G}, v(K), \mathrm{Allow})$ that never gets out of $W(\mathcal{G}, v(K), \mathrm{Allow})$, along which* `Adam` *always obeys* `Eve`*, $u^A \lesssim_A v^A(K) \lesssim_A w^A$ for all $A \in \mathrm{Agt}$, and $\pi_1(\mathrm{Inf}(\rho) \cap V_\exists) = K$ (which implies that $\rho \in v^A(K)$ for all A).*

Proof. The first implication is shown by building a path in $W(\mathcal{G}, v(K), \mathrm{Allow})$ that successively visits all the states in $K \times \{\mathrm{Agt}\}$ forever. Thanks to 5, 2 and 4 (and the fact that `Adam` obeys `Eve`), such a path exists, and from 3 and 4, this path remains in the winning region. From 1, we have the condition on the preferences. Conversely, consider such a path ρ, and let $K = \pi_1(\mathrm{Inf}(\rho) \cap V_\exists)$ and $E = \{(k,k') \in K^2 \mid \exists (k, \mathrm{Agt}, m_{\mathrm{Agt}}) \in \mathrm{Inf}(\rho).\ \mathrm{Tab}(k, m_{\mathrm{Agt}}) = k'\}$. Condition 5 clearly holds. Conditions 1, 3 and 4 are easy consequences of the hypotheses and construction. We prove that $\langle K, E\rangle$ is strongly connected. First, since `Adam` obeys `Eve` and ρ starts in (k, Agt), we have $L(\rho) = \mathrm{Agt}$. Now, take any two states k and k' in K: then ρ visits (k, Agt) and (k', Agt) infinitely often, and there is a subpath of ρ between those two states, all of which states appear infinitely often along ρ. Such a subpath gives rise to a path between k and k', as required. ☐

As a consequence, if $\langle K, E\rangle$ satisfies the five previous conditions, by Theorem 3.4, there is a Nash equilibrium whose outcome lies between the bounds u^A and w^A. Our aim is to compute in polynomial time all maximal pairs $\langle K, E\rangle$ that satisfy the conditions. We first need to compute the set $W(\mathcal{G}, v(K), \mathrm{Allow})$, given $v(K)$. In our particular case where each player has a single Büchi objective, this is done by computing the winning region of a co-Büchi game thanks to Lemma 4.2.

Now, we define a recursive function SSG (standing for "solve sub-game"), working on transition systems:

- if $K \times \{\mathrm{Agt}\} \subseteq W(\mathcal{G}, v(K), \mathrm{Allow})$, and for all $(k,k') \in E$, there is a $(k, \mathrm{Agt}, m_{\mathrm{Agt}})$ in $W(\mathcal{G}, v(K), \mathrm{Allow})$ s.t. $\mathrm{Tab}(k, m_{\mathrm{Agt}}) = k'$, and finally $\langle K, E\rangle$ is strongly connected, then we set $\mathsf{SSG}(\langle K, E\rangle) = \{\langle K, E\rangle\}$;
- otherwise, we let

$$\mathsf{SSG}(\langle K, E\rangle) = \bigcup_{\langle K', E'\rangle \in \mathrm{SCC}(\langle K, E\rangle)} \mathsf{SSG}(T(\langle K', E'\rangle))$$

[1] Formally the suspect game has been defined with a play as reference, and not a equivalence class. However, if π and π' are equivalent, the games $\mathcal{H}(\mathcal{G}, \pi, \mathrm{Allow})$ and $\mathcal{H}(\mathcal{G}, \pi', \mathrm{Allow})$ are identical.

where $\mathrm{SCC}(\langle K, E\rangle)$ is the set of strongly connected components of $\langle K, E\rangle$ (which can be computed in linear time), and where $T(\langle K', E'\rangle)$ is the transition system whose set of states is $\{k \in K' \mid (k, \mathrm{Agt}) \in W(\mathcal{G}, v(K'), \mathrm{Allow})\}$ and whose set of edges is

$$\{(k, k') \in E' \mid \exists (k, \mathrm{Agt}, m_{\mathrm{Agt}}) \in W(\mathcal{G}, v(K'), \mathrm{Allow}).\ \mathrm{Tab}(k, m_{\mathrm{Agt}}) = k'\}.$$

Notice that this set of edges is never empty, but $T(\langle K', E'\rangle)$ might not be strongly connected anymore, so that this is really a recursive definition.

We have to ensure that the outcome does not exceed w^A. For that we need to assume that given a constraint w^A, we are able to construct (in polynomial time) a set of states S^A, such that $\mathrm{Inf}(\rho) \subseteq S^A \Leftrightarrow \rho \precsim_A w^A$. In our particular case of a single objective for each player, this is simply done by removing the target set of A from States, if this player has to be losing (that is, if w^A does not satisfy the Büchi objective). We then define

$$\mathsf{Sol} = \mathsf{SSG}\big(\langle \bigcap_{A \in \mathrm{Agt}} S^A, \mathrm{Edg}'\rangle\big) \cap \big\{\langle K, E\rangle \mid \forall A \in \mathrm{Agt}.\ u^A \lesssim v^A(K)\big\}$$

where Edg' restricts Edg to $\bigcap_{A \in \mathrm{Agt}} S^A$.

To prove the correctness of this construction we need to assume some monotonicity of the preference relation, informally, seeing more states infinitely often is not bad for the players. This is indeed the case for the particular case of single Büchi objectives that we consider in this section.

Lemma 4.4. *Assume the preference relation for each player A is such that it only depends on the set of states visited infinitely often, $K \subseteq K' \Rightarrow v^A(K) \lesssim_A v^A(K')$ and $\mathrm{Inf}(\rho) \subseteq S^A \Leftrightarrow \rho \precsim_A w^A$. Then, if $\langle K, E\rangle \in \mathsf{Sol}$ then it satisfies conditions 1 to 4. Conversely, if $\langle K, E\rangle$ satisfies conditions 1 to 4, then there exists $\langle K', E'\rangle \in \mathsf{Sol}$ such that $\langle K, E\rangle \subseteq \langle K', E'\rangle$.*

Proof. Let $\langle K, E\rangle \in \mathsf{Sol}$. By definition of SSG, all (k, Agt) for $k \in K$ are in $W(\mathcal{G}, v(K), \mathrm{Allow})$, and for all $(k, k') \in E$, there is a state $(k, \mathrm{Agt}, m_{\mathrm{Agt}})$ in $W(\mathcal{G}, v(K), \mathrm{Allow})$ such that $\mathrm{Tab}(k, m_{\mathrm{Agt}}) = k'$, and $\langle K, E\rangle$ is strongly connected. Also, for all A, $u^A \lesssim v^A(K)$ because $\mathsf{Sol} \subseteq \{\langle K, E\rangle \mid u^A \lesssim v^A(K)\}$. Finally, for any $A \in \mathrm{Agt}$, $v^A(K) \lesssim w^A$ because the set K is included in S^A.

Conversely, assume that $\langle K, E\rangle$ satisfies the conditions. We show that if $\langle K, E\rangle \subseteq \langle K', E'\rangle$ then there is $\langle K'', E''\rangle$ in $\mathsf{SSG}(\langle K', E'\rangle)$ such that $\langle K, E\rangle \subseteq \langle K'', E''\rangle$. The proof is by induction on the size of $\langle K', E'\rangle$.

The basic case is when $\langle K', E'\rangle$ satisfies conditions 2, 3, and 4. Under these conditions $\mathsf{SSG}(\langle K', E'\rangle) = \{\langle K', E'\rangle\}$, and by letting $\langle K'', E''\rangle = \langle K', E'\rangle$ we get the expected result.

We now analyze the other case. There is a strongly connected component of $\langle K', E'\rangle$, say $\langle K'', E''\rangle$, which contains $\langle K, E\rangle$, because $\langle K, E\rangle$ satisfies condition 2. We have $v^A(K) \lesssim v^A(K'')$ (because $K \subseteq K''$) for every A, and thus $W(\mathcal{G}, v(K), \mathrm{Allow}) \subseteq W(\mathcal{G}, v(K''), \mathrm{Allow})$. This ensures that $T(\langle K'', E''\rangle)$ contains $\langle K, E\rangle$ as a subgraph. Since $\langle K'', E''\rangle$ is a subgraph of $\langle K', E'\rangle$, the graph

$T(\langle K'', E''\rangle)$ also is. We show that they are not equal, so that we can apply the induction hypothesis to $T(\langle K'', E''\rangle)$. For this, we exploit the fact that $\langle K', E'\rangle$ does not satisfy one of conditions 2 to 4:

- first, if $\langle K', E'\rangle$ is not strongly connected while $\langle K'', E''\rangle$ is, they cannot be equal;
- if there is some $k \in K'$ such that (k, Agt) is not in $W(\mathcal{G}, v(K'), \mathrm{Allow})$, then k is not a vertex of $T(\langle K'', E''\rangle)$;
- if there some edge (k, k') in E' such that there is no state $(k, \mathrm{Agt}, m_{\mathrm{Agt}})$ in $W(\mathcal{G}, v(K'), \mathrm{Allow})$ such that $\mathrm{Tab}(k, m_{\mathrm{Agt}}) = k'$, then the edge (k, k') is not in $T(\langle K'', E''\rangle)$.

We then apply the induction hypothesis to $T(\langle K'', E''\rangle)$, and get the expected result. Now, because of condition 1, $u^A \lesssim v^A(K) \lesssim w^A$. Hence, due to the previous analysis, there exists $\langle K', E'\rangle \in \mathsf{SSG}\left(\langle \bigcap_{A\in\mathrm{Agt}} S^A, \mathrm{Edg}'\rangle\right)$ such that $\langle K, E\rangle \subseteq \langle K', E'\rangle$. This concludes the proof of the lemma. $\square$

This lemma implies in particular the equivalence between the existence of a Nash equilibrium and the non-emptyness of Sol.

Lemma 4.5. *The set Sol can be computed in polynomial time.*

Proof. Each recursive call to SSG applies to a decomposition in strongly connected components of the current transition system under consideration. Hence the number of recursive calls is bounded by $|\mathrm{States}|^2$. Computing the decomposition in SCCs can be done in linear time. Furthermore, thanks to Lemma 4.2, $W(\mathcal{G}, v(K), \mathrm{Allow})$ can be computed in polynomial time. S^A is obtained by removing the target of the losers (for w^A) from States. Hence globally we can compute Sol in polynomial time. $\square$

To conclude the algorithm, we need to check that condition 5 holds for one of the solutions $\langle K, E\rangle$ in Sol. It can be done in polynomial time by looking for a path in the winning region of $\mathtt{Eve}$ in $\mathcal{H}(\mathcal{G}, v(K), \mathrm{Allow})$ that reaches $K \times \{\mathrm{Agt}\}$ from (s, Agt). The correctness of the algorithm is ensured by the fact that if some $\langle K, E\rangle$ satisfies the five conditions, there is a $\langle K', E'\rangle$ in Sol with $K \subseteq K'$ and $E \subseteq E'$. Since $K \subseteq K'$ implies $v^A(K) \lesssim_A v^A(K')$, the winning region of $\mathtt{Eve}$ in $\mathcal{H}(\mathcal{G}, v(K'), \mathrm{Allow})$ is larger than that $\mathcal{H}(\mathcal{G}, v(K'), \mathrm{Allow})$, which implies that the path from (s, Agt) to $K \times \{\mathrm{Agt}\}$ is also a path from (s, Agt) to $K' \times \{\mathrm{Agt}\}$. Hence, $\langle K', E'\rangle$ also satisfies condition 5, and therefore the five expected conditions.

Hardness. We recall a possible proof of P-hardness of the value problem, from which we will infer the other lower bounds. The circuit value problem can be easily encoded into a deterministic turn-based game with Büchi objectives: a circuit (which we assume w.l.o.g. has only AND- and OR-gates) is transformed into a two-player turn-based game, where one player controls the AND-gates and the other player controls the OR-gates. We add self-loops on the leaves.

Positive leaves of the circuit are the (Büchi) objective of the OR-player, and negative leaves are the (Büchi) objective of the AND-player. Then obviously, the circuit evaluates to true if, and only if, the OR-player has a winning strategy for satisfying his Büchi condition, which in turn is equivalent to the fact that there is an equilibrium with payoff 0 for the AND-player, by Lemma 2.3. We obtain P-hardness for the existence problem, using Lemma 2.4: the preference relations in the game constructed in Lemma 2.4 are Büchi objectives.

4.4 Safety Objectives

The value problem for safety objectives is known to be P-complete [28]. We next show that the existence problem with constrained moves can be solved in NP, and conclude with NP-hardness of the existence problem.

Reduction to a conjunction of reachability objectives. We assume the preference relation of each player in $A \in \text{Agt}$ is defined as a single safety objective Ω_A given by target set T_A. In the corresponding suspect game, we show that the goal of Eve is equivalent to a conjunction of reachability objectives. Let $L \subseteq \text{Agt}$. In suspect game $\mathcal{H}(\mathcal{G}, L, \text{Allow})$, we define several reachability objectives as follows: for each $A \in L$, we define $T'_A = T_A \times \{P \mid P \subseteq \text{Agt}\} \cup \text{States} \times \{P \mid A \notin P\}$, and we write Ω'_A for the corresponding reachability objectives.

Lemma 4.6. *A play ρ is winning for Eve in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$ if, and only if, $\rho \in \bigcap_{A\in L} \Omega'_A$.*

Proof. Let ρ be a play in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$, and assume it is winning for Eve. Then, for each $A \in L(\rho) \cap L$, $\rho \notin \Omega_A$, which means that the target set T_A is visited along $\pi_1(\rho)$, and therefore T'_A is visited along ρ. If $A \notin L(\rho)$, then a state (s, P) with $A \notin P$ is visited by ρ: the target set T'_A is visited. This implies that $\rho \in \bigcap_{A\in L} \Omega'_A$.

Conversely let $\rho \in \bigcap_{A\in L} \Omega'_A$. For every $A \in L$, T'_A is visited by ρ. Then, either T_A is visited by $\pi_1(\rho)$ (which means that $\rho \notin \Omega_A$) or $A \notin L(\rho)$. In particular, ρ is a winning play for Eve in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$. $\square$

Algorithm for solving zero-sum games with a conjunction of reachability objectives. We now give a simple algorithm for solving zero-sum games with a conjunction of reachability objectives. This algorithm works in exponential time with respect to the size of the conjunction (we will see in [8] that the problem is PSPACE-complete). However for computing Nash equilibria in safety games we will only use it for small (logarithmic) conjunctions. Let $\mathcal{G}$ be a two-player turn-based game with a winning objective for Eve given as a conjunction of reachability objectives $\Omega_1, \ldots, \Omega_k$. We assume vertices of Eve and Adam in $\mathcal{G}$ are $V_\exists$ and $V_\forall$ respectively, and that the initial vertex is v_0. The idea is to construct a new game $\mathcal{G}'$ that remembers the objectives that have been visited so far. The vertices of game $\mathcal{G}'$ controlled by `Eve` and `Adam` are $V'_\exists = V_\exists \times 2^{[\![1,k]\!]}$ and $V'_\forall = V_\forall \times 2^{[\![1,k]\!]}$ respectively. There is a transition from (v, S) to (v', S') if, and only if, there is a transition from v to v' in the original game and $S' = S \cup \{i \mid v' \in \Omega_i\}$. The reachability objective Ω for `Eve` is given by target set $\text{States} \times [\![1,k]\!]$. It is clear that there is a winning strategy in $\mathcal{G}$ from v_0 for the conjunction of reachability objectives $\Omega_1, \ldots, \Omega_k$ if, and only if, there is a winning strategy in game $\mathcal{G}'$ from $(v_0, \{i \mid v_0 \in \Omega_i\})$ for the reachability objective Ω. The number of vertices of this new game is $|V'_\exists \cup V'_\forall| = |V_\exists \cup V_\forall| \cdot 2^k$, and the size of the new transition table Tab' is bounded by $|\text{Tab}| \cdot 2^k$, where Tab is the transition table of $\mathcal{G}$. An attractor computation on $\mathcal{G}'$ is then done in time $\mathcal{O}(|V'_\exists \cup V'_\forall| \cdot |\text{Tab}'|)$, we obtain an algorithm for solving zero-sum games with a conjunction of reachability objectives, running in time $\mathcal{O}(2^{2k} \cdot (|V_\exists \cup V_\forall| \cdot |\text{Tab}|))$.

Algorithm. The algorithm for solving the existence problem with constrained moves for single reachability objectives could be copied and would then be correct. It would however not be running in NP. We therefore propose a refined algorithm:

(i) guess a lasso-shaped play $\rho = \tau_1 \cdot \tau_2^\omega$ (with $|\tau_i| \leq |\text{States}|^2$) in $\mathcal{H}(\mathcal{G}, \text{Allow})$ such that `Adam` obeys `Eve` along ρ, and $\pi = \pi_1(\rho)$ satisfies the constraint on the payoff;

NB: if $\text{Los}(\pi)$ is the set of players losing in π, computing $W(\mathcal{G}, \text{Los}(\pi), \text{Allow})$ would require exponential time. We will avoid this expensive computation.

(ii) check that any `Adam`-deviation along ρ, say at position i (for any i), leads to a state from which `Eve` has a strategy $\sigma^i_\exists$ to ensure that any play in $\rho_{\leq i} \cdot \text{Out}(\sigma^i_\exists)$ is winning for her.

Step (ii) can be done as follows: pick an `Adam`-state $(s, \text{Agt}, m_{\text{Agt}})$ along ρ and a successor (t, P) such that $t \neq \text{Tab}(s, m_{\text{Agt}})$; we only need to show that $(t, P) \in W(\mathcal{G}, (\text{Los}(\pi) \setminus \text{Los}(\pi_{\leq i})) \cap P, \text{Allow})$. We can compute this set efficiently (in polynomial time) using the algorithm of the previous paragraph since $2^{|P|} \leq |\text{Tab}|$ (using the same argument as in Lemma 3.2).

This algorithm, which runs in NP, precisely implements Theorem 3.4, and therefore correctly decides the existence problem with constrained moves.

Hardness. The NP-hardness for the existence problem with constrained outcomes, can be proven by encoding an instance of 3SAT using a game similar to that for reachability objectives, see Section 4.2. We only change the constraint which is now that all players C_i should be losing, and we get the same equivalence.

The reduction of Lemma 2.4.4 cannot be used to deduce the hardness of the existence problem, since it assumes a lower bound on the payoff. Here the constraint is an upper bound ("each player should be losing"). We therefore provide an ad-hoc reduction in this special case. We add some module at the end of the game to enforce that in an equilibrium, all players are losing. We add concurrent states between A and each C_i. All players C_i are trying to avoid t, and A is trying to avoid u.

Since A has no target in $\mathcal{G}_\phi$ she cannot lose before seeing u, and then she can always change her strategy in the concurrent states in order to go to t. Therefore an equilibrium always ends in t. A player C_i whose target was not seen during game $\mathcal{G}_\phi$, can change her strategy in order to go u instead of t. That means that if there is an equilibrium, there was one in $\mathcal{G}_\phi$ where all C_i where losing. Conversely, if there was such an equilibrium in $\mathcal{G}_\phi$, we can extend this strategy profile by one whose outcome goes to t and it is an equilibrium in the new game.

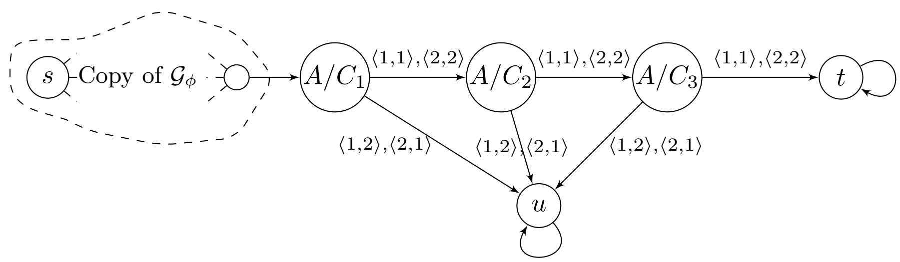


Figure 4.3: Extending the game with final concurrent modules

4.5 Co-Büchi Objectives

The value problem for co-Büchi objectives is known to be P-complete since [28]. We will now prove that the existence problem with constrained moves is in NP and then NP-hardness of the (constrained) existence problem.

Equivalence with a conjunction of Büchi conditions. We assume the preference relation of each player $A \in \text{Agt}$ is a single co-Büchi objective Ω_A given by target set T_A. In the corresponding suspect game, we show that the goal of player A_1 is equivalent to a conjunction of Büchi objectives. Let $L \subseteq \text{Agt}$. In suspect game $\mathcal{H}(\mathcal{G}, L, \text{Allow})$, we define several Büchi objectives as follows: for each $A \in L$, we define $T'_A = T_A \times \{P \mid P \subseteq \text{Agt}\} \cup \text{States} \times \{P \mid A \notin P\}$, and we write Ω'_A for the corresponding Büchi objective.

Lemma 4.7. *A play ρ is winning for* `Eve` *in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$ if, and only if, $\rho \in \bigcap_{A\in L} \Omega'_A$.*

Proof. Let ρ be a play in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$, and assume it is winning for `Eve`. Then, for each $A \in L(\rho) \cap L$, $\rho \notin \Omega_A$, which means that the target set T_A is visited along $\pi_1(\rho)$, and therefore T'_A is visited infinitely often along ρ. If $A \notin L(\rho)$, then a state (s, P) with $A \notin P$ is visited infinitely often by ρ: the target set T'_A is visited infinitely often. This implies that $\rho \in \bigcap_{A\in L} \Omega'_A$.

Conversely let $\rho \in \bigcap_{A \in L} \Omega'_A$. For every $A \in L$, T'_A is visited infinitely often by ρ. Then, either T_A is visited infinitely often by $\pi_1(\rho)$ (which means that $\rho \notin \Omega_A$) or $A \notin L(\rho)$. In particular, ρ is a winning play for `Eve` in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$. $\square$

Algorithm for solving zero-sum games with a conjunction of Büchi objectives. We will adapt the algorithm for conjunctions of reachability objectives (page 53) to conjunctions of Büchi objectives. Let $\mathcal{G}$ be a two-player turn-based game with a winning objective for `Eve` given as a conjunction of Büchi objectives $\Omega_1, \ldots, \Omega_k$. The idea is to construct a new game $\mathcal{G}'$ which checks that each objective Ω_i is visited infinitely often. The vertices of $\mathcal{G}'$ controlled by `Eve` and `Adam` are $V'_\exists = V_\exists \times [\![0, k]\!]$ and $V'_\forall = V_\forall \times [\![0, k]\!]$ respectively. There is a transition from (v, k) to $(v', 0)$ if, and only if, there is a transition from v to v' in the original game and for $0 \leq i < k$, there is a transition from (v, i) to $(v', i+1)$ if, and only if, there is a transition from v to v' in the original game and $v' \in \Omega_{i+1}$. In $\mathcal{G}'$, the objective for `Eve` is the Büchi objective Ω given by target set $\text{States} \times \{k\}$, where $\text{States} = V_\exists \cup V_\forall$ is the set of vertices of $\mathcal{G}$. It is clear that there is a winning strategy in $\mathcal{G}$ from v_0 for the conjunction of Büchi objectives $\Omega_1, \ldots, \Omega_k$ if, and only if, there is a winning strategy in $\mathcal{G}'$ from $(v_0, 0)$ for the Büchi objective Ω. The number of states of game $\mathcal{G}'$ is $|\text{States}'| = |\text{States}| \cdot k$, and the size of the transition table $|\text{Tab}'| = |\text{Tab}| \cdot k$. Using a standard algorithm for turn-based Büchi objectives, which works in time $\mathcal{O}(|\text{States}'| \cdot |\text{Tab}'|)$, we obtain an algorithm for solving zero-sum games with a conjunction of Büchi objectives running in time $\mathcal{O}(k^2 \cdot |\text{States}| \cdot |\text{Tab}|)$ (hence in polynomial time).

Algorithm. The algorithm is the same as for reachability objectives. Only the computation of the set of winning states in the suspect game is different. Since we just showed that this part can be done in polynomial time, the global algorithm still runs in (non-deterministic) polynomial time.

Hardness. The hardness result for the existence problem with constrained outcomes with co-Büchi objectives was already proven in [49]. The idea is to encode an instance of 3SAT into a game with co-Büchi objectives. For completeness we describe the reduction below, and explain how it can be modified for proving NP-hardness of the existence problem.

Let us consider an instance $\phi = C_1 \wedge \cdots \wedge C_n$ of SAT, where $C_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$, and $\ell_{i,j} \in \{x_k, \neg x_k \mid 1 \leq k \leq p\}$. The game $\mathcal{G}$ is obtained from module $M(\phi)$ depicted on Figure 4.4, by joining the outgoing edge of C_{n+1} to C_1. Each module $M(\phi)$ involves a set of players B_k, one for each variable x_k, and a player A_1. Player A_1 controls the clause states. Player B_k control the literal states $\ell_{i,j}$ when $\ell_{i,j} = \neg x_k$, then having the opportunity to go to state $\perp$. There is no transition to $\perp$ for literals of the form x_k. In $M(\phi)$, assuming that the players B_k will not play to $\perp$, then A_1 has a strategy that does not visit both x_k and $\neg x_k$ for every k if, and only if, formula ϕ is satisfiable. Finally, the co-Büchi objective of B_k is given by $\{x_k\}$. In other terms, the aim of B_k is to visit x_k only a finite number of times. This way, in a Nash equilibrium, it cannot be the case that both x_k and $\neg x_k$ are visited infinitely often: it would imply that B_k loses but could improve her payoff by going to $\perp$ (actually, $\neg x_k$ should not be visited at all if x_k is visited infinitely often). Therefore setting the objective of A_1 to $\{\perp\}$, there is a Nash equilibrium where she wins if, and only if, ϕ is satisfiable. This shows NP-hardness for the existence problem with constrained outcomes.

For the existence problem, we use the transformation described in Section 2.4.4. We add an extra player A_2 to $\mathcal{G}$ and consider the game $\mathcal{G}' = E(\mathcal{G}, A_1, A_2, \rho)$, where ρ is a winning path for A_1. The objective of the players in $\mathcal{G}'$ can be described by co-Büchi objectives: A_2 has to avoid seeing $T = \{s_1\}$ infinitely often and keep the same target set for A_1. Applying Lemma 2.5, there is a Nash equilibrium in $\mathcal{G}'$ if, and only if, there is one in $\mathcal{G}$ where A_1 wins, this shows NP-hardness for the existence problem.

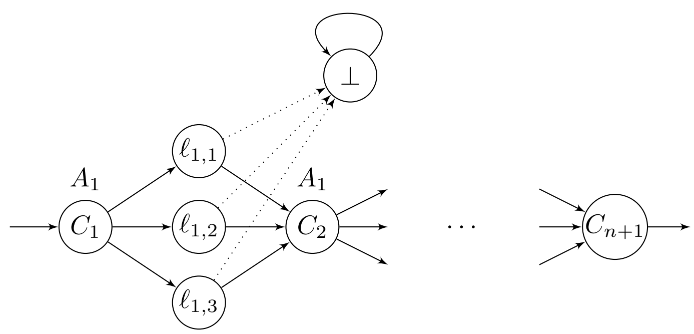


Figure 4.4: Module $M(\phi)$, where $\phi = C_1 \wedge \cdots \wedge C_n$ and $C_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$

4.6 Objectives Given as Circuits

The value problem is known to be PSPACE-complete for turn-based games and objectives given as circuits [20]. We will show that the (constrained) existence problem is also PSPACE-complete in this framework.

Equivalence with a circuit objective. We assume the preference relation of each player $A \in \text{Agt}$ is given by a circuit C_A. Let $L \subseteq \text{Agt}$. We will define a Boolean circuit defining the winning condition of Eve in the suspect game $\mathcal{H}(\mathcal{G}, L, \text{Allow})$.

We define for each player $A \in \text{Agt}$ and each set P of players (such that $\text{States} \times P$ is reachable in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$), a circuit $D_{A,P}$ which outputs true for the plays ρ with $L(\rho) = P$ (i.e. whose states that are visited infinitely often are some in $\text{States} \times \{P\}$), and whose value by C_A is true. We do so by making a copy of the circuit C_A, adding $|\text{States}| - 1$ OR gates $g_1 \cdots g_{|\text{States}|}$ and one AND gate h. There is an edge from (s_i, P) to g_i and from g_{i-1} to g_i if $i < |\text{States}|$ then there is an edge from the output gate of C_A to h and from h to the output gate of the new circuit. Inputs of C_A are now the (s, P)'s (instead of the s's). The circuit $D_{A,P}$ is given on Figure 4.5.

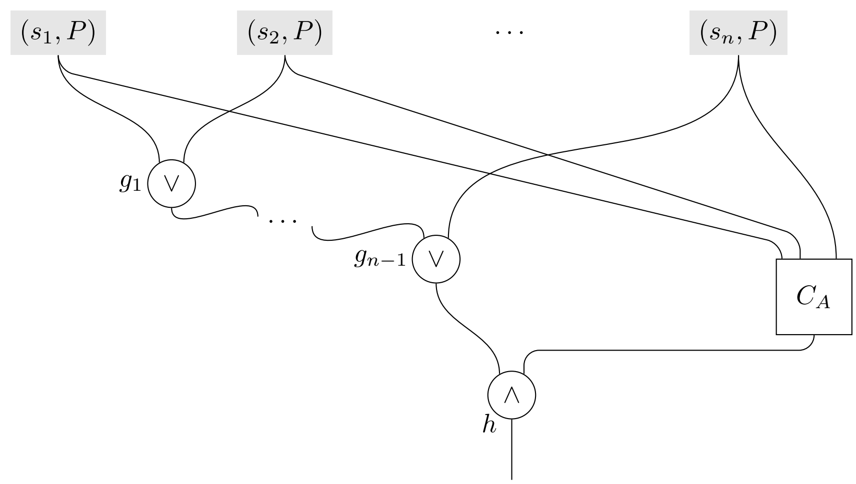


Figure 4.5: Circuit $D_{A,P}$

We then define a circuit E_A which outputs `true` for the plays ρ with $A \in L(\rho)$ and whose output by C_A is `true`. We do so by taking the disjunction of the circuits $D_{A,P}$. Formally, for each set of players P such that States $\times$ P is reachable in the suspect game and $A \in P$, we include the circuit $D_{A,P}$ and writing $o_{A,P}$ for its output gate, we add `OR` gates so that there is an edge from $o_{A,P}$ to g_i and from g_i to g_{i+1}, and then from g_{n+1} to the output gate.

Finally we define the circuit F_L, which outputs `true` for the plays ρ such that there is no $A \in L$ such that $A \in L(\rho)$ and the output of $\pi_1(\rho)$ by C_A is `true`. This corresponds exactly to the plays that are winning for `Eve` in suspect game $\mathcal{H}(\mathcal{G}, L, \text{Allow})$. We do so by negating the disjunction of all the circuits E_A for $A \in L$.

The next lemma follows from the construction:

Lemma 4.8. *A play ρ is winning for `Eve` in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$ if, and only if, ρ evaluates circuit F_L to `true`.*

We should notice that circuit F_L has size polynomial in the size of $\mathcal{G}$, thanks to Lemma 3.2.

Algorithm and complexity analysis. To solve the existence problem with constrained moves we apply the same algorithm as for reachability objectives (see section 4.2). For complexity matters, the only difference stands in the computation of the set of winning states in the suspect game. Thanks to Lemma 4.8, we know it reduces to the computation of the set of winning states in a turn-based game with an objective given as a circuit (of polynomial-size). This can be done in PSPACE [20], which yields a PSPACE upper bound for the existence problem with constrained moves (and therefore for the existence problem and the value problem 2.3). PSPACE-hardness of all problems follows from that of the value problem in turn-based games [20], and from Lemma 2.3 and 2.4 (we notice that the preference relations in the new games are easily definable by circuits).

4.7 Rabin and Parity objectives

The value problem is known to be NP-complete for Rabin conditions [23] and in UP$\cap$ co-UP for parity conditions [34].

We then notice that a parity condition is a Rabin condition with half as many pairs as the number of priorities: assume the parity condition is given by $p\colon \text{States} \mapsto [\![0,d]\!]$ with $d \in \mathbb{N}$; take for i in $[\![0,\frac{d}{2}]\!]$, $Q_i = p^{-1}\{2i\}$ and $R_i = p^{-1}\{2j+1 \mid j \geq i\}$. Then the Rabin objective $(Q_i, R_i)_{0\leq i \leq \frac{d}{2}}$ is equivalent to the parity condition given by p.

We will design an algorithm that solves the existence problem with constrained moves in $\mathsf{P}^{\mathsf{NP}}_{\parallel}$ for Rabin objectives. This algorithm uses a lot non-determinism. We will then propose a deterministic algorithm which runs in exponential time, but will be useful in Section 4.8. This section will end with $\mathsf{P}^{\mathsf{NP}}_{\parallel}$-hardness of the (constrained) existence problem for parity objectives. This will imply all expected results.

Equivalence with a Streett game. We assume that the preference relation of each player $A \in \text{Agt}$ is given by the Rabin condition $(Q_{i,A}, R_{i,A})_{i\in[\![1,k_A]\!]}$. Let $L \subseteq \text{Agt}$. In the suspect game $\mathcal{H}(\mathcal{G}, L, \text{Allow})$, we define the Streett objective $(Q'_{i,A}, R'_{i,A})_{i\in[\![1,k_A]\!],A\in L}$, where $Q'_{i,A} = (Q_{i,A} \times \{P \mid A \in P\}) \cup (\text{States} \times \{P \mid A \notin P\})$ and $R'_{i,A} = R_{i,A} \times \{P \mid A \in P\}$, and we write Ω_L for the corresponding set of winning plays.

Lemma 4.9. *A play ρ is winning for* `Eve` *in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$ if, and only if, $\rho \in \Omega_L$.*

Proof. Assume ρ is winning for `Eve` in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$. For all $A \in L(\rho) \cap L$, $\pi_1(\rho)$ does not satisfy the Rabin condition given by $(Q_{i,A}, R_{i,A})_{i\in[\![1,k_A]\!]}$. For all $1 \leq i \leq k_A$, $\text{Inf}(\pi_1(\rho)) \cap Q_{i,A} = \varnothing$ or $\text{Inf}(\pi_1(\rho)) \cap R_{i,A} \neq \varnothing$. We infer that for all $1 \leq i \leq k_A$, $\text{Inf}(\rho) \cap Q'_{i,A} = \varnothing$ or $\text{Inf}(\rho) \cap R'_{i,A} \neq \varnothing$. Now, if $A \notin L(\rho)$ then all $Q'_{i,A}$ are seen infinitely often along ρ. Therefore for every $A \in L$, the Streett conditions $(Q'_{i,A}, R'_{i,A})$ is satisfied along ρ (that is, $\rho \in \Omega_L$).

Conversely, if the Streett condition $(Q'_{i,A}, R'_{i,A})_{i\in[\![1,k_A]\!],A\in L}$ is satisfied along ρ, then either the Rabin condition $(Q_{i,A}, R_{i,A})$ is not satisfied along $\pi_1(\rho)$ or $A \notin L(\rho)$. This means that `Eve` is winning in $\mathcal{H}(\mathcal{G}, L, \text{Allow})$. $\square$

Algorithm. We now describe a $\mathsf{P}_{\|}^{\mathsf{NP}}$ algorithm for solving the existence problem with constrained moves in games where each player has a single Rabin objective. As in the previous cases, our algorithm relies on the suspect game construction.

Write $\mathcal{P}$ for the set of sets of players of Agt that appear as the second item of a state of $\mathcal{H}(\mathcal{G}, \mathrm{Allow})$:

$$\mathcal{P} = \{P \subseteq \mathrm{Agt} \mid \exists s \in \mathrm{States}.\ (s, P) \text{ is a state of } \mathcal{H}(\mathcal{G}, \mathrm{Allow})\}.$$

Since $\mathcal{H}(\mathcal{G}, \mathrm{Allow})$ has size polynomial, so does $\mathcal{P}$. Also, for any path ρ, $L(\rho)$ is a set of $\mathcal{P}$. Hence, for a fixed L, the number of sets $L(\rho) \cap L$ is polynomial. Now, as recalled on page 46, the winning condition for `Eve` is that the players in $L(\rho) \cap L$ must be losing along $\pi_1(\rho)$ in $\mathcal{G}$ for their Rabin objective. We have seen that this can be seen as a Streett objective.

Now, deciding whether a state is winning in a turn-based game for a Streett condition can be decided in coNP [23]. Hence, given a state $s \in$ States and a set L, we can decide in coNP whether s is winning for `Eve` in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$. This will be used as an oracle in our algorithm below.

Now, pick a set $P \subseteq$ Agt of suspects, i.e., for which there exists $(s, t) \in \mathrm{States}^2$ and m_{Agt} s.t. $P = \mathrm{Susp}((s, t), m_{\mathrm{Agt}})$. Using the same arguments as in the proof of Lemma 3.2, it can be shown that $2^{|P|} \leq |\mathrm{Tab}|$, so that the number of subsets of P is polynomial. Now, for each set P of suspects and each $L \subseteq P$, write $w(L)$ for the size of the winning region of `Eve` in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$. Then the sum $\sum_{P \in \mathcal{P} \setminus \{\mathrm{Agt}\}} \sum_{L \subseteq P} w(L)$ is at most $|\mathrm{States}| \times |\mathrm{Tab}|^2$.

Assume that the exact value M of this sum is known, and consider the following algorithm:

1. for each $P \subseteq \mathcal{P} \setminus \{\mathrm{Agt}\}$ and each $L \subseteq P$, guess a set $W(\mathcal{G}, L, \mathrm{Allow}) \subseteq$ States, which we intend to be the exact winning region for `Eve` in the game $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$.

2. check that the sizes of those sets sum up to M;

3. for each $s \notin W(\mathcal{G}, L, \mathrm{Allow})$, check that `Eve` does not have a winning strategy from s in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$. This can be checked in NP, as explained above.

4. guess a lasso-shaped path $\rho = \pi \cdot \tau^\omega$ in $\mathcal{H}(\mathcal{G}, \mathrm{Allow})$ starting from (s, Agt), with $|\pi|$ and $|\tau|$ less than $|\mathrm{States}|^2$ (following Lemma 2.2) visiting only states where the second item is Agt. This path can be seen as the outcome of some strategy of `Eve` when `Adam` obeys. For this path, we then check the following:

   - along ρ, the sets of winning and losing players satisfy the original constraint (remember that in the problem we aim at solving there are constraints on the outcome);

- any deviation along ρ leads to a state that is winning for `Eve`. In other terms, pick a state $h = (s, \mathrm{Agt}, m_{\mathrm{Agt}})$ of `Adam` along ρ, and pick a successor $h' = (t, P)$ of h such that $t \neq \mathrm{Tab}(s, m_{\mathrm{Agt}})$. Then the algorithm checks that $t \in W(\mathcal{G}, L \cap P, \mathrm{Allow})$.

The algorithm accepts the input M if it succeeds in finding the sets W and the path ρ such that all the checks are successful. This algorithm is in NP, and will be used as a second oracle.

We now show that if M is exactly the sum of the $w(L)$, then the algorithm accepts M if, and only if, there is a Nash equilibrium satisfying the constraint, i.e. if, and only if, `Eve` has a winning strategy from (s, Agt) in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$.

First assume that the algorithm accepts M. This means that it is able, for each L, to find sets $W(\mathcal{G}, L, \mathrm{Allow})$ of states whose complement does not intersect the winning region of $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$. Since M is assumed to be the exact sum of $w(\mathcal{G}, L, \mathrm{Allow})$ and the size of the sets $W(\mathcal{G}, L, \mathrm{Allow})$ sum up to M, we deduce that $W(\mathcal{G}, L, \mathrm{Allow})$ is exactly the winning region of `Eve` in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$. Now, since the algorithm accepts, it is also able to find a (lasso-shaped) path ρ only visiting states having Agt as the second component. This path has the additional property that any "deviation" from a state of `Adam` along this path ends up in a state that is winning for `Eve` for players in $L \cap P$, where P is the set of suspects for the present deviation. This way, if during ρ, `Adam` deviates to a state (t, P), then `Eve` will have a strategy to ensure that along any subsequent play, the objectives of players in $L \cap P$ (in $\mathcal{G}$) are not fulfilled, so that along any run ρ', the players in $L \cap L(\rho')$ are losing for their objectives in $\mathcal{G}$, so that `Eve` wins in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$.

Conversely, assume that there is a Nash equilibrium satisfying the constraint. Following Lemma 2.2, we assume that the outcome of the corresponding strategy profile has the form $\pi \cdot \tau^{\omega}$. From Lemma 3.3, there is a winning strategy for `Eve` in $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$ whose outcome when `Adam` obeys follows the outcome of the Nash equilibrium. As a consequence, the outcome when `Adam` obeys is a path ρ that the algorithm can guess. Indeed, it must satisfy the constraints, and any deviation from ρ with set of suspects P ends in a state where `Eve` wins for the winning condition of $\mathcal{H}(\mathcal{G}, L, \mathrm{Allow})$, hence also for the winning condition of $\mathcal{H}(\mathcal{G}, L \cap P, \mathrm{Allow})$, since any path ρ' visiting (t, P) has $L(\rho') \subseteq P$.

Finally, our global algorithm is as follows: we run the first oracle for all the states and all the sets L that are subsets of a set of suspects (we know that there are polynomially many such inputs). We also run the second algorithm on all the possible values for M, which are also polynomially many. Now, from the answers of the first oracle, we compute the exact value M, and return the value given by the second on that input. This algorithm runs in $\mathsf{P}^{\mathsf{NP}}_{\parallel}$ and decides the existence problem with constrained moves.

Deterministic algorithm. In the next section we will need a deterministic algorithm to solve games with objectives given as deterministic Rabin automata.

We therefore present it right now. The deterministic algorithm works by successively trying all the possible payoffs, there are $2^{|\mathrm{Agt}|}$ of them. Then it computes the winning strategies of the suspect game for that payoff. In [33] an algorithm for Streett games is given, which works in time $\mathcal{O}(n^k \cdot k!)$, where n is the number of vertices in the game, and k the size of the Streett condition. The algorithm has to find, in the winning region of `Eve` in $\mathcal{H}(\mathcal{G}, \mathrm{Allow})$, a lasso that satisfies the Rabin winning conditions of the winners and do not satisfy whose of the losers. To do so it tries all the possible choices of elementary Rabin condition that are satisfied to make the players win, there are at most $\prod_{A \in \mathrm{Agt}} k_A$ possible choices. And for the losers, we try the possible choices for whether $Q_{i,A}$ is visited of not, there are $\prod_{A \in \mathrm{Agt}} 2^{k_A}$ such choices. It then looks for a lasso cycle that, when A is a winner, does not visit $Q_{i_A,A}$ and visits $R_{i_A,A}$, and when A is a loser, visits $R_{i_A,A}$ when it has to, or does not visit $Q_{i_A,A}$. This is equivalent to finding a path satisfying a conjunction of Büchi conditions and can be done in polynomial time $\mathcal{O}(n \times \sum_{A \in \mathrm{Agt}} k_A)$. The global algorithm works in time

$$\mathcal{O}\left(2^{|\mathrm{Agt}|} \cdot \left(|\mathrm{Tab}|^{3 \sum_A k_A} \cdot (\sum_A k_A)! + \left(\prod_{A \in \mathrm{Agt}} k_A \cdot 2^{k_A}\right) \cdot |\mathrm{Tab}|^3 \cdot \sum_A k_A\right)\right)$$

Notice that the exponential does not come from the size of the graph but from the number of agents and the number of elementary Rabin conditions, this will be important when in the next section we will reuse the algorithm on a game whose size is exponential.

$\mathsf{P}_{\parallel}^{\mathsf{NP}}$**-hardness.** We now prove $\mathsf{P}_{\parallel}^{\mathsf{NP}}$-hardness of the existence problem with constrained outcomes in the case of parity objectives. The main reduction is an encoding of the $\oplus\mathsf{SAT}$ problem, where the aim is to decide whether the number of satisfiable instances among a set of formulas is even. This problem is known to be complete for $\mathsf{P}_{\parallel}^{\mathsf{NP}}$ [26].

Before tackling the whole reduction, we first develop some preliminaries on single instances of SAT, inspired from [13]. Let us consider an instance $\phi = C_1 \wedge \cdots \wedge C_n$ of SAT, where $C_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$, and $\ell_{i,j} \in \{x_k, \neg x_k \mid 1 \leq k \leq p\}$. With ϕ, we associate a three-player game $N(\phi)$, depicted on Figure 4.6 (where the first state of $N(\phi)$ is controlled by A_1, and the first state of each $N'(C_j)$ is concurrently controlled by A_2 and A_3). For each variable x_j, players A_2 and A_3 have the following target sets:

$$T_{2j}^{A_2} = \{x_j\} \qquad T_{2j+1}^{A_2} = \{\neg x_j\} \qquad\qquad T_{2j+1}^{A_3} = \{x_j\} \qquad T_{2j}^{A_3} = \{\neg x_j\}$$

This construction enjoys interesting properties, given by the following lemma:

Lemma 4.10. *If the formula ϕ is not satisfiable, then there is a strategy for player A_1 in $N(\phi)$ such that players A_2 and A_3 lose. If the formula ϕ is satisfiable, then for any strategy profile σ_{Agt}, one of A_2 and A_3 can change her strategy and win.*

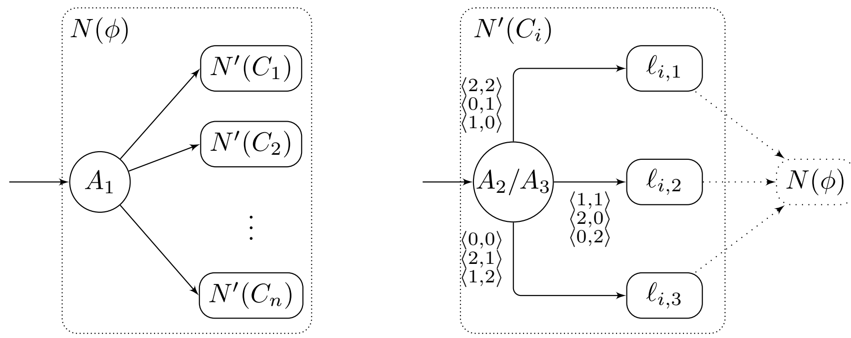


Figure 4.6: The game $N(\phi)$ (left), where $N'(C_i)$ is the module on the right.

Proof. We begin with the first statement, assuming that ϕ is not satisfiable and defining the strategy for A_1. With a history h in $N(\phi)$, we associate a valuation $v^h\colon \{x_k \mid k \in [1,p]\} \to \{\top, \bot\}$ (where p is the number of distinct variables in ϕ), defined as follows:

$$v^h(x_k) = \top \;\Leftrightarrow\; \exists m.\ h_m = x_k \wedge \forall m' > m.\ h_{m'} \neq \neg x_k \qquad \text{for all } k \in [1,p]$$

We also define $v^h(\neg x_k) = \neg v^h(x_k)$. Under this definition, $v^h(x_k) = \top$ if the last occurrence of x_k or $\neg x_k$ along h was x_k. We then define a strategy σ_1 for player A_1: after a history h ending in an A_1-state, we require $\sigma_1(h)$ to go to $N'(C_i)$ for some C_i (with least index, say) that evaluates to false under v^h (such a C_i exists since ϕ is not satisfiable). This strategy enforces that if $h \cdot \sigma_1(h) \cdot \ell_{i,j}$ is a finite outcome of σ_1, then $v^h(\ell_{i,j}) = \bot$, because A_1 has selected a clause C_i whose literals all evaluate to $\bot$. Moreover, $v^{h\cdot\sigma_1(h)\cdot\ell_{i,j}}(\ell_{i,j}) = \top$, so that for each j, any outcome of σ_1 will either alternate between x_k and $\neg x_k$ (hence visit both of them infinitely often), or no longer visit any of them after some point. Hence both A_2 and A_3 lose.

We now prove the second statement. Let v be a valuation under which ϕ evaluates to true, and σ_{Agt} be a strategy profile. From σ_{A_2} and σ_{A_3}, we define two strategies σ'_{A_2} and σ'_{A_3}. Consider a finite history h ending in the first state of $N'(C_i)$, for some i. Pick a literal $\ell_{i,j}$ of C_i that is true under v (the one with least index, say). We set

$$\sigma'_{A_2}(h) = [j - \sigma_{A_3}(h) \pmod 3] \qquad \sigma'_{A_3}(h) = [j - \sigma_{A_2}(h) \pmod 3].$$

It is easily checked that, when σ_{A_2} and σ'_{A_3} (or σ'_{A_2} and σ_{A_3}) are played simultaneously in the first state of some $N'(C_i)$, then the game goes to $\ell_{i,j}$. Thus under those strategies, any visited literal evaluates to true under v, which means that at most one of x_k and $\neg x_k$ is visited (infinitely often). Hence one of A_2 and A_3 is winning, which proves our claim. $\square$

We now proceed by encoding an instance

$$\exists x_1^1, \dots x_k^1.\ \phi^1(x_1^1, \dots, x_k^1)$$
$$\dots$$
$$\exists x_1^m, \dots x_k^m.\ \phi^m(x_1^m, \dots, x_k^m)$$

of $\oplus\mathsf{SAT}$ into a parity game. The game involves the three players A_1, A_2 and A_3 of the game $N(\phi)$ defined above, and it will contain a copy of $N(\phi^r)$ for each $1 \leq r \leq m$. The objectives of A_2 and A_3 are the unions of their objectives in each $N(\phi^r)$, e.g. $p^{A_2}(x_j^1) = p^{A_2}(x_j^2) = \dots = p^{A_m}(x_j^m) = 2j$.

For each such r, the game will also contain a copy of the game $M(\phi^r)$ depicted on Figure 4.4. Each game $M(\phi^r)$ involves an extra set of players B_k^r, one for each variable x_k^r. As we have seen in Section 4.5, in a Nash equilibrium, it cannot be the case that both x_k^r and $\neg x_k^r$ are visited infinitely often.

In order to test the parity of the number of satisfiable formulas, we then define two families of modules, depicted on Figure 4.7 to 4.10. Finally, the whole game $\mathcal{G}$ is depicted on Figure 4.11. In that game, the objective of A_1 is to visit infinitely often the initial state init.

Lemma 4.11. *There is a Nash equilibrium in the game $\mathcal{G}$ where A_2 and A_3 lose and A_1 wins if, and only if, the number of satisfiable formulas is even.*

Proof. Assume that there is a Nash equilibrium in $\mathcal{G}$ where A_1 wins and both A_2 and A_3 lose. Let ρ be its outcome. As already noted, if ρ visits module $M(\phi^r)$ infinitely often, then it cannot be the case that both x_k^r and $\neg x_k^r$ are visited infinitely often in $M(\phi^r)$, as otherwise B_k^r would be losing and have the opportunity to improve her payoff. This implies that ϕ^r is satisfiable. Similarly, if ρ visits infinitely often the states of $H(\phi^r)$ or $G(\phi^r)$ that is controlled by A_2 and A_3, then it must be the case that ϕ^r is not satisfiable, since from Lemma 4.10 this would imply that A_2 or A_3 could deviate and improve her payoff by going to $N(\phi^r)$.

We now show by induction on r that if ρ goes infinitely often in module $G(\phi^r)$ then $\#\{j \leq r \mid \phi^r \text{ is satisfiable}\}$ is even, and that (if $n > 1$) this number is odd if ρ goes infinitely in module $H(\phi^r)$.

When $r = 1$, since $H(\phi^1)$ is $M(\phi^1)$, ϕ^1 is satisfiable, as noted above. Similarly, if ρ visits $G(\phi^1)$ infinitely often, it also visits its A_2/A_3-state infinitely often, so that ϕ^1 is not satisfiable. This proves the base case.

Assume that the result holds up to some $r-1$, and assume that ρ visits $G(\phi^r)$ infinitely often. Two cases may occur:

- it can be the case that $M(\phi^r)$ is visited infinitely often, as well as $H(\phi^{r-1})$. Then ϕ^r is satisfiable, and the number of satisfiable formulas with index less than or equal to $r-1$ is odd. Hence the number of satisfiable formulas with index less than or equal to r is even.
- it can also be the case that the state A_2/A_3 of $G(\phi^r)$ is visited infinitely often. Then ϕ^r is not satisfiable. Moreover, since A_1 wins, the play

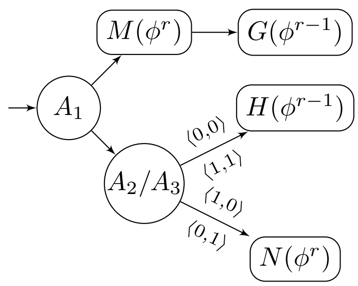


Figure 4.7: Module $H(\phi^r)$ for $r \geq 2$

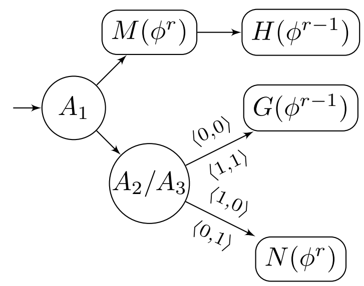


Figure 4.8: Module $G(\phi^r)$ for $r \geq 2$

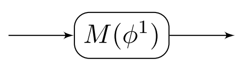


Figure 4.9: Module $H(\phi^1)$

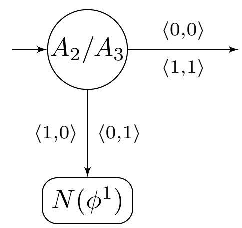


Figure 4.10: Module $G(\phi^1)$

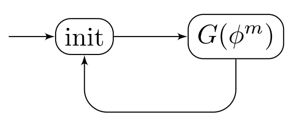


Figure 4.11: The game $\mathcal{G}$

will also visit $G(\phi^{r-1})$ infinitely often, so that the number of satisfiable formulas with index less than or equal to r is even.

If ρ visits $H(\phi^r)$ infinitely often, using similar arguments we prove that the number of satisfiable formulas with index less than or equal to r is odd.

To conclude, since A_1 wins, the play visits $G(\phi^m)$ infinitely often, so that the total number of satisfiable formulas is even.

Conversely, assume that the number of satisfiable formulas is even. We build a strategy profile, which we prove is a Nash equilibrium in which A_1 wins and A_2 and A_3 lose. The strategy for A_1 in the initial states of $H(\phi^r)$ and $G(\phi^r)$ is to go to $M(\phi^r)$ when ϕ^r is satisfiable, and to state A_2/A_3 otherwise. In $M(\phi^r)$, the strategy is to play according to a valuation satisfying ϕ^r. In $N(\phi^r)$, it follows a strategy along which A_2 and A_3 lose (this exists according to Lemma 4.10). This defines the strategy for A_1. Then A_2 and A_3 are required to always play the same move, so that the play never goes to some $N(\phi^r)$. In $N(\phi^r)$, they can play any strategy (they lose anyway, whatever they do). Finally, the strategy of B^r_k never goes to $\perp$.

We now explain why this is the Nash equilibrium we are after. First, as A_1 plays according to fixed valuations for the variables x^r_k, either B^r_k wins or she does not have the opportunity to go to $\perp$. It remains to prove that A_1 wins, and that A_2 and A_3 lose and cannot improve (individually). To see this, notice that between two consecutive visits to init, exactly one of $G(\phi^r)$ and $H(\phi^r)$ is visited. More precisely, it can be observed that the strategy of A_1 enforces that $G(\phi^r)$ is visited if $\#\{r < r' \leq m \mid \phi^{r'} \text{ is satisfiable}\}$ is even, and that $H(\phi^r)$ is visited otherwise. Then if $H(\phi_1)$ is visited, the number of satisfiable formulas with index between 2 and m is odd, so that ϕ_1 is satisfiable and A_1 can return to init. If $G(\phi^1)$ is visited, an even number of formulas with index between 2 and m is satisfiable, and ϕ^1 is not. Hence A_1 has a strategy in $N(\phi^1)$ to make A_2 and A_3 lose, so that A_2 and A_3 cannot improve their payoffs. □

This proves hardness for the existence problem with constrained outcomes for parity objectives. For the existence problem we will use the construction of Section 2.4.4, but since it can only be used to get rid of constraint of the type "A_1 is winning", we will add to the game two players, A_4 and A_5, whose objectives are opposite to A_2 and A_3 respectively, and one player A_6 that will be playing matching-penny games. The objectives for A_4 and A_5 are definable by parity objectives, by adding 1 to all the priorities. Then, we consider game $\mathcal{G}' = E(E(E(\mathcal{G}, A_1, A_6, \rho_1), A_4, A_6, \rho_4), A_5, A_6, \rho_5)$ where ρ_1, ρ_4 and ρ_5 are winning paths for A_1, A_4 and A_5 respectively. Thanks to Lemma 2.5, there is a Nash equilibrium in $\mathcal{G}'$ if, and only if, there is a Nash equilibrium in $\mathcal{G}$ where A_1 wins and A_2 and A_3 lose. We deduce $\mathsf{P}^{\mathsf{NP}}_{\parallel}$-hardness for the existence problem with parity objectives.

4.8 Objectives Given as Deterministic Rabin Automata

In order to find Nash equilibria when objectives are given as deterministic Rabin automata, we define the product of a game with automata (defining the objectives of the players), and show that it game-simulates the original game. This reduces the case of games with objectives are defined as Rabin automata to games with Rabin objectives, which we handled at the previous section; the resulting algorithm is in $\mathsf{EXPTIME}$. We end the section by showing PSPACE-hardness in the restricted case of Büchi automata.

Fix a game $\mathcal{G} = \langle \text{States}, \text{Agt}, \text{Act}, \text{Mov}, \text{Tab}, (\precsim_A)_{A\in\text{Agt}}\rangle$. Assume that some player A has her objective given by a deterministic Rabin automaton $\mathcal{A} = \langle Q, \text{States}, \delta, q_0, (Q_i, R_i)_{i\in[\![1,n]\!]}\rangle$; this automaton reads sequences of states of $\mathcal{G}$, and accepts the paths that are winning for player A. We show how to compute Nash equilibria in $\mathcal{G}$ by building a product $\mathcal{G}'$ of $\mathcal{G}$ with the automaton $\mathcal{A}$ and computing the Nash equilibria in the resulting game, with a Rabin winning condition for A.

We define the product of the game $\mathcal{G}$ with the automaton $\mathcal{A}$ as the game

$\mathcal{G} \ltimes \mathcal{A} = \langle \text{States}', \text{Agt}, \text{Act}, \text{Mov}', \text{Tab}', (\precsim'_A)_{A\in\text{Agt}}\rangle$, where:

- $\text{States}' = \text{States} \times Q$;
- $\text{Mov}'((s,q), A_j) = \text{Mov}(s, A_j)$ for every $A_j \in \text{Agt}$;
- $\text{Tab}'((s,q), m_{\text{Agt}}) = (s', q')$ where $\text{Tab}(s, m_{\text{Agt}}) = s'$ and $\delta(q,s) = q'$;
- If $B = A$ then $\precsim'_B$ is given by the internal Rabin condition $Q'_i = \text{States}\times Q_i$ and $R'_i = \text{States} \times R'_i$.

  Otherwise $\precsim'_B$ is derived from $\precsim_B$, defined by $\rho \precsim'_B \overline{\rho}$ if, and only if, $\pi(\rho) \precsim_B \pi(\overline{\rho})$ (where π is the projection of States$'$ on States). Notice that if $\precsim_B$ is an internal Rabin condition, then so is $\precsim'_B$.

Given a constraint Allow on moves in $\mathcal{G}$, we define the constraint Allow$'$ in $\mathcal{G}'$ by $\text{Allow}'((s,q), m_{\text{Agt}}) = \text{Allow}(s, m_{\text{Agt}})$.

Lemma 4.12. *$\mathcal{G} \ltimes \mathcal{A}$ game-simulates $\mathcal{G}$ with respect to constraints* Allow$'$ *and* Allow*, with game simulation defined according to the projection: $s \lhd (s', q)$ if, and only if, $s = s'$. This game simulation is preference-preserving. Conversely, $\mathcal{G}$ game-simulates $\mathcal{G} \ltimes \mathcal{A}$ with respect to constraints* Allow *and* Allow$'$*, with $(s,q) \lhd' s'$ if, and only if, $s = s'$, which is also preference-preserving.*

Proof. We begin with proving that both relations are preference-preserving. First notice that if $((s_n, q_n))_{n\geq 0}$ is a play in $\mathcal{G} \ltimes \mathcal{A}$, then its π-projection $(s_n)_{n\geq 0}$ is a play in $\mathcal{G}$. Conversely, if $\rho = (s_n)_{n\geq 0}$ is a play in $\mathcal{G}$, then there is a unique path $(q_n)_{n\geq 0}$ from initial state q_0 in $\mathcal{A}$ which reads it, and $((s_n, q_n))_{n\geq 0}$ is then a path in $\mathcal{G} \ltimes \mathcal{A}$ that we write $\pi^{-1}(\rho) = ((s_n, q_n))_{n\geq 0}$. That way, π defines a one-to-one correspondence between plays in $\mathcal{G}$ and plays in $\mathcal{G} \ltimes \mathcal{A}$ where the second component starts in q_0. For a player $B \neq A$, the objective is defined so that $\pi(\rho)$ has the same payoff as ρ. Consider now player A, she is winning in $\mathcal{G}$ for $\rho = (s_n)_{n\geq 0}$ if, and only if, $(s_n)_{n\geq 0} \in \text{L}(\mathcal{A})$ if, and only if, the unique path $(q_n)_{n\geq 0}$ from initial state q_0 that reads $(s_n)_{n\geq 0}$ satisfies the Rabin condition $(Q_i, R_i)_{i\in[\![1,n]\!]}$ in $\mathcal{A}$ if, and only if, $\pi^{-1}(\rho)$ satisfies the internal Rabin condition $(Q'_i, R'_i)_{i\in[\![1,n]\!]}$ in $\mathcal{G} \ltimes \mathcal{A}$. This proves that $\lhd$ is winning-preserving.

It remains to show that both relations are is game simulations. Assume $s \lhd (s,q)$ and pick an allowed move m_{Agt} in $\mathcal{G}$. It is also allowed in $\mathcal{G} \ltimes \mathcal{A}$. Take $(s', q') \in \text{States}'$, by definition $s' \lhd (s', q')$.

- If $\delta(q', s) \neq q'$ then $\text{Susp}(((s,q),(s',q')), m_{\text{Agt}}) = \varnothing$, and condition (2) trivially holds.
- Otherwise $\delta(q,s) = q'$. For any move m'_{Agt}, we have that $\text{Tab}(s, m'_{\text{Agt}}) = s'$ if, and only if, $\text{Tab}'((s,q), m'_{\text{Agt}}) = (s', \delta(q,s))$. Hence we have that $\text{Susp}(((s,q),(s',q')), m_{\text{Agt}}) = \text{Susp}((s,s'), m_{\text{Agt}})$, which implies condition (2).

Condition (1) obviously holds since, $(s,s') \in \text{Tab}(s, m_{\text{Agt}})$ if, and only if, $((s,q),(s',\delta(q,s))) \in \text{Tab}'((s,q), m_{\text{Agt}})$ by definition of $\mathcal{G} \ltimes \mathcal{A}$.

We now assume $(s,q) \lhd' s$ and pick an allowed move m_{Agt} in $\mathcal{G} \ltimes \mathcal{A}$. It is also allowed in $\mathcal{G}$. Take $s' \in \text{States}$. We define $q' = \delta(q,s)$, and we have

$(s', q') \lhd s'$ by definition of $\lhd'$. As before, condition (1) obviously holds, and we get condition (2) because $\text{Susp}(((s, q), (s', q')), m_{\text{Agt}}) = \text{Susp}((s, s'), m_{\text{Agt}})$. $\square$

Assume that for each player $A_i \in \text{Agt}$ the objective in $\mathcal{G}$ is given by a deterministic Rabin automaton $\mathcal{A}_i$. Applying the above result inductively, we can transform $\mathcal{G}$ into a game $\mathcal{G}'$ where each player has an internal Rabin winning condition. Applying Prop. 3.5 each time, we get the following result:

Proposition 4.13. *Let* $s \in \text{States}$. *There is a Nash equilibrium* σ_{Agt} *in* $\mathcal{G}$ *from* s *which respects the constraint on moves* Allow *and with outcome* ρ *if, and only if, there is a Nash equilibrium* σ'_{Agt} *in* $\mathcal{G}'$ *from* $(s, q_{01}, \ldots, q_{0n})$ *which respects the constraint on moves* Allow$'$ *and with outcome* ρ', *where* q_{0i} *is the initial state of* $\mathcal{A}_i$. *Moreover, the projection of* ρ' *on* $\mathcal{G}$ *is precisely* ρ.

Algorithm

The algorithm starts by computing the product of the game with the automata. The resulting game has size $|\mathcal{G}| \times \prod_{j \in [\![1,n]\!]} |\mathcal{A}_j|$, which is exponential in the number of players. For each player A_j ($1 \leq j \leq n$), the number of Rabin pairs in the product game is that of the original specification $\mathcal{A}_j$, say k_j. We apply the deterministic algorithm that we have designed for Rabin objectives (see page 61), which yields an exponential-time algorithm in our framework.

Hardness

We prove PSPACE-hardness in the restricted case of deterministic Büchi automata, by a reduction from the (complement of the) problem of the emptiness of the intersection of several language given by finite automata. This problem is known to be PSPACE-complete [37].

We fix finite automata $\mathcal{A}_1, \ldots, \mathcal{A}_n$ over alphabet Σ. For every $j \in [\![1, n]\!]$, we construct the Büchi automaton $\mathcal{A}'_j$ from $\mathcal{A}_j$ as follows. The alphabet contains Σ and two fresh special symbols i and f, $\Sigma' = \{i, f\} \cup \Sigma$. We add a state F with a self-loop labeled by f, an initial state I with a transition labeled by i to the original initial state. We add transitions labeled by f from every terminal states to F. We set the Büchi condition to $\{F\}$. If $\mathcal{L}_j$ is the language recognized by $\mathcal{A}_j$, then the language recognized by the Büchi automaton $\mathcal{A}'_j$ is $\mathcal{L}'_j = i \cdot \mathcal{L}_j \cdot f^\omega$. The intersection of the languages recognized by the automata $\mathcal{A}_j$ is empty if, and only if, the intersection of the languages recognized by the automata $\mathcal{A}'_j$ is empty.

We construct the game $\mathcal{G}$, with States $= \Sigma'$. For each $j \in [\![1, n]\!]$, there is a player A_j whose objective is given by $\mathcal{A}'_j$ and one special player A_0 whose objective is States$^\omega$ (she is always winning). Player A_0 controls all the states and there are transitions from any state to the states of $\Sigma \cup \{f\}$. Formally $\text{Act} = \Sigma \cup \{f\} \cup \bot$, for all state $s \in \text{States}$, $\text{Mov}(s, A_0) = \text{Act}$, and if $j \neq 0$ then $\text{Mov}(s, A_j) = \{\bot\}$ and for all $\alpha \in \Sigma \cup \{f\}$, $\text{Tab}(s, (\alpha, \bot, \ldots, \bot)) = \alpha$.

Lemma 4.14. *There is a Nash equilibrium in game $\mathcal{G}$ from i where every player wins if, and only if, the intersection of the languages recognized by the automata $\mathcal{A}'_j$ is not empty.*

Proof. If there is such a Nash equilibrium, let ρ be its outcome. The path ρ forms a word of Σ', it is accepted by every automata $\mathcal{A}'_j$ since every player wins. Hence the intersection of the languages $\mathcal{L}_j$ is not empty. Conversely, if a word $w = i \cdot w_1 \cdot w_2 \cdots$ is accepted by all the automata, player A_0 can play in a way such that everybody is winning: if at each step j she plays w_j, then the outcome is w which is accepted by all the automata. It is a Nash equilibrium since A_0 controls everything and cannot improve her payoff. $\square$

Since PSPACE is stable by complementation, this proves that the existence problem with constrained outcomes is PSPACE-hard for objectives described by Büchi automata.

In order to prove hardness for the existence problem we use results from Section 2.5. Winning conditions in $E(E(\ldots(E(\mathcal{G}, A_n, A_0, \rho_n), \ldots, A_2, A_0, \rho_2), A_1, A_0, \rho_1)$, where ρ_j is a winning play for A_i, can be defined by slightly modifying automata $\mathcal{A}'_1, \ldots, \mathcal{A}'_n$ to take into account the new states. By Lemma 2.5, there exists a Nash equilibrium in this game if and only if there is one in $\mathcal{G}$ where all the players win. Hence PSPACE-hardness also holds for the existence problem.

Chapter 5

Ordered Objectives

5.1 Ordering Several Objectives

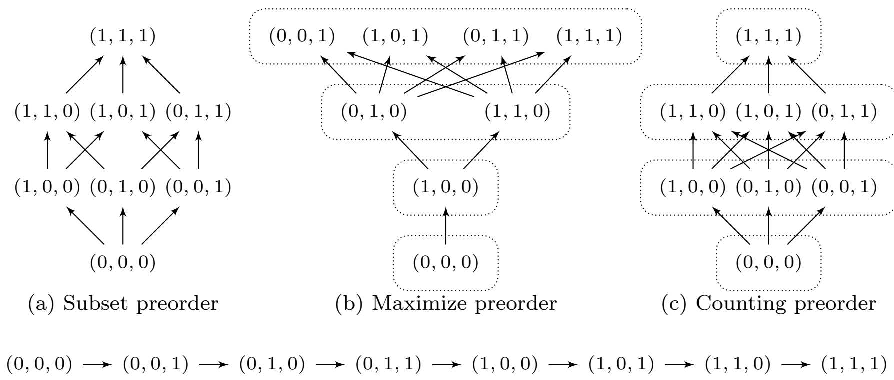
In this chapter we are interested in preference relations given as ordered objectives. In a game $\mathcal{G}$, an *ordered objective* is a pair $\omega = \langle(\Omega_i)_{1\leq i\leq n}, \lesssim\rangle$, where, for every $1 \leq i \leq n$, Ω_i is an objective, and $\lesssim$ is a preorder on $\{0,1\}^n$. A play ρ is assigned a *payoff vector* w.r.t. that ordered objective, which is defined as $\mathsf{payoff}_\omega(\rho) = \mathbf{1}_{\{i \mid \rho\in\Omega_i\}} \in \{0,1\}^n$ (where $\mathbf{1}_S$ is the vector v such that $v_i = 1 \Leftrightarrow i \in S$). The corresponding preference relation $\precsim_\omega$ is then defined by $\rho \precsim_\omega \rho'$ if, and only if, $\mathsf{payoff}_\omega(\rho) \lesssim \mathsf{payoff}_\omega(\rho')$.

We fix a game $\mathcal{G} = \langle \mathrm{States}, \mathrm{Agt}, \mathrm{Act}, \mathrm{Mov}, \mathrm{Tab}, (\precsim_A)_{A\in\mathrm{Agt}}\rangle$, and we assume that each preference relation $\precsim_A$ is given by an ordered objective $\omega_A = \langle(\Omega_i^A)_{1\leq i\leq n_A}, \lesssim_A\rangle$, where all Ω_i^A's are either reachability objectives or Büchi objectives. In the following we will write payoff_A instead of $\mathsf{payoff}_{\omega_A}$, and if ρ is a play, $\mathsf{payoff}(\rho) = (\mathsf{payoff}_A(\rho))_{A\in\mathrm{Agt}}$.

Examples of preorders. We now describe the preorders on $\{0,1\}^n$ that we consider in the sequel (Figures 5.1a–5.1d display four of these preorders for $n = 3$). For the purpose of these definitions, we assume that $\max\varnothing = -\infty$.

- *Conjunction*: $v \lesssim w$ if, and only if, either $v_i = 0$ for some $1 \leq i \leq n$, or $w_i = 1$ for all $1 \leq i \leq n$. This corresponds to the case where a player wants to achieve all her objectives.
- *Disjunction*: $v \lesssim w$ if, and only if, either $v_i = 0$ for all $1 \leq i \leq n$, or $w_i = 1$ for some $1 \leq i \leq n$. The aim here is to satisfy at least one objective.
- *Counting*: $v \lesssim w$ if, and only if, $|\{i \mid v_i = 1\}| \leq |\{i \mid w_i = 1\}|$. The aim is to maximize the number of conditions that are satisfied;
- *Subset*: $v \lesssim w$ if, and only if, $\{i \mid v_i = 1\} \subseteq \{i \mid w_i = 1\}$: in this setting, a player will always struggle to satisfy a larger (for inclusion) set of objectives.

- *Maximize*: $v \lesssim w$ if, and only if, $\max\{i \mid v_i = 1\} \leq \max\{i \mid w_i = 1\}$. The aim is to maximize the highest index of the objectives that are satisfied.
- *Lexicographic*: $v \lesssim w$ if, and only if, either $v = w$, or there is $1 \leq i \leq n$ such that $v_i = 0$, $w_i = 1$ and $v_j = w_j$ for all $1 \leq j < i$.
- *Boolean Circuit*: given a Boolean circuit, with input from $\{0,1\}^{2n}$, $v \lesssim w$ if, and only if, the circuit evaluates 1 on input $v_1 \dots v_n w_1 \dots w_n$.
- *Monotonic Boolean Circuit*: same as above, with the restriction that the input gates corresponding to v are negated, and no other negation appear in the circuit.



(a) Subset preorder (b) Maximize preorder (c) Counting preorder

(d) Lexicographic order

Figure 5.1: Examples of preorders (for $n = 3$): dotted boxes represent equivalence classes for the relation $\sim$, defined as $a \sim b \Leftrightarrow a \lesssim b \wedge b \lesssim a$; arrows represent the preorder relation $\lesssim$, forgetting about $\sim$-equivalent elements

In terms of expressiveness, any preorder over $\{0,1\}^n$ can be given as a Boolean circuit: for each pair (v, w) with $v \lesssim w$, it is possible to construct a circuit whose output is 1 if and only if the input is $v_1 \dots v_n w_1 \dots w_n$; taking the disjunction of all these circuits we obtain a Boolean circuit defining the preorder. Its size can be bounded by $2^{2n+3}n$, which is exponential in general, but all the above examples can be specified with a circuit of polynomial size. In Figure 5.2 we give a polynomial-size Boolean circuit for the subset preorder

A preorder $\lesssim$ is *monotonic* if it is compatible with the subset ordering, i.e. if $\{i \mid v_i = 1\} \subseteq \{i \mid w_i = 1\}$ implies $v \lesssim w$. Hence, a preorder is monotonic if fulfilling more objectives never results in a lower payoff. All our examples of preorders except for the Boolean circuit preorder are monotonic. Moreover, any monotonic preorder can be expressed as a monotonic Boolean circuit: for a pair (v, w) with $v \lesssim w$, we can build a circuit whose output is 1 if, and only if, the input is $v_1 \dots v_n w_1 \dots w_n$. We can require this circuit to have negation at the leaves. Indeed, if the input w_j appears negated, and if $w_j = 0$, then by monotonicity, also the input $(v, \tilde{w})$ is accepted, with $\tilde{w}_i = w_i$ when $i \neq j$ and

$\tilde{w}_j = 1$. Hence the negated input gate can be replaced with `true`. Similarly for positive occurrences of any v_j. Hence any monotonic preorder can be written as a monotonic Boolean circuit. Notice that with Definition 2.2, and Remark 2.2, any Nash equilibrium σ_{Agt} for the subset preorder is also a Nash equilibrium for any monotonic preorder.

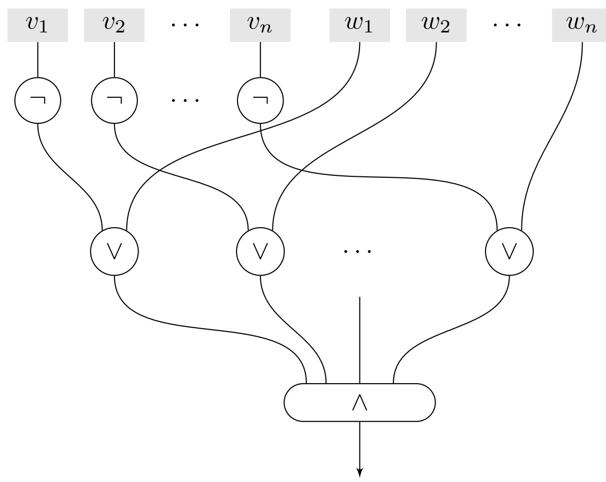


Figure 5.2: Boolean circuit defining the subset preorder

We will focus on ordered objectives where the objectives are reachability or Büchi objectives, since classical objectives such as Muller or parity can be equivalently described with a preorder given by a Boolean circuit over Büchi objectives with polynomial size.

5.2 Ordered Büchi Objectives

We begin with ordered Büchi objectives, for which we prove the results listed in Table 5.1. We will first consider the general case of preorders given as Boolean circuits, and then exhibit several simpler cases. We should also notice that ordered Büchi objectives are prefix-independent: Remark 3.2 therefore applies.

5.2.1 General Case

Theorem 5.1. *For ordered Büchi objectives with preorders given as Boolean circuits, the value, existence and constrained existence problems are* PSPACE-*complete.*

Proof. We fix a game $\mathcal{G} = \langle \mathrm{States}, \mathrm{Agt}, \mathrm{Act}, \mathrm{Allow}, \mathrm{Tab}, (\precsim_A)_{A \in \mathrm{Agt}} \rangle$, and we assume that for each player A, the preorder $\lesssim_A$ is given by a Boolean circuit C_A. The algorithm proceeds by trying all the possible payoffs for the players.

Table 5.1: Summary of the results for Büchi objectives

Preorder	Value	(Constrained) Existence
Maximize	P-c (Sect.5.2.2)	P-c (Sect.5.2.2)
Disjunction	P-c (Sect.5.2.2)	P-c (Sect.5.2.2)
Subset	P-c (Sect. 5.2.3)	P-c (Sect.5.2.2)
Conjunction,Lexicographic	P-c (Sect. 5.2.3)	P-h and in NP (Sect. 5.2.4) [a]
Counting	coNP-c (Sect. 5.2.4)	NP-c (Sect. 5.2.4)
Monotonic Boolean Circuit	coNP-c (Sect. 5.2.4)	NP-c (Sect. 5.2.4)
Boolean Circuit	PSPACE-c (Sect. 5.2.1)	PSPACE-c (Sect. 5.2.1)

[a] The constrained existence problem is actually NP-complete.

Fix such a payoff $(v^A)_{A\in\mathrm{Agt}}$, with $v^A \in \{0,1\}^{n_A}$ for every player A. We will build a circuit D_A which will represent a single objective for player A. Inputs to circuit D_A will be states of the game. This circuit is constructed from C_A as follows: We set all input gates $w_1 \cdots w_n$ of circuit C_A to the value given by payoff v^A; The former input v_i receives the disjunction of all the states in Ω_i; We negate the output. It is not hard to check that the new circuit D_A is such that for every play ρ, $D_A[\mathrm{Inf}(\rho)]$ evaluates to true if and only if $\mathsf{payoff}_A(\rho) \not\lesssim_A v^A$, i.e. if ρ is an improvement for player A.

Circuit D_A is now viewed as a single objective for player A, we write $\mathcal{G}'$ for the new game. We will look for Nash equilibria in this new game, with payoff 0 for each player. Indeed, a Nash equilibrium σ_{Agt} in $\mathcal{G}$ with payoff $(v^A)_{A\in\mathrm{Agt}}$ will be a Nash equilibrium in game $\mathcal{G}'$ with payoff $(0,\dots,0)$. Conversely a Nash equilibrium σ_{Agt} in game $\mathcal{G}'$ with payoff $(0,\dots,0)$ will be a Nash equilibrium in $\mathcal{G}$ as soon as the payoff of its outcome (in $\mathcal{G}$) is $(v^A)_{A\in\mathrm{Agt}}$.

We use the algorithm described in Section 4.6 for computing Nash equilibria with single objectives given as Boolean circuits, and we slightly modify it to take into account the constraint that it has payoff v^A for each player A. This can be done in polynomial space, thanks to Lemma 2.2: it is sufficient to look for plays of the form $\pi \cdot \tau^\omega$ with $|\pi| \le |\mathrm{States}|^2$ and $|\tau| \le |\mathrm{States}|^2$.

PSPACE-hardness was proven for single objectives given as a Boolean circuits (the circuit evaluates by setting to `true` all states that are visited infinitely often, and to `false` all other states) in Section 4.6. This kind of objective can therefore be seen as an ordered Büchi objective with a preorder given as a Boolean circuit. This was actually a consequence of the PSPACE-hardness of the value problem in turn-based games [20]. □

5.2.2 Reduction to a Single Büchi Objective

For some ordered objectives, the preference relation can (efficiently) be reduced to a single objective. For instance, a disjunction of several Büchi objectives can obviously be reduced to a single Büchi objective, by considering the union of the target sets. Formally, we say that an ordered Büchi objective $\omega = \langle (\Omega_i)_{1\le i\le n}, \lesssim \rangle$ is *reducible* to a single Büchi objective if, given any payoff vector v, we can construct in polynomial time a target set $\widehat{T}(v)$ such that for all paths ρ, $v \lesssim \mathsf{payoff}_\omega(\rho)$ if, and only if, $\mathrm{Inf}(\rho) \cap \widehat{T}(v) \neq \varnothing$. It means that *securing* payoff v corresponds to ensuring infinitely many visits to the new target set. Similarly, we say that ω is *co-reducible* to a single Büchi objective if for any vector v we can construct in polynomial time a target set $\widehat{T}(v)$ such that $\mathsf{payoff}_\omega(\rho) \not\lesssim v$ if, and only if $\mathrm{Inf}(\rho) \cap \widehat{T}(v) \neq \varnothing$. It means that *improving* on payoff v corresponds to ensuring infinitely many visits to the new target

We will prove the following proposition, which exploits (co-)reducibility for efficiently solving the various problems.

Proposition 5.2. *For ordered Büchi objectives which are reducible to single Büchi objectives, and where the preorders are non-trivial and monotonic, the value problem is* P*-complete. For ordered Büchi objectives which are co-reducible to single Büchi objectives, and where the preorders are non-trivial and monotonic the existence and constrained existence problems are* P*-complete.*

First note that hardness results follow from hardness of the same problems for single Büchi objectives, proven in Section 4.3.

We now prove the two upper bounds.

Reducibility to single Büchi objectives.

Lemma 5.3. *For ordered Büchi objectives which are reducible to single Büchi objectives, and where the preorders are monotonic, the value problem is in* P.

Proof. We transform the ordered Büchi objectives of the considered player into a single Büchi objective, and use a polynomial-time algorithm [28] to solve the resulting zero-sum Büchi game. □

Co-reducibility to single Büchi objectives.

In the case where the ordered objectives are all co-reducible to single Büchi objectives, we will show how one can adapt the algorithm for single Büchi objectives, which was presented in Section 4.3.

Let $\mathcal{G}$ be a game. We write $\langle (\Omega_i^A)_{1\le i\le n_A}, \lesssim_A \rangle$ for the ordered Büchi objective of player A. We assume that those ordered objectives are all co-reducible to single Büchi objectives.

For every $K \subseteq \mathrm{States}$, we write $v^A(K)$ for the player A payoff of any play π where $\mathrm{Inf}(\pi) = K$. We set $v(K) = (v^A(K))_{A\in\mathrm{Agt}}$. We can first notice that the winning condition of the suspect game $\mathcal{H}(\mathcal{G}, \pi, \mathrm{Allow})$ only depends on $\mathrm{Inf}(\pi)$, we therefore simply write $\mathcal{H}(\mathcal{G}, K, \mathrm{Allow})$.

In Section 4.3, we gave a characterization of Nash equilibria based on strongly-connected subgraphs of the arena of $\mathcal{G}$, when the preference relations only depend on the set of states that are visited infinitely often (Lemma 4.3). Then we proposed a recursive algorithm to compute 'all' such strongly-connected subgraph (Lemma 4.4), and proved (Lemma 4.5) constrained existence problem in polynomial time, provided each set $W(\mathcal{G}, v(K), \text{Allow})$ can be computed in polynomial time and for every player-A payoff w^A, we can construct in polynomial time a set of states S^A such that for every play ρ, $\text{Inf}(\rho) \subseteq S^A$ if, and only if, $\rho \precsim_A w^A$. This last condition obviously holds, since the ordered objectives are co-reducible to single Büchi objectives. The first condition also holds, as stated below.

Lemma 5.4. *The set $W(\mathcal{G}, v(K), \text{Allow})$ can be computed in polynomial time.*

Proof. As the ordered objectives are co-reducible to single Büchi objectives, we can construct in polynomial time target sets $\widehat{T}^A(v(K))$ for each player A. The objective of `Eve` in the suspect game $\mathcal{H}(\mathcal{G}, K, \text{Allow})$ is equivalent to a co-Büchi objective with target set $\{(\widehat{T}^A(v(K), P) \mid A \in P\}$. The winning region can then be determined using the polynomial time algorithm of Lemma 5.3 for Büchi games. □

We therefore deduce the following result:

Corollary 5.5. *For ordered Büchi objectives which are co-reducible to single Büchi objectives, the constrained existence problem is in* P.

Applications.

We will give preorders to which the above applies, allowing to infer several P-completeness results in Table 5.1 (those written with reference "Sect.5.2.2").

We first show that reducibility and co-reducibility coincide when the preorder is total.

Lemma 5.6. *Let $\omega = \langle (\Omega_i)_{1 \leq i \leq n}, \lesssim \rangle$ be an ordered Büchi objective, and assume that $\lesssim$ is total. Then, ω is reducible to a single Büchi objective if, and only if, ω is co-reducible to a single Büchi objective.*

Proof. Let $u \in \{0,1\}^n$ be a vector. If u is a maximal element, the new target set is empty, which satisfies the property for co-reducibility. Otherwise we pick a vector v among the smallest elements that is strictly larger than u. Since the preorder is reducible to a single Büchi objective, there is a target set $\widehat{T}$ that is reached infinitely often whenever the payoff is greater than v. Since the preorder is total and by choice of v, we have $w \not\lesssim u \Leftrightarrow v \lesssim w$. Thus the target set $\widehat{T}$ is visited infinitely often when u is not larger than the payoff. Hence ω is co-reducible to a single Büchi objective.

The proof of the other direction is similar (we only distinguish the case where u is minimal, and then pick v that is the smallest among those that are larger than u). □

Lemma 5.7. *Ordered Büchi objectives with disjunction or maximize preorders are reducible to single Büchi objectives. Ordered Büchi objectives with disjunction, maximize or subset preorders are co-reducible to single Büchi objectives.*

Proof. Let $\omega = \langle (\Omega_i)_{1\leq i\leq n}, \lesssim \rangle$ be an ordered Büchi objective. Assume T_i is the target set for Ω_i.

Assume $\lesssim$ is the disjunction preorder. If the payoff v is different from $\mathbf{0}$ then we define $\widehat{T}(v)$ as the union of all the target sets: $\widehat{T}(v) = \bigcup_{i=1}^{n} T_i$. Then, for every run ρ,

$$\begin{aligned} v \lesssim \mathsf{payoff}_\omega(\rho) \quad &\Leftrightarrow \quad \text{there is some } i \text{ for which } \mathrm{Inf}(\rho) \cap T_i \neq \varnothing \\ &\Leftrightarrow \quad \mathrm{Inf}(\rho) \cap \widehat{T}(v) \neq \varnothing \end{aligned}$$

If the payoff v is $\mathbf{0}$ then we get the expected result with $\widehat{T}(v) = \text{States}$. Disjunction being a total preorder, it is also co-reducible (from Lemma 5.6).

We assume now that $\lesssim$ is the maximize preorder. Given a payoff v, consider the index $i_0 = \max\{i \mid v_i = 1\}$. We then define $\widehat{T}(v)$ as the union of the target sets that are above i_0: $\widehat{T}(v) = \bigcup_{i\geq i_0} T_i$. The following four statements are then equivalent, if ρ is a run:

$$\begin{aligned} v \lesssim \mathsf{payoff}_\omega(\rho) \quad &\Leftrightarrow \quad v \lesssim \mathbf{1}_{\{i \mid \mathrm{Inf}(\rho)\cap T_i \neq \varnothing\}} \\ &\Leftrightarrow \quad i_0 \leq \max\{i \mid \mathrm{Inf}(\rho) \cap T_i \neq \varnothing\} \\ &\Leftrightarrow \quad \exists i \geq i_0.\ \mathrm{Inf}(\rho) \cap T_i \neq \varnothing \end{aligned}$$

Hence ω is reducible, and also co-reducible as it is total, to a single Büchi objective.

Finally, we assume that $\lesssim$ is the subset preorder, and we will show that ω is then co-reducible to a single Büchi objective. Given a payoff v, the new target is the union of the target sets that are not reached infinitely often for that payoff: $\widehat{T}(v) = \bigcup_{\{i \mid v_i = 0\}} T_i$. Then the following statements are equivalent, if ρ is a run:

$$\begin{aligned} \mathsf{payoff}_\omega(\rho) \not\lesssim u \quad &\Leftrightarrow \quad \mathbf{1}_{\{i \mid \mathrm{Inf}(\rho)\cap T_i \neq \varnothing\}} \not\lesssim u \\ &\Leftrightarrow \quad \exists i.\ \mathrm{Inf}(\rho) \cap T_i \neq \varnothing \text{ and } u_i = 0 \\ &\Leftrightarrow \quad \mathrm{Inf}(\rho) \cap \widehat{T}(v) \neq \varnothing \end{aligned}$$

□

Remark. Note that we cannot infer P-completeness of the value problem for the subset preorder since the subset preorder is not total, and ordered objectives with subset preorder are not reducible to single Büchi objectives. Such an ordered objective is actually reducible to a generalized Büchi objective (several Büchi objectives should be satisfied).

5.2.3 Reduction to a Deterministic Büchi Automaton Objective

For some ordered objectives, the preference relation can (efficiently) be reduced to the acceptance by a deterministic Büchi automaton. Formally, we say that an ordered objective $\omega = \langle (\Omega_i)_{1\leq i\leq n}, \lesssim \rangle$ is *reducible* to a deterministic Büchi automaton whenever, given any payoff vector u, we can construct in polynomial time a deterministic Büchi automaton over States which accepts exactly all plays ρ with $u \lesssim \mathsf{payoff}_\omega(\rho)$. For such preorders, we will see that the value problem can be solved efficiently by constructing the product of the deterministic Büchi automaton and the arena of the game. This construction does not help for solving the (constrained) existence problems since the number of players is a parameter of the problem, and the size of the resulting game will then be exponential.

Proposition 5.8. *For ordered Büchi objectives which are reducible to deterministic Büchi automata, the value problem is in* P.

Proof. Given the payoff v^A for player A, the algorithm proceeds by constructing the automaton that recognizes the plays with payoff higher than v^A. By performing the product with the game as described in Section 4.8, we obtain a new game, in which there is a winning strategy if and only if there is a strategy in the original game to ensure payoff v^A. In this new game, player A has a single Büchi objective, so that the existence of a winning strategy can be decided in polynomial time. □

We now give preorders to which the above result applies, that is, which are reducible to deterministic Büchi automata objectives.

Lemma 5.9. *An ordered objective where the preorder is the conjunction is reducible to a deterministic Büchi automaton objective.*

Proof. Let $\omega = \langle (\Omega_i)_{1\le i\le n}, \lesssim\rangle$ be an ordered Büchi objective, where $\lesssim$ is the conjunction. For every $1 \le i \le n$, let T_i be the target set defining the Büchi condition Ω_i. There are only two possible payoffs: either all objectives are satisfied, or one objective is not satisfied. For the second payoff case, any play has a larger payoff: hence the trivial automaton (which accepts all plays) witnesses the property. For the first payoff case, we construct a deterministic Büchi automaton $\mathcal{B}$ as follows. There is one state for each target set, plus one accepting state: $Q = \{q_0, q_1, \dots, q_n\}$; the initial state is q_0, and the unique repeated state is q_n. For all $1 \le i \le n$, the transitions are $q_{i-1} \xrightarrow{s} q_i$ when $s \in T_i$ and $q_{i-1} \xrightarrow{s} q_{i-1}$ otherwise. There are also transitions $q_n \xrightarrow{s} q_0$ for every $s \in$ States. Automaton $\mathcal{B}$ describes the plays that goes through each set T_i infinitely often, hence witnesses the property. It can furthermore be computed in polynomial time. The construction is illustrated in Figure 5.3. □

Lemma 5.10. *An ordered objective where the preorder is the subset preorder is reducible to a deterministic Büchi automaton objective.*

Proof. Let $\omega = \langle (\Omega_i)_{1\le i\le n}, \lesssim\rangle$ be an ordered Büchi objective, where $\lesssim$ is the subset preorder. For every $1 \le i \le n$, let T_i be the target set defining the Büchi condition Ω_i. Fix a payoff u. A play ρ is such that $u \lesssim \mathsf{payoff}_\omega(\rho)$ if, and only if, ρ visits infinitely often all sets T_i with $u_i = 1$. This is then equivalent to the conjunction of all Ω_i's with $u_i = 1$. We therefore apply the construction of Lemma 5.9 and get the expected result. □

Lemma 5.11. *An ordered Büchi objective where the preorder is the lexicographic preorder is reducible to a deterministic Büchi automaton objective.*

Proof. Let $\omega = \langle (\Omega_i)_{1\le i\le n}, \precsim\rangle$ be an ordered Büchi objective, where $\precsim$ is the lexicographic preorder. For every $1 \le i \le n$, let T_i be the target set defining the Büchi condition Ω_i. Let $u \in \{0,1\}^n$ be a payoff vector. We construct the following deterministic Büchi automaton which recognizes the runs whose payoff is greater than or equal to u.

In this automaton there is a state q_i for each i such that $u_i = 1$, and a state q_0 that is both initial and repeated: $Q = \{q_0\} \cup \{q_i \mid u_i = 1\}$. We write $I = \{0\} \cup \{i \mid u_i = 1\}$. For every $i \in I$, we write $\mathsf{succ}(i) = \min(I \setminus \{j \mid j \le i\})$, with the convention that $\min \varnothing = 0$. The transition relation is defined as follows:

- for every $s \in \text{States}$, there is a transition $q_0 \xrightarrow{s} q_{\mathsf{succ}(0)}$;
- for every $i \in I \setminus \{0\}$, we have the following transitions:
  - $q_i \xrightarrow{T_i} q_{\mathsf{succ}(i)}$;
  - $q_i \xrightarrow{T_k \setminus T_i} q_0$ with $k < i$ and $u_k = 0$;
  - $q_i \xrightarrow{s} q_i$ for every $s \in \text{States} \setminus (T_i \cup \bigcup_{k<i, u_k=0} T_k)$.

An example of the construction is given in Figure 5.4.

We now prove correctness of this construction. Consider a path that goes from q_0 to q_0: if the automaton is currently in state q_i, then since the last occurrence of q_0, at least one state for each target set T_j with $j < i$ and $u_j = 1$ has been visited. When q_0 is reached again, either it is because we have seen all the T_j with $u_j = 1$, or it is because the run visited some target T_i with $u_i = 0$ and all the T_j such that $u_j = 1$ and $j < i$; in both cases, the set of targets that have been visited between two visits to q_0 describes a payoff greater than u. Assume the play π is accepted by the automaton; then there is a sequence of q_i as above that is taken infinitely often, therefore $\mathsf{payoff}_\omega(\pi)$ is greater than or equal to u for the lexicographic order.

Conversely assume $v = \mathsf{payoff}_\omega(\pi)$ is greater than or equal to u, that we already read a prefix $\pi_{\le k}$ for some k, and that the current state is q_0. Reading the first symbol in π after position k, the run goes to the state q_i where i is the least integer such that $u_i = 1$. Either the path visits T_i at some point, or it visits a state in a target T_j, with j smaller than i and $v_j = 0$, in which case the automaton goes back to q_0. Therefore from q_0 we can again come back to q_0 while reading the following of π, and the automaton accepts. $\square$

Corollary 5.12. *For ordered Büchi objectives with either of the conjunction, the lexicographic or the subset preorders, the value problem is* P*-complete.*

Proof. The upper bound is a consequence of all the results proven above. Hardness in P already holds for games with a single Büchi objective. $\square$

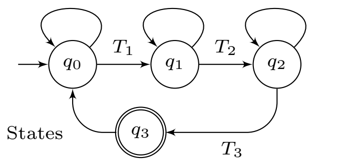


Figure 5.3: The automaton for the conjunction preorder, $n = 3$

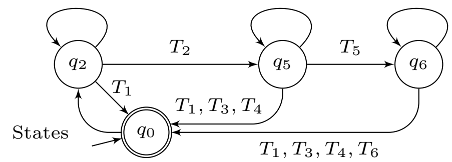


Figure 5.4: The automaton for the lexicographic order, $n = 7$ and $u = (0, 1, 0, 0, 1, 1, 0)$

5.2.4 Monotonic Preorders

We will see in this part that monotonic preorders will lead to more efficient algorithms.

When monotonicity implies memorylessness.

We say that a strategy σ is *memoryless* (resp. from state s_0) if there exists a function $f\colon \text{States} \to \text{Act}$ such that $\sigma(h \cdot s) = f(s)$ for every $h \in \text{Hist}$ (resp. for every $h \in \text{Hist}(s_0)$). A strategy profile is said memoryless whenever all strategies of single players are memoryless. We show that when the preorders (in ordered Büchi objectives) are monotonic, our problems are also easier than in the general case. This is because we can find memoryless *trigger strategies*: we recall that a strategy profile σ_{Agt} is a trigger strategy for a play π from state s if, for any strategy σ'_A of any player $A \in \text{Agt}$, the path π is at least as good as the outcome of $\sigma_{\text{Agt}}[A \mapsto \sigma'_A]$ from s (that is, $\text{Out}(s, \sigma_{\text{Agt}}[A \mapsto \sigma'_A]) \precsim_A \pi$).

Lemma 5.13. *Let $\mathcal{H}$ be a turn-based two-player game. Call* `Eve` *one player, and let $\sigma_\exists$ be a strategy for* `Eve`*, and s_0 be a state of $\mathcal{H}$. There is a memoryless strategy $\sigma'_\exists$ such that for every $\rho' \in \text{Out}_{\mathcal{H}}(s_0, \sigma'_\exists)$, there exists $\rho \in \text{Out}_{\mathcal{H}}(s_0, \sigma_\exists)$ such that $\text{Inf}(\rho') \subseteq \text{Inf}(\rho)$.*

Proof. This proof is by induction on the size of the set $S(\sigma_1) = \{(s, m) \mid \exists h \in \text{Hist}(\sigma_1).\ \sigma_1(h) = m \text{ and } \text{last}(h) = s\}$. If its size is the same as that of $\{s \mid \exists h \in \text{Hist}(\sigma_1).\ \text{last}(h) = s\}$ then the strategy is memoryless. Otherwise, let s be a state at which σ_1 takes several different actions (i.e., $|(\{s\} \times \text{Act}) \cap S(\sigma_1)| > 1$).

We will define a new strategy σ'_1 that takes fewer different actions in s and such that for every outcome of σ'_1, there is an outcome of σ_1 that visits (at least) the same states infinitely often.

If σ is a strategy and h a history, we let $\sigma \circ h\colon h' \mapsto \sigma(h \cdot h')$ for any history h'. For every m such that $(s, m) \in S(\sigma_1)$ we define the set $H_m = \{h \in \text{Hist}(\sigma_1) \mid \text{last}(h) = s \text{ and } \sigma_1(h) = m\}$, and for every h, $h^{-1} \cdot H_m = \{h' \mid h \cdot h' \in H_m\}$.

We pick m such that H_m is not empty.

- Assume that there is $h_0 \in \text{Hist}(\sigma_1)$ with $\text{last}(h_0) = s$, such that $h_0^{-1} \cdot H_m$ is empty. We define a new strategy σ'_1 as follows. If h is an history which does not visit s, then $\sigma'_1(h) = \sigma_1(h)$. If h is an history which visits s, then decompose h as $h' \cdot h''$ where $\text{last}(h') = s$ is the first visit to s and define $\sigma'_1(h) = \sigma_1(h_0 \cdot h'')$. Then, strategy σ'_1 does not use m at state s, and therefore at least one action has been "removed" from the strategy. More precisely, $|(\{s\} \times \text{Act}) \cap S(\sigma'_1)| \leq |(\{s\} \times \text{Act}) \cap S(\sigma_1)| - 1$. Furthermore the conditions on infinite states which are visited infinitely often by outcomes of σ'_1 is also satisfied.

- Otherwise for any $h \in \text{Hist}(\sigma_1)$ with $\text{last}(h) = s$, $h^{-1} \cdot H_m$ is not empty. We will construct a strategy σ'_1 which plays m at s. Let h be an history, we first define the extension $e(h)$ inductively in that way:

  - $e(\varepsilon) = \varepsilon$, where ε is the empty history;
  - $e(h \cdot s) = e(h) \cdot h'$ where $h' \in (e(h))^{-1} \cdot H_m$;
  - $e(h \cdot s') = e(h) \cdot s'$ if $s' \neq s$.

  We extend the definition of e to infinite outcomes in the natural way: $e(\rho)_i = e(\rho_{\leq i})_i$. We then define the strategy $\sigma'_1 \colon h \mapsto \sigma_1(e(h))$. We show that if ρ is an outcome of σ'_1, then $e(\rho)$ is an outcome of σ_1. Indeed assume h is a finite outcome of σ'_1, that $e(h)$ is an outcome of σ_1 and $\text{last}(h) = \text{last}(e(h))$. If $h \cdot s$ is an outcome of σ'_1, by construction of e, $e(h \cdot s) = e(h) \cdot h'$, such that $\text{last}(h') = s$, and h' is an outcome of $\sigma_1 \circ e(h)$ and as $e(h)$ is an outcome of σ_1 by hypothesis, that means that $e(h \cdot s)$ is an outcome of σ_1. If $h \cdot s'$ with $s' \neq s$ is an outcome of σ'_1, $e(h \cdot s') = e(h) \cdot s'$, $s' \in \text{Tab}(\text{last}(h), \sigma'_1(h))$, and $\sigma'_1(h) = \sigma_1(e(h))$. Using the hypothesis $\text{last}(h) = \text{last}(e(h))$, and $e(h)$ is an outcome of σ_1, therefore $e(h \cdot s')$ is an outcome of σ_1. This shows that if ρ is an outcome of σ'_1 then $e(\rho)$ is an outcome of σ_1. The property on states visited infinitely often follows. Several moves have been removed from the strategy at s (since the strategy is now memoryless at s, playing m).

In all cases we have $S(\sigma'_1)$ strictly included in $S(\sigma_1)$, and an inductive reasoning entails the result. □

Lemma 5.14. *For ordered Büchi objectives with monotonic preorders, if there is a trigger strategy that respect the constraint* Allow *for some play π from s, then there is a memoryless winning strategy for* `Eve` *in $\mathcal{H}(\mathcal{G}, \pi, \text{Allow})$ from state (s, Agt).*

Proof. Assume there is a trigger strategy for π. We have seen in Lemma 3.3 that there is then a winning strategy $\sigma_\exists$ in game $\mathcal{H}(\mathcal{G}, \pi, \text{Allow})$ for `Eve`. Consider the memoryless strategy $\sigma'_\exists$ constructed as in Lemma 5.13. Let ρ' be an outcome of $\sigma'_\exists$, there is an outcome ρ of $\sigma_\exists$ such that $\text{Inf}(\rho') \subseteq \text{Inf}(\rho)$. As $\sigma_\exists$ is winning in $\mathcal{H}(\mathcal{G}, \pi, \text{Allow})$, for every $A \in L(\rho)$, $\pi_1(\rho) \precsim_A \pi$. We assume the Büchi conditions are given by the target sets $(T_i^A)_{A,i}$. For each player A, $\{i \mid \text{Inf}(\pi_1(\rho')) \cap T_i^A\} \subseteq \{i \mid \text{Inf}(\pi_1(\rho)) \cap T_i^A\}$. As the preorder is monotonic the payoff of $\pi_1(\rho')$ is smaller than that of $\pi_1(\rho)$: $\pi_1(\rho') \precsim_A \pi_1(\rho)$. So the play is winning for any player A and $\sigma'_\exists$ is a memoryless winning strategy in game $\mathcal{H}(\mathcal{G}, \pi, \text{Allow})$ for `Eve`. □

Lemma 5.15. *For ordered Büchi objectives with monotonic preorders given by monotonic Boolean circuits, given a path π, we can decide in polynomial time if a memoryless strategy for* `Eve` *in $\mathcal{H}(\mathcal{G}, \pi, \text{Allow})$ is winning.*

Proof. Let $\sigma_\exists$ be a memoryless strategy in $\mathcal{H}(\mathcal{G}, \pi, \text{Allow})$ for `Eve`. By keeping only the edges that are taken by $\sigma_\exists$, we define a subgraph of the game. We can compute in polynomial time the strongly connected components of this graph. If one component is reachable and does not satisfy the objective of `Eve`, then the strategy is not winning. Conversely if all the reachable strongly connected components satisfy the winning condition of `Eve`, since the preorder is monotonic, $\sigma_\exists$ is a winning strategy. Notice that since the preorder is given as a Boolean circuit, we can check in polynomial time whether a strongly connected component is winning or not. Globally the algorithm is therefore polynomial-time. □

Main general result.

The previous analysis allows to get the following results.

Proposition 5.16. *For ordered Büchi objectives with monotonic given by monotonic Boolean circuits, the value problem is in* coNP*, and the existence and constrained existence problems are in* NP*. Completeness holds in both cases for preorders given by monotonic Boolean circuits or for the counting preorder.* NP*-completeness also holds for the constrained existence problem for preorders $\lesssim$ with furthermore an element v such that for every v', $v' \neq \mathbf{1} \Leftrightarrow v' \lesssim v$,*[1]

Proofs of the upper bounds.

We show that the value problem is in coNP for ordered Büchi objectives with monotonic preorders given by monotonic Boolean circuits.

For the value problem, we can make the concurrent game turn-based: since player A must win against any strategy of the coalition $P = \text{Agt} \setminus \{A\}$, she must also win in the case where the opponents' strategies can adapt to what A plays. This turn-based game is determined, so that there is a strategy σ whose outcomes are always better (for A) than v^A if and only if, for any strategy σ' of coalition P, there is an outcome with payoff (for A) better than v^A. If there is a counterexample to this fact, then thanks to Lemma 5.13 there is one with a memoryless strategy σ'. The coNP algorithm proceeds by checking that all the memoryless strategies of coalition P have an outcome better than v^A, which is achievable in polynomial time, with a method similar to Lemma 5.15.

We show now that the constrained existence problem is in NP for ordered Büchi objectives given by monotonic Boolean circuits.

The algorithm for the constrained existence problem proceeds by guessing:

[1] To be fully formal, the preorder $\lesssim$ is in fact a family $(\lesssim_n)_{n\in\mathbb{N}}$ (where $\lesssim_n$ compares two vectors of size n), and this condition should be stated as "*if, for all n, there is an element $v_n \in \{0,1\}^n$, v_n, such that for all $v' \in \{0,1\}^n$, it holds $v' \neq \mathbf{1} \Leftrightarrow v' \lesssim v_n$* ".

- the payoff for each player,
- a play of the form $\pi \cdot \tau^\omega$, where $|\pi| \leq |\text{States}|^2$ and $|\tau| \leq |\text{States}|^2$,
- an under-approximation W of the set of winning states in $\mathcal{H}(\mathcal{G}, \pi{\cdot}\tau^\omega, \text{Allow})$
- a memoryless strategy profile σ_{Agt} in $\mathcal{H}(\mathcal{G}, \pi \cdot \tau^\omega, \text{Allow})$.

We check that σ_{Agt} is a witness for the fact that the states in W are winning; thanks to Lemma 5.15, this can be done in polynomial time. We also verify that the play $\pi \cdot \tau^\omega$ has the expected payoff, that the payoff satisfies the constraints, and that it never gets out of W. If these conditions are fulfilled, then the play $\pi \cdot \tau^\omega$ meets the conditions of Theorem 3.4, and there is a Nash equilibrium with outcome $\pi \cdot \tau^\omega$. Lemmas 5.14 and 2.2 ensure that if there is a Nash equilibrium, we can find it this way.

Proofs of the hardness results.

Lemma 5.17. *For ordered Büchi objectives with the counting preorder, the value problem is* coNP*-hard.*

Proof. We reduce (the complement of) 3SAT into the value problem for two-player turn-based games with Büchi objectives with the counting preorder. Consider an instance

$$\phi = C_1 \wedge \cdots \wedge C_m$$

with $C_j = \ell_{j,1} \vee \ell_{j,2} \vee \ell_{j,3}$, over a set of variables $\{x_1, \ldots, x_n\}$. With ϕ, we associate a two-player turn-based game $\mathcal{G}$. Its set of states is made of

- a set containing the unique initial state $V_0 = \{s_0\}$,
- a set of two states $V_k = \{x_k, \neg x_k\}$ for each $1 \leq k \leq n$,
- and a set of three states $V_{n+j} = \{t_{j,1}, t_{j,2}, t_{j,3}\}$ for each $1 \leq j \leq m$.

Then, for each $0 \leq l \leq n+m$, there is a transition between any state of V_l and any state of V_{l+1} (assuming $V_{n+m+1} = V_0$).

The game involves two players: player B owns all the states, but has no objectives (she always loses). Player A has a set of Büchi objectives defined by $T^A_{2\cdot k} = \{x_k\} \cup \{t_{j,p} \mid \ell_{j,p} = x_k\}$, $T^A_{2\cdot k+1} = \{\neg x_k\} \cup \{t_{j,p} \mid \ell_{j,p} = \neg x_k\}$, for $1 \leq k \leq n$. Notice that at least n of these objectives will be visited infinitely often along any infinite play. We prove that if the formula is not satisfiable, then at least $n+1$ objectives will be fulfilled, and conversely.

Assume the formula is satisfiable, and pick a witnessing valuation v. We define a strategy σ_B for B that "follows" valuation v: from states in V_{k-1}, for any $1 \leq k \leq n$, the strategy plays towards x_k if $v(x_k) = \texttt{true}$ (and to $\neg x_k$ otherwise). Then, from a state in V_{n+l-1} with $1 \leq l \leq m$, it plays towards one of the $t_{j,p}$ that evaluates to true under v (the one with least index p, say). This way, the number of targets of player A that are visited infinitely often is n.

Conversely, pick a play in $\mathcal{G}$ s.t. at most (hence exactly) n objectives of A are fulfilled. In particular, for any $1 \leq k \leq n$, this play never visits one of x_k

and $\neg x_k$, so that it defines a valuation v over $\{x_1, \ldots, x_n\}$. Moreover, any state of V_{n+l}, with $1 \leq l \leq p$, that is visited infinitely often must correspond to a literal that is made true by v, as otherwise this would make one more objective that is fulfilled for A. As a consequence, each clause of ϕ evaluates to true under v, and the result follows. □

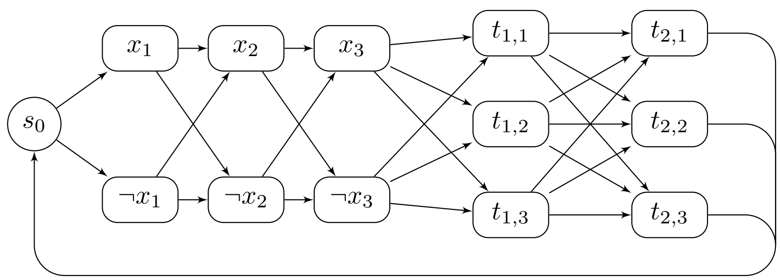


Figure 5.5: The game $\mathcal{G}$ associated with formula ϕ of 5.1

Example 9. We illustrate the construction of the previous proof in Figure 5.5 for the formula

$$\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)\,. \tag{5.1}$$

The targets for player A are $T_1 = \{x_1, t_{1,1}\}$, $T_2 = \{\neg x_1, t_{2,1}\}$, $T_3 = \{x_2, t_{1,2}, t_{2,2}\}$, $T_42 = \{\neg x_2\}$, $T_5 = \{x_3\}$, $T_6 = \{\neg x_3, t_{1,3}, t_{2,3}\}$. Player A cannot ensure visiting infinitely often four target sets, therefore the formula is satisfiable.

Lemma 5.18. *For ordered Büchi objectives with the counting preorder, the existence problem is* NP*-hard.*

Proof. Let $\mathcal{G}$ be the game we constructed for Lemma 5.17. We construct the game $\mathcal{G}''$ from $\mathcal{G}$ as described in Section 2.4.3. The preference in $\mathcal{G}'$ can still be described with ordered Büchi objectives and the counting preorder: the only target set of B is $\{s_1\}$ and we add s_1 to n different targets of A, where n is the number of variables as in Lemma 5.17. From Lemma 2.4 there is a Nash equilibrium in $\mathcal{G}''$ from s_0 if, and only if, A cannot ensure visiting at least $n+1$ targets infinitely often. Hence the existence problem is NP-hard. □

This proves also NP-hardness for the constrained existence problem for ordered Büchi objectives with the counting preorder. Hardness results for preorders given by monotonic Boolean circuits follow from the above since the counting preorder is a special case of preorder given as a monotonic Boolean circuit.

Lemma 5.19. *For ordered Büchi objectives with a monotonic preorder for which there is an element v such that for every v', $v' \neq \mathbf{1} \Leftrightarrow v' \lesssim v$, the existence problem with constrained outcomes for turn-based games is* NP*-hard.*

Proof. Let us consider a formula $\phi = C_1 \wedge \cdots \wedge C_m$ For each variable x_i, our game has one player B_i and three states s_i, x_i and $\neg x_i$. The objectives of B_i are the sets $\{x_i\}$ and $\{\neg x_i\}$. Transitions go from each s_i to x_i and $\neg x_i$, and from x_i and $\neg x_i$ to s_{i+1} (with $s_{n+1} = s_0$). Finally, an extra player A has full control of the game (i.e., she owns all the states) and has n objectives, defined by $T_i^A = \{\ell_{i,1}, \ell_{i,2}, \ell_{i,3}\}$ for $1 \leq i \leq n$. The construction is illustrated in Figure 5.6.

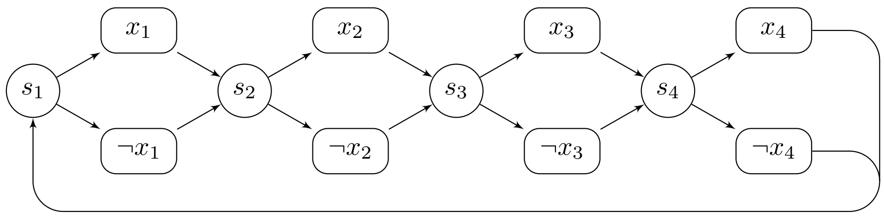


Figure 5.6: The Büchi game for a formula with 4 variables

We show that formula ϕ is satisfiable if, and only if, there is a Nash equilibrium where each player B_i gets payoff β_i satisfying $\beta_i \lesssim v$ (hence $\beta_i \neq (1,1)$), and player A gets payoff $\mathbf{1}$.

First assume that the formula is satisfiable, and pick a witnessing valuation u. By playing according to u, player A can satisfy all of her objectives (hence she cannot improve her payoff, since the preorder is monotonic). Since she alone controls all the game, the other players cannot improve their payoff, so that this is a Nash equilibrium. Moreover, since A plays memoryless, only one of x_i and $\neg x_i$ is visited for each i, so that the payoff β_i for B_i satisfies $\beta_i \lesssim v$. Conversely, if there is a Nash equilibrium with the desired payoff, then by hypothesis, exactly one of each x_i and $\neg x_i$ is visited infinitely often (so that the payoff for B_i is not $(1,1)$), which defines a valuation u. Since in this Nash equilibrium, player A satisfies all its objectives, one state of each target is visited, which means that under valuation u, formula ϕ evaluates to true.

□

Applications.

We now describe examples of preorders which satisfy the conditions on the existence of an hypotheses of an element v such that $v' \neq \mathbf{1} \Leftrightarrow v' \lesssim v$.

Lemma 5.20. *Conjunction, counting and lexicographic preorders have an element v such that $v' \neq \mathbf{1} \Leftrightarrow v' \lesssim v$.*

Proof. Consider $v = (1, \ldots, 1, 0)$, and $v' \neq \mathbf{1}$. For conjunction, there is i such that $v'_i = 0$, so $v' \lesssim v$. For counting, $|\{i \mid v'_i = 1\}| < n$, so $v' \lesssim v$. For the lexicographic preorder, let i be the smallest index such that $v'_i = 0$, and either $v_i = 1$ and $v_j = v'_j$ for all $j < i$, or for all $j \in \{1, \ldots, n\}$, $v_j = v'_j$. In both cases $v' \lesssim v$. □

As a consequence, the result of Lemma 5.19 applies in particular to the conjunction and lexicographic preorders, for which the constrained existence problem is thus NP-complete.

5.3 Ordered Reachability Objectives

We now assume that all considered objectives are reachability objectives, that is, if we consider an ordered objective $\langle(\Omega_i)_{1\leq i\leq n}, \lesssim\rangle$, then all Ω_i's are reachability objectives. In the general case where the preorders are given as Boolean circuits, we show that the various decision problems are PSPACE-complete, and we even notice that the hardness result holds for several simpler preorders. We then improve this result in a number of cases. The results are summarized in Table 5.2.

Table 5.2: Summary of the results for reachability objectives

Preorder	Value	(Constrained) Existence
Disjunction, Maximize	P-c (Sect. 5.3.2)	NP-c (Sect. 5.3.2)
Subset	PSPACE-c (Sect. 5.3.1)	NP-c (Sect. 5.3.2)
Conjunction	PSPACE-c (Sect. 5.3.1)	PSPACE-c (Sect. 5.3.1)
Counting	PSPACE-c (Sect. 5.3.1)	PSPACE-c (Sect. 5.3.1)
Lexicographic	PSPACE-c (Sect. 5.3.1)	PSPACE-c (Sect. 5.3.1)
(Monotonic) Boolean Circuit	PSPACE-c (Sect. 5.3.1)	PSPACE-c (Sect. 5.3.1)

5.3.1 General Case

Reduction to a game with ordered Büchi objectives.

We show how to transform a game $\mathcal{G}$ with preferences given by Boolean circuits over reachability objectives into a new game $\mathcal{G}'$, with preferences given by Boolean circuits over Büchi objectives. Although the size of $\mathcal{G}'$ will be exponential, circuit order with Büchi objectives define prefix-independent preference relations and thus checking condition 3 of Theorem 3.4 can be made more efficient.

States of $\mathcal{G}'$ remember the states of $\mathcal{G}$ which already occurred. Its set of states is $\text{States}' = \text{States} \times 2^{\text{States}}$. The transitions are $(s, S) \to (s', S')$ when there is a transition $s \to s'$ in $\mathcal{G}$ and $S' = S \cup \{s'\}$. We keep the same circuits to define the preference relations, but the reachability objectives are transformed into Büchi objectives: a target set T is transformed into $T' = \{(s, S) \mid S \cap T \neq \varnothing\}$. Although the game has exponential size, the preference relations only depend on the strongly connected components the path ends in, so that we will be able to use a special algorithm, that we will describe after this lemma.

We define the relation $s \lhd s'$ over states of $\mathcal{G}$ and $\mathcal{G}'$ if, and only if, $s' = (s, S)$ with $S \subseteq \text{States}$.

We define $\text{Allow}'((s,S),m_{\text{Agt}}) = \text{Allow}(s,m_{\text{Agt}})$ and we prove that $\lhd$ is a game simulation, in the sense of Section 3.3.

Lemma 5.21. *The relation $\lhd$ (resp. $\lhd^{-1}$) is a game simulation between $\mathcal{G}$ and $\mathcal{G}'$ with respect to the constraints* Allow *and* Allow$'$*, and it is preference-preserving from $(s_0,(s_0,\{s_0\}))$ (resp. $((s_0,\{s_0\}),s_0)$).*

Proof. Let m_{Agt} be an allowed move from s, then m_{Agt} is also allowed from (s,S), let $t = \text{Tab}(s,m_{\text{Agt}})$, $\text{Tab}'((s,S),m_{\text{Agt}}) = (t, S\cup\{t\})$, therefore $\text{Tab}(s,m_{\text{Agt}}) \lhd \text{Tab}'(s',m_{\text{Agt}})$. Let (t,S') be a state of $\mathcal{G}'$, we have that $t \lhd (t,S')$. If $S' = S\cup\{t\}$ then $\text{Susp}((s,t),m_{\text{Agt}}) = \text{Susp}(((s,S),(t,S')),m_{\text{Agt}})$, and otherwise $\text{Susp}(((s,S),(t,S')),m_{\text{Agt}}) = \varnothing$. In both cases, condition (2) that defines a game simulation is obviously satisfied.

In the other direction, let $(s', S\cup\{s'\}) = \text{Tab}((s,S),m_{\text{Agt}})$, we have that $s' \lhd (s', S\cup\{s'\})$. Let $t \in \text{States}$, $t \lhd (t, S\cup\{t\})$ and $\text{Susp}((s,t),m_{\text{Agt}}) = \text{Susp}(((s,S),(t,S\cup\{t\})),m_{\text{Agt}})$.

Let ρ and ρ' be two paths, from s_0 and $(s_0,\{s_0\})$ respectively, and such that $\rho \lhd \rho'$. We show preference preservation, by showing that ρ reaches target set T if and only if ρ' visits infinitely often T'. If ρ visits some state $s \in T$, then from that point, states visited by ρ' are of the form (s',S') with $s \in S'$, all these states are in T', therefore ρ' visits infinitely often T'. Conversely, if ρ' visits infinitely often T', some state of T' have been visited by ρ. $\square$

As a corollary, and thanks to Proposition 3.5, we get that there is a correspondence between Nash equilibria in $\mathcal{G}$ and Nash equilibria in $\mathcal{G}'$.

Corollary 5.22. *If there is a Nash equilibrium σ_{Agt} in $\mathcal{G}$ from s_0 which respects the constraint* Allow *, then there is a Nash equilibrium σ'_{Agt} in $\mathcal{G}'$ from $(s_0,\{s_0\})$ which respect the constraint* Allow$'$ *and such that $\text{Out}_{\mathcal{G}}(s_0,\sigma_{\text{Agt}}) \lhd \text{Out}_{\mathcal{G}'}((s_0,\{s_0\}),\sigma'_{\text{Agt}})$. And vice-versa: if there is a Nash equilibrium σ'_{Agt} in $\mathcal{G}'$ from $(s_0,\{s_0\})$ which respect the constraint* Allow$'$ *, then there is a Nash equilibrium σ_{Agt} in $\mathcal{G}$ from s_0 which respect the constraint* Allow *and such that $\text{Out}_{\mathcal{G}'}((s_0,\{s_0\}),\sigma'_{\text{Agt}}) \lhd^{-1} \text{Out}_{\mathcal{G}}(s_0,\sigma_{\text{Agt}})$.*

Note that, if $\text{Out}_{\mathcal{G}}(s_0,\sigma_{\text{Agt}}) \lhd \text{Out}_{\mathcal{G}'}((s_0,\{s_0\}),\sigma'_{\text{Agt}})$, then $\text{Out}_{\mathcal{G}}(s_0,\sigma_{\text{Agt}})$ satisfies the reachability objective defined with target set T if, and only if, $\text{Out}_{\mathcal{G}'}((s_0,\{s_0\}),\sigma'_{\text{Agt}})$ satisfies the Büchi objective with target set $T' = \{(s,S) \mid S\cap T \neq \varnothing\}$. From this strong correspondence between $\mathcal{G}$ and $\mathcal{G}'$, we get that it is sufficient to look for Nash equilibria in game $\mathcal{G}'$.

How to efficiently solve the suspect game of $\mathcal{G}'$

In game $\mathcal{G}'$, preference relations are prefix-independent. Applying Remark 3.2 the preference relation in the suspect game is then also prefix-independent, and the payoff of a play only depends on which strongly-connected component the path ends in. We now give an alternating algorithm which runs in polynomial time and solves the game $\mathcal{H}(\mathcal{G}',\pi',\text{Allow}')$, where π' is an infinite path in $\mathcal{G}'$.

Lemma 5.23. *The winner of* $\mathcal{H}(\mathcal{G}', \pi', \mathrm{Allow}')$ *can be decided by an alternating algorithm which runs in time polynomial in the size of* $\mathcal{G}$.

Proof. Let C^A be the circuit defining the preference relation of player A. Let $\rho = (s_i, S_i)_{i\geq 0}$ be a path in $\mathcal{G}'$, the sequence $(S_i)_{i\geq 0}$ is non-decreasing and converges to a limit $S(\rho)$. We have $\mathsf{payoff}_A(\rho) = \mathbf{1}_{\{i | T^i_A \cap S(\rho) = \varnothing\}}$. Therefore the winning condition of `Eve` in $\mathcal{H}(\mathcal{G}', \pi', \mathrm{Allow}')$ for a play ρ only depends on the limits $L(\rho)$ and $S(\pi_1(\rho))$. It can be described by a single Büchi condition given by the target set $T = \{((s, S), P) \mid \forall A \in P.\ C^A[v^A(S), w^A] \text{ evaluates to } \mathsf{true}\}$ where $v^A(S) = \mathbf{1}_{\{i | T^i_A \cap S = \varnothing\}}$ and $w^A = \mathsf{payoff}_A(\pi')$. We now describe the algorithm.

Initially the current state is set to $((s_0, \{s_0\}), \mathrm{Agt})$. We also keep a list of the states which have been visited, and we initialize it with $\mathrm{Occ} \leftarrow \{(s_0, \{s_0\}), \mathrm{Agt}\}$. Then,

- if the current state is $((s, S), P)$, the algorithm existentially guesses a move m_{Agt} of `Eve` and we set $t = ((s, S), P, m_{\mathrm{Agt}})$;
- otherwise if the current state is of the form $((s, S), P, m_{\mathrm{Agt}})$, it universally guesses a state s' which corresponds to a move of `Adam` and we set $t = ((s', S \cup \{s'\}), P \cap \mathrm{Susp}((s, s'), m_{\mathrm{Agt}}))$.

If t was already seen (that is, if $t \in \mathrm{Occ}$), the algorithm returns true when $t \in T$ and false when $t \notin T$, otherwise the current state is set to t, and we add t to the list of visited states: $\mathrm{Occ} \leftarrow \mathrm{Occ} \cup \{t\}$, and we repeat this step. Because we stop when the same state is seen, the algorithm stops after at most $\ell + 1$ steps, where ℓ is the length of the longest acyclic path. Since the size of S can only increase and the size of P only decrease, we bound ℓ by $|\mathrm{States}|^2 \cdot |\mathrm{Agt}|$.

We now prove the correctness of the algorithm. First, $\mathcal{H}(\mathcal{G}', \pi', \mathrm{Allow}')$ is a turn-based Büchi game, which is a special case of parity game. Parity games are known to be determined with memoryless strategies [43, 24], hence $\mathcal{H}(\mathcal{G}', \pi', \mathrm{Allow}')$ is determined with memoryless strategies.

If the algorithm answers true, then there exist a strategy $\sigma_\exists$ of `Eve` such that for all the strategies $\sigma_\forall$ of `Adam`, any outcome ρ of $\mathrm{Out}(\sigma_\exists, \sigma_\forall)$ is such that there exist $i < j \leq \ell + 1$ with $\rho_i = \rho_j \in T$ and all ρ_k with $k < j$ are different. We extend this strategy $\sigma_\exists$ to a winning strategy $\sigma'_\exists$ for `Eve`. We do so by ignoring the loops we see in the history, formally we inductively define a reduction r of histories by:

- $r(\varepsilon) = \varepsilon$;
- if $((s, S), P)$ does not appear in $r(h)$ then $r(h \cdot ((s, S), P)) = r(h) \cdot ((s, S), P)$;
- otherwise $r(h \cdot ((s, S), P)) = r(h)_{\leq i}$ where i is the smallest index such that $r(h)_i = ((s, S), P)$.

We then define $\sigma'_\exists$ for any history h by $\sigma'_\exists(h) = \sigma_\exists(r(h))$.

We show by induction that if h is a history compatible with $\sigma'_\exists$ from the state $((s_0, \{s_0\}), \mathrm{Agt})$ then $r(h)$ is compatible with $\sigma_\exists$ from $((s_0, \{s_0\}), \mathrm{Agt})$. It is true when $h = ((s_0, \{s_0\}), \mathrm{Agt})$, now assuming it holds for all history of length $\leq k$, we show it for history of length $k+1$. Let $h \cdot s$ be a history of length $k+1$ compatible with $\sigma'_\exists$. By hypothesis $r(h)$ is compatible with h and since $\sigma'_\exists(h) = \sigma_\exists(r(h))$, $r(h) \cdot s$ is compatible with $\sigma_\exists$. If $r(h \cdot s) = r(h) \cdot s$ then $r(h \cdot s)$ is compatible with $\sigma_\exists$. Otherwise $r(h \cdot s)$ is a prefix of $r(h)$ and therefore of length $\leq k$, we can apply the induction hypothesis to conclude that $r(h \cdot s)$ is compatible with $\sigma_\exists$.

We now show that the strategy $\sigma'_\exists$ that we defined, is winning. Let ρ be a possible outcome of $\sigma'_\exists$, let $i < j$ be the first indexes such that $\rho_i, \rho_j \in (\mathrm{States} \times S(\rho)) \times L(\rho)$ and $\rho_i = \rho_j$. Because there is no repetition between i and $j-1$: $r(\rho_{\leq j-1}) = r(\rho_{\leq i-1})\rho_i \cdots \rho_{j-1}$. We have that $\sigma_\exists(r(\rho_{\leq i-1})\rho_i \cdots \rho_{j-1}) = \sigma'_\exists(\rho_{j-1})$. From this move, ρ_j is a possible next state, so $r(\rho_{\leq i-1})\rho_i \cdots \rho_j$ is a possible outcome of $\sigma_\exists$. As $\rho_i = \rho_j$ and all other states are different, by the hypothesis on $\sigma_\exists$ we have that $\rho_j \in T$. This shows that ρ ultimately loops in states of T and therefore ρ is a winning run for `Eve`.

Reciprocally, if `Eve` has a winning strategy, she has a memoryless one $\sigma_\exists$ since this is a Büchi game. We can see this strategy as an oracle for the various existential choices in the algorithm. Consider some universal choices in the algorithm, it corresponds to a strategy $\sigma_\forall$ for `Adam`. The branch corresponding to $(\sigma_\exists, \sigma_\forall)$ ends the first time we encounter a loop, we write this history $h \cdot h'$ with $\mathrm{last}(h') = \mathrm{last}(h)$. Since the strategy $\sigma_\exists$ is memoryless, $h \cdot h'^\omega$ is a possible outcome. Since it is winning, $\mathrm{last}(h')$ is in T and therefore the branch is accepting. This being true for all the branches given by the choices of $\sigma_\exists$, the algorithm answers `true`. ☐

Main general result when preorders are given as Boolean circuits

Proposition 5.24. *For ordered reachability objectives with preorders given by Boolean circuits, the value, existence and constrained existence problems are in* PSPACE. *For ordered reachability objectives with preorders having* $\mathbf{1}$ *as a unique maximal element, the value problem is* PSPACE*-hard (even for two-player turn-based games). If moreover the preorders have an element* v *such that for every* v', $v' \neq \mathbf{1} \Leftrightarrow v' \lesssim v$, *then the existence and constrained existence problems are* PSPACE*-hard (even for two-player games).*

PSPACE-completeness therefore holds for conjunction, counting and lexicographic preorders (thanks to the fact that $\mathbf{1}$ is the unique maximal element for theses orders and to Lemma 5.20). As conjunction (for instance) can easily be encoded using a (monotonic) Boolean circuit in polynomial time, the hardness results are also valid if the preorder is given by a (monotonic) Boolean circuit. On the other hand, the disjunction and maximize preorders do not have a unique maximal element, so the hardness result does not carry over to these preorders. In the same way, for the subset preorder, there is no v such that $v' \neq \mathbf{1} \Leftrightarrow v' \lesssim v$, so the hardness result does not apply. We prove later (in Section 5.3.2) that in these special cases, the complexity is actually lower.

It remains to complete the proof of proposition 5.24. We do so in the next paragraphs.

Proof of the PSPACE upper bounds.

We describe a PSPACE algorithm for solving the constrained existence problem. The algorithm proceeds by trying all plays π in $\mathcal{G}$ of the form described in Lemma 2.2. This corresponds to a (unique) play π' in $\mathcal{G}'$. We check that π' has a payoff satisfying the constraints, and that there is a path ρ in $\mathcal{H}(\mathcal{G}', \pi', \mathrm{Allow}')$, whose projection is π', along which `Adam` obeys `Eve`, and which stays in the winning region of `Eve`. This last step is done by using the algorithm of Lemma 5.23 on each state ρ goes through. All these conditions are satisfied exactly when the conditions of Theorem 3.4 are satisfied, in which case there is a Nash equilibrium within the given bounds.

The PSPACE upper bound for the value problem can be inferred from Lemma 2.3.

Proof of PSPACE-hardness for the value problem.

We show PSPACE-hardness of the value problem when the preorder has $\mathbf{1}$ as a unique maximal element.

We reduce QSAT to the value problem, where QSAT is the satisfiability problem for quantified Boolean formula. For an instance of QSAT, we assume without loss of generality that the Boolean formula is a conjunction of disjunctive clauses[2].

Let $\phi = Q_1 x_1 \ldots Q_n x_n.\ \phi'$, where $Q_i \in \{\forall, \exists\}$ and $\phi' = C_1 \wedge \cdots \wedge C_m$ with $C_j = \bigwedge_{1\le k\le p} \ell_{j,k}$ and $\ell_{j,k} \in \{x_i, \neg x_i \mid 1 \le i \le n\} \cup \{\top, \bot\}$. We define a turn-based game $\mathcal{G}(\phi)$ in the following way (illustrated in Example 10 below). There is one state for each quantifier, one for each literal, and two additional states $\top$ and $\bot$:

$$\text{States} = \{Q_i \mid 1 \le i \le n\} \cup \{x_j, \neg x_j \mid 1 \le j \le m\} \cup \{\top, \bot\}.$$

The game involves two players, A and B. Both states $\top$ and $\bot$, the existential-quantifier states and the literal states are controlled by A, while the universal-quantifier states belong to player B. The state corresponding to quantifier Q_i has two outgoing transitions, going to x_i and $\neg x_i$ respectively. The literal states only have one transition to the next quantifier state, or to the final state for the last literal state. Finally, states $\top$ and $\bot$ both carry a self-loop (notice that $\bot$ is not reachable, while $\top$ will always be visited).

Player A has one target set for each clause: if $C_j = \bigwedge_{1\le k\le p} \ell_{j,k}$ then $T_j^A = \{\ell_{j,k} \mid 1 \le k \le p\}$. The j-th objective Ω_j^A is to reach target set T_j^A. The following result is then straightforward:

Lemma 5.25. *Formula ϕ is valid if, and only if, player A has a strategy whose outcomes from state Q_1 all visit each target set T_j^A.*

Proof. We begin with the direct implication, by induction on n. For the base case, $\phi = Q_1 x_1.\ \bigwedge_j C_j$ where C_j only involves x_1. We consider two cases:

[2] With the convention that en empty disjunction is equivalent to $\bot$.

- $Q_1 = \exists$: since we assume ϕ be true, there must exist a value for x_1 which makes all clauses true. If this value is $\top$, consider the strategy $\sigma_\top$ of Player A such that $\sigma_\top(s_1) = x_1$. Then each clause C_j must have x_i as one of its literals, so that the objective Ω_j^A is satisfied with this strategy. The same argument applies if the value for x_1 were $\bot$.
- $Q_1 = \forall$: in that case, Player A has only one strategy. For both x_1 and $\neg x_1$ all the clauses are satisfied. It follows that each clause C_j must contain x_1 and $\neg x_1$, so that objective Ω_j^A is satisfied for any strategy of player B.

Now, assume that the result holds for all QSAT instances with at most $n-1$ quantifiers.

- if $Q_1 = \exists$, then one of $Q_2 x_2 \ldots Q_n x_n \phi'[x_1 \leftarrow \top]$ and $Q_2 x_2 \ldots Q_n x_n \phi'[x_1 \leftarrow \bot]$ is valid. We handle the first case, the second one being symmetric. For a literal ℓ_i, we write L_{ℓ_i} for the set of clauses containing ℓ_i as a literal, and T_{ℓ_i} for the corresponding set of target sets.

  Assume $Q_2 x_2 \ldots Q_n x_n \phi'[x_1 \leftarrow \top]$ is valid; by induction we know that there exists a strategy σ^{x_1} such that all the targets in T_{ℓ_i} are visited along any outcome from state Q_2 (because $\mathcal{G}(Q_2 x_2 \ldots Q_n x_n \phi'[x_1 \leftarrow \top])$ is the same game as $\mathcal{G}(\phi)$, but with Q_2 as the initial state, and with the targets in T_{x_1} containing $\{\top\}$ in place of x_1). We define the strategy σ by $\sigma(Q_1) = x_1$ and $\sigma(Q_1 \cdot x_1 \cdot \rho) = \sigma^{x_1}(\rho)$. An outcome of σ will necessarily visit x_1, hence visiting all the targets in T_{x_1}; because σ follows σ^{x_1}, all the objectives not in T_{x_1} are met as well.
- if $Q_1 = \forall$, then $Q_2 x_2 \ldots Q_n x_n \phi'[x_1 \leftarrow \top]$ is valid. Using the induction hypothesis we know that from Q_2 there is a strategy σ^{x_1} that enforces a visit to all the targets in T_{x_1}. Similarly, $Q_2 x_2 \ldots Q_n x_n \phi'[x_1 \leftarrow \bot]$ is valid, and there is a strategy $\sigma^{\neg x_1}$ that visits all the objectives not in $T_{\neg x_1}$. We define a new strategy σ as follows: $\sigma(s_1 \cdot x_1 \cdot \rho) = \sigma^{x_1}(\rho)$ and $\sigma(Q_1 \cdot \neg x_1 \cdot \rho) = \sigma^{\neg x_1}(\rho)$. Consider an outcome of σ: if it visits x_1, then all the objectives in T_{x_1} are visited, and because the path follows σ^{x_1}, the objectives not in T_{x_1} are also visited. The other case is similar.

We now turn to the converse implication. Assume the formula is not valid. We prove that for any strategy σ of player A, there is an outcome ρ of this strategy such that some objective Ω_j^A is not satisfied. We again proceed by induction, beginning with the case where $n = 1$.

- if $Q_1 = \exists$, then both $Q_2 x_2 \ldots Q_n x_n \phi'[x_1 \leftarrow \top]$ and $Q_2 x_2 \ldots Q_n x_n \phi'[x_1 \leftarrow \bot]$ are false. This entails that one of the clauses only involves $\bot$ (no other disjunction involving x_1 and/or $\neg x_1$ is always false), and the corresponding reachability condition is $\bot$, which is not reachable.
- if $Q_1 = \forall$, then one of $Q_2 x_2 \ldots Q_n x_n \phi'[x_1 \leftarrow \top]$ and $Q_2 x_2 \ldots Q_n x_n \phi'[x_1 \leftarrow \bot]$ is false. In the first case, one of the clauses contains $\neg x_1$, or only contains $\bot$. Then along the run $Q_1 \cdot x_1 \cdot \top^\omega$, the objective C_j is not visited. The other case is similar.

Now, assuming that the result holds for formulas with $n-1$ quantifiers, we prove the result with n quantifiers.

- if $Q_1 = \exists$, then both $\phi'[x_1 \leftarrow \top]$ and $\phi'[x_1 \leftarrow \bot]$ are false; using the induction hypothesis, any run from Q_2 fails to visit some objective not in $T_{x_1} \cup T_{\neg x_1}$. Hence no strategy from Q_1 can enforce a visit to all the objectives.
- if $Q_1 = \forall$, then one of $\phi'[x_1 \leftarrow \top]$ and $\phi'[x_1 \leftarrow \bot]$ is false. We handle the first case, the second one being symmetrical. By induction hypothesis, for any strategy σ of player A in the game $\mathcal{G}(\phi'[x_1 \leftarrow \top])$, one of the outcome fails to visit all the objective not in T_{x_1}. Then along the path $\rho = Q_1 \cdot x_1 \cdot \rho'$, some objectives not in T_{x_1} are not visited.

□

We can directly conclude from this lemma that the value of the game for A is $\mathbf{1}$ (the unique maximal payoff for our preorder) if, and only if, the formula ϕ is valid, hence this problem is PSPACE-hard.

Example 10. As an example of the construction, let us consider the formula

$$\phi = \forall x_1.\ \exists x_2.\ \forall x_3.\ \exists x_4.\ (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge \neg x_4 \tag{5.2}$$

The target sets for player A are given by $T^A_1 = \{x_1; \neg x_2; \neg x_3\}$, $T^A_2 = \{x_1; x_2; x_4\}$, and $T^A_3 = \{\neg x_4\}$. The structure of the game is represented in Figure 5.7. Player B has a strategy that falsifies one of the clauses whatever A does, which means that the formula is not valid.

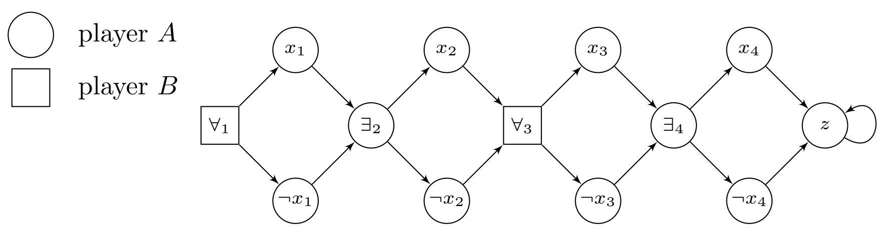


Figure 5.7: Reachability game associated with the formula (5.2)

Proof of PSPACE-hardness for the (constrained) existence problem.

We will now prove PSPACE-hardness for the existence problem, under the conditions specified in the statement of Proposition 5.24, using Lemma 2.4. We specify the new preference relation for the construction of Section 2.4.3. We give B one objective, which is to reach s_1 (s_1 is the sink state introduced by the construction). In terms of preferences for A, going s_1 should be just below visiting all targets. For this we use the statement in Proposition 5.24, that there is v such that for every v', $v' \neq \mathbf{1} \Leftrightarrow v' \lesssim v$, and add s_1 as a target to each T^A_i

such that $v_i = 1$. This defines a preference relation equivalent to the one in the game constructed in Section 2.4.3, therefore we deduce with Lemma 2.4 that the existence problem is PSPACE-hard.

5.3.2 Simple cases

As for ordered Büchi objectives, for some ordered reachability objectives, the preference relation can be (efficiently) (co-)reduced to a single reachability objective. We do not give the formal definitions, they can easily be inferred from that for Büchi objectives on page 73.

Proposition 5.26. *For ordered reachability objectives which are reducible to single reachability objectives, and where the preorders are non-trivial, the value problem is* P*-complete. For ordered reachability objectives which are co-reducible to single reachability objectives, and where these preorders are non-trivial, the existence and constrained existence problems are* NP*-complete.*

Proof. Since P-hardness (resp. NP-hardness) already holds for the value (resp. existence) problem with a single reachability objective (see [28]), we only focus on the upper bounds.

We begin with the value problem: given a payoff vector u for player A, we build the new target set $\widehat{T}$ in polynomial time, and then use a classical algorithm for deciding whether A has a winning strategy (see [28, Chapter 2]). If she does, then she can secure payoff u.

Consider now the constrained existence problem, and assume that the preference relation for each player A is given by target sets $(T_i^A)_{1 \le i \le n_A}$. The NP-algorithm consists in guessing the payoff vector $(v_A)_{A \in \mathrm{Agt}}$ and an ultimately periodic play $\rho = \pi \cdot \tau^\omega$ with $|\pi|, |\tau| \le |\mathrm{States}|^2$, which, for each A, visits T_i^A if, and only if, $v_i^A = 1$. We then co-reduce the payoff to a new target set $\widehat{T}^A(v^A)$ for each player A.

The run ρ is the outcome of a Nash equilibrium with payoff $(v_A)_{A \in \mathrm{Agt}}$ for the original preference relation if, and only if, ρ is the outcome of a Nash equilibrium with payoff 0 with the single reachability objective $\widehat{T}^A(v^A)$ for each $A \in \mathrm{Agt}$. Indeed, in both cases, this is equivalent to the property that no player A can enforce a payoff greater than v^A. Thanks to the algorithm presented in Section 4.2 this condition can be checked in polynomial time. $\square$

We will now see to which ordered objectives this result applies. It is not difficult to realize that the same transformations as those made in the proof of Lemma 5.7 can be made as well for reachability objectives. We therefore get the following lemma, from which we get the remaining results in Table 5.2.

Lemma 5.27. *Ordered reachability objectives with disjunction or maximize preorders are reducible to single reachability objectives. Ordered reachability objectives with disjunction, maximize or subset preorders are co-reducible to single reachability objectives.*

Chapter 6

Timed Games

In this chapter we will apply our procedure to timed games, which was our original motivation. We begin with some definitions.

6.1 Definitions

Clocks. We consider a finite set of *clocks* denoted by X. A *valuation* over X is an application $v\colon X \to \mathbb{R}_+$. If v is a valuation and $t \in \mathbb{R}_+$, then $v + t$ is the valuation that assigns to each $x \in X$ the value $v(x) + t$. If v is a valuation and $Y \subseteq X$, then $[Y \leftarrow 0]v$ is the valuation that assigns 0 to each $y \in Y$ and $v(x)$ to each $x \in X \setminus Y$. We write $\mathbf{0}$ for the valuation that assigns 0 to all the clocks.

A *clock constraint* over X is a formula built on the following grammar:

$$\mathfrak{C}(X) \ni g \ ::= \ x \sim c \ \mid \ g \wedge g$$

where x ranges over X, $\sim \in \{<, \leq, =, \geq, >\}$, and c is an integer. The interpretation of clock constraints over valuations is given inductively by:

$$\begin{cases} v \models x \sim c & \text{if } v(x) \sim c \\ v \models g_1 \wedge g_2 & \text{if } v \models g_1 \text{ and } v \models g_2 \end{cases}$$

When $v \models g$ we say that the valuation v satisfies the clock constraint g.

We now define the classical notion of timed games, following that of [21].

Timed game. A *timed game* is a 8-tuple $\mathcal{G} = \langle \mathrm{Loc}, X, \mathrm{Inv}, \mathrm{Trans}, \mathrm{Agt}, \mathrm{Owner}, (\leq_\ell)_{\ell \in \mathrm{Loc}}, (\precsim_A)_{A \in \mathrm{Agt}} \rangle$ where:

- Loc is a finite set of *locations*;
- X is a finite set of *clocks*;
- $\mathrm{Inv}\colon \mathrm{Loc} \to \mathfrak{C}(X)$ assigns an *invariant* to each location;
- $\mathrm{Trans} \subseteq \mathrm{Loc} \times \mathfrak{C}(\mathit{clocks}) \times 2^X \times \mathrm{Loc}$ is the set of *transitions*;

- Agt is the finite set of *players*;
- Owner: Trans $\to$ Agt assigns a player to each transition, if $\text{Owner}(t) = A$ we will say that A *owns* t;
- for each $\ell \in \text{Loc}$, $\leq_\ell$ is a total order[1] over Agt, it is the *priority order* of the players at that location;
- for each $A \in \text{Agt}$, $\precsim_A$ is a preorder over $(\text{Loc} \times \mathbb{R}_+^X)^\omega$, it is the *preference relation* of player A.

Remark. A problem with the classical definition of timed games, is to decide what happens when two players choose the same delay. In the two player zero-sum case, a solution is to consider that the opponent always has the priority. There is no such solution in multiplayer games. In previous works [6, 7], we considered that such a situation would be resolved by non-determinism. However the concept of Nash equilibrium does not apply in the resulting game since a given strategy profile can have several different outcomes. To extend Nash equilibrium to non-deterministic games, we proposed the concept of pseudo Nash equilibrium. However, in this work, we decided to focus on the time aspect of the model and not on non-determinism. We consider a simple and natural solution by introducing a priority order among the players which depend on the current state. This avoids the problem of collision after choosing a delay, and we can use the previous results on the computation of Nash equilibria in this context.

A timed game is played as follows: a state of the game is a pair (ℓ, v) where ℓ is a location and v is a clock valuation, provided that $v \models \text{Inv}(\ell)$. From each state (starting from an initial state $s_0 = (\ell, \mathbf{0})$) each player A chooses a non-negative real number d and a transition $\delta = (\ell, g, z, \ell')$, with the intended meaning that she wants to delay for d time units and then fire transition δ, this forms a *timed action* $m_A = (d, \delta)$. There are several (natural) restrictions on these choices:

- spending d time units in ℓ must be allowed[2] i.e. $v \models \text{Inv}(\ell)$ and $v + d \models \text{Inv}(\ell)$;
- δ belongs to player A, i.e. $\text{Owner}(\delta) = A$;
- the transition is firable after d time units, i.e. $v + d \models g$;
- the invariant is satisfied when entering ℓ', i.e. $[z \leftarrow 0](v + d) \models \text{Inv}(\ell')$.

If (and only if) there is no such possible choice for some player A (for instance if no transition from ℓ belongs to A), then she chooses a special move, denoted by $\perp$. When these conditions are respected we say that the action is *legal*.

Given a set of legal choices m_{Agt} for all the players, the shortest delay will be selected. If $m_A \neq \perp$, we write $m_A = (d_A, \delta_A)$. Let $d(m_{\text{Agt}}) = \min\{d_A \mid$

[1] Recall that a total order is a transitive, antisymmetric and total relation.

[2] Formally, this should be written $v + d' \models \text{Inv}(\ell)$ for all $0 \leq d' \leq d$, but this is equivalent to having only $v \models \text{Inv}(\ell)$ and $v + d \models \text{Inv}(\ell)$ since invariants are convex.

$A \in \text{Agt}$ and $m_A = (d_A, \delta_A)\}$ be the *shortest delay* that was chosen by a player. Among the players who chose the shortest delay, we select the one with the highest priority, we do this according to the priority order of the current state $\leq_\ell$, that is $\text{Select}(m_{\text{Agt}}) = \max_{\leq_\ell}\{A \in \text{Agt} \mid m_A = (d_A, \delta_A) \text{ and } d_A = d(m_{\text{Agt}})\}$. We say that $\text{Select}(m_{\text{Agt}})$ is *selected* from the action profile m_{Agt}. Then, all the clocks grow at the same rate during $d(m_{\text{Agt}})$ time unit, and the transition $\delta_{\text{Select}(m_{\text{Agt}})} = (\ell, g, z, \ell')$ of the selected player is applied, resetting the clocks of z to 0. This leads to a new state $(\ell', [z \leftarrow 0](v + d(m_{\text{Agt}})))$.

In the following, and to simplify notations, we define for each location ℓ a total order $\leq^2_\ell$ over pairs of $\mathbb{R}_+ \times \text{Agt}$, defined by $(d_A, A) \leq^2_\ell (d_B, B)$ when:

- either $d_A < d_B$;
- or $d_A = d_B$ and $A \geq_\ell B$;

This way, if we write for each player $A \in \text{Agt}$ her action $m_A = (d_A, \delta_A)$, then the selected player $\text{Select}(m_{\text{Agt}})$ is the player A for which the pair (d_A, A) is minimal with respect to $\leq^2_\ell$. We also define the associated strict total order $<^2_\ell$, which is defined by $(d_A, A) <^2_\ell (d_B, B)$ if, and only if, $(d_A, A) \leq^2_\ell (d_B, B)$ and $(d_A, A) \neq (d_B, B)$.

6.1.1 Semantics as an Infinite Concurrent Game

To formalize the way timed games are played, we express their semantics in terms of an infinite-state concurrent game. With a timed game $\mathcal{G} = \langle \text{Loc}, X, \text{Inv}, \text{Trans}, \text{Agt}, \text{Owner}, (\leq_\ell)_{\ell \in \text{Loc}}, (\precsim_A)_{A \in \text{Agt}} \rangle$ we associate the infinite concurrent game $\mathcal{G}' = \langle \text{States}, \text{Agt}, \text{Act}, \text{Mov}, \text{Tab}, (\precsim_A)_{A \in \text{Agt}} \rangle$ such that

- the set of states is the set of configurations of the timed game: $\text{States} = \{(\ell, v) \mid \ell \in \text{Loc}, v\colon X \to \mathbb{R}_+ \text{ such that } v \models \text{Inv}(\ell)\}$;
- $s_0 = (\ell_0, \mathbf{0})$ is the initial state;
- the set of actions is $\text{Act} = \{(d, \delta) \mid d \in \mathbb{R}_+, \delta \in \text{Trans}\} \cup \{\perp\}$;
- an action (d, δ) is allowed to player A in state (ℓ, v) if, and only if, writing $\delta = (\ell, g, z, \ell')$, the following conditions hold:
  - $v + d \models \text{Inv}(\ell)$;
  - $\text{Owner}(\delta) = A$;
  - $v + d \models g$;
  - $[z \leftarrow 0](v + d) \models \text{Inv}(\ell')$.

  Then $\text{Mov}((\ell, v), A)$ is the set of actions available to player A when this set is non empty, and it is $\{\perp\}$ otherwise;
- finally, given a state (ℓ, v) and a legal move m_{Agt}, $\text{Tab}((\ell, v), m_{\text{Agt}}) = (\ell', v')$ such that if $A = \text{Select}(m_{\text{Agt}})$ is the selected player, writing $m_A = (d_A, \delta_A)$, we have that $\delta_A = (\ell, g, z, \ell')$ and $v' = [z \leftarrow 0](v + d_A)$;
- the preference relations $(\precsim_A)_{A \in \text{Agt}}$ are inherited from $\mathcal{G}$.

Timed games inherit the notions of history, play, path, strategy, profile, outcome and Nash equilibrium from concurrent games via this correspondence.

In the sequel, we consider only *non-blocking* timed games, i.e., timed games in which, for any reachable state (ℓ, v), at least one player A has an available action, i.e. $\mathrm{Mov}((\ell, v), A) \neq \{\perp\}$.

6.2 The Region Game

In this section, we explain how Nash equilibria can be computed in timed games with region invariant preferences. This result is based on the construction that we explain now.

6.2.1 Regions

The construction relies on the classical notion of *regions* [2].

Regions. If $M \in \mathbb{N}$, we write $\mathfrak{C}_M(X)$ for the set of constraints in $\mathfrak{C}(X)$ in which constants are integers within the interval $[\![0; M]\!]$. Let $\mathcal{G}$ be a timed game, and M be the maximal constant appearing in $\mathcal{G}$: $M = \max\{c \mid x \sim c \text{ constraint in } \mathcal{G}\}$. For a real number δ, we write $\lfloor \delta \rfloor$ the integral part of δ and $\mathrm{fr}(\delta)$ its fractional part. We define the equivalence relation $\equiv_{X,M}$ over $\mathbb{R}_+^X$ by $v \equiv_{X,M} v'$ if, and only if:

1. for all clocks $x \in X$, either $\lfloor v(x) \rfloor$ and $\lfloor v'(x) \rfloor$ are the same, or both $v(x)$ and $v(x)$ exceed M;
2. for all clocks $x, y \in X$ with $v(x) \leq M$ and $v(y) \leq M$, $\mathrm{fr}(v(x)) \leq \mathrm{fr}(v(y))$ if, and only if, $\mathrm{fr}(v'(x)) \leq \mathrm{fr}(v'(y))$;
3. for all clocks $x \in X$ with $v(x) \leq M$, $\mathrm{fr}(v(x)) = 0$ if, and only if, $\mathrm{fr}(v'(x)) = 0$;

This equivalence relation naturally induces a partition $\mathcal{R}_{X,M}$ of $\mathbb{R}_+^X$. This partition has the following properties:

- it is compatible with constraints in $\mathfrak{C}_M(X)$, i.e. for every $r \in \mathcal{R}_{X,M}$, and constraint $g \in \mathfrak{C}_M(X)$ either all valuations in r satisfy the clock constraint g, or no valuation in r satisfies it.
- it is compatible with time elapsing, i.e. if there is $v \in r$ and $t \in \mathbb{R}_+$ such that $v + t \in r'$, then for all $v' \in r$ there is t' such that $v' + t' \in r'$;
- it is compatible with resets, i.e. if $z \subseteq X$ then if $[z \leftarrow 0]r \cap r' \neq \varnothing$ then $[z \leftarrow 0]r \subseteq r'$.

Elements of $\mathcal{R}_{X,M}$ are called *regions*. We denote $[v]_{X,M}$ the region containing valuation v, or simply $[v]$ if X and M are clear from the context. A region r is said to be *time-elapsing*, if for any $v \in r$ there is $t > 0$ such that $v + t \in r$. We write Succ(r) the *successors* of r by time elapsing, it is defined by $r' \in \text{Succ}(r)$ if there is $v \in r$ and $t \in \mathbb{R}_+$ such that $(v + t) \in r'$.

The number of region is bounded by $|X|! \cdot (4M + 4)^{|X|}$, note that this is exponential both in the number of clocks in X and in the size of the maximal constant, if constants are encoded in binary.

In the following, we will use an abstraction of the timed game, which is based on regions. For this to be correct, we need the preference of the players to be preserved by the region abstraction. This is expressed by the following definition:

Region-invariance. A preorder $\lesssim$ over $(\text{Loc} \times \mathbb{R}_+^X)^\omega$ is said to be *region-invariant* when the following holds: for any two plays $\rho = (\ell_i, v_i)_{i\geq 0}$ and $\rho' = (\ell'_i, v'_i)_{i\geq 0}$, if for all $i \in \mathbb{N}$, $\ell_i = \ell'_i$ and v_i and v'_i belong to the same region, then ρ and ρ' are equivalent for $\lesssim$, i.e. $\rho \lesssim \rho'$ and $\rho' \lesssim \rho$.

6.2.2 Construction of the Region Game

Let $\mathcal{G} = \langle \text{Loc}, X, \text{Inv}, \text{Trans}, \text{Agt}, \text{Owner}, (\leq_\ell)_{\ell\in\text{Loc}}, (\precsim_A)_{A\in\text{Agt}}\rangle$ be a timed game, where all preference relations $\precsim_A$ for $A \in \text{Agt}$ are region invariant. Let M be the maximal constant appearing in $\mathcal{G}$. We define a finite concurrent game, that we call the *region game*, $\mathcal{R} = \langle \text{States}_\mathcal{R}, \text{Agt}, \text{Act}_\mathcal{R}, \text{Mov}_\mathcal{R}, \text{Tab}_\mathcal{R}, (\preccurlyeq_A^\mathcal{R})_{A\in\text{Agt}}\rangle$. We first give the idea of the construction, emphasizing the differences with a classical region construction.

Concerning the actions, instead of giving a real delay, players will specify a region and an integer index p in the interval $[\![0, 4]\!]$. In the equilibrium we will constrain the strategies to use only integers 1, 2 and 3. Roughly, the index p allows the players to say if they want to play first ($p = 1$), second ($p = 2$) or later ($p = 3$) if their region is selected. In time-elapsing regions, deviators will have the possibility to use indexes $p = 0$ and $p = 4$ to play before or after everyone else. This correspondence is illustrated in Fig. 6.1.

Concerning the states of the game, they will be composed by a tuple (ℓ, r, q, p), with $\ell \in \text{Loc}$, $r \in \mathfrak{R}$, $q \in \mathfrak{R}$ and $p \in [\![0, 4]\!]$. The first two components ℓ and r correspond to the current location and the region containing the current valuation as is classically done in region automaton construction, they characterize which actions are available to the players. The last two are needed to preserve information on the delay selected in the last transition. We illustrate the use of this two last components with two examples.

Consider Fig. 6.2 as part of a timed game with two players. We aim at finding an equilibrium which goes through location ℓ_1. If one of the players changes her strategy in order to go to ℓ_2, it may be useful to know which of the two did so. Starting in ℓ_0, with a valuation $v : x \to 0, y \to 0.5$, player A_1 can wait for a delay shorter than 0.5 and take a transition, and player A_2 can wait for a delay in $]0.5, 1[$ for instance. These two possibilities are represented in Fig. 6.3. In the timed game, thanks to the final value of x, we are able to determine in which region the transition was taken and so which player deviated: if the valuation

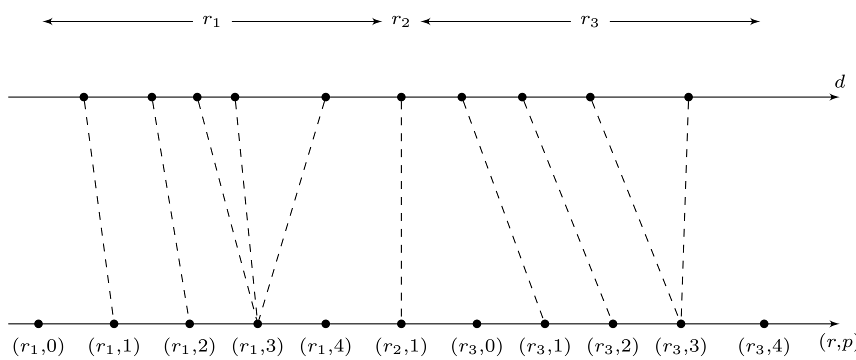


Figure 6.1: A correspondence of actions between the timed game and the region game. Indexes 0 and 4 are to be used only by deviators and not in the equilibria. Regions r_1 and r_3 are time-elapsing in contrast with r_2.

is on the left of the dotted line then the unique suspect is A_1 and if it is on the right the suspect is A_2. Let us look at what happens in the region game: as y is reset, for transitions of both players the final valuation will be in the same region where $x \in]0,1[$ and $y = 0$. This information is not enough to know in which region the transition was taken and who is suspect. To remedy this, we added the component $q \in \mathfrak{R}$ that gives the information of the region in which the last transition happened.

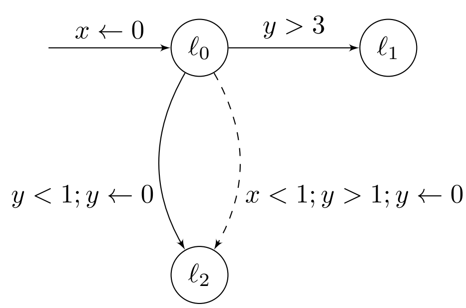


Figure 6.2: Part of a timed game. Player A_1 controls the plain transitions, and A_2 the dashed one.

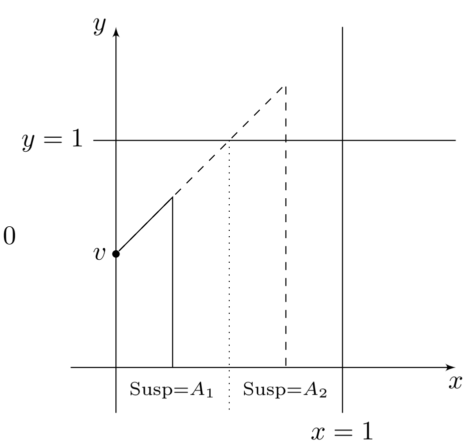


Figure 6.3: Two possible deviations

Now let us look at transitions fired in the same region, for another game, represented in Fig. 6.4. The timing information in the timed game can help us discriminate between two kinds of deviations represented in Fig. 6.5. Assume the original strategy is to play a delay of 0.2 for player A_1 and of 0.6 for player A_2. For a deviation that goes to ℓ_2 instead of ℓ_1, we can recover from the valuation of x which of the players deviated. If x is smaller than 0.2 then A_2 is the suspect, and if x is greater than 0.2, A_1 is the suspect. Since the region reached is always the same, we again need to add some information in the state of the region game. This is the role of the p component, that allows to recover the index played by the selected player during the last transition. Assume that in the region game A_1 is playing with an index $p = 1$, and A_2 with an index $p = 2$. The first deviation when A_2 plays before the player that was supposed to play first will correspond to $p = 0$, and the second deviation when A_1 plays after the second player will correspond to $p = 3$ or $p = 4$.

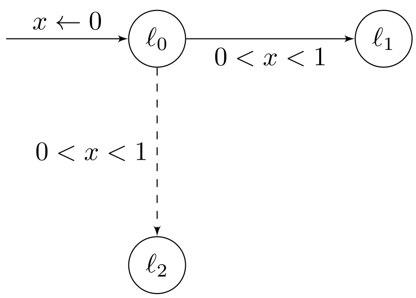


Figure 6.4: Part of a timed game. Player A_1 controls the plain transitions, and A_2 the dashed one.

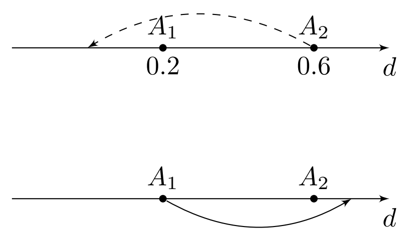


Figure 6.5: Two possible deviations.

Region game. The region game is formally defined as follows:

- $\mathrm{States}_{\mathcal{R}} = \mathrm{Loc} \times \mathfrak{R} \times \mathfrak{R} \times [\![0, 4]\!]$, where $\mathfrak{R} = \mathcal{R}_{X,M}$ is the set of clock regions;
- $\mathrm{Act}_{\mathcal{R}} = \mathfrak{R} \times [\![0, 4]\!] \times \mathrm{Trans}$;
- $\mathrm{Mov}_{\mathcal{R}}((\ell, r, q, p), A)$ is the set composed of all (r', p', δ) such that writing (ℓ, g, z, ℓ') for δ we have that:
  - $r' \in \mathrm{Succ}(r)$;
  - $r' \models \mathrm{Inv}(\ell)$;
  - $p' \in [\![0, 4]\!]$ if r' is time-elapsing, and $p' = 1$ otherwise;
  - $\mathrm{Owner}(\delta) = A$;
  - $r' \models g$;
  - $[z \leftarrow 0]r' \models \mathrm{Inv}(\ell')$.

  if this set is not empty. If there is no such (r', p', δ) then $\mathrm{Mov}_{\mathcal{R}}((\ell, r, q, p), A)$ is equal to $\{\perp\}$;
- Given a state $(\ell, r, q, p) \in \mathrm{States}_{\mathcal{R}}$ and an action profile legal at that state $n_{\mathrm{Agt}} \in \prod_{A \in \mathrm{Agt}} \mathrm{Mov}_{\mathcal{R}}((\ell, r, q, p), A)$, we write $n_A = (r_A, p_A, \delta_A)$ for actions different from $\perp$ and:

– $r(n_{\mathrm{Agt}}) = \min\{r_A \mid A \in \mathrm{Agt}\}$ the first (w.r.t. time elapsing) region that was chosen by a player;
– $p(n_{\mathrm{Agt}}) = \min\{p_A \mid A \in \mathrm{Agt} \text{ and } r_A = r(n_{\mathrm{Agt}})\}$, the smallest index that was chosen together with the region $r(n_{\mathrm{Agt}})$;
– $\mathrm{Select}(n_{\mathrm{Agt}}) = \max_{\geq_\ell}\{A \in \mathrm{Agt} \mid r_A = r(n_{\mathrm{Agt}}) \text{ and } p_A = p(n_{\mathrm{Agt}})\}$ the player with the highest priority among those who chose the minimal pair of region and index;

Then $\mathrm{Tab}_{\mathcal{R}}((\ell, r, q, p), n_{\mathrm{Agt}}) = (\ell', r', q', p')$ such that:

– $\delta_{\mathrm{Select}(n_{\mathrm{Agt}})} = (\ell, g, z, \ell')$;
– $r' = [z \leftarrow 0]r(n_{\mathrm{Agt}})$;
– $(q', p') = (r(n_{\mathrm{Agt}}), p(n_{\mathrm{Agt}}))$ if $z \neq X$ and $(q', p') = ([\mathbf{0}], 1)$ otherwise. This distinction is important since if one clock was not reset during the last transition we can deduce the exact time that was spent before the transition actually happened.

- For a player A, we define the preference relation $\preccurlyeq_A^{\mathcal{R}}$ by $(\ell_i, r_i, q_i, p_i)_{i\geq 0} \preccurlyeq_A^{\mathcal{R}} (\ell'_i, r'_i, q'_i, p'_i)_{i\geq 0}$ if there exist two runs $(\ell_i, v_i)_{i\geq 0}$ and $(\ell'_i, v'_i)_{i\geq 0}$ such that $v_i \in r_i$ and $v'_i \in r'_i$ for all $i \geq 0$, and $(\ell_i, v_i)_{i\geq 0} \preccurlyeq_A (\ell'_i, v'_i)_{i\geq 0}$.

In order to simplify notations, similarly as we did for the timed game, we define for each location a partial orders $\leq_\ell^3$ over triples of $\mathfrak{R} \times [\![0, 4]\!] \times \mathrm{Agt}$. It is defined by $(r_A, p_A, A) \leq_\ell^3 (r_B, p_B, B)$ when:

- either $r_A < r_B$;
- or $r_A = r_B$ and $p_A < p_B$;
- or $r_A = r_B$, $p_A = p_B$ and $A \geq_\ell B$.

This way, if we write for each player $A \in \mathrm{Agt}$ her action $n_A = (r_A, p_A, \delta_A)$, then the selected player $\mathrm{Select}(n_{\mathrm{Agt}})$ is the player A for which the triple (r_A, p_A, A) is minimal with respect to $\leq_\ell^3$. We also define $<_\ell^3$ the associated strict total order.

In the next section, we show how the region game relates to the original timed game. Roughly Nash equilibria with a specific constraint on the action allowed, will correspond to Nash equilibria in the timed game.

6.2.3 Proof of Correctness

The correctness of the construction is captured by the next proposition.

Proposition 6.1. *Let $\mathcal{G}$ be a timed game with region-invariant preference relations, and $\mathcal{R}$ its associated region game. Then there is a Nash equilibrium in $\mathcal{G}$ from $(s, \mathbf{0})$ if, and only if, there is a Nash equilibria in $\mathcal{R}$ from $(s, [\mathbf{0}])$, with constraint* Allow *on the moves, where* Allow *is defined for every state (ℓ, r, q, p) by* $\mathrm{Allow}((\ell, r, q, p), (r_A, p_A, \delta_A)_{A\in\mathrm{Agt}}) = \mathsf{true}$ *if, and only if, for all $A \in \mathrm{Agt}$, $p_A \in [\![1, 3]\!]$. Furthermore, this equivalence is constructive.*

To establish this result, we rely on the notion of game simulation we defined in Section 3.3. This proof is quite long and several cases have to be distinguished, so it will be split in several lemmas.

We make a first remark which will be useful when establishing a relationship between the suspects in one game and the other one.

Lemma 6.2. *Let $\mathcal{G}$ be a timed game, (ℓ, v) be a configuration and m_{Agt} a legal move. Let A_0 be the selected player in m_{Agt}, then if $A \neq A_0$, for any m'_A the selected player from $m_{\mathrm{Agt}}[A \mapsto m'_A]$, is either A or A_0. Similarly, in the region game $\mathcal{R}$, let (ℓ, r, q, p) be a state and n_{Agt} a legal move. Let A_0 is the selected player in n_{Agt}, if $A \neq A_0$, for any n'_A the selected player from $n_{\mathrm{Agt}}[A \mapsto n'_A]$, is either A or A_0.*

Proof. A_0 is the player that minimizes (d_A, A) for the order $\leq^2_\ell$ We write (d'_B, δ'_B) for the action of a player B in the profile $m_{\mathrm{Agt}}[A \mapsto m'_A]$. In $m_{\mathrm{Agt}}[A \mapsto m'_A]$ only the action of A is changed, so

$$\min_{\leq^2_\ell}\{(d'_B, B) \mid B \in \mathrm{Agt}\} = \min_{\leq^2_\ell}(\{(d_B, B) \mid B \in \mathrm{Agt} \setminus A\} \cup \{(d'_A, A)\})$$

If $A_0 \neq A$, then $\min_{\leq^2_\ell}\{(d_B, B) \mid B \in \mathrm{Agt} \setminus A\} = (d_{A_0}, A_0)$. Therefore $\min_{\leq^2_\ell}\{(d'_B, B) \mid B \in \mathrm{Agt}\}$ is either (d_{A_0}, A_0) or (d'_A, A) and the selected player in $m_{\mathrm{Agt}}[A \mapsto m'_A]$ is either A or A_0.

Similarly, writing (r'_B, p'_B, δ'_B) for the action of a player B in the profile $n_{\mathrm{Agt}}[A \mapsto n'_A]$, only the action of A can change in that profile, and so

$$\min_{\leq^3_\ell}\{(r'_B, p'_B, B) \mid B \in \mathrm{Agt}\} = \min_{\leq^3_\ell}(\{(r_B, p_B, B) \mid B \in \mathrm{Agt} \setminus A\} \cup \{(r'_A, p'_A, A)\})$$

Therefore, if $A_0 \neq A$, the selected player in $n_{\mathrm{Agt}}[A \mapsto n'_A]$ is either A or A_0. □

In the remaining of this section we will show that a timed game and its associated region game simulate each other in the sense of Section 3.3. This is achieved by considering the relation $\lhd$ given by $(\ell, v) \lhd (\ell, r, q, p)$ for any q and p if r is the region containing v, and showing that $\lhd$ is a simulation. Since the preference relations in $\mathcal{G}$ are region invariant and by definition of the preference relations in $\mathcal{R}$, the relation $\lhd$ is preference preserving. We will then define two functions λ and μ. The function λ maps moves in $\mathcal{G}$ to *equivalent* moves in $\mathcal{R}$, furthermore the index p used in the image of λ are restricted to $[\![1, 3]\!]$. The function μ, maps moves in $\mathcal{R}$ that use the indexes in $[\![1, 3]\!]$, to *equivalent* moves in $\mathcal{G}$.

For this, we use a partial function $f \colon \mathbb{R}^X_+ \times \mathfrak{R} \times [\![0, 4]\!] \to \mathbb{R}_+$. We pick it such that for every valuation v, and every region r, if there is some $t \in \mathbb{R}_+$ such that $v + t \in r$, then

- if r is time-elapsing this function is defined at (v, r, p) for all $p \in [\![0, 4]\!]$, and we require that $v + f(v, r, p) \in r$ for all $p \in [\![0, 4]\!]$, and that $f(v, r, 0) < f(v, r, 1) < f(v, r, 2) < f(v, r, 3) < f(v, r, 4)$;

- if r is not time-elapsing this function is defined at (v, r, p) for $p = 1$, and $f(v, r, 1)$ is the unique delay t such that $v + t \in r$.

If no such t exists, the function is undefined at (v, r, p).

6.2.4 From Timed Game $\mathcal{G}$ to Region Game $\mathcal{R}$.

For any configuration (ℓ, v) and any move vector m_{Agt} in $\mathcal{G}$, we define the move vector λ_{Agt} in $\mathcal{R}$, using only indexes in $[\![1, 3]\!]$, as follows. For all $A \in \mathrm{Agt}$, if $m_A = \perp$ then $\lambda_A = \perp$; otherwise we write $m_A = (d_A, \delta_A)$, and r_A the region corresponding to valuation $v + d_A$. If $d_A = d(m_{\mathrm{Agt}})$, or if r_A is not time-elapsing then $\lambda_A = (r_A, 1, \delta_A)$; otherwise we write $d^2 = \min\{d \mid A \in \mathrm{Agt},\ m_A = (d, \delta) \text{ and } d > d(m_{\mathrm{Agt}})\}$ the second shortest delay, this is well defined since $d(m_{\mathrm{Agt}}) < d_A$. Then, we let,

- if $d = d^2$, then $\lambda_A = (r_A, 2, \delta_A)$;
- if $d > d^2$, then $\lambda_A = (r_A, 3, \delta_A)$.

If (d_A, δ_A) is allowed to A in (ℓ, v), then λ_A is allowed to A in (ℓ, r, p) where r is the region containing v, since it corresponds to the same transition played in the correct region.

Let first remark that this construction selects the same player in the following sense:

Lemma 6.3. $\mathrm{Select}(m_{\mathrm{Agt}}) = \mathrm{Select}(\lambda_{\mathrm{Agt}})$

Proof. Let $A = \mathrm{Select}(m_{\mathrm{Agt}})$, the delay d_A is minimal among the delays $d_A = d(m_{\mathrm{Agt}})$, hence its corresponding region $v + d_A = r_A$ is also minimal among the regions played in λ_{Agt}, $r_A = r(\lambda_{\mathrm{Agt}})$. Now $d_A = d^1$ hence $\lambda_A = (r_A, 1, \delta_A)$, p_A is minimal among the index $p_A = p(\lambda_{\mathrm{Agt}})$. Any player B such that $(r_B, p_B) = (r_A, p_A)$, also played a delay $d_B = d^1$. Since A was selected in m_{Agt}, it is that $A \geq_\ell B$ and therefore A is also selected in λ_{Agt}. □

The following two lemmas will show that the region game $\mathcal{R}$ with action profile limited to indexes in $[\![1, 3]\!]$ simulates the timed game $\mathcal{G}$.

Lemma 6.4. *Let (ℓ, v) and (ℓ', v') be two configurations in the timed game $\mathcal{G}$, m_{Agt} be a legal move from (ℓ, v) and λ_{Agt} be the corresponding move in the region game. If $\mathrm{Tab}((\ell, v), m_{\mathrm{Agt}}) = (\ell', v')$, then for any $q \in \mathfrak{R}$ and $p \in [\![0, 4]\!]$, $\mathrm{Tab}_{\mathcal{R}}((\ell, r, q, p), \lambda_{\mathrm{Agt}}) = (\ell', r', q', 1)$, where r and r' are the regions containing v and v' respectively.*

Proof. Assuming $\mathrm{Tab}((\ell, v), m_{\mathrm{Agt}}) = (\ell', v')$, let $A = \mathrm{Select}(m_{\mathrm{Agt}})$ be the selected player in m_{Agt}, as showed in Lemma 6.3, she is also selected in λ_{Agt}. Hence, writing for each player $A \in \mathrm{Agt}$ her action $\lambda_A = (r_A, p_A, \delta_A)$, we have that $\mathrm{Tab}_{\mathcal{R}}((\ell, r, q, p), \lambda_{\mathrm{Agt}}) = (\ell', r', q', p_A)$ where $r' = [z_A \leftarrow 0]r_A$, and $p_A = 1$ by construction of λ_{Agt}. It is the case that r' contains $[z_A \leftarrow 0](v + d_A) = v'$ since regions are compatible with clock resets. □

We now compare the suspects players in $\mathcal{G}$ and $\mathcal{R}$.

Lemma 6.5. *Let (ℓ, r, q, p) and (ℓ', r', q', p') be two states in the region game. If $v \in r$, m_{Agt} is an allowed move from (ℓ, v), and λ_{Agt} is the corresponding move in the region game, then there is a valuation $v' \in r'$ such that:*

$$\mathrm{Susp}_{\mathcal{R}}(((\ell, r, q, p), (\ell', r', q', p')), \lambda_{\mathrm{Agt}}) \subseteq \mathrm{Susp}_{\mathcal{G}}(((\ell, v), (\ell', v')), m_{\mathrm{Agt}})$$

Proof. If $\mathrm{Susp}_{\mathcal{R}}(((\ell, r, q, p), (\ell', r', q', p')), \lambda_{\mathrm{Agt}})$ is empty then the property is trivially true, otherwise we first need to decide on valuation v'. For any player $A \in \mathrm{Agt}$, we write her action $(r_A, p_A, \delta_A) = \lambda_A$ in the move λ_{Agt}, and $(d_A, \delta_A) = n_A$ in the move n_{Agt}. Let A_0 be the selected player from λ_{Agt}, we have $p_{A_0} = 1$ by construction of λ_{Agt}. Let A be the suspect with the smallest priority in $\mathrm{Susp}_{\mathcal{R}}(((\ell, r, q, p), (\ell', r', q', p')), \lambda_{\mathrm{Agt}})$. There is a move $\lambda'_A = (r'_A, p'_A, \delta'_A)$ such that $\mathrm{Tab}_{\mathcal{R}}((\ell, r, q, p), \lambda_{\mathrm{Agt}}[A \mapsto \lambda'_A]) = (\ell', r', q', p')$. Let C be the selected player from $\lambda_{\mathrm{Agt}}[A \mapsto \lambda'_A]$, we write her action (r'_C, p'_C, δ'_C) in the strategy profile $\lambda_{\mathrm{Agt}}[A \mapsto \lambda'_A]$, it is equal to λ'_A if $C = A$, and to λ_C otherwise.

The valuation v' will be decided by choosing a delay d'. Notice that from Lemma 6.2, if $A \neq A_0$ then C is either A or A_0, we distinguish the following cases:

- if $C \neq A_0 = A$ we let $d' = d_C$;
- if $C = A_0 \neq A$, we let $d' = d_{A_0}$;
- if $C = A$ and there is $E \in \mathrm{Agt}$ such that $(r'_A, p'_A) = (r_E, p_E)$, we let $d' = d_E$. This delay is uniquely defined in this way because by construction of λ_{Agt}, if $(r_E, p_E) = (r_F, p_F)$ for two players E and F then this means that $d_E = d_F$;
- if $C = A$ and $(r'_A, p'_A) < (r_{A_0}, p_{A_0})$, it is possible to find a delay $d' < d_{A_0}$ such that $v + d' \in r'_A$;
- otherwise $C = A$, $(r'_A, p'_A) > (r_{A_0}, p_{A_0})$, and $(r'_A, p'_A) \neq (r_E, p_E)$ for all $E \in \mathrm{Agt}$, it is possible to find a delay $d' > d_{A_0}$ and $d' < d_E$ for all $E \neq A$, such that $v + d' \in r'_A$;

We write $\delta_C = (\ell, g_C, z_C, \ell')$. We then take v' as the valuation resulting from the transition taken by C after delay d', that is: $[z'_C \leftarrow 0](v + d')$. The valuation v' belongs to $r' = [z'_C \leftarrow 0]r'_C$. Also notice that the valuation $v + d'$ belongs to r'_C.

Now let B be any suspect in $\mathrm{Susp}_{\mathcal{R}}(((\ell, r, q, p), (\ell', r', q', p')), \lambda_{\mathrm{Agt}})$, there is a move λ'_B such that $\mathrm{Tab}_{\mathcal{R}}((\ell, r, q, p), \lambda_{\mathrm{Agt}}[B \mapsto \lambda'_B]) = (\ell', r', q', p')$. Let D be the selected player from $\lambda_{\mathrm{Agt}}[B \mapsto \lambda'_B]$, we write her action (r'_D, p'_D, δ'_D) in the profile $\lambda_{\mathrm{Agt}}[B \mapsto \lambda'_B]$ and $\delta'_D = (\ell, g'_D, z'_D, \ell')$. We want an action (d'_B, δ'_B) such that $\mathrm{Tab}((\ell, v), m_{\mathrm{Agt}}[B \mapsto (d'_B, \delta'_B)]) = (\ell', v')$. For that it is enough to find a delay d'_B such that the same player gets selected in $m_{\mathrm{Agt}}[B \mapsto (d'_B, \delta'_B)]$ than in $\lambda_{\mathrm{Agt}}[B \mapsto \lambda'_B]$, and that $d'_D = d(m_{\mathrm{Agt}}[B \mapsto (d'_B, \delta'_B)])$ is such that the valuation after the reset $[z'_D \leftarrow 0](v + d'_D)$ is the same than v'. We make a distinction according to whether all the clocks are reset or not.

First consider the case where not all the clocks are reset, i.e. $z'_D \neq X$. In that case we have $r'_D = q' = r'_C$, $p'_D = p' = p'_C$. Notice that by Lemma 6.2

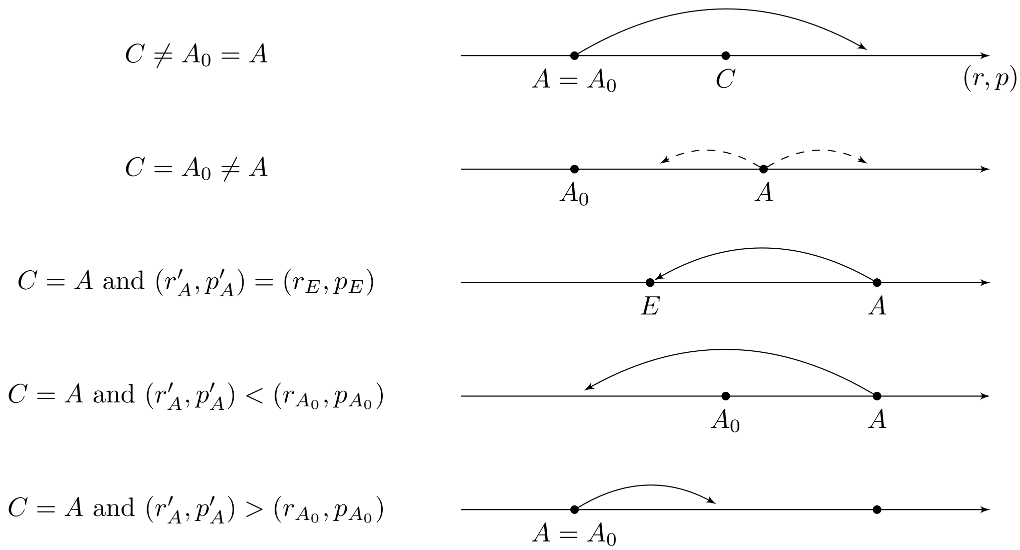


Figure 6.6: Five kinds of deviations

if $B \neq A_0$ then D is either B or A_0, so will select the delay by distinguishing these different cases:

- if $D = B$, we consider delay $d'_B = d'$. First, we notice that playing (d'_B, δ_B) is allowed to B, this is because $v + d' \in r'_C = r'_B$ and (r'_B, δ'_B) is allowed to B in $\mathcal{R}$. We now show that $[z'_B \leftarrow 0](v + d'_B) = v'$. Consider a clock $x \in X$, let us write $v'' = [z'_B \leftarrow 0](v + d'_B)$. If $v''(x) = 0$, then all the valuations in the region containing v'' also evaluates x to 0. The region containing v'' is r' and in particular $v' \in r'$, so $v'(x) = 0$. Similarly if $v'(x) = 0$ then $v''(x) = 0$. Now if $v''(x) \neq 0$, then also $v'(x) \neq 0$, this means that this clock was not reset by z'_B or z'_C, so $v''(x) = (v + d')(x) = v'(x)$. Therefore $[z'_B \leftarrow 0](v + d'_B) = v'$. It remains to show B is also selected in $m_{\mathrm{Agt}}[B \mapsto (d'_B, \delta'_B)]$, for this we examine the 5 different kinds of deviations:
  - if $C = A_0 \neq A$, then $d'_B = d_{A_0}$. We have the following equalities:
    - $(r'_B, p'_B) = (r'_D, p'_D)$ because $B = D$;
    - $(r'_D, p'_D) = (r'_C, p'_C)$ because $z'_D \neq X$;
    - $(r'_C, p'_C) = (r_C, p_C)$ because $C \neq A$;
    - $(r_C, p_C) = (r_{A_0}, p_{A_0})$ because $C = A_0$.

    From this, we deduce that $(r'_B, p'_B) = (r_{A_0}, p_{A_0})$. Now, since B is selected in $\lambda_{\mathrm{Agt}}[B \mapsto \lambda'_B]$, it is that $B \geq_\ell A_0$. As $d'_B = d_{A_0}$, B is also selected in $m_{\mathrm{Agt}}[B \mapsto (d'_B, \delta'_B)]$.
  - if $C \neq A_0 = A$, then $d' = d_C$. We have the following (in)equalities:
    - $(r'_B, p'_B) = (r'_D, p'_D)$ because $B = D$;

    - $(r'_D, p'_D) = (r'_C, p'_C)$ because $z'_D \neq X$;
    - $(r'_C, p'_C) = (r_C, p_C)$ because $C \neq A$;
    - $(r_C, p_C) \geq (r_{A_0}, p_{A_0})$ because A_0 is selected in λ_{Agt};
    - $(r_{A_0}, r_{A_0}) \geq (r'_B, p'_B)$ because $B = D$.

  From this we deduce that all these couples are equal. Notice from the construction of λ_{Agt} that if $p_C = p_{A_0}$ (which is the case here) then $d_C = d_{A_0}$. Hence $d'_B = d' = d_C = d_{A_0}$. Moreover, because A is minimal among the suspects and $A = A_0$, the player B has the priority over A_0, so B is selected in $m_{\mathrm{Agt}}[B \mapsto (d'_B, \delta'_B)]$.
  - if $C = A$ and there is a player E such that $(r'_A, p'_A) = (r_E, p_E)$, then $d' = d_E$. Hence $d'_B = d' = d_E$. Since A is selected in $\lambda_{\mathrm{Agt}}[A \mapsto \lambda'_A]$ this means that $A \geq_\ell E$. Since A is minimal among the suspects $B \geq_\ell E$, so B is selected in $m_{\mathrm{Agt}}[B \mapsto (d'_B, \delta'_B)]$.
  - if $C = A$ and $(r'_A, p'_A) < (r_{A_0}, p_{A_0})$, then $d' < d_{A_0}$. Therefore $d'_B = d' < d_{A_0}$, so B is selected in $m_{\mathrm{Agt}}[B \mapsto (d'_B, \delta'_B)]$.
  - If $C = A$ and $(r'_A, p'_A) > (r_{A_0}, p_{A_0})$ and $(r'_A, p'_A) \neq (r_E, p_E)$ for all E, then d' is smaller than d_E for all $E \neq A$. If $B \neq A_0$, we have the following (in)equalities:
    - $(r_{A_0}, p_{A_0}) \geq (r'_B, p'_B)$ because B is selected in $\lambda_{\mathrm{Agt}}[B \mapsto \lambda'_B]$,
    - $(r'_B, p'_B) = (r'_D, p'_D)$ because $B = D$;
    - $(r'_D, p'_D) = (r'_C, p'_C)$ because $z'_D \neq X$;
    - $(r'_C, p'_C) = (r'_A, p'_A)$ because $C = A$.

    Hence $(r'_A, p'_A) \leq (r_{A_0}, p_{A_0})$, which is a contradiction;

    If $A \neq A_0$, then $(r'_A, p'_A) \leq (r_{A_0}, p_{A_0})$ because A is selected in $\lambda_{\mathrm{Agt}}[A \mapsto \lambda'_A]$, this is again a contradiction;

    Otherwise $C = A = B = D = A_0$, and d'_B is smaller than all d'_E for $E \neq B$, therefore B is selected in $m_{\mathrm{Agt}}[B \mapsto (d'_B, \delta'_B)]$

- if $D = A_0 \neq B$, then we take $d'_B > d_{A_0}$ when $B >_\ell A_0$, $d'_B \geq d_{A_0}$ when $B <_\ell A_0$, this is possible since B is not selected in $\lambda_{\mathrm{Agt}}[B \mapsto \lambda'_B]$, and A_0 is selected in $m_{\mathrm{Agt}}[B \mapsto (d'_B, \delta'_B)]$. It remains to show that $d_{A_0} = d'$ in order to obtain the correct valuation after the transition. Notice that we have that $p'_C = p'_D = p_{A_0}$ and $r'_C = r'_D = r_{A_0}$,
  - if $C = A_0 \neq A$, then $d' = d_{A_0}$;
  - if $C \neq A_0 = A$, then $(r'_D, p'_D) = (r_{A_0}, p_{A_0})$ and $(r_C, p_C) = (r'_C, p'_C) = (r'_D, p'_D)$ therefore $(r_{A_0}, p_{A_0}) < (r_C, p_C)$ is not possible, which means that $(r_{A_0}, p_{A_0}) = (r_C, p_C)$, by construction of λ_{Agt}, as we already noticed before, this means $d_{A_0} = d_C$ and therefore $d' = d_{A_0}$;
  - otherwise $C = A$ and then $(r'_A, p'_A) = (r_{A_0}, p_{A_0})$, and therefore $d' = d_{A_0}$;

- otherwise $D \neq A_0 = B$, we take for B a delay $d'_B > d_D$ if $B \geq_\ell D$ or $d'_B \geq d_D$ if $B <_\ell D$, this is possible since B is selected in $\lambda_{\mathrm{Agt}}[B \mapsto \lambda'_B]$. We have to show that $d_D = d'$,

- if $C = A_0 \neq A$, then $d' = d_{A_0}$. We have the following equalities:
  - $(r_{A_0}, p_{A_0}) = (r_C, p_C)$ because $C = A_0$;
  - $(r_C, p_C) = (r'_C, p'_C)$ because $C \neq A$;
  - $(r'_C, p'_C) = (r'_D, p'_D)$ because $z'_D \neq X$;
  - $(r'_D, p'_D) = (r_D, p_D)$ because $B \neq D$.

  Therefore $(r_{A_0}, p_{A_0}) = (r_D, p_D)$, so by construction of λ_{Agt}, as we already noticed before, $d_D = d_{A_0}$ and therefore $d_D = d'$;
- if $C \neq A_0 = A$, then $d' = d_C$. We have that $A = A_0 = B$. $(r_C, p_C, C) = \min\{(r_E, p_E, E) \mid E \in \mathrm{Agt} \setminus \{A\}\}$ because C is selected in $\lambda_{\mathrm{Agt}}[A \mapsto \lambda'_A]$ and $C \neq A$. Likewise, since $D \neq A$, $(r_D, p_D, D) = \min\{(r_E, p_E, E) \mid E \in \mathrm{Agt} \setminus \{A\}\}$. Therefore $C = D$ and $d_D = d'$;
- if $C = A$ then:
  - $(r_D, p_D) = (r'_D, p'_D)$ because $B \neq D$;
  - $(r'_D, p'_D) = (r'_C, p'_C)$ because $z'_D \neq X$;
  - $(r'_C, p'_C) = (r'_A, p'_A)$ because $C = A$;

  therefore $(r'_A, p'_A) = (r_D, p_D)$. Hence $d' = d_D$.

Now, if all the clocks are reset by the transition, i.e. $z'_D = X$, only the fact that the correct player (i.e. D) is selected matters (and not the exact delay).

- if $D = B$, we have for all $E \neq B$ that $(r'_B, p'_B, B) \leq^3_\ell (r_E, p_E, E)$, it is possible to take d'_B such that $v + d'_B \in r'_B$ and $(d'_B, B) \leq^2_\ell (d_E, E)$, for all E:
  - if $r'_B < r_E$ then any d'_B such that $v + d'_B \in r'_B$ works;
  - if $r'_B = r_E$ and $p'_B < p_E$, then it means the region is time elapsing since otherwise the only available choice for p'_B and p_E would be 1, therefore it is possible to take a delay shorter than d_E and still end in region r'_E;
  - if $r'_B = r_E$, $p'_B < p_E$ and $B >_\ell E$; then we can take a delay $d'_B \leq d_E$;

  Hence we can select a delay such that $(d'_B, B) \leq^2_\ell (d_E, E)$, for all E, and then B is selected in $\lambda_{\mathrm{Agt}}[B \mapsto \lambda'_B]$;
- if $D = A_0 \neq B$, we take d'_B such that $v + d'_B \in r'_B$ and $(d_{A_0}, A_0) <^2_\ell (d'_B, B)$, similarly to the previous case we can show that this possible since $(r_{A_0}, p_{A_0}, A_0) <^3_\ell (r'_B, p'_B, B)$. Then A_0 is selected in $\lambda_{\mathrm{Agt}}[B \mapsto \lambda'_B]$;
- if $D \neq A_0 = B$, we take d'_B such that $v + d'_B \in r'_B$ and $(d_D, D) <^2_\ell (d'_B, B)$, this is possible since $(r_D, p_D, D) <^3_\ell (r'_B, p'_B, B)$, Then D is selected in $\lambda_{\mathrm{Agt}}[B \mapsto \lambda'_B]$; □

Thanks to the two last lemmas, we showed that the relation we defined complies with the definition of a game simulation.

Corollary 6.6. *The region game $\mathcal{R}$ from state $(\ell_0, [\mathbf{0}], [\mathbf{0}], 1)$ with constraint* Allow *on the moves, simulates the game $\mathcal{G}$ from configuration $(\ell_0, \mathbf{0})$.*

6.2.5 From Region Game $\mathcal{R}$ to Timed Game $\mathcal{G}$.

We will now prove simulation in the other direction. For this we consider a move n_{Agt} in $\mathcal{R}$ and a clock valuation v, and define the move μ_{Agt} in $\mathcal{G}$ as follows:

- if $n_A = \perp$, then $\mu_A = \perp$;
- if $n_A = (r_A, p_A, \delta_A)$, then $\mu_A = (f(v, r_A, p_A), \delta_A)$.

If n_A is allowed to player A in (ℓ, r), then μ_A is also allowed to A in (ℓ, v), since it corresponds to playing the same transition in the same region.

A first step towards the correctness of this construction is that the selected players are the same for both profiles.

Lemma 6.7. $\mathrm{Select}(n_{\mathrm{Agt}}) = \mathrm{Select}(\mu_{\mathrm{Agt}})$

Proof. Let $A = \mathrm{Select}(n_{\mathrm{Agt}})$, $(r_A, p_A, A) = \min_{\leq^3_\ell}\{(r_B, p_B, B) \mid B \in \mathrm{Agt}\}$. Let B be a player different from A, we have that $(r_A, p_A, A) <^3_\ell (r_B, p_B, B)$, there are three possibilities:

- if $r_A < r_B$ then $f(v, r_A, p_A) < f(v, r_B, p_B)$ because $f(v, r_A, p_A) \in r_A$ and $f(v, r_B, p_B) \in r_B$;
- if $r_A = r_B$ and $p_A < p_B$ then by construction of f we have that $f(v, r_A, p_A) < f(v, r_B, p_B)$;
- if $r_A = r_B$, $p_A = p_B$ and $B <_\ell A$ then $f(v, r_A, p_A) = f(v, r_B, p_B)$;

Writing $\mu_B = (d_B, \delta_B)$ for each player B, we have $d_B = f(v, r_B, p_B)$, and in all cases $(d_A, A) = \min_{\leq^2_\ell}\{(d_B, B) \mid B \in \mathrm{Agt}\}$, hence A is selected in μ_{Agt}. □

Lemma 6.8. *Let (ℓ, r, q, p) be a state in $\mathcal{R}$, v be a valuation in r, n_{Agt} be a move in $\mathcal{R}$ and μ_{Agt} be the corresponding move in $\mathcal{G}$. If $\mathrm{Tab}_{\mathcal{R}}((\ell, r, q, p), n_{\mathrm{Agt}}) = (\ell', r', q', p')$ then $\mathrm{Tab}((\ell, v), \mu_{\mathrm{Agt}}) = (\ell', v')$ for some v' in the region r'.*

Proof. Since $\mathrm{Tab}_{\mathcal{R}}((\ell, r, q, p), n_{\mathrm{Agt}}) = (\ell', r', q', p')$, the action (r_A, p_A, δ_A) of the selected player $A = \mathrm{Select}(n_{\mathrm{Agt}})$ is such that:

- $\delta_A = (\ell, g_A, z_A, \ell')$;
- $r' = [z_A \leftarrow 0] r_A$;
- $v + f(v, r_A, p_A) \in r_A$ by definition of f;
- $v' = [z_A \leftarrow 0](v + f(v, r_A, p_A))$ because A is also selected in μ_{Agt} by Lemma 6.7.

Therefore $\mathrm{Tab}((\ell, v), \mu_{\mathrm{Agt}}) = (\ell', v')$ with $v' \in r'$. □

Lemma 6.9. *Let $\mathcal{G}$ be a timed game, (ℓ, v) and (ℓ', v') be two configurations, q be a region and p an index in $[\![0, 4]\!]$. We write r and r' for the region containing v and v' respectively. If n_{Agt} is legal and allowed by* Allow *in (ℓ, r, q, p), let μ_{Agt} be the corresponding move from (ℓ, v) in $\mathcal{G}$, then there is a region q' and an index $p' \in [\![0, 4]\!]$ such that*

$$\mathrm{Susp}_{\mathcal{G}}(((\ell, v), (\ell', v')), \mu_{\mathrm{Agt}}) \subseteq \mathrm{Susp}_{\mathcal{R}}(((\ell, r, q, p), (\ell', r', q', p')), n_{\mathrm{Agt}})$$

Proof. We first need to select the right index p'. Let A be the minimal (with respect to $\leq_\ell$) suspect in $\mathrm{Susp}_{\mathcal{G}}(((\ell, v), (\ell', v')), \mu_{\mathrm{Agt}})$, and $\mu'_A = (d'_A, \delta'_A)$ be such that $\mathrm{Tab}((\ell, v), \mu_{\mathrm{Agt}}[A \mapsto \mu'_A]) = (\ell', v')$. We write A_0 the player selected in μ_{Agt} and C the player selected in $\mu_{\mathrm{Agt}}[A \mapsto \mu'_A]$, we also write (d'_C, δ'_C) the action of C in the action profile $\mu_{\mathrm{Agt}}[A \mapsto \mu'_A]$. We distinguish three cases according to the way A deviates:

- if $C \neq A$ then $p' = p_C$;
- if $C = A$ and there is a player E such that $d'_A = d_E$ then $p' = p_E$. This index is uniquely defined by construction of μ_{Agt} because if $d_E = d_F$ for two players E and F then this means that $p_E = p_F$;
- otherwise $p' = 0$ if the region is time elapsing and $p' = 1$ otherwise;

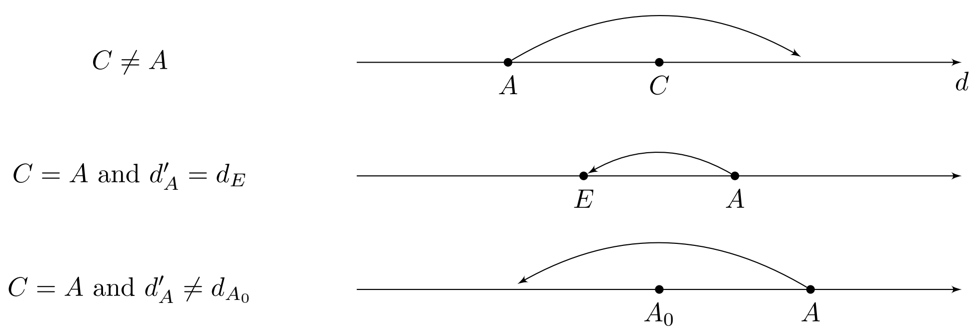


Figure 6.7: Three kinds of deviations.

Now let B be any suspect in $\mathrm{Susp}_{\mathcal{G}}(((\ell, v), (\ell', v')), \mu_{\mathrm{Agt}})$, there is an action $\mu'_B = (d'_B, \delta'_B)$ such that $\mathrm{Tab}((\ell, v), \mu_{\mathrm{Agt}}[B \mapsto \mu'_B]) = (\ell', v')$. We look for an action $n'_B = (r'_B, p'_B, \delta'_B)$ for player B such that $\mathrm{Tab}_{\mathcal{R}}((\ell, r, p), n_{\mathrm{Agt}}[B \mapsto n'_B]) = (\ell', r', p')$. In all cases r'_B is the region containing $v + d'_B$. Then we choose p'_B. Let D be the selected player in $\mu_{\mathrm{Agt}}[B \mapsto \mu'_B]$, we write (d'_D, δ'_D) her action in the profile $\mu_{\mathrm{Agt}}[B \mapsto \mu'_B]$, and $\delta'_D = (\ell, g'_D, z'_D, \ell')$.

First consider the case where $z'_D \neq X$. In that case, we show $d'_D = d'_C$, this is because timing information is kept during the transition. Let $c \in X \setminus z'_D$ be a clock that is not reset, then $v'(c) = [z_D \leftarrow 0](v + d'_D)(c) = v(c) + d'_D$, so $d'_D = v'(c) - v(c)$ and similarly this is also equal to d'_C, hence $d'_D = d'_C$.

- if $B = D$, we take $p'_B = p'$, we show that B is selected in $n_{\mathrm{Agt}}[B \mapsto n'_B]$,
  - if $C \neq A$, then $p' = p_C$. First we have:
    - $d'_B = d'_D$ because $B = D$;
    - $d'_D = d'_C$ because $z'_D \neq X$;
    - $d'_C = d_C$ because $C \neq A$.

    So $d'_B = d_C$ and since B is selected in $\mu_{\mathrm{Agt}}[B \mapsto \mu'_B]$ this means that $B \geq_\ell C$. Now we will distinguish two cases:

If $B = A$ then : $(r_C, p_C) \le (r_E, p_E)$ for all $E \ne A$ because C is selected in $\mu_{\mathrm{Agt}}[A \mapsto \mu'_A]$ and $A \ne C$. Therefore, as $(r'_B, p'_B) = (r_C, p_C)$, $B = A$ and $B \ge_\ell C$, B is selected in $n_{\mathrm{Agt}}[B \mapsto n'_B]$.

If $B \ne A$ then: $d_A \ge d'_B$ because B is selected in $\mu_{\mathrm{Agt}}[B \mapsto \mu'_B]$ and $A \ne B$. As we showed $d_C = d'_B$, this means $d_A \ge d_C$. Moreover, for all $E \ne A$, $(d_C, C) <^2_\ell (d_E, E)$ because C is selected in $\mu_{\mathrm{Agt}}[A \mapsto \mu'_A]$. By construction of μ_{Agt} we have that for any player E, $(r_C, p_C) \le (r_E, p_E)$ and $(r_C, p_C) < (r_E, p_E)$ if $C <_\ell E \ne A$. As $(r'_B, p'_B) = (r_C, p_C)$, $B \ge_\ell C$ and $B >_\ell A$, for any player E, $(r'_B, p'_B) \le (r_E, p_E)$ and $(r'_B, p'_B) < (r_E, p_E)$ if $B <_\ell E$, so B is selected in $n_{\mathrm{Agt}}[B \mapsto n'_B]$.

  - if $C = A$ and $d'_A \ne d_E$ for any E then:

    If the region is time-elapsing $p' = 0$. For all the players $E \ne B$, $d'_B \le d_E$ because B is selected in $\mu_{\mathrm{Agt}}[B \mapsto \mu'_B]$. Therefore $r'_B \le r_E$, and as these players are restricted to play indexes $p_E \in [\![1, 3]\!]$, we also have $p'_B < p_E$. Hence B is selected in $n_{\mathrm{Agt}}[B \mapsto n'_B]$.

    Otherwise the region is not time-elapsing, then for all $E \ne B$:

    * if $d'_B = d_E$ and $B >_\ell E$, then $r'_B = r_E$, $p'_B = p_E$ and we have $(r'_B, p'_B, B) <^3_\ell (r_E, p_E, E)$;
    * otherwise $d'_B < d_E$, and then $r'_B < r_E$.

    Hence B is selected in $n_{\mathrm{Agt}}[B \mapsto n'_B]$.

  - if $C = A$ and $d'_A = d_{A_0}$ then $p' = p_{A_0}$. As A is selected in $\mu_{\mathrm{Agt}}[A \mapsto \mu'_A]$, it means that $A \ge_\ell A_0$ and as A is minimal among the suspect $B \ge_\ell A_0$. Moreover $p'_B = p' = p_{A_0}$, therefore B is selected in $n_{\mathrm{Agt}}[B \mapsto n'_B]$;

- otherwise, $B \ne D$, we take $p'_B = 4$ if the region is open, and $p'_B = 1$ otherwise, then D is selected in $n_{\mathrm{Agt}}[B \mapsto n'_B]$, we show that $p_D = p'$:

  - if $C \ne A$, then $p' = p_C$.

    * $d_D = d'_D$ because $B \ne D$;
    * $d'_D = d'_C$ because $z'_D \ne X$;
    * $d'_C = d_C$ because $C \ne A$.

    Hence $d_D = d_C$ and by construction of μ_{Agt} this means that $p_D = p_C = p'$;

  - if $C = A \ne A_0$, then:

    * $d_{A_0} \le d_D$ because A_0 is selected in μ_{Agt};
    * $d_D = d'_D$ because $D \ne B$;
    * $d'_D = d'_C$ because $z'_D \ne X$;
    * $d'_C \le d'_{A_0}$ because C is selected in $\mu_{\mathrm{Agt}}[A \mapsto \mu'_A]$;
    * $d'_{A_0} = d_{A_0}$ because $A \ne A_0$;

    Hence all these delays are equal and $d'_A = d_{A_0}$ and then $p' = p_{A_0}$. Moreover $d_D = d_{A_0}$, and by construction of μ_{Agt} this means that $p_D = p_{A_0}$. Therefore $p_D = p'$.

  - Otherwise $C = A = A_0$,

* $d_D = d'_D$ because $D \neq B$;
* $d'_D = d'_C$ because $z'_D \neq X$;
* $d'_C = d'_A$ because $C = A$;

Hence $d_D = d'_A$ and $p' = p_D$ by construction of p'.

Now, if $z'_D = X$, only the fact that the correct player (i.e. D) is selected matters, and not the chosen index, since p' will be equal to 1 anyway.

- if $D = B$, then:
  - if r'_B is time elapsing, we take $p'_B = 0$. Since B is selected in $\mu_{\mathrm{Agt}}[B \mapsto \mu'_B]$, for all $E \neq B$, $d'_B \leq d_E$, hence we also have that $r'_B \leq r_E$ and therefore $(r'_B, p'_B, B) <^3_\ell (r_E, p_E, E)$;
  - otherwise r'_B is not time elapsing, we have to take $p'_B = 1$. Since B is selected in $\mu_{\mathrm{Agt}}[B \mapsto \mu'_B]$, for all $E \neq B$, $(d'_B, B) <^2_\ell (d_E, E)$. If $d'_B = d_E$ then $B >_\ell E$ and $(r'_B, p'_B, B) <^3_\ell (r_E, p_E, E)$. Otherwise $r'_B < r_E$ because r'_B is not time elapsing, and we also have $(r'_B, p'_B, B) <^3_\ell (r_E, p_E, E)$.

  In both cases B is selected;
- if $D \neq B$, then:
  - if r'_B is time elapsing, we take $p'_B = 4$. Since D is selected in $\mu_{\mathrm{Agt}}[B \mapsto \mu'_B]$, $d_D \leq d'_B$, we also have that $r_D \leq r'_B$. Since D is restricted to actions with $p_D \in [\![1, 3]\!]$, $p_D < p'_B$. Hence $(r_D, p_D, D) <^3_\ell (r'_B, p'_B, B)$, and
  - otherwise r'_B is not time elapsing, we have to take $p'_B = 1$. Since D is selected in $\mu_{\mathrm{Agt}}[B \mapsto \mu'_B]$, $(d_D, D) \leq^2_\ell (d'_B, B)$. If $d'_B = d_D$ then $r'_B = r_D$, $p'_B = p_D = 1$ and $B <_\ell D$, hence $(r_D, p_D, D) <^3_\ell (r'_B, p'_B, B)$. Otherwise $r_D < r'_B$ because r'_B is not time elapsing, and we also have $(r'_B, p'_B, B) <^3_\ell (r_E, p_E, E)$.

  By construction of μ_{Agt} we also have that $(r_D, p_D, D) <^3_\ell (r_E, p_E, E)$ for all $E \in \mathrm{Agt} \setminus \{B, D\}$ so in this case D is selected. $\square$

We conclude from the two last lemmas, the simulation in this direction.

Corollary 6.10. *The region game $\mathcal{G}$ from configuration $(\ell_0, \mathbf{0})$ simulates the game $\mathcal{R}$ from state $(\ell_0, [\mathbf{0}], [\mathbf{0}], 1)$ with constraint* Allow *on the moves in $\mathcal{R}$.*

6.2.6 Conclusion of the Proof

We now conclude the proof of Proposition 6.1.

Proof. Lemmas 6.4, 6.5, 6.8, and 6.9 show that the relation $\triangleleft$ between the timed game $\mathcal{G}$ and its associated region game $\mathcal{R}$ is a simulation in both directions when actions in the region game are restricted to use indexes $p \in [\![1, 3]\!]$. Since the preference relation is region invariant, the simulation is preference preserving.

Hence by the Proposition 3.5 there is a Nash equilibrium in $\mathcal{G}$ from $(s, \mathbf{0})$ if, and only if, there is a Nash equilibria in $\mathcal{R}$ from $(s, [\mathbf{0}])$, with constraint Allow on the moves. $\square$

6.3 Complexity Analysis

6.3.1 Size of the Region Game

The region game $\mathcal{R}$ has size exponential in the size of $\mathcal{G}$:

$$
\begin{aligned}
|\mathfrak{R}| &= |X|! \cdot (4M+4)^{|X|} \\
|\mathrm{States}| &= 5 \cdot |\mathrm{Loc}| \cdot |\mathfrak{R}|^2 \\
|\mathrm{Act}| &= 5 \cdot |\mathfrak{R}| \cdot |\mathrm{Trans}| \\
|\mathrm{Mov}| &\leq |\mathrm{States}| \cdot |\mathrm{Agt}| \cdot |\mathrm{Act}| \\
|\mathrm{Tab}| &\leq |\mathrm{States}|^2 \cdot |\mathrm{Act}|^{|\mathrm{Agt}|}
\end{aligned}
$$

It is exponential both because the number of regions is exponential, and because the size of the transition table can be exponential in the number of agents. If for instance, for one location, each player has 2 outgoing edges, then the number of edges in the timed games is $2 \cdot |\mathrm{Agt}|$ but in the transition table of the corresponding region game there must be $2^{|\mathrm{Agt}|}$ cells: one for all possible moves.

6.3.2 Algorithm

We will consider in this part objectives given by deterministic Rabin automata reading the locations of the timed game $\mathcal{G}$. We recall that deterministic Rabin automata can describe any ω-regular condition and we presented in Section 4.8 an exponential algorithm to decide the constraint existence problem in concurrent games.

Theorem 6.11. *The existence problem with constrained outcomes, in timed game with objectives given by deterministic Rabin automata can be solved in* EXPTIME.

Proof. The algorithm consists in constructing the region game and solving the constrained existence problem on the region game. Thanks to Prop. 6.1, we can recover Nash equilibria in the original timed game. The execution time of the algorithm we gave for Rabin automata, was only exponential in the number of agents and the number of Rabin pairs, but not in the size of the arena. The blow-up induced by the region transformation is therefore orthogonal and the global execution time remains a simple exponential. To be precise the execution time is bounded (up to constant factor) by the following expression:

$$2^{|\mathrm{Agt}|} \cdot \left(|\mathrm{Loc}|^2 \cdot |\mathfrak{R}|^{4+|\mathrm{Agt}|} \cdot |\mathrm{Trans}|^{|\mathrm{Agt}|}\right)^{3\sum_A k_A} \cdot \left(\sum_A k_A\right)!$$

$$+2^{|\mathrm{Agt}|} \cdot \left(\prod_{A\in\mathrm{Agt}} k_A \cdot 2^{k_A}\right) \cdot \left(|\mathrm{Loc}|^2 \cdot |\mathfrak{R}|^{4+|\mathrm{Agt}|} \cdot |\mathrm{Trans}|^{|\mathrm{Agt}|}\right)^3 \cdot \sum_A k_A$$

where k_A is the number of Rabin pairs describing the objective of player A.

□

6.3.3 Hardness

From the point of view of complexity classes, our algorithm is optimal, as we will prove EXPTIME-hardness for the restricted case of Büchi conditions. This is proved by encoding *countdown games* [35], that we now introduce.

Countdown games. A *countdown game* $\mathcal{C}$ is played on a weighted graph (N, E), whose edges are labeled with positive integer weights encoded in binary. A *move* of the game from configuration $(n, c) \in N \times \mathbb{Z}$ is determined jointly by both players, as follows. First, `Eve` chooses a number d such that $(n, d, n') \in E$ for some node n'. Then `Adam` chooses a node $n' \in N$ such that $(n, d, n') \in E$. The resulting configuration is $(n', c - d)$. If a configuration (n, c) with $c = 0$ is reached, then the game stops and `Eve` wins.

Example 11. An example of a countdown game is represented in Fig. 6.8. In node n_1, `Eve` chooses an integer among 3 and 5, then `Adam` has to choose one of the outgoing edges which is labeled by this integer. For instance, if `Eve` chooses 5, `Adam` has the choice to either go to n_2 or n_4, and then the counter is decremented by 5. `Eve` wins if the counter reaches exactly 0.

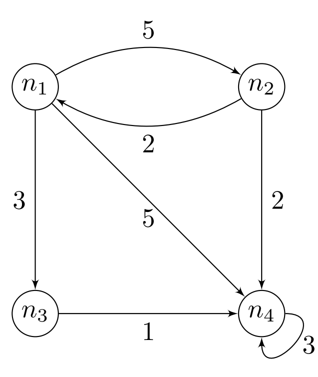


Figure 6.8: A countdown game $\mathcal{C}$.

Given a countdown game, we express its semantics in terms of a (infinite) turn-based game with a Büchi objective. We only give the function Mov for legal actions, and as the game is turn-based, this is enough to deduce the transition table Tab.

- $\mathrm{Agt} = \{\mathtt{Eve}, \mathtt{Adam}\}$;
- $\mathrm{States} = \mathrm{States}_\exists \cup \mathrm{States}_\forall \cup \{w_\exists\}$ where the states controlled by Eve are $\mathrm{States}_\exists = N \times \mathbb{Z}$, the states controlled by Adam are $\mathrm{States}_\forall = \{(n, c, d) \mid c \in \mathbb{Z} \text{ and } \exists (n, d, n') \in E\}$, and the state $w_\exists$ corresponds to Eve winning and the game being stopped;
- $\mathrm{Owner}(s) = \mathtt{Eve}$ if $s \in \mathrm{States}_\exists \cup \{w_\exists\}$ and $\mathrm{Owner}(s) = \mathtt{Adam}$ otherwise;
- if $c \neq 0$, $\mathrm{Mov}((n, c), \mathtt{Eve}) = \{(n, c, d) \mid \exists (n, d, n') \in E\}$;
- if $c = 0$, $\mathrm{Mov}((n, c), \mathtt{Eve}) = \{w_\exists\}$;
- $\mathrm{Mov}(w_\exists, \mathtt{Eve}) = \{w_\exists\}$;
- $\mathrm{Mov}((n, c, d), \mathtt{Adam}) = \{(n', c - d) \mid \exists (n, d, n') \in E\}$;
- the preference relation is given by a Büchi objective for Eve, with the target $T_\exists = \{w_\exists\}$.

Given an initial configuration (n, c), if we forget about those with $c < 0$ which are always winning for Adam, then the number of reachable configurations is finite. To be more precise, it is exponential, because integer constants like c and labels of transitions are written in binary. We note that given a countdown game and an initial configuration, the existence of a winning strategy for Eve is EXPTIME-complete [35].

We first prove the result for the value problem and reachability objectives. EXPTIME-hardness of this problem was already known for timed games in general [30]. Using a reduction from countdown games, we provide here a simpler proof which only uses two clocks.

Proposition 6.12. *The value problem for timed games with Büchi objectives and only two clocks is* EXPTIME*-hard.*

Proof. Let $\mathcal{C}$ be a countdown game, (n_0, c_0) be a configuration $\mathcal{C}$. We build a timed game $\mathcal{G}$ such that there is a winning strategy for Eve in $\mathcal{G}$ from initial state $(n_0, \mathbf{0})$ if, and only if, there is a winning strategy for Eve in $\mathcal{C}$ from configuration (n_0, c_0). It is defined as follow:

- $\mathrm{Loc} = N \cup \{(n, d) \in N \times \mathbb{N} \mid \exists (n, d, n') \in E\} \cup \{w_\exists\}$, the locations in N correspond to the nodes of the countdown game and will be controlled by Eve, the locations of the form (n, d) are controlled by Adam is the winning state for Eve;
- $X = \{x, y\}$, the counter of the game will be encoded in the valuation of clock y by $c_0 - v(y)$, clock x will be used to decrement the value of the counter by the integer corresponding to the selected transition;
- $\mathrm{Agt} = \{\mathtt{Adam}, \mathtt{Eve}\}$;

- Trans = $\text{Trans}_\exists \cup \text{Trans}_\forall$, where :

$$\begin{aligned}\text{Trans}_\exists =& \{(s,(x=0 \wedge \ y \neq c_0), \varnothing, (s,d)) \mid \exists (s,d,s') \in T\}\\ &\cup \{(s,(x=0 \ \wedge \ y=c_0), \varnothing, w_\exists)\}\\ &\cup \{(w_\exists, \textsf{true}, \varnothing, w_\exists)\}\\ \text{Trans}_\forall =& \{((s,d),(x=d),\{x\},s') \mid (s,d,s') \in T\}\end{aligned}$$

- $\text{Owner}(t) = \begin{cases} \texttt{Eve} \text{ if } t \in \text{Trans}_\exists \\ \texttt{Adam} \text{ if } t \in \text{Trans}_\forall \end{cases}$
- The preference relation is given by a Büchi objective for `Eve`, with the target $T_\exists = \{w_\exists\}$.

Remark. Compared to the configurations of $\mathcal{C}$ we do not keep the values of the counter in the locations, since there are an exponential number of possible value for it, we will encode it in the clock valuation.

This construction is illustrated in Fig. 6.9, for the game we presented in Example 11.

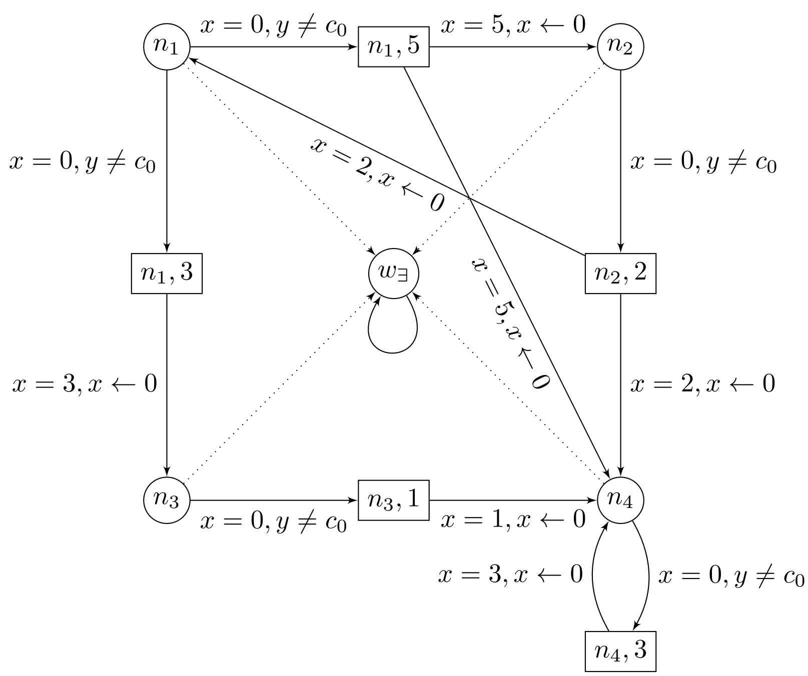


Figure 6.9: The encoding of the countdown game $\mathcal{C}$ as a timed game. Locations controlled by `Eve` are represented with circles and those controlled by `Adam` with rectangles. All the dotted transitions are labeled by $x = 0 \wedge y = c$.

We will show that these two games simulate each other in the sense of Section 3.3, although game simulation was define for Nash equilibrium we can deduce from Prop. 3.5, this simple corollary.

Corollary 6.13 (of Prop. 3.5)**.** *Let $\mathcal{G}$ and $\mathcal{G}'$ be two zero-sum games involving the same players* `Eve` *and* `Adam` *with a reachability objective for* `Eve`*. Fix two states s_0 and s'_0 in $\mathcal{G}$ and $\mathcal{G}'$ respectively. Assume that $\lhd$ is a preference-preserving game simulation from (s_0, s'_0). If* `Eve` *has a winning strategy in $\mathcal{G}$ from s_0 then she has a winning strategy in $\mathcal{G}'$ from s'_0.*

Proof. `Eve` has a winning strategy in $\mathcal{G}$ corresponds to the fact that there is a Nash equilibrium where she wins in the game where `Adam` has the opposite objective. Hence this equivalence is a direct consequence of Prop. 3.5. $\square$

Moreover, when the games are turn-based, which is the case of $\mathcal{C}$ and $\mathcal{G}$ here, we can simplify the definition of a game simulation as follows. If $\mathcal{G}$ and $\mathcal{G}'$ are two turn-based games, a relation $\lhd$ is a game simulation between $\mathcal{G}$ and $\mathcal{G}'$, if $s \lhd s'$ implies that:

1. $\mathrm{Owner}(s) = \mathrm{Owner}(s')$;
2. for each t successor of s there is t' successor of s' with $t \lhd t'$;
3. for each t' successor of s' there is t successor of s with $t \lhd t'$.

We then define the relation $\lhd$ between $\mathcal{C}$ and $\mathcal{G}$ in the following manner. For the states controlled by `Eve`, $(n, c) \lhd (n', v)$ if, and only if, $n = n'$, $v(x) = 0$ and $v(y) = c_0 - c$. For the states controlled by `Adam`, $(n, c, d) \lhd ((n', d'), v)$ if, and only if, $n = n'$, $d = d'$ and $v(x) = 0$ and $v(y) = c_0 - c$, and $w_\exists \lhd (w_\exists, v)$ for any valuation v.

First remark that $\lhd$ is preference preserving. Let ρ be a path in $\mathcal{C}$ and ρ' be a path in $\mathcal{G}$ such that $\rho \lhd \rho'$. In $\mathcal{C}$, ρ is winning if, and only if, it reaches $w_\exists$ which if and only if, ρ' reaches $w_\exists$ since they are equivalent, and then ρ' is winning in $\mathcal{G}$.

We now prove that it is a game simulation.

Lemma 6.14. *The relation $\lhd$ is a game simulation between $\mathcal{C}$ and $\mathcal{G}$.*

Proof. For a configuration $(n, 0)$ in $\mathcal{C}$, the corresponding configuration in $\mathcal{G}$ is (n, v) with $v(x) = 0$ and $v(y) = c_0$, then the only available transition goes to $w_\exists$ in both cases.

Let (n, c) with $c \neq 0$ be a configuration controlled by `Eve` in $\mathcal{C}$, and (s, v) the corresponding configuration in $\mathcal{G}$, $v(x) = 0$ and $v(y) = c_0 - c$. For a successor configuration (n, c, d), there is a transition $(n, (x = 0 \wedge y \neq c_0), \varnothing, (n, d))$ in $\mathcal{G}$. We note that $v(y) \neq c_0$ since $c \neq 0$, hence there is a successor $((n, d), v')$ of (n, v) such that $v'(y) = c_0 - c$, so that it is equivalent to (n, c, d). Now for the configuration (n, c, d) controlled by `Adam`, let $((n, d), v)$ be the corresponding configuration in $\mathcal{G}$. For a successor $(n', c - d)$, there is a transition $((n, d), (x =$

$d), \{x\}, n')$ in $\mathcal{G}$. Hence there is a successor (n', v') of $((n, d), v)$ such that $v'(y) = v(y) + d = c_0 - (c - d)$, so that it is equivalent to $(n', c - d)$.

Now in the other direction, let (n, v) be a configuration controlled by `Eve` in $\mathcal{G}$, and (n, c) the corresponding configuration in $\mathcal{C}$: we have $v(x) = 0$ and $v(y) = c_0 - c$. For a transition to a location (n, d), there exists (n, d, n') in E. The constraint ensures that the valuation does not change, so (n, c, d) is equivalent to the next state of $\mathcal{G}$. For a state $((n, d), v)$ controlled by `Adam`, and a new location n', we have that $(n, d, n') \in E$, and because of the constraint on x the new valuation is v' such that $v'(y) = v(y) + d$, hence $(n', c - d)$ is a successor of (n, c, d) and it is equivalent to the new state of $\mathcal{G}$. □

From this lemma, and as $(n_0, c_0) \lhd (n_0, \mathbf{0})$, we deduce that there is a winning strategy for `Eve` in $\mathcal{C}$ from the configuration (n_0, c_0) if, and only if, there is a winning strategy for `Eve` in $\mathcal{G}$ from location n_0 and valuation $\mathbf{0}$. This proves the EXPTIME-hardness of the value problem for timed games with Büchi objectives. □

We now prove the result for existence problems.

Proposition 6.15. *The existence problem for timed games with Büchi objectives, only two clocks and two players is* EXPTIME*-hard.*

Proof. Note that we cannot directly apply Lem. 2.4, since timed games do not encode concurrent games in general. However, in this special case we will replace the initial concurrent module by a timed one, as is shown in Fig. 6.10. Let $\mathcal{G}$ be a two-player two-clocks turn-based zero-sum game. If `Eve` has a winning strategy in $\mathcal{G}$ then there is no equilibrium, since her interest is to play before `Adam` and play her winning strategy in $\mathcal{G}$, but then `Adam` can change his strategy to play a shorter delay from ℓ_0 and win. If `Eve` has no winning strategy in $\mathcal{G}$ then `Adam` has one, since turn-based Büchi games are determined. Then a strategy profile that consists in going to $w_\forall$ in the initial state, and for `Adam` to play his winning strategy in $\mathcal{G}$, forms a Nash equilibrium. Hence the existence problem for timed games with Büchi objectives is at least as hard as the value problem. □

We conclude this section by the following corollary that summarizes the results.

Corollary 6.16. *The value problem, the existence problem, the existence problem with constrained outcomes and the existence problem with constrained moves for timed games with Büchi objectives or objectives given by deterministic Rabin automata, are* EXPTIME*-complete.*

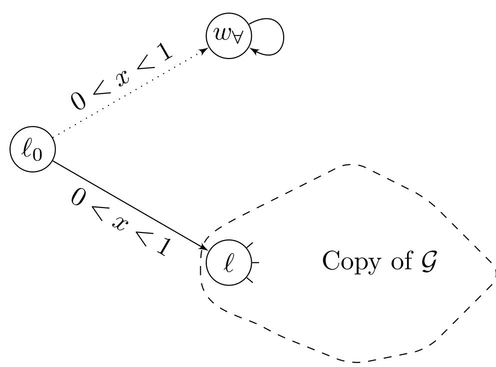


Figure 6.10: Extending game $\mathcal{G}$ with an initial module. The plain transition is controlled by `Eve` and the dotted one by `Adam`. `Eve` wins if she reaches the winning state of game $\mathcal{G}$, `Adam` wins if the winning state of `Eve` is never reached.

Chapter 7

Implementation

In this chapter, we present the implementation we made of some of the algorithms presented in this thesis. They are available as a tool called Praline, that can be downloaded from `http://www.lsv.ens-cachan.fr/Software/praline/`.

7.1 Algorithmic and Implementation Details

Praline is implemented in Ocaml[1]. The first version of Praline works on explicit graphs and implements the polynomial algorithm of Section 5.2.2. To represent and manipulate graphs we use the ocamlgraph library [16]. The game files are imported into the ocamlgraph representation, and then analyzed using the algorithm of Section 5.2.2 for Büchi games with maximize order. The product of the arena with deterministic Büchi automata is also implemented. The tool can thus handle objectives given by deterministic Büchi automata, and in particular reachability and safety objectives.

As explained in Section 4.3, a strongly connected component of the game defines a payoff. It is the payoff that is obtained by visiting all states of the component infinitely often. The algorithm looks for a strongly connected component of the suspect game, whose states are in the winning region of `Eve` with respect to its payoff. The algorithm works in polynomial time by recursively looking at the intersection of the strongly connected component with `Eve`'s winning region.

A difference between the algorithm we presented and its implementation, is that the suspect game is not computed explicitly. Instead it works on copies of the arena. We have one such copy for each possible set of suspects P such that a set (s, P) is accessible in the suspect game. There is a direct correspondence between a state (s, P) controlled by `Eve` in the suspect game and state s in the copy corresponding to the set P. There is also a correspondence between a state (s, P, m_{Agt}) controlled by `Adam` and the outgoing edge of s with label m_{Agt} in the copy corresponding to the set P. The computation of `Eve`'s winning

[1] `http://caml.inria.fr/`

region corresponds to what we called the *repellor transition system* in an earlier version of the work [6, 7].

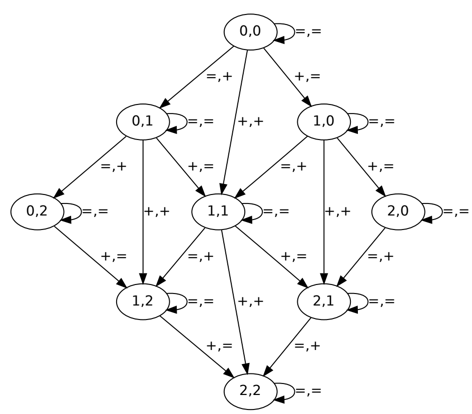
7.2 Input and Output

Praline looks for pure Nash equilibria in concurrent games. Objectives for the players are given by reachability, safety and Büchi objectives, or by deterministic Büchi automata. Each player can have several objectives; they are ordered according to an integer index. The goal for a player is then to satisfy the objective with the highest index.

The whole game is given to the tool in a file as the one in Fig. 7.1a, which describes the payoffs and the arena. The arena can either be given by a GML[2] or a Graphviz[3] file as in Fig. 7.1b,. The edges of this graph should be labeled by a tuple composed of the actions for each player. If the game contains a Nash equilibrium with some payoff v, then Praline returns at least one equilibrium with payoff w such that for every player i, $v_i \leq w_i$. For each solution, the tool outputs a file containing the full strategy profile. The profile is represented as an automaton indicating the actions that should be played by each player, one example is given in Fig. 7.2c. As can be seen on the figure those graphs are usually big. A more readable view of the equilibrium is the *shape* of the solution, which represents the outcome of the equilibrium, as in Fig. 7.2a and 7.2b. This gives an overview of what the evolution of the system should be.

```
arena "power_control.dot"
start "0,0"
objective 1 buchi
  "1,0" -> 140  ;
  "1,1" -> 44  ;
  "1,2" -> 21  ;
  "2,0" -> 73  ;
  "2,1" -> 38  ;
  "2,2" -> 23
objective 2 buchi
  "0,1" -> 140  ;
  "0,2" -> 73  ;
  "1,1" -> 44  ;
  "1,2" -> 38  ;
  "2,1" -> 21  ;
  "2,2" -> 23
```

(a) Game file “power_control.game”



(b) Arena file “power_control.dot”

Figure 7.1: Example of an input game for Praline

[2] `http://www.infosun.fim.uni-passau.de/Graphlet/GML/`
[3] `http://www.graphviz.org/`

7.3 Examples

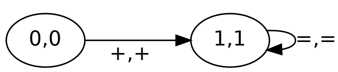
7.3.1 Power Control

We consider the problem of power control. At each step of the game, each agent i can choose to increase or not its emitting power p_i. The payoff for each state is given by the expression from [47]:

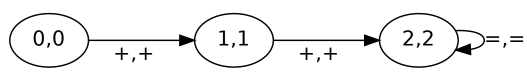
$$\frac{R}{p_i}\left(1 - e^{-0.5\gamma_i}\right)^L \tag{7.1}$$

The arena on which the game is played is represented in Fig. 7.1b for an instance with two agents, three possible levels of emission and some arbitrarily chosen parameters. The objectives of the game are described on Fig. 7.1a: each state is assigned a payoff according to Equation (7.1). This reads as follows: assuming $p_1 = 1$ and $p_2 = 2$, the current configuration corresponds to the node labeled by $(1,2)$ and its payoff is 21 for player 1 and 38 for player 2. The states that are not mentioned in the file have payoff 0 by default.

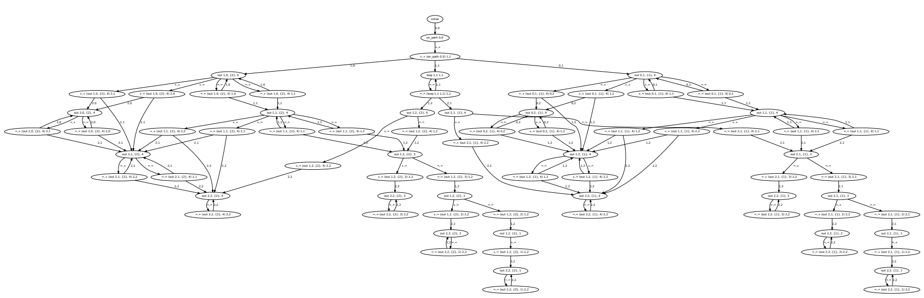
For this example, our tool gives two solutions, one Nash equilibrium with payoff 44 for each player and another one with payoff 23 for each. We give the shape of the two solutions in Fig. 7.2a and 7.2b. The first solution suggests that the players should limit their power to 1. Now if one player does not behave as expected, for instance she raises her power to 2, then the strategy for the other one is to use her maximal power to emit, which can be read only from the full strategy in Fig. 7.2c. This prevents the former player from achieving a payoff better than 23.



(a) Shape of solution 1



(b) Shape of solution 2



(c) Strategies for solution 1

Figure 7.2: Solutions for the power control game

7.3.2 Medium Access Control

The second example is based on the problem of medium access control. We consider that if more than one player is trying to emit in a given slot then no frame is transmitted during that slot. We also assume that each player has a limited energy and can therefore only emit on the network a limited number of times. Each of them is then trying to maximize the number of successful attempts. In the experiments, we considered as a parameter their initial level of energy.

7.3.3 Shared File System

Our last example models a shared file system with locks. In this games, once a file is locked by a player it can no longer be accessed by the other until it is unlocked. The objective for each player is then given by a deterministic Büchi automaton to describe the order in which the files should be accessed by the player. We experimented with different numbers of players and files. We also tried Büchi automata of different sizes; the number of states and edges indicated in Table 7.1 are those of the product.

7.4 Experiments

In order to show the influence of the size of the graph on the time taken to compute Nash equilibria, we ran our tool on several sets of examples. The experimental results are given in Table 7.1.

We observe from these experiments that our prototype works well for games up to one hundred states. The execution time then quickly increases. This is because the procedure as described in Section 4.3, requires computation of the winning regions in a number of subgames that might be quadratic in the number of states of the game. The computation of the winning region in itself is done in time quadratic with respect to the size of the suspect game. For future implementations, we hope to improve that part of the computation by using symbolic methods, instead of enumerative methods.

Table 7.1: Experiments

Power Control					
Players	Emission Levels	States	Edges	Solutions	Time (sec.)
2	2	9	25	2	0.01
2	5	36	121	19	0.54
3	2	27	125	5	0.43
3	5	216	1331	83	162.75
4	2	81	625	6	18.56
5	2	243	3125	17	941.73
Medium Access Control					
Players	Initial Energy	States	Edges	Solutions	Time (sec.)
2	2	14	35	1	0.01
2	4	55	165	1	0.37
3	2	99	339	1	1.72
3	4	1359	6295	1	1209.85
4	2	756	3661	1	335.39
Shared File System					
Players	Number of files	States	Edges	Solutions	Time (sec.)
2	2	9	47	1	0.01
2	2	33	175	1	0.01
2	2	121	652	1	0.07
3	2	16	132	1	0.03
3	2	196	1759	1	0.77
4	2	25	333	1	0.70
4	3	125	3656	1	27.01

Chapter 8

Conclusion

8.1 Summary

In this work, we reduced the computation of Nash equilibria in concurrent multiplayer semi-quantitative games to the computation of winning strategies in turn-based qualitative two-player games. This transformation is based on the notion of suspect. We showed that the size of the resulting game is polynomial.

This suspect game is a powerful tool. We used it to describe the precise complexity classes of the different problems in many cases. The algorithms are often simple thanks to this transformation. For instance the NP algorithm for reachability objective is obtained simply by guessing a path in the winning region of the suspect game. The computation of the winning region itself is done by an attractor computation, since the suspect game turns out to be a simple safety game. The complexity for internal objectives in general, lies between PTIME and PSPACE, the "simplest" being Büchi objectives and the most difficult are objectives defined by Boolean circuits, which can encode Muller winning conditions. For objectives described by automata, we only have an exponential-time algorithm.

We extended the approach to a more quantitative context by allowing several reachability or Büchi objectives for each player. We analyzed the complexity with respect to the order that was chosen. In the general case where the order is given by a Boolean circuit we showed PSPACE-completeness for both Büchi and reachability objectives. An interesting restriction is the case of monotonic orders over Büchi objectives for which we showed NP-completeness.

Finally, we applied our results to timed games, through a refinement of the region abstraction, that allowed to reduce these games to finite concurrent games. The region game is exponential, and we showed that all the decisions problem are EXPTIME-complete, for objectives given by Büchi conditions and objectives given by Rabin automata.

8.2 Perspectives

The perspectives of this work are multiple. From the point of view of preferences of the players, it is natural next to consider a more quantitative setting, such as mean-payoff and discounted-games. For these games, we do not know yet whether the constrained existence problem is decidable. For instance for mean-payoff, in the suspect game, the objective of $\texttt{Eve}$ is a multidimensional mean-payoff game. Chatterjee, Doyen, Henzinger, and Raskin studied this multidimensional games and gave an algorithm for the value problem [12]. However this is not sufficient in our framework, since there is an infinite number of possible values, it makes it impossible to guess the payoff of the Nash equilibrium. We would first need a way to find all the values of the game. A restriction that would elude this problem is that of action-visible. In that case, for any deviation there is only one suspect and therefore the game is a simple mean-payoff game. Ummels and Wojtczak already showed that for these games, the constraint existence problem is $\texttt{NP}$-complete [52].

Concerning the solution concepts, notions other than Nash equilibria have been proposed in game theory to represent rational behaviors. In particular, subgame perfect equilibria are relevant for repeated games. In a subgame perfect equilibrium, we require that the strategy profile is a Nash equilibrium starting from any history. Ummels showed results similar to Nash equilibria for turn-based and stochastic games [50]. We expect a refinement of the suspect game to be useful in that case. Moreover, other solution concepts may have to be defined in order to answer problems specific to computer science.

Among other extensions of Nash equilibria are resilient and immune equilibria. In a k-resilient equilibrium, a coalition of k-players can change its strategy, and we have to ensure that none of the players can improve her outcome. In a t-immune equilibria, we allow t players to be irrational and the payoff of the others should not be harmed by deviation of the irrational players. We believe that a transformation similar to the suspect game could help solve these problems. However the size of that construction might no more be polynomial.

Concerning our model of games, it is not well suited in some situations where players do not have access to the same information. However, in general adding imperfect information makes the problems we are studying undecidable. This is due to the problem of *information fork*. There is still hope that some interesting restrictions make the problem decidable. In our context for instance, players do not see each others actions, allowing to model a bit of imperfect information while preserving decidability.

Bibliography

[1] R. Alur, C. Courcoubetis, and D. Dill. Model-checking in dense real-time. *Information and computation*, 104(1):2–34, 1993.

[2] R. Alur and D. L. Dill. A theory of timed automata. *TCS*, 126(2):183–235, 1994.

[3] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, September 2002.

[4] G. Behrmann, A. Cougnard, A. David, E. Fleury, K. G. Larsen, and D. Lime. UPPAAL-Tiga: Time for playing games! In *CAV'07*, volume 4590 of *LNCS*, pages 121–125. Springer, 2007.

[5] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, and W. Yi. Uppaal—a tool suite for automatic verification of real-time systems. *Hybrid Systems III*, pages 232–243, 1996.

[6] P. Bouyer, R. Brenguier, and N. Markey. Nash equilibria for reachability objectives in multi-player timed games. 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 192–206. Springer-Verlag, September 2010.

[7] P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Nash equilibria in concurrent games with Büchi objectives. 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 375–386, Mumbai, India, December 2011. Leibniz-Zentrum für Informatik.

[8] P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Pure Nash equilibria in concurrent games – part II: Ordered objectives, 2012. In preparation.

[9] P. Bouyer, D. D'Souza, P. Madhusudan, and A. Petit. Timed control with partial observability. In Warren A. Hunt, Jr and Fabio Somenzi, editors, *Proceedings of the 15th International Conference on Computer Aided Verification (CAV'03)*, volume 2725 of *Lecture Notes in Computer Science*, pages 180–192, Boulder, Colorado, USA, July 2003. Springer.

[10] T. Brihaye, V. Bruyère, and J. De Pril. Equilibria in quantitative reachability games. In Farid Ablayev and Ernst Mayr, editors, *Computer Science – Theory and Applications*, volume 6072 of *Lecture Notes in Computer Science*, pages 72–83. Springer Berlin / Heidelberg, 2010.

[11] J.R. Burch, E.M. Clarke, K.L. McMillan, and D.L. Dill. Sequential circuit verification using symbolic model checking. In *Design Automation Conference, 1990. Proceedings., 27th ACM/IEEE*, pages 46–51. IEEE, 1990.

[12] K. Chatterjee, L. Doyen, T.A. Henzinger, and J.F. Raskin. Generalized mean-payoff and energy games. *Arxiv preprint arXiv:1007.1669*, 2010.

[13] K. Chatterjee, T. A. Henzinger, and N. Piterman. Generalized parity games. In *Proceedings of the 10th International Conference on Foundations of Software Science and Computation Structure (FoSSaCS'07)*, volume 4423 of *Lecture Notes in Computer Science*, pages 153–167. Springer-Verlag, 2007.

[14] K. Chatterjee, R. Majumdar, and M. Jurdziński. On Nash equilibria in stochastic games. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, *Proceedings of the 18th International Workshop on Computer Science Logic (CSL'04)*, volume 3210 of *Lecture Notes in Computer Science*, pages 26–40. Springer-Verlag, September 2004.

[15] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri. Nusmv: A new symbolic model verifier. In *Computer Aided Verification*, pages 682–682. Springer, 1999.

[16] S. Conchon, J.C. Filliâtre, and J. Signoles. Designing a generic graph library using ml functors. *TFP*, 7:124–140, 2008.

[17] A.A. Cournot and I. Fisher. *Researches into the Mathematical Principles of the Theory of Wealth*. Macmillan, 1897.

[18] J. Cristau, C. David, and F. Horn. How do we remember the past in randomised strategies? *Electronic Proceedings in Theoretical Computer Science*, 25, 2010.

[19] C. Daskalakis, P.W. Goldberg, and C.H. Papadimitriou. The complexity of computing a nash equilibrium. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 71–78. ACM, 2006.

[20] A. Dawar, F. Horn, and P. Hunter. Complexity bounds for Muller games. *Theoretical Computer Science*, 2011. Submitted.

[21] L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In Roberto Amadio and Denis Lugiez, editors, *Proceedings of the 14th International Conference on Concurrency Theory (CONCUR'03)*, volume 2761 of *Lecture Notes in Computer Science*, pages 142–156. Springer-Verlag, August-September 2003.

[22] L. De Alfaro, T. Henzinger, and R. Majumdar. Symbolic algorithms for infinite-state games. *CONCUR 2001—Concurrency Theory*, pages 536–550, 2001.

[23] E. A. Emerson and C. S. Jutla. The complexity of tree automata and logics of programs. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science (FOCS'88)*, pages 328–337. IEEE Comp. Soc. Press, October 1988.

[24] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy. In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, pages 368–377. IEEE, 1991.

[25] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1(1):80 – 93, 1989.

[26] G. Gottlob. NP trees and Carnap's modal logic. *Journal of the ACM*, 42(2):421–457, March 1995.

[27] G. Gottlob, G. Greco, and F. Scarcello. Pure nash equilibria: hard and easy games. In *Proceedings of the 9th conference on Theoretical aspects of rationality and knowledge*, TARK '03, pages 215–230, New York, NY, USA, 2003. ACM.

[28] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*, volume 2500 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[29] T.A. Henzinger, P.H. Ho, and H. Wong-Toi. Hytech: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer (STTT)*, 1(1):110–122, 1997.

[30] T.A. Henzinger and P.W. Kopke. Discrete-time control for rectangular hybrid automata. *Theoretical Computer Science*, 221(1-2):369–392, 1999.

[31] G.J. Holzmann. Design and validation of protocols: a tutorial. *Computer Networks and ISDN Systems*, 25(9):981–1017, 1993.

[32] G.J. Holzmann. The model checker spin. *Software Engineering, IEEE Transactions on*, 23(5):279–295, 1997.

[33] F. Horn. Streett games on finite graphs. In Marcin Jurdziński and Rupak Majumdar, editors, *Proceedings of the 2nd Workshop on Games in Design and Verification (GDV'05)*, July 2005.

[34] M. Jurdziński. Deciding the winner in parity games is in UP$\cap$co-UP. *Information Processing Letters*, 68(3):119–124, November 1998.

[35] M. Jurdziński, F. Laroussinie, and J. Sproston. Model checking probabilistic timed automata with one or two clocks. In Orna Grumberg and Michael Huth, editors, *Proceedings of the 13th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'07)*, volume 4424 of *Lecture Notes in Computer Science*, pages 170–184. Springer, March 2007.

[36] M. Klimoš, K. Larsen, F. Štefaňák, and J. Thaarup. Nash equilibria in concurrent priced games. *Language and Automata Theory and Applications*, pages 363–376, 2012.

[37] D. Kozen. Lower bounds for natural proof systems. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS'77)*, pages 254–266. IEEE Comp. Soc. Press, October-November 1977.

[38] F. Laroussinie, N. Markey, and G. Oreiby. On the expressiveness and complexity of ATL. *Logicical Methods in Computer Science*, 4(2), May 2008.

[39] A.B. MacKenzie, S. Wicker. Game theory and the design of self-configuring, adaptive wireless networks. *Communications Magazine, IEEE*, 39(11):126–131, 2001.

[40] A.B. MacKenzie, S. Wicker. Game theory in communications: Motivation, explanation, and application to power control. In *Global Telecommunications Conference, 2001. IEEE*, volume 2, pages 821–826. IEEE, 2001.

[41] A.B. MacKenzie, S. Wicker. Stability of multipacket slotted aloha with selfish users and perfect information. *IEEE INFOCOM*, 3:1583–1590, 2003.

[42] O. Morgenstern and J. Von Neumann. Theory of games and economic behavior. 1953.

[43] A.W. Mostowski. *Games with forbidden positions.* UG, 1991.

[44] J. F. Nash, Jr. Equilibrium points in n-person games. *Proc. National Academy of Sciences of the USA*, 36(1):48–49, January 1950.

[45] PJ Ramadge and WM Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, 25(1):206–230, 1987.

[46] D. Rosenberg, E. Solan, and N. Vieille. Stochastic games with imperfect monitoring. *Advances in Dynamic Games*, pages 3–22, 2006.

[47] C.U. Saraydar, N.B. Mandayam, and D.J. Goodman. Pareto efficiency of pricing-based power control in wireless data networks. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 231–235. IEEE, 1999.

[48] J.M. Smith and G.R. Price. The logic of animal conflict. *Nature*, 246:15, 1973.

[49] M. Ummels. The complexity of Nash equilibria in infinite multiplayer games. In Roberto Amadio, editor, *Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structure (FoSSaCS'08)*, volume 4962 of *Lecture Notes in Computer Science*, pages 20–34. Springer-Verlag, March-April 2008.

[50] M. Ummels. *Stochastic multiplayer games: Theory and algorithms.* Pallas Publications, 2010.

[51] M. Ummels and D. Wojtczak. The complexity of nash equilibria in simple stochastic multiplayer games. *Automata, Languages and Programming*, pages 297–308, 2009.

[52] M. Ummels and D. Wojtczak. The complexity of nash equilibria in limit-average games. *CONCUR 2011–Concurrency Theory*, pages 482–496, 2011.

[53] J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

[54] N. Wallmeier, P. Hütten, and W. Thomas. Symbolic synthesis of finite-state controllers for request-response specifications. *Implementation and Application of Automata*, pages 113–127, 2003.

[55] S. Yovine. Kronos: A verification tool for real-time systems. *International Journal on Software Tools for Technology Transfer (STTT)*, 1(1):123–133, 1997.

Index