UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II"

Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione

Corso di Laurea in Informatica

Graded Modalities in Strategic Reasoning

Relatori:

Prof. Aniello Murano

Dott. Fabio Mogavero

Co-relatore:

Prof. Luigi Sauro

Candidato:

Vadim Malvone

N97/132

Anno Accademico 2013/14

Contents

Abstract (in Italian)

I metodi formali sono uno strumento ampiamente utilizzato per la specifica e la verifica di sistemi. Tra i metodi di verifica la tecnica di maggior successo è sicuramente il *model checking*. Concepito alla fine degli anni 70, il model checking permette di verificare la correttezza di un sistema rispetto ad una certa proprietà controllando che un modello del sistema soddisfi una rappresentazione formale della proprietà. Il model checking è stato dapprima introdotto per sistemi monolitici per i quali il modello è spesso rappresentato in termini di *Kripke structure* e la specifica attraverso una logica temporale. Le logiche temporali sono un particolare tipo di logiche modali che permettono di analizzare proprietà che variano nel tempo senza introdurre il tempo esplicitamente.

Negli ultimi anni, sulle logiche temporali sono state ampiamente studiate le *modalità graded*. Nelle logiche temporali classiche, tali modalità sono state ottenute sostituendo i quantificatori esistenziali $\mathtt{E}\psi$ e universali $\mathtt{A}\psi$ con quelli della forma $\mathtt{E}^{\geq g}\psi$ (*ci sono almeno g percorsi*) e $\mathtt{A}^{<g}\psi$ (*a meno di g percorsi*). Dato un modello monolitico $\mathcal{K}$ di un sistema, per valutare la verità di una formula graded, è necessario contare il numero di diverse computazioni che soddisfano ψ in $\mathcal{K}$. Nel caso più generale di sistemi multi-agente, le computazioni valide corrispondono ai play indotti dall'interazione degli agenti in base alle proprie strategie. In questo ambito, le modalità graded permettono di determinare quante strategie un agente può sfruttare per soddisfare una certa proprietà.

In questa tesi definiamo la nozione di *Graded Strategy Game* (GSG) come un'estensione del framework *Strategy Game* (SG) recentemente introdotto per descrivere giochi in cui il solution concept è definito come una formula di *Strategy Logic* (SL). Quest'ultima è una logica potente per il ragionamento strategico, nel quale le strategie sono esplicitamente trattate come oggetti del primo ordine. Formalmente, SG si compone di tre entità: un'arena, che riassume le regole del gioco, un'estensione che descrive le proprietà di interesse, e uno schema, che riporta i ruoli degli agenti e le loro interazioni. Un GSG si ottiene tramite un'estensione graded di SL (GSL) come logica per descrivere i solution concept. Questa logica utilizza gli operatori strategici esistenziale $\langle\langle x \geq g \rangle\rangle\psi$ e universale $[[x < g]]\psi$ per richiedere se *ci sono almeno g* o *tutte a meno di g* strategie

x soddisfano ψ. Inoltre per associare una strategia x a uno specifico agente a usiamo un operatore di binding (a, x).

L'introduzione formale di GSL richiede una notevole quantità di lavoro per la definizione di una relazione di equivalenza sugli assegnamenti strategici. La caratteristica fondamentale di questa relazione è quella di classificare come equivalenti tutti gli assegnamenti che riflettono lo stesso "*ragionamento strategico*" anche se possono avere strutture diverse. Per dare un'intuizione di cosa intendiamo, si considerino due assegnamenti e le corrispondenti strategie coinvolte. Si supponga che tali strategie si differenziano solo per storie mai incontrate da un play a causa di una combinazione specifica di azioni degli agenti. Risulta evidente che questi assegnamenti inducono gli stessi comportamenti, il che significa riflettere lo stesso ragionamento strategico, pertanto è naturale impostarli come equivalenti.

Oltre all'introduzione formale di GSG, in questa tesi abbiamo studiato le versioni graded corrispondenti ai frammenti one-goal e boolean-goal di SG. Questi frammenti sono stati denominati GSG[1G] e GSG[BG] e sono stato oggetto di studio in relazione a problemi classici sulla teoria dei giochi. Nello specifico, abbiamo analizzato la conversione da un gioco concorrente alla sua versione turn-based in GSG[1G] e la conversione da un gioco GSG[BG] a SG[BG]. Per la prima conversione è stato necessario costruire una nuova arena e, di conseguenza, una nuova estensione. Abbiamo diviso questo processo in tre fasi. La prima, detta normalizzazione, riduce il numero di agenti al numero di variabili. La seconda, detta di minimizzazione, riduce lo spazio delle azioni, eliminando quelle ridondanti. La terza esegue la costruzione di un gioco turn-based. Per la seconda conversione, l'arena non cambia mentre l'estensione è arricchita con nuove proprietà. La trasformazione principale riguarda il solution concept visto che da GSL occorre passare ad SL. Per fare questo sono state introdotte due funzioni, una di trasformazione sulla formula ed un'altra utilizzata per la relazione d'equivalenza tra assegnamenti.

Un aspetto fondamentale per i nostri risultati riguarda la determinatezza sui giochi GSG. In particolare abbiamo introdotto una prova per giochi con solution concept in GSL[1G]. Vale la pena ricordare che questa proprietà è stata dimostrata per il classico gioco Boreliano turn-based, ma la prova applicata in tale situazione non può essere applicata direttamente al nostro tipo di gioco. Per dare una prova delle sostanziali differenze tra i due framework, è utile osservare che in SG una formula del tipo $\langle\langle x\rangle\rangle[\![y]\!]\psi$ implica $[\![y]\!]\langle\langle x\rangle\rangle\psi$, mentre in GSG $\langle\langle x \geq g_1\rangle\rangle[\![y < g_2]\!]\psi$ non implica $[\![y<g_2]\!]\langle\langle x\geq g_1\rangle\rangle\psi$. La proprietà di determinatezza che dimostriamo è $\wp\psi \Rightarrow \wp'\psi$, dove il prefisso di quantificazione $\wp'$ è ottenuto da $\wp$, mettendo tutti i quantificatori esistenziali prima di quelli universali e preservando l'ordine relativo tra i quantificatori dello stesso tipo.

Da una variante delle prova di determinatezza abbiamo potuto definire il problema decisionale per i giochi studiati, detto *fulfillment*. Il fulfillment rappresenta una generalizzazione del model checking. Per chiarire meglio il concetto, in termini di proporzioni potremmo dire che il model checking sta ai sistemi chiusi come il fulfillment sta ai giochi che abbiamo definito. In particolare, abbiamo dimostratato come il problema del fulfilling per GSG[1G] su predicati di reachability e safety risulta essere PTIME-COMPLETE nella dimensione sia dell'arena che del solution concept.

1 Introduction

The problem of the correctness of systems is particularly felt in hardware and software design, especially in the context of critical systems. When we talk about a critical system, we mean one in which failure is not an option. Mainly, these systems are divided into: safety critical, where errors can cost lives; mission critical, where unreliability can cost in terms of objectives; business critical, where failure can cost in loss of money.

In recent decades, there have been several examples of critical systems with significant unexpected behavior having disastrous consequences. In the eighties, for example, several people were killed and seriously injured by a system malfunction of Therac-25, a radiation machine used in the medical field. Specifically, the malfunction has been produced by flaws in the software and in a wrong interface with the hardware machine. As an other example, in 1995 the new Denver airport designed to be at the forefront with its complex computerized system for baggage handling was forced to open 16 months after the date set due to flaws in the system. The malfunctioning of the software has result in a cost of 3.2 million of dollar. Another example is the shuttle Arianne 5, launched into space in 1996, has a flagship of the European Space Programme. After 36.7 seconds of flight, the system has diverted its route and exploded right after. The disaster was caused by a program of the navigation system that attempted to cast a long int into short int. The cost of the damage was approximately one million of Euros, not counting the loss of reputation.

Describing the above examples let us to realize that verifying the correctness of a system, prior to its use, is a vital aspect. The main methods for software verification are: testing, simulation, deductive verification, and formal verification. Testing is an operation carried out throughout the development phase in which the produced system is checked to find any faults, and if so, to resolve them before release. Although widely used, this technique presents many disadvantages due to its empirical structure. First of all, it only works on a real implementation of the system. This means that to correct a error in the system design we may need to implement the whole system from scratch. Thus, it would trigger a wasting of time and money. It requires to be tested on a large amount of data, that is all possible behaviors that the system can exhibit. Obviously, to retrieve all such data is not an easy task, as well as the set of all possible system behaviors is clearly to big to be feasibly tested. Infact, in some cases, this number is exponentially larger than the size of the system (consider for example real-time and concurrent systems). On the other hand, in the safety and business critical systems described above, a whole check is a "must", since it is not possible to miss out any behavior that could hide a dangerous and unknown bug. Simulation is similar to testing except that the verification process is performed on an abstract model of the system. This is an advantage as errors can be discovered before the system is implemented, saving a lot of money and time. Conversely, under this approach there is no guarantee that all possible executions are simulated. This means that both testing and simulation can detect errors but can not determine its absence. Deductive verification uses axioms and proof rules to verify the correctness of the system. This method is very expensive regarding time execution and therefore is mainly used for highly sensitive systems. To face with the disadvantages of all above methods, several methodologies have been proposed. Among others, *formal methods* result to be very useful. This approach provides a formal-based methodology to model systems, specify properties, and verify that a system satisfies a given specification. Basically, in order to check the correctness of a system with respect to a desired behavior, we consider a model M of the system, a formal specification φ of its behavior, and a formal technique to show that M meets φ. The latter is done by means of suitable decision procedures. Using formal methods, beside checking system incorrectness, one can also get a counterexample that certifies it. In particular, a counterexample will give information regarding which part of the design contains a bug.

Formal verification has several advantages over testing. First of all it works on a model of the system rather than using a real implementation. This means that in case of an error in the system design, it can be repaired immediately, reducing the high cost of correcting errors in a digital design after its production. Second, this method is exhaustive, as it allows to check the system with respect to all its possible behaviors, in any possible environment. This avoids the common problem in testing of overlooking some critical executions that could hide unexpected errors. Indeed, verification is an *automatic method*. Another advantage regarding formal methods relies on the possibility of using formal specification languages to express the desired properties of a system. This is fundamental to gain a non ambiguity characterization in what it is meant for an execution to be correct. Finally, system correctness is reduced to some well-founded decision procedures, which enforce a rigorous checking. In formal verification, the specification is usually based on modal logics and specifically on temporal logics. Modal logics study different "modes" of truth that determine a more involved value structure with respect to simple Boolean values. Temporal logics are a special type of modal logics, originally developed by philosophers to investigate the ways in which the time can be used in natural language speaking. In particular, temporal logics can describe the order of events without introducing the time explicitly. Therefore, it can formalize opportunely how the system evolves during the time, and this makes this logic very appropriate to the formal verification task. In temporal logics, we mainly distinguish between linear- and branching-time logics, which reflect the underlying nature of the time we consider. In *linear-time logics*, at each moment there is only one possible future moment, while in *branching-time logics*, time has a tree-like structure and, at each moment, time may split into several courses representing different possible futures. Accordingly, the semantic of linear logics is given with respect to *linear structures* while for branching logics we use *branching structures*. Temporal modalities of a temporal logic reflect the kind of time assumed in the semantics. Thus, in a linear-time logic, temporal modalities are provided for describing events along a single time line. In contrast, in a branching-time logic, the modalities reflect the branching nature of time by allowing existential and universal quantifiers over possible futures. In literature, many types of temporal logics have been considered, in both linear- and branching-time. Between them, the most popular ones are the linear-time temporal logic LTL [Pnu77], the branching-time temporal logic CTL [CE81], and their extension CTL* [EH86].

An outstanding development in the area of temporal logics has been the discovery of algorithmic methods to verify properties of finite-state systems represented by Kripke structure $\mathcal{K}$.

Kripke structures also represent the underlaying structure on which to define the semantics for temporal logics. Hence, the formal verification of a system modeled by $\mathcal{K}$ with respect a temporal logic specification φ can be rephrased as "Is $\mathcal{K}$ a model of φ?", which explains the name *model checking*, as it was coined by Clarke and Emerson in [CE81], used to denote this problem and the verification methods derived from this point of view. Another interesting question to answer in temporal logic is whether a given formula φ is *satisfiable* or not, that is, if there exists a Kripke structure that is a model of φ. Thus, this decision problem can be used to verify whether the specification of a system can be implemented.

Mainly, two main modalities are considered to perform model checking and satisfiability in practice. The first option is a classical use of ad-hoc algorithms. For example, the PSPACE-COMPLETE recursive algorithms have been carried out to solve the model checking and satisfiability problems for the linear-time logic LTL. Similarly, for CTL, it has been shown a linear algorithm for model checking and a EXPTIME-COMPLETE algorithm for the satisfiability questions. The second option involves instead a systematic use of the automata-theoretic approach on infinite objects. In particular, we operate a translation from a temporal logic formula φ to an automaton ensuring bijection between the models of φ and the (infinite) objects recognized by the constructed automaton. In this way, we reduce the satisfiability problem to the emptiness problem of the automaton. The model checking question, instead, reduces to the emptiness problem of the intersection between the automaton corresponding to the system and the one for the complement of the property.

In the multi-agent system design and verification, temporal logics have recently assumed a prominent role for the strategic reasoning [AHK02, JvdH04, CHP10, MMV10, Lor10, vE13]. Specifically, classical temporal logics have been extended in such a way that they can check open system reliability. This has been done by rephrasing the decision question as an interaction among different players in a game-strategic setting, which may compete or collaborate for the achievement of specific *goals*. One of the most important developments in this field is *Alternating-Time Temporal Logic* (ATL*, for short), introduced by Alur, Henzinger, and Kupferman [AHK02]. ATL* allows to reason about strategies of agents having the satisfaction of temporal goals as payoff criterion. Formally, it is obtained as a generalization of CTL*, in which the existential E and the universal A *path quantifiers* are replaced with *strategic modalities* of the form $\langle\langle \mathrm{A} \rangle\rangle$ and $[[\mathrm{A}]]$, where A is a set of *agents*. Despite its expressiveness, ATL* suffers from the strong limitation that strategies are treated only implicitly in the semantics of such modalities. This restriction makes the logic less suited to formalize several important solution concepts, such as the *Nash Equilibrium*. These considerations led to the introduction and study of *Strategy Logic* (SL, for short) [CHP07, MMV10], a more powerful formalism for strategic reasoning. As a key aspect, this logic treats strategies as *first-order objects* that can be determined by means of the existential $\langle\langle x \rangle\rangle$ and universal $[[x]]$ quantifiers, which can be respectively read as *"there exists a strategy x"* and *"for all strategies x"*. Remarkably, in SL [MMV10], a strategy is a generic conditional plan that at each step prescribes an action on the base of the history of the play. Therefore, this plan is not intrinsically glued to a specific agent, but an explicit binding operator (a, x) allows to link an agent a to the strategy associated with a variable x. Unfortunately, the high expressivity of SL comes at a price. Indeed, it has been shown that the model-checking problem for SL becomes non-elementary complete and the satisfiability undecidable. To gain back elementariness, several fragments of SL, strictly subsuming ATL* have been considered. Among the others, One-Goal Strategy Logic (SG[1G], for short) considers SL formulas in a special prenex normal form having a single temporal goal at a time. For a goal, it is specifically meant a sequence of bindings followed by a temporal logic formula. It has been shown that for SG[1G] both the model checking question and the satisfiability problem are 2-EXPTIME-COMPLETE, as it is for ATL. If one allows for a Boolean combination of goals, then the resulting logic is named Boolean goal Strategy Logic (SG[BG]), for which the satisfiability problem has been proved to be undecidable, while the exact complexity of the model checking problem is an open question.

A common aspect about all logics mentioned above is that quantifications are made either existential or universal or combinations of both. *Per contra*, there are several real scenarios in which "more precise" quantifications are crucially needed. For example, in a planning scenario it maybe useful know that more than one possible paths exists to reach a specific target. As another example, in formal verification setting knowing how many path fail to satisfy a path formula can accelerate the process of correcting them. Finally, in the security scenario assuming that one can control up to n computations, one may want to know whether all but n computations are safe. See also [BMM12], for other argumentations. This has attracted the interest of the formal verification community to *graded modalities*. They have been first studied in classic modal logic [Fin72] and then imported in the field of *knowledge representation* to allow quantitative bounds on the set of individuals satisfying a certain property. In particular, they are considered as *counting quantifiers* in first-order logics [GOR97] and *number restrictions* in *description logics* [HB91].

The first applications of graded modalities in formal verification concern closed systems. Specifically, in [KSV02], *graded* μCALCULUS has been introduced in order to express statements about a given number of immediately accessible worlds. Successively [FNP09, BMM09, BMM10, BMM12], the notion of graded modalities have been extended in order to handle a number of paths instead of successive worlds. In particular, in [BMM12] graded CTL (GCTL, for short), an extension of CTL with graded path modalities, has been introduced along with a suitable axiomatization of a counting.

In open systems verification, we are aware of just two cases in which graded modalities have been investigated: the module checking with respect to graded μCALCULUS [FMP08] and the extension of ATL with graded path modalities (GATL, for short) [FNP10]. These two orthogonal approaches have the merit of introducing, for the first time, a counting on strategies. In particular, while the former involves a counting on one-step moves among two-agents, the latter allows for a more sophisticated counting on the histories of the game in a multi-player setting. Nevertheless, GATL suffer of several limitations. First, not surprisingly, it cannot express powerful game reasonings due to the limited power of the underline logic ATL. Second, it is based on a very rigid and restricted counting of existential strategies only.

In this thesis, we overcome the above limitations by introducing and studying *Graded Strategy Game* (GSG) as an extension, along with graded quantifiers, of the recently introduced framework of *Strategy Game* [MMS14]. Formally, a Strategy Game consists of three entities: the arena, shaping the rules of the game, the extension describing the property of interest of each possible play, and the schema, representing the agent roles and their interaction. This formalism turns out to be a powerful machinery to describe games whose solution concepts are given via a *Strategy Logic* sentence. Specifically, a given property is verified (formally *fulfilled*) by checking whether an extension satisfies it. Therefore, in Strategy game one can express via Strategy logic whether an extension admits a strategy or all strategies are capable to fulfill a given solution concept. Additionally, by means of the existential $\langle\langle x \geq g\rangle\rangle\varphi$ and universal $[[x < g]]\varphi$ graded strategy quantifiers, one can require that *there are at least* g or *all but less than* g strategies x satisfying the property φ. Finally, by using these new modalities in SL, we get a graded extension of this logic, which we call *Graded Strategy Logic* (GSL, for short) and the generalized mentioned framework of GSG derives accordingly. Also, as it has been done for SL, we define and study in this thesis the following fragments of GSL: GSL[BG] and GSL[1G], and the corresponding fragments in GSG: GSG[BG] and GSG[1G].

The formal introduction of GSL comprises a considerable amount of work for the definition of a suitable equivalence relation over strategy assignments. The key feature of this relation is to classify as equivalent all assignments that reflect the same "*strategic reasoning*", although they may have completely different structures. Just to get an intuition about what we mean, consider for example two assignments and the corresponding involved strategies. Assume now that all such strategies differ only on histories never met by a play because of a specific combination of agent actions. Clearly, these assignments induce the same agent behaviors, which means to reflect the same strategic reasoning. Therefore, it is natural to set them as equivalent, as we do.

Apart from the formal introduction of GSG, classical game-theoretic questions have been also investigated. Among the others, we study the conversion from concurrent game one-goal to its turn-based version and, only for Boolean fragment, from GSG to SG. The first conversion requires to build a new arena and, consequently, a new extension. We divide the conversion in a three-step process. The first phase, called normalization, reduces the number of agents to the number of variables. The second phase, called minimization, reduces the space of the actions by merging the actions that involve the same strategic reasoning. The third and final step performs the construction of turn-based game. For the second conversion, the arena remains the same, but the extension is enriched with new predicates. Regarding the solution concept the transformation makes use of a specifics two functions. The first on takes can of a semantic transformation. The other one is used to enforce the equivalence among assignments holding for the input target. It is important to observe that starting with a GSG[1G], the algorithm return in general a SG[BG].

Furthermore, we deal with the determinacy of GSG[1G]. It is worth recalling that this property has been first proved for classic Borelian turn-based two-player games in [Mar75], but the proof considered for that setting does not directly apply to our graded games. To give an evidence of the substantial differences between the two frameworks, it is useful to observe that in SG $\langle\langle x\rangle\rangle[[y]]\eta$ implies $[[y]]\langle\langle x\rangle\rangle\eta$, while in GSG the corresponding implication $\langle\langle x \geq g_1\rangle\rangle[[y < g_2]]\eta \Rightarrow [[y < g_2]]\langle\langle x \geq g_1\rangle\rangle\eta$ does not hold. The determinacy property we are interested in is exactly the converse direction. More formally, we prove that $\wp\eta \Rightarrow \wp'\eta$, where the quantification prefix $\wp'$ is obtained from $\wp$, by putting forward all existential quantifiers *w.r.t.* the universal ones and preserving the relative order between those of the same kind.

A slight variant of procedure for the determinacy proof may work as a tool to determine the degrees with which a given GSL[1G] is satisfied. We finally describe the solution of the fulfilling problem for GSG[1G] with reachability and safety predicates that is PTIME-COMPLETE in the size of both the arena and the solution concept.

Outline In Chapter 2, we recall graded modalities for closed and open systems. Then, we have Chapter 3, in which we describe a game-theoretic framework in [MMS14] and we introduce GSL and define its syntax and semantics, followed by Chapter 4 in which there are studied the main properties of strategy equivalence relations. In Chapter 5, we describe the two game type conversions. Finally, in Chapter 6 we construct the grading function for determinacy and the solution of the fulfilling problem.

2

Graded Modalities in Formal Verification

In formal verification one can guarantee system accomplishes a desired behavior by checking whether a mathematical model of the former conforms with a formal property specifying the latter. One of the most common framework to specify properties is temporal logic. Specifically, in the branching setting, temporal logics allow to enforce that *there exist* or that *for all* system executions a certain property holds. These logics have been recently extended under graded modalities allowing to refine the number of system executions on which to predicate. In the process of modeling the system, one needs to focus on the main aspects of its behavior. Mainly, we distinguish between *closed* and *open* systems [HP85]. While the behavior of a closed system is completely determined by the state of the system, the behavior of an open system depends on the ongoing interaction with its environment [Hoa85]. Unfortunately, the methods used for closed systems are often useless for open systems. For this reason several ad-hoc methods have been introduced to handle open systems. The same has happened in the case of graded logics.

In Section 2.1 we analyze the graded modalities for closed systems, in particular we describe the extension applied to the logic CTL. In Section 2.2 we analyze the concept of graded applied for μCALCULUS and ATL in the open systems. As far as we know these are the only two cases in which graded modalities have been applied to open system verification.

Finally, we report that most of the material reported in this introductory chapter comes from the works [BMM09, BMM10, BMM12, FMP08, FNP10] and we refer to them for more details, examples and motivations.

2.1 Graded in Closed Systems

In this section we talk about the use of graded modalities for the verification of closed systems. Specifically we will discuss GCTL, an extension of CTL with graded mode, which has been studied for both the model checking problem that for satisfiability. In this section we recall the basic concepts and results, organized as follows. In Subsection 2.1.1 we describe the Kripke structures and other basic concepts. In Subsection 2.1.2 we describe the syntax and semantics of logic GCTL. Finally, in Subsection 2.1.3 we describe the results for decision problems.

2.1.1 Preliminaries

We introduce some preliminary definitions and further notation used in this section.

Kripke structures. A *Kripke structure* (*KS*, for short) is a tuple $\mathcal{K} \triangleq \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_{0} \rangle$, where AP is a finite non-empty set of *atomic propositions*, W is an enumerable non-empty set of *worlds*, $w_0 \in \mathrm{W}$ is a designated *initial world*, $R \subseteq \mathrm{W} \times \mathrm{W}$ is a *transition relation*, and $\mathsf{L} : \mathrm{W} \to 2^{\mathrm{AP}}$ is a *labeling function* that maps each world to the set of atomic propositions true in that world. A KS is said *total* iff it has a *total* transition relation R, i.e., for all $w \in \mathrm{W}$, there is $w' \in \mathrm{W}$ such that $(w, w') \in R$. By $\|\mathcal{K}\| \triangleq |R| \leq |\mathrm{W}|^2$ we denote the *size* of $\mathcal{K}$, which also corresponds to the size of the transition relation. A *finite* KS is a structure of finite size.

Tracks and paths. A *track* in $\mathcal{K}$ is a finite sequence of worlds $\rho \in \mathrm{W}^*$ such that, for all $0 < i \leq |\rho|$, it holds that $((\rho)_i, (\rho)_{i+1}) \in R$. Furthermore, a *path* in $\mathcal{K}$ is a finite or infinite sequence of worlds $\pi \in \mathrm{W}^\infty$ such that, for all $0 < i \leq |\pi|$, it holds that $((\pi)_i, (\pi)_{i+1}) \in R$ and if $|\pi| < \infty$ then there is no world $w \in \mathrm{W}$ such that $(\mathsf{lst}(\pi), w) \in R$ (where $\mathsf{lst}(\pi)$ is last world of the path π), i.e., it is *maximal*. Intuitively, tracks and paths of a KS $\mathcal{K}$ are legal sequences of reachable worlds in $\mathcal{K}$ that can be seen as a partial or complete description of the possible *computations* of the system modeled by $\mathcal{K}$. A track ρ is said *non-trivial* iff $|\rho| > 0$, i.e., $\rho \neq \epsilon$. We use $\mathrm{Trk}(\mathcal{K}) \subseteq \mathrm{W}^{+}$ and $\mathrm{Pth}(\mathcal{K}) \subseteq \mathrm{W}^{\infty}$ to indicate, respectively, the sets of all non-trivial tracks and paths of the KS $\mathcal{K}$. Moreover, by $\mathrm{Trk}(\mathcal{K}, w) \subseteq \mathrm{Trk}(\mathcal{K})$ and $\mathrm{Pth}(\mathcal{K}, w) \subseteq \mathrm{Pth}(\mathcal{K})$ we denote the subsets of tracks and paths starting at the world w.

Kripke trees. A *Kripke tree* (KT, for short) is a KS $\mathcal{T} = \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_{\mathrm{o}} \rangle$, where *(i)* $\mathrm{W} \subseteq \mathrm{Dir}^{*}$ is a Dir-tree for a given set Dir of directions and *(ii)*, for all $t \in \mathrm{W}$ and $d \in \mathrm{Dir}$, it holds that $t \cdot d \in \mathrm{W}$ iff $(t, t \cdot d) \in R$.

Bisimulation. Let $\mathcal{K}_1 = \langle \mathrm{AP}, \mathrm{W}_1, R_1, \mathsf{L}_1, w_{0_1} \rangle$ and $\mathcal{K}_2 = \langle \mathrm{AP}, \mathrm{W}_2, R_2, \mathsf{L}_2, w_{0_2} \rangle$ be two KSs. Then, $\mathcal{K}_1$ and $\mathcal{K}_2$ are *bisimilar* iff there is a relation $\sim \subseteq \mathrm{W}_1 \times \mathrm{W}_2$ between worlds, called *bisimulation relation*, such that $w_{0_1} \sim w_{0_2}$ and if $w_1 \sim w_2$ then *(i)* $\mathsf{L}_1(w_1) = \mathsf{L}_2(w_2)$, *(ii)* for all $v_1 \in \mathrm{W}_1$ such that $(w_1, v_1) \in R_1$, there is $v_2 \in \mathrm{W}_2$ such that $(w_2, v_2) \in R_2$ and $v_1 \sim v_2$, and *(iii)* for all $v_2 \in \mathrm{W}_2$ such that $(w_2, v_2) \in R_2$, there is $v_1 \in \mathrm{W}_1$ such that $(w_1, v_1) \in R_1$ and $v_1 \sim v_2$.

Unwinding. Let $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_{\mathrm{o}} \rangle$ be a KS. Then, the *unwinding* of $\mathcal{K}$ is the KT $\mathcal{K}_U \triangleq \langle \mathrm{AP}, \mathrm{W}', R', \mathsf{L}', \epsilon \rangle$, where *(i)* W is the set of directions, *(ii)* the states in $\mathrm{W}' \triangleq \{\rho \in \mathrm{W}^{*} : w_0 \cdot \rho \in \mathrm{Trk}(\mathcal{K})\}$ are the suffixes of the tracks starting in w_0, *(iii)* $(\rho, \rho \cdot w) \in R'$ iff $(\mathsf{lst}(w_0 \cdot \rho), w) \in R$.

Note that, there is a surjective function $\mathsf{unw} : \mathrm{W}' \to \mathrm{W}$, called *unwinding function*, such that *(i)* $\mathsf{unw}(\rho) \triangleq \mathsf{lst}(w_0 \cdot \rho)$ and *(ii)* $\mathsf{L}'(\rho) \triangleq \mathsf{L}(\mathsf{unw}(\rho))$, for all $\rho \in \mathrm{W}'$ and $w \in \mathrm{W}$. It is easy to note that a KS is always bisimilar to its unwinding, since the unwinding function is a particular relation of bisimulation.

2.1.2 Graded Computation Tree Logic

Temporal logics are a special kind of *modal logics* that provide a formal framework for qualitatively describing and reasoning about how the truth values of assertions change over time. First pointed out by Pnueli in 1977 [Pnu77], these logics turn out to be particularly suitable for reasoning about correctness of concurrent programs [Pnu81].

Depending on the view of the underlying nature of time, two types of temporal logics are mainly considered [Lam80]. In *linear-time temporal logics*, such as LTL [Pnu77], time is treated as if each moment in time has a unique possible future. Conversely, in *branching-time temporal logics*, such as CTL [CE81] and CTL* [EH86], each moment in time may split into various possible futures and *existential* and *universal quantifiers* are used to express properties along one or all the possible futures. In modal logics, such as μCALCULUS [Koz83], these kinds of quantifiers have been generalized by means of *graded (worlds) modalities* [Fin72, Tob01], which allow to express properties such as "there exist at least n accessible worlds satisfying a certain formula" or "all but n accessible worlds satisfy a certain formula". Despite its high expressive power, the μCALCULUS is considered some what low-level logic and "unfriendly" logic for users, whereas simpler logics, such as CTL, can naturally express useful properties of computation trees. Therefore, an interesting and natural question is how graded modalities can affect CTL expressiveness and decidability. Moreover in the μCALCULUS, and standard modal logics, existential and universal quantifiers range over successors, therefore it is easy extend them with number restrictions. On the other side in CTL the underlying objects are both states and paths. Thus, graded modalities have to cover both of them. In [BMM12] solve this problem by generalizing the usual graded modalities, which act locally to words and their successor, to path of worlds, i.e., they allow to express properties such as "there are at least n classes of paths satisfying a formula". We call the logic CTL extended with graded path modalities GCTL, for short.

The *graded full computation tree logic* (GCTL*, for short) extends CTL* by using two special path quantifiers, the existential $\mathtt{E}^{\geq g}$ and the universal $\mathtt{A}^{<g}$, where $g \in \mathbb{N}$ denotes the corresponding *degree*. As in CTL*, these quantifiers can prefix a linear-time formula composed of an arbitrary Boolean combination and nesting of the temporal operators $\mathtt{X}$*"next"*, $\mathtt{U}$*"until"*, and $\mathtt{R}$*"release"*. The quantifiers $\mathtt{E}^{\geq g}$ and $\mathtt{A}^{<g}$ can be respectively read as *"there exist at least g paths"* and *"all but g paths"*. The formal syntax of GCTL* follows.

Definition 2.1.1 (GCTL* Syntax)**.** GCTL* state *(φ) and* path *(ψ)* formulas *are defined inductively as follows:*

1. $\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathtt{E}^{\geq g}\psi \mid \mathtt{A}^{<g}\psi$;

2. $\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \mathtt{X}\psi \mid \psi\, \mathtt{U}\psi \mid \psi\, \mathtt{R}\psi$.

Where $p \in \mathrm{AP}$ *and* $g \in \mathbb{N}$. GCTL *formulas are obtained by forcing each temporal operator* *occurring into a formula to be coupled with a path quantifier, as in the classical case of* CTL.

We now define the semantics of GCTL* w.r.t. a KS $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_0 \rangle$. For a world $w \in \mathrm{W}$, we write $\mathcal{K}, w \models \varphi$ to indicate that a state formula φ holds on $\mathcal{K}$ at w. Moreover, for a path $\pi \in \mathrm{Pth}(\mathcal{K})$, we write $\mathcal{K}, \pi \models \psi$ to indicate that a path formula ψ holds on π. The semantics of GCTL* state formulas simply extends that of CTL*. In particular, in the definition of graded quantifiers, we make use of a generic equivalence relation $\equiv_{\mathcal{K}}^{\psi}$ on the set of paths $\mathrm{Pth}(\mathcal{K})$ that depends on both the KS $\mathcal{K}$ and the path formula ψ. This equivalence is used to reasonably count the number of ways a structure has to satisfy a path formula w.r.t. an a priori criterion. The semantics of the GCTL* path formulas is defined as usual for LTL and is omitted here.

Definition 2.1.2 (GCTL* Semantics)**.** *Given a KS $\mathcal{K} = \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_0 \rangle$, for all* GCTL* *state formulas φ and worlds $w \in \mathrm{W}$, the relation $\mathcal{K}, w \models \varphi$ is inductively defined as follows.*

1. $\mathcal{K}, w \models p$ *iff* $p \in \mathsf{L}(w)$, *with* $p \in \mathrm{AP}$.
2. *For all state formulas φ, φ_1, and φ_2, it holds that:*
   (a) $\mathcal{K}, w \models \neg\varphi$ *iff not* $\mathcal{K}, w \models \varphi$, *that is* $\mathcal{K}, w \not\models \varphi$;
   (b) $\mathcal{K}, w \models \varphi_1 \wedge \varphi_2$ *iff* $\mathcal{K}, w \models \varphi_1$ *and* $\mathcal{K}, w \models \varphi_2$;
   (c) $\mathcal{K}, w \models \varphi_1 \vee \varphi_2$ *iff* $\mathcal{K}, w \models \varphi_1$ *or* $\mathcal{K}, w \models \varphi_2$.
3. *For a number $g \in \mathbb{N}$ and a path formula ψ, it holds that:*
   (a) $\mathcal{K}, w \models \mathtt{E}^{\geq g}\psi$ *iff* $|(\mathrm{Pth}(\mathcal{K}, w, \psi)/\equiv_{\mathcal{K}}^{\psi})| \geq g$;
   (b) $\mathcal{K}, w \models \mathtt{A}^{< g}\psi$ *iff* $|(\mathrm{Pth}(\mathcal{K}, w, \neg\psi)/\equiv_{\mathcal{K}}^{\neg\psi})| < g$;

   where $\mathrm{Pth}(\mathcal{K}, w, \psi) \triangleq \{\pi \in \mathrm{Pth}(\mathcal{K}, w) : \mathcal{K}, \pi \models \psi\}$ *is the set of paths of $\mathcal{K}$ starting in w that satisfy the path formula ψ and* $(\mathrm{Pth}(\mathcal{K}, w, \psi)/\equiv_{\mathcal{K}}^{\psi})$ *denotes the* quotient *set of* $\mathrm{Pth}(\mathcal{K}, w, \psi)$ *w.r.t. the equivalence relation* $\equiv_{\mathcal{K}}^{\psi}$, *i.e., the set of all the equivalence classes.*

2.1.3 Decision Problems on GCTL

The introduced framework of graded path modalities turns out to be very efficient in terms of expressiveness and complexity. Indeed, it has been proved that GCTL is more expressive than

CTL, it retains the tree and the finite model properties, and its satisfiability problem is solvable in EXPTIME, therefore it is not harder than that for CTL [EH85]. This, along with the fact that GCTL is at least exponentially more succinct than graded μCALCULUS (formally defined in the next section), makes GCTL even more appealing. The upper bound for the satisfiability complexity result is obtained by exploiting an automata-theoretic approach [KVW00]. To develop a decision procedure for a logic with the tree model property, one first develops an appropriate notion of tree automata and studies their emptiness problem. Then, the satisfiability problem for the logic is reduced to the emptiness problem of the automata. To this aim, has been introduced a new automaton model: *partitioning alternating tree automata* (PATA). While a nondeterministic automaton on visiting a node of the input tree sends exactly one copy of itself to each successor of the node, an alternating automaton can send several copies of itself to the same successor. In particular, in *symmetric alternating automata* [JW95, Wil99] it is not necessary to specify the direction of the tree on which a copy is sent. In [KSV02], *graded alternating tree automata* (GATA) has been introduced as a generalization of symmetric alternating tree automata, in such a way that the automaton can send copies of itself to a given number n of state successors, either in existential or universal way, without specifying which successors these exactly are. PATA further extend GATA in such a way that the automaton can send copies of itself to a given number n of paths. It has been shown, for each GCTL formula φ, it is always possible to build in linear time a PATA $\mathcal{A}_\varphi$ along with a Büchi condition (PABT) accepting all the tree models of φ. In [BMM09], addresses the specific case of GCTL where numbers are coded in unary. In particular, it has first shown that unary GCTL has the tree model property by showing that any formula φ is satisfiable on a Kripke structure iff it has a tree model whose branching degree is polynomial in the size of φ. Then, a corresponding tree automaton model named *partitioning alternating Büchi tree automata* (PABT) has been introduced and shown that:

Theorem 2.1.1. *Given a* GCTL *formula φ with degree b, we can construct in time* $\mathrm{O}(|\varphi|)$ *a* PABT $\mathcal{A}_\varphi$*, with* $\mathrm{O}(|\varphi|)$ *states and counting branching bound b, such that* $\mathcal{L}(\mathcal{A}_\varphi)$ *is exactly the set of all tree models of φ.*

Then, by using a nontrivial extension of the Miyano and Hayashi technique [MH84] it has been shown that:

Theorem 2.1.2. *Let $\mathcal{A}$ be a* PABT *with n states and counting branching bound b. Then, there exists a NBT $\mathcal{A}'$ with $2^{2n*(b+1)}$ states and $n * b(b+1)/2$ directions such that $\mathcal{A}$ is not empty iff $\mathcal{A}'$ is not as well.*

Now, recall that for the NBT $\mathcal{A}'$ with Q' as state set and branching degree d' the emptiness problem is solvable in PTIME [VW86] and, precisely, in $\mathrm{O}(|Q'|^{2d'})$ (we directly consider the one-letter automaton associated to $\mathcal{A}'$). Then, by Theorem 2.1.2, the following result follows.

Theorem 2.1.3. *The emptiness problem for a* PABT *$\mathcal{A}$ with n states and counting branching bound b can be decided in time $2^{\mathrm{O}(n^2*b^3)}$.*

By Theorems 2.1.1 and 2.1.3, and by $n = |\mathsf{ecl}(\varphi)| = \mathrm{O}(|\varphi|)$ and $b = \mathsf{deg}(\varphi) = \mathrm{O}(|\varphi|)$, we get that the satisfiability problem for GCTL is in EXPTIME and precisely solvable in time $2^{\mathrm{O}(|\varphi|^5)}$. Since GCTL subsumes CTL, the following holds.

Theorem 2.1.4. *The satisfiability problem for* GCTL *is* EXPTIME-COMPLETE.

The satisfiability was also analyzed for case of GCTL where numbers are coded in binary. For simplicity reasoning, we have reported only the unary case.

2.2 Graded in Open Systems

In open systems verification, we are aware of just two cases in which graded modalities have been investigated. Chronologically, they regard module checking with respect to graded μ-calculs [FM07, FMP08] specifications and the extension of ATL with graded path modalities (GATL, for short) [FNP10]. These two orthogonal approaches have the merit of introducing, for the first time, a counting on strategies. In particular, while the former involves a counting on one-step moves among two-agents, the latter allows for a more sophisticated counting on the histories of the game in a multi-player setting. In this Section we report both studies giving some details about their representation and their solution. This section is organized as follows. In subsection 2.2.1 we will discuss the graded module checking by presenting the syntax and semantics of graded μCALCULUS and its resolution. While, in Section 2.2.2 we will discuss graded ATL and its syntax, semantics, decision problems, and their solutions.

2.2.1 Graded in Module Checking

In model checking open systems, introduced and called *module-checking* in [KVW01], one should check the system with respect to arbitrary environments and should take into account uncertainty regarding the environment. In such a framework, the open finite–state system is described by a labeled state–transition graph, called in fact *module*, whose set of states is partitioned into *system states* (where the system makes a transition) and *environment states* (where the environment makes a transition). Given a module $\mathcal{M}$, describing the system to be verified, and a temporal logic formula φ, specifying the desired behavior of the system, module checking asks whether for all possible environments, $\mathcal{M}$ satisfies φ. Therefore, in module checking it is not sufficient to check whether the full computation tree obtained by unwinding $\mathcal{M}$ satisfies φ, but it is also necessary to verify that all trees obtained from the full computation tree by pruning some subtrees rooted in nodes corresponding to choices disabled by the environment (those trees represent the interactions of $\mathcal{M}$ with all the possible environments), satisfy φ. We collect all such trees in a set named $exec(\mathcal{M})$. It is worth noticing that each tree in $exec(\mathcal{M})$ represents a "memoryful" behavior of the environment. Indeed, the unwinding of a module $\mathcal{M}$ induces duplication of nodes, which allow different pruning of subtrees. To see an example, consider a two-drink dispenser machine that serves, upon customer request, tea or coffee. The machine is an open system and an environment for the system is an infinite line of thirsty people. Since each person in the line can prefer both tea and coffee, or only tea, or only coffee, each person suggests a different disabling of the external choices. Accordingly, there are many different possible environments to consider. In [KV97, KVW01], it has been shown that while for linear–time logics model and module checking coincide, module checking for specification given in CTL and CTL* is exponentially harder than model checking in the size of the formula and preserves the linearity in the size of the model. Indeed, CTL and CTL* module checking is EXPTIME–complete and 2-EXPTIME–complete, respectively.

Among the various formalisms used for specifying properties, a valid candidate is the μ-*calculus*, a very powerful propositional modal logic augmented with least and greatest fixpoint operators [Koz83] (for a recent survey, see also [BS06]). The *Graded enriched* μ*–calculus* [BP04] is the extension of the μ–calculus with *graded modalities*. Intuitively, graded modalities enable statements about the number of successors of a state [KSV02]. It has been shown in [SV01, BLMV06] that satisfiability *Graded enriched μ–calculus* is decidable and EXPTIME-complete. The upper bound result is based on an automata–theoretic approach via *graded alternating parity tree automata (*GAPT*)*, along with the fact that this logic enjoys the *tree model property*. Intuitively, *GAPT* generalize alternating automata on infinite trees graded modalities enrich the standard μ–calculus. Using *GAPT* and the tree model property, it has been shown in [SV01, BLMV06] that given a formula φ of graded μ-calculus, it is possible to construct a *GAPT* accepting all trees models of φ. Then, the exponential-upper bound follows from the fact that the emptiness problem for *GAPT* is solvable in PTIME [KPV02].

In [FMP08] the module checking problem to the *graded μ-calculus* has been investigate. To see an example of module checking a finite-state open system w.r.t. graded μ-calculus specification, consider again the above two-drink dispenser machine with the following extra feature: "whenever the customer comes at a choice, he can choose for both". This property can be formalized by the graded μ–calculus formula $\nu x.(choose \rightarrow \langle 1, service \rangle)$.

By exploiting an automata–theoretic approach via tree automata, graded μ–calculus module checking is decidable and solvable in EXPTIME in the size of the formula and PTIME in the size of the system. Therefore, in the module checking the size of a formula induces an exponential blow-up *w.r.t.* the model checking problem. In particular, we reduce the addressed module checking problem to the emptiness problem (*GAPT*).

Now, we recall the graded μ–calculus. We refer to [BLMV06] for more technical definitions and motivating examples.

Graded μ–Calculus. Let AP, Var, and $Prog$ be finite and pairwise disjoint sets of *atomic propositions*, *propositional variables*, and *atomic programs* (which allow to travel the system along accessibility relations). The set of *graded μ–calculus* formulas is the smallest set such that

- **true** and **false** are formulas;
- p and $\neg p$, for $p \in AP$, are formulas;
- $x \in Var$ is a formula;

- if φ_1 and φ_2 are formulas, $\alpha \in Prog$, n is a non negative integer, and $y \in Var$, then the following are also formulas:

$$\varphi_1 \vee \varphi_2, \varphi_1 \wedge \varphi_2, \langle n, \alpha \rangle \varphi_1, [n, \alpha]\varphi_1,\ \mu y.\varphi_1(y), \text{and } \nu y.\varphi_1(y).$$

Observe that we use positive normal form, i.e., negation is applied only to atomic propositions.

We call μ and ν *fixpoint operators*. A propositional variable y occurs *free* in a formula if it is not in the scope of a fixpoint operator. A *sentence* is a formula that contains no free variables. We refer often to the *graded modalities* $\langle n, \alpha \rangle \varphi_1$ and $[n, \alpha]\varphi_1$ as respectively *atleast formulas* and *allbut formulas* and assume that the integers in these operators are given in binary code.

The semantics of the graded μ–calculus is defined with respect to a *Kripke structure*, i.e., a tuple $\mathcal{K} = \langle W, W_0, R, L \rangle$ where W is a non–empty set of *states*, $W_0 \subseteq W$ is the set of initial states, $R : Prog \to 2^{W \times W}$ is a function that assigns to each atomic program a transition relation over W, and $L : AP \to 2^W$ is a labeling function that assigns to each atomic proposition and nominal a set of states such that the sets assigned to nominals are singletons and subsets of W_0. If $(w, w') \in R(\alpha)$, we say that w' is an α*–successor* of w. Informally, an *atleast* formula $\langle n, \alpha \rangle \varphi$ holds at a state w of $\mathcal{K}$ if φ holds in at least $n + 1$ α–successors of w. Dually, the *allbut* formula $[n, \alpha]\varphi$ holds in a state w of $\mathcal{K}$ if φ holds in all but at most n α–successors of w. Note that $\neg\langle n, \alpha \rangle \varphi$ is equivalent to $[n, \alpha]\neg\varphi$, and the modalities $\langle \alpha \rangle \varphi$ and $[\alpha]\varphi$ of the standard μ–calculus can be expressed as $\langle 0, \alpha \rangle \varphi$ and $[0, \alpha]\varphi$, respectively.

To formalize semantics, we introduce valuations. Given a Kripke structure $\mathcal{K} = \langle W, W_0, R, L \rangle$ and a set $\{y_1, \ldots, y_n\}$ of variables in Var, a *valuation* $\mathcal{V} : \{y_1, \ldots, y_n\} \to 2^W$ is an assignment of subsets of W to the variables $y_1, \ldots, y_n$. For a valuation $\mathcal{V}$, a variable y, and a set $W' \subseteq W$, we denote by $\mathcal{V}[y \leftarrow W']$ the valuation obtained from $\mathcal{V}$ by assigning W' to y. A formula φ with free variables among $y_1, \ldots, y_n$ is interpreted over $\mathcal{K}$ as a mapping $\varphi^{\mathcal{K}}$ from valuations to 2^W, i.e., $\varphi^{\mathcal{K}}(\mathcal{V})$ denotes the set of points that satisfy φ under valuation $\mathcal{V}$. The mapping $\varphi^{\mathcal{K}}$ is defined inductively as follows:

- $\textbf{true}^{\mathcal{K}}(\mathcal{V}) = W$ and $\textbf{false}^{\mathcal{K}}(\mathcal{V}) = \emptyset$;

- for $p \in AP$, we have $p^{\mathcal{K}}(\mathcal{V}) = L(p)$ and $(\neg p)^{\mathcal{K}}(\mathcal{V}) = W \setminus L(p)$;

- for $y \in Var$, we have $y^{\mathcal{K}}(\mathcal{V}) = \mathcal{V}(y)$;
- $(\varphi_1 \wedge \varphi_2)^{\mathcal{K}}(\mathcal{V}) = \varphi_1^{\mathcal{K}}(\mathcal{V}) \cap \varphi_2^{\mathcal{K}}(\mathcal{V})$ and $(\varphi_1 \vee \varphi_2)^{\mathcal{K}}(\mathcal{V}) = \varphi_1^{\mathcal{K}}(\mathcal{V}) \cup \varphi_2^{\mathcal{K}}(\mathcal{V})$;
- $(\langle n, \alpha \rangle \varphi)^{\mathcal{K}}(\mathcal{V}) = \{w : |\{w' \in W : (w, w') \in R(\alpha)$ and $w' \in \varphi^{\mathcal{K}}(\mathcal{V})\}| \geq n + 1\}$;
- $([n, \alpha]\varphi)^{\mathcal{K}}(\mathcal{V}) = \{w : |\{w' \in W : (w, w') \in R(\alpha)$ and $w' \notin \varphi^{\mathcal{K}}(\mathcal{V})\}| \leq n\}$;
- $(\mu y.\varphi(y))^{\mathcal{K}}(\mathcal{V}) = \bigcap\{W' \subseteq W : \varphi^{\mathcal{K}}([y \leftarrow W']) \subseteq W'\}$;
- $(\nu y.\varphi(y))^{\mathcal{K}}(\mathcal{V}) = \bigcup\{W' \subseteq W : W' \subseteq \varphi^{\mathcal{K}}([y \leftarrow W'])\}$.

For a state w of a Kripke structure $\mathcal{K}$, we say that $\mathcal{K}$ *satisfies* φ at w if $w \in \varphi^{\mathcal{K}}$. In what follows, a formula φ *counts* up to b if the maximal integer in *atleast* and *allbut* formulas used in φ is $b - 1$.

Graded μ-Calculus Module Checking. In [FMP08] has been considered open systems, *i.e.*, systems that interact with their environment and whose behavior depends on this interaction. The (global) behavior of such a system is described by a *module* $\mathcal{M} = \langle W_s, W_e, W_0, R, L \rangle$, which is a Kripke structure where the set of states $W = W_s \cup W_e$ is partitioned in *system states* W_s and *environment states* W_e.

Given a module $\mathcal{M}$, we assume that its states are ordered and the number of successors of each state w is finite. For each $w \in W$, we denote by $succ(w)$ the ordered tuple (possibly empty) of w's α-successors, for all $\alpha \in Prog$. When $\mathcal{M}$ is in a system state w_s, then all states in $succ(w_s)$ are possible next states. On the other hand, when $\mathcal{M}$ is in an environment state w_e, the possible next states (that are in $succ(w_e)$) depend on the current environment. Since the behavior of the environment is not predictable, we have to consider all the possible sub–tuples of $succ(w_e)$. The only constraint, since we consider environments that cannot block the system, is that not all the transitions from w_e are disabled.

The set of all (maximal) computations of $\mathcal{M}$, starting from W_0, is described by a $(W, Prog)$–labeled tree $\langle F_{\mathcal{M}}, \mathrm{V}_{\mathcal{M}}, \mathrm{E}_{\mathcal{M}} \rangle$, called *computation tree*, which is obtained by unwinding $\mathcal{M}$ in the usual way. The problem of deciding, for a given branching–time formula φ over AP, whether $\langle F_{\mathcal{M}}, L \circ \mathrm{V}_{\mathcal{M}}, \mathrm{E}_{\mathcal{M}} \rangle$ satisfies φ at a root node, denoted $\mathcal{M} \models \varphi$, is the usual *model–checking* *problem* [CE81, QS81]. On the other hand, for an open system $\mathcal{M}$, the tree $\langle F_{\mathcal{M}}, \mathrm{V}_{\mathcal{M}}, \mathrm{E}_{\mathcal{M}} \rangle$ corresponds to a very specific environment, *i.e.*, a maximal environment that never restricts the set of its next states. Therefore, when we examine a branching–time formula φ *w.r.t.* $\mathcal{M}$, the formula φ should hold not only in $\langle F_{\mathcal{M}}, \mathrm{V}_{\mathcal{M}}, \mathrm{E}_{\mathcal{M}} \rangle$, but in all tree obtained by pruning from $\langle F_{\mathcal{M}}, \mathrm{V}_{\mathcal{M}}, \mathrm{E}_{\mathcal{M}} \rangle$ subtrees rooted at children of environment nodes, as well as inhibiting some of their jumps to roots (that is, successor nodes labeled with nominals), if there are any. The set of these tree, which collects all possible behaviors of the environment, is denoted by $exec(\mathcal{M})$ and is formally defined as follows. A tree $\langle F, \mathrm{V}, \mathrm{E} \rangle \in exec(\mathcal{M})$ iff

- for each $w_i \in W_0$, we have $\mathrm{V}(i) = w_i$;
- for each $x \in F$, with $\mathrm{V}(x) = w$, $succ(w) = \langle w_1, \ldots, w_n, w_{n+1}, \ldots, w_{n+m} \rangle$, and $succ(w) \cap W_0 = \langle w_{n+1}, \ldots, w_{n+m} \rangle$, there exists $\mathrm{St} = \langle w'_1, \ldots, w'_p, w'_{p+1}, \ldots, w'_{p+q} \rangle$ sub-tuple of $succ(w)$ such that $p + q \geq 1$ and the following hold:
  - $\mathrm{St} = succ(w)$ if $w \in W_s$;
  - $children(x) = \{x \cdot 1, \ldots, x \cdot p\}$ and, for $1 \leq j \leq p$, we have $\mathrm{V}(x \cdot j) = w'_j$ and $\mathrm{E}(x, x \cdot j) = \alpha$ if $(w, w'_j) \in R(\alpha)$;
  - for $1 \leq j \leq q$, let $x_j \in \mathbb{N}$ such that $\mathrm{V}(x_j) = w'_{p+j}$, then $\mathrm{E}(x, x_j) = \alpha$ if $(w, w'_{p+j}) \in R(\alpha)$.

In the following, we consider tree in $exec(\mathcal{M})$ as labeled with $(2^{AP}, Prog)$, i.e., taking the label of a node x as $L(\mathrm{V}(x))$. For a module $\mathcal{M}$ and a formula φ of the graded μ–calculus, we say that $\mathcal{M}$ *reactively* satisfies φ, denoted $\mathcal{M} \models_r \varphi$ (where "r" stands for *reactively*), if all tree in $exec(\mathcal{M})$ satisfy φ. The problem of deciding whether $\mathcal{M} \models_r \varphi$ is called *graded μ–calculus module checking*.

Let $\mathcal{M}$ be a module and φ an graded μ–calculus formula. We decide the module checking problem for $\mathcal{M}$ against φ by building a *GAPT* $\mathtt{A}_{\mathcal{M} \times \not\models \varphi}$ as the intersection of two automata. Essentially, the first automaton, denoted by $\mathtt{A}_{\mathcal{M}}$, is a Büchi automaton that accepts trees of $exec(\mathcal{M})$, and the second automaton is a *GAPT* $\mathtt{A}_{\not\models \varphi}$ that accepts all tree model that do not satisfy φ (i.e, $\neg\varphi$ is satisfied at all initial nodes). Thus, $\mathcal{M} \models_r \varphi$ iff $\mathcal{L}(\mathtt{A}_{\mathcal{M} \times \not\models \varphi})$ is empty. First we recall some important statements.

Theorem 2.2.1. *[BLMV06] The emptiness problem for a* GAPT $\mathtt{A} = \langle \Sigma, b, Q, \delta, q_0, \mathtt{F} \rangle$ *can be solved in time linear in the size of* Σ *and* b*, and exponential in the index of the automaton and number of states.*

Lemma 2.2.1. *[BLMV06] Given two* GAPT $\mathtt{A}_1$ *and* $\mathtt{A}_2$*, there exists a* GAPT $\mathtt{A}$ *such that* $\mathcal{L}(\mathtt{A}) = \mathcal{L}(\mathtt{A}_1) \cap \mathcal{L}(\mathtt{A}_2)$ *and whose size is linear in the size of* $\mathtt{A}_1$ *and* $\mathtt{A}_2$*.*

We now recall a result on *GAPT* and graded μ-calculus formulas.

Lemma 2.2.2 ([BLMV06])**.** *Given an graded* μ*-calculus sentence* φ *with* ℓ atleast *subsentences and counting up to* b*, it is possible to construct a* GAPT *with* $\mathcal{O}(|\varphi|^2)$ *states, index* $|\varphi|$*, and counting bound* b *that accepts exactly each tree that encodes a tree model of* φ*.*

By using the above result, the following statement holds.

Theorem 2.2.2. *The module checking problem with respect to graded* μ*–calculus formulas is* EXPTIME*–complete.*

2.2.2 Graded ATL

In [FNP09] the quantifiers of ATL, like GCTL, have been enriched with an integer that represents the degree. This extension has been called graded ATL and now we describe this. First, we introduce the concept of turn based game. Later, we show the syntax and semantics of graded ATL. Finally, we give the results on the problem of model checking.

Turn Based Games. A *Turn Based Game* is a tuple $\mathcal{G} \triangleq (m, \mathrm{St}, pl, \tau, \lambda)$ such that: *(i)* $m > 0$ is the number of players; *(ii)* St is a finite set of states; *(iii)* $pl : \mathrm{St} \to \{1, ..., m\}$ is a function mapping each state s to the player who owns it; *(iv)* $\tau \subseteq \mathrm{St} \times \mathrm{St}$ is the transition relation and *(v)* $\lambda : \mathrm{St} \to 2^{\mathrm{AP}}$ is the function assigning to each state s the set of atomic propositions that are true at s. We assume that games are non-blocking, *i.e.* each state has at least one successor in τ. In the following, unless otherwise noted, we consider a fixed game $\mathcal{G}$. A (finite or infinite) path in $\mathcal{G}$ is a (finite or infinite) path in the directed graph (St, τ).

A strategy in $\mathcal{G}$ is a pair (X, f), where $\mathrm{X} \subseteq \{1, ..., m\}$ is the team to which the strategy belongs, and $\mathsf{f} : \mathrm{St}^+ \to \mathrm{St}$ is a function such that for all $\rho \in \mathrm{St}^+$, $(\mathsf{lst}(()\rho), \mathsf{f}(\rho)) \in \tau$. Note, our strategies are deterministic. For a team $\mathrm{X} \subseteq \{1, ..., m\}$, we denote by St_{X} the set of states belonging to team X, *i.e.* $\mathrm{St}_{\mathrm{X}} = \{s \in \mathrm{St} | pl(s) \in \mathrm{X}\}$, and we denote by $\neg\mathrm{X}$ the opposite team, *i.e.* $\neg\mathrm{X} = \{1, ..., m\} \setminus \mathrm{X}$. We say that an infinite path $s_0 s_1 ...$ in $\mathcal{G}$ is consistent with a strategy $\sigma = (\mathrm{X}, \mathsf{f})$ if for all $i \geq 0$, if $s_i \in \mathrm{St}_{\mathrm{X}}$ then $s_{i+1} = \mathsf{f}(s_0 s_1 ... s_i)$. We denote by $\textsc{Out}_{\mathcal{G}}(s, \sigma)$ the set of all infinite paths in $\mathcal{G}$ which start from s and are consistent with σ (we omit the subscript $\mathcal{G}$ when it is obvious from the context). For two strategies $\sigma_1 = (\mathrm{X}, \mathsf{f})$ and $\sigma_2 = (\neg\mathrm{X}, \mathsf{g})$, and a state s, we denote by $\textsc{Out}(s, \sigma_1, \sigma_2)$ the unique infinite path which starts from s and is consistent with both σ_1 and σ_2.

Syntax and Semantics. The *graded alternating-time temporal logics* (GATL, for short) extends ATL by using a special team quantifiers $\langle\langle \mathrm{X} \rangle\rangle^{g}$, where $\mathrm{X} \subseteq \{1, ..., m\}$ is a team and $g \in \mathbb{N}$ denotes the corresponding *degree*. The formal syntax of GATL follows.

Definition 2.2.1. GATL *state (ψ) and path (φ) formulas are built inductively from the sets of atomic propositions* AP *in the following way, where $p \in$ AP is an atomic proposition, $\mathrm{X} \subseteq \{1, ..., m\}$ is a team, and g is a natural number:*

1. $\psi ::= p \mid \neg\psi \mid \psi \vee \psi \mid \langle\langle \mathrm{X} \rangle\rangle^{g} \varphi$;
2. $\varphi ::= \mathtt{X}\psi \mid \psi \, \mathtt{U} \psi \mid \mathtt{G}\psi$.

The operators U (until), G (globally) and X (next) are temporal operators. The syntax of ATL is the same as the one of GATL, except that the team quantifier exhibits no natural superscript. Note that GATL formulas with degrees $g = 1$ are ATL formulas.

In [FNP09] present two alternative semantics for graded ATL, called *off-line* semantics and *on-line* semantics. GATL have the ability to count how many different strategies (in the off-line semantics) or paths (in the on-line semantics) satisfy a certain property. Their satisfaction relations are denoted by $\models^{\mathrm{off}}$ and $\models^{\mathrm{on}}$, respectively, and they only differ in the interpretation of the team quantifier. The meaning of the other operators is invariant in the two semantics. The formal semantic of GATL follows.

Definition 2.2.2. *Let ρ be an infinite path in the game, s be a state, and ψ_1 and ψ_2 be state formulas. For $x \in \{on, off\}$, the satisfaction relations are defined as follows.*

- *$s \models^x p$ iff $p \in \lambda(s)$,*
- *$s \models^x \neg\psi_1$ iff $s \not\models^x \psi_1$,*
- *$s \models^x \psi_1 \vee \psi_2$ iff $s \models^x \psi_1$ or $s \models^x \psi_2$,*
- *$\rho \models^x \mathtt{X}\psi_1$ iff $\rho(1) \models^x \psi_1$,*
- *$\rho \models^x \mathtt{G}\psi_1$ iff $\forall i \in \mathbb{N}$, $\rho(i) \models^x \psi_1$,*
- *$\rho \models^x \psi_1 \mathtt{U} \psi_1$ iff $\exists j \in \mathbb{N}$, $\rho(j) \models^x \psi_2$ and $\forall 0 \leq i < j$, $\rho(i) \models^x \psi_1$.*

Off-line semantics. *The meaning of the team quantifier is defined as follows, for a path formula φ.*

- *$s \models^{off} \langle\langle \mathrm{X} \rangle\rangle^g \varphi$ iff there exist g strategies $\sigma_i = (\mathrm{X}, \mathsf{f}_i)$ such that for all $k \neq j$, σ_k and σ_j are $(\langle\langle \mathrm{X} \rangle\rangle^g \varphi, off)$-dissimilar at s and for all $\rho \in \textsc{Out}(s, \sigma_k)$, $\rho \models^{off} \varphi$.*

On-line semantics. *The meaning of the team quantifier is defined as follows, for a path formula φ.*

- *$s \models^{on} \langle\langle \mathrm{X} \rangle\rangle^g \varphi$ iff for all $\sigma' = (\neg\mathrm{X}, \mathsf{f})$ there exist g pairwise $(\langle\langle \mathrm{X} \rangle\rangle^g \varphi, on)$-dissimilar paths $\rho \in \textsc{Out}(s, \sigma')$ such that $\rho \models^{on} \varphi$.*

In the two semantics was introduced the concept of dissimilar, now illustrate three definitions that explain this concept. First, we define the property of dissimilarity for two finite paths.

Definition 2.2.3. *Two finite paths ρ and ρ' are dissimilar iff there exists $0 \leq i \leq \mathsf{min}\{|\rho|, |\rho'|\}$ such that $\rho(i) \neq \rho'(i)$.*

Observe that if ρ is a prefix of ρ' , then ρ and ρ' are not dissimilar. Now, we analyze the property of dissimilarity for two infinite paths on a generic graded ATL formula with team quantifier.

Definition 2.2.4. *Given a GATL formula $\langle\langle \mathrm{X} \rangle\rangle \varphi$, where φ is a path formula, a team $\mathrm{X} \subseteq \{1, ..., m\}$, and $x \in \{on, off\}$, we say that two infinite paths ρ and ρ' are (φ, x)-dissimilar iff:*

- *$\varphi = \mathtt{X}\psi$ and $\rho(1) \neq \rho'(1)$, or*
- *$\varphi = \mathtt{G}\psi$ and $\rho(i) \neq \rho'(i)$ for some i, or*
- *$\varphi = \psi_1 \, \mathtt{U} \, \psi_2$ and there are two integers j and k such that:*

– $\rho(j) \models^x \psi_2$,

– $\rho^{'}(k) \models^x \psi_2$,

– *for all* $0 \leq i < j$, $\rho(i) \models^x \psi_1$ *and* $\rho(i) \models^x \langle\langle \mathrm{X} \rangle\rangle \psi_1 \, \mathtt{U} \psi_2$, *and*

– *for all* $0 \leq h < k$, $\rho^{'}(h) \models^x \psi_1$ *and* $\rho^{'}(h) \models^x \langle\langle \mathrm{X} \rangle\rangle \psi_1 \, \mathtt{U} \psi_2$, *and*

– $\rho_{\leq j}$ *and* $\rho^{'}_{\leq k}$ *are dissimilar.*

Finally, we analyze the case on two sets of infinite paths.

Definition 2.2.5. *Two sets of infinite paths are* (φ, x)*-dissimilar iff one set contains a path which is* (φ, x)*-dissimilar to all the paths in the other set, and, given a state* s*, two strategies* σ_1*,* σ_2 *are* (φ, x)*-dissimilar at* s *if the sets* $\textsc{Out}(s, \sigma_1)$ *and* $\textsc{Out}(s, \sigma_2)$ *are* (φ, x)*-dissimilar.*

Decision problems. For GATL was studied the model checking problem for both off-line and on-line semantics. Were developed two algorithms in [FNP09], one for each semantic that solve the model checking problem for a generic GATL formula. It has been shown that a algorithm for the off-line semantics is performed in linear time, precisely, the following theorem holds.

Theorem 2.2.3. *Given a game* $\mathcal{G}$*, a state* s *in* $\mathcal{G}$ *and a graded* ATL *formula* ψ*, the graded model checking problem,* $s \models^{\mathit{off}} \psi$*, can be solved in time* $\mathrm{O}(|\tau| \cdot |\psi|)$*, where* $|\psi|$ *is the number of operators occurring in* ψ*.*

Similarly it was shown that a algorithm for on-line semantic runs in quadratic time, precisely, the following theorem holds.

Theorem 2.2.4. *Given a game* $\mathcal{G}$*, a state* s *in* $\mathcal{G}$ *and a graded* ATL *formula* ψ*, the graded model checking problem,* $s \models^{\mathit{on}} \psi$*, can be solved in time* $\mathrm{O}(|\mathrm{St}| \cdot |\tau| \cdot |\psi|)$*, where* $|\psi|$ *is the number of operators occurring in* ψ*.*

Finally, from the PTIME hardness of the reachability problem for AND-OR graphs [14], this corollary follows for both off-line and on-line semantics.

Corollary 2.2.1. *The graded* ATL *model checking problem is* PTIME-COMPLETE.

Now, we evaluate the decision problems on concurrent games. Intuitively, a game $\mathcal{G}$ is concurrent if it is possible to associate any number of agent in any state. In [AHK02], there is a construction that reduces a concurrent game in a turn-based game with two player. That said, the following theorem holds.

Theorem 2.2.5. *The model-checking problem for graded* ATL *on concurrent games is* PTIME-COMPLETE*, and can be solved in time* $\mathrm{O}(|\tau| \cdot |\psi|)$ *in the off-line semantics and in time* $\mathrm{O}(|\tau|^2 \cdot |\psi|)$ *in the on-line semantics, for a game with transition function* τ *and for a formula* ψ*.*

3 Graded Strategy Games

In this Chapter, we describe a reformulation of the game-theoretic framework defined in [MMS14] and introduce an extension of Strategy Logics (SL, for short) with graded modalities (GSL, for short). The syntax of GSL extends the one of SL by replacing the two classic *strategy quantifiers* with their graded version. Similarly, the semantics of GSL differs from the one of SL on the evaluation of the two quantifiers, in which strategies are counted by means of suitable equivalence relation. Finally, as it has been done for SL, we define the following four fragments of GSL: GSL[BG], GSL[1G], GSL[CG], and GSL[DG].

3.1 Game Framework

In this section, we describe a reformulation proposed in [MMS14] of the game-theoretic framework. Differently from classic approaches, the basic aspects of a game are factorized into three components: *arena*, *extension*, and *schema*. The arena describes the world where agents act. The extension specifies the properties of interests of the plays. Finally, a schema abstractly represents the solution concepts to analyze.

3.1.1 Arenas

To describe a game, we first need to define the space of its configurations, identify the players, and declare the necessary rules according to which a play has to evolve. In the classic game of chess, for example, two people plays on a 8×8 chessboard that, together with the positions of the 32 pieces equally split in the white and black sets, determines all configurations. The style of moving for each of the 6 different types of pieces, moreover, prescribes the rules that the players have to follow. In general, the formal description of the dynamics of a game can be done by means of the concept of *arena* that combines all the discussed information. It is important to observe that the configurations cannot be considered as global states of the players, but states of the world in which they operate. In particular, since we are only interested in games of perfect information, we are not taking into account the local states of the players when defining their legal moves [FHMV95].

Definition 3.1.1 (Arena). *A multi-agent concurrent* arena *is a tuple* $\mathcal{A} \triangleq \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr} \rangle \in \mathrm{Ar}(\mathrm{Ag})$*, where* Ag *is the finite set of* agents, a.k.a. players*,* Ac *is the set of* actions, a.k.a. moves*, and* St *is the non-empty set of* states, a.k.a. positions*. Assume* $\mathrm{Dc} \triangleq \mathrm{Ag} \rightharpoonup \mathrm{Ac}$ *to be the set of* decisions*,* i.e.*, partial functions describing the choices of an action by some agent. Then,* $\mathsf{tr} : \mathrm{Dc} \to 2^{\mathrm{St} \times \mathrm{St}}$ *denotes the* transition function *mapping every decision* $\delta \in \mathrm{Dc}$ *to a relation* $\mathsf{tr}(\delta) \subseteq \mathrm{St} \times \mathrm{St}$ *between states. Finally,* $\mathrm{Ar}(\mathrm{Ag})$ *is the class of all arenas over* Ag.

Intuitively, an arena can be seen as a generic *labeled transition graph* [Kel76], where labels are possibly incomplete agent decisions, which determine the transitions to be executed at each step of a play in dependence of the choices made by the agents in the relative state. In particular, incomplete decisions allow us to represent any kind of legal move in a state, where some agents or a particular combination of actions may not be active. It can be interesting to note that an arena actually corresponds, in the jargon of modal logic, to a *frame* representing the naked structure underlying a model. Observe also that, due to the loose definition of the transition function, an arena is in general nondeterministic. Indeed, two different but indistinguishable decisions may enable different transitions for the same state. Even more, a single decision may induce a non-functional relation. However, due to the focus of this work, we restrict to the case of deterministic arenas, by describing later on few conditions that rule out how the transition function has to map partial decisions to transitions.

An arena $\mathcal{A}$ naturally induces a graph $\mathcal{G}(\mathcal{A}) \triangleq \langle \mathrm{St}, Ed \rangle$, whose vertexes are represented by the states and the edge relation $Ed \triangleq \bigcup_{\delta \in \mathrm{Dc}} \mathsf{tr}(\delta)$ is obtained by rubbing out all labels on the transitions. Note that there could be states where no transitions are available, *i.e.*, $\mathsf{dom}(Ed) \subset \mathrm{St}$. If this is the case, those ones in $\mathrm{St} \setminus \mathsf{dom}(Ed)$ are called *sink-states*. A *path* $\pi \in \mathrm{Pth} \triangleq \{\pi \in \mathrm{St}^{\omega} : \forall i \in \mathbb{N} . ((\pi)_i, (\pi)_{i+1}) \in Ed\}$ in $\mathcal{A}$ is simply an infinite path in $\mathcal{G}(\mathcal{A})$. Similarly, the *order* $|\mathcal{A}| \triangleq |\mathcal{G}(\mathcal{A})|$ (resp., *size* $\|\mathcal{A}\| \triangleq \|\mathcal{G}(\mathcal{A})\|$) of $\mathcal{A}$ is the order (resp., size) of its induced graph. As usual in the study of extensive-form games, finite paths also describe the possible evolutions of a play up to a certain point. For this reason, they are called in the game-theoretic jargon *histories*, whose corresponding set is denoted by $\mathrm{Hst} \triangleq \{\rho \in \mathrm{St}^* : \forall i \in [0, |\rho| - 1[\, . ((\rho)_i, (\rho)_{i+1}) \in Ed\}$.

We now introduce the sets of decisions, agents, and actions that trigger some transition in a given state $s \in \mathrm{St}$ by means of the three functions $\mathsf{dc} : \mathrm{St} \to 2^{\mathrm{Dc}}$, $\mathsf{ag} : \mathrm{St} \to 2^{\mathrm{Ag}}$, and $\mathsf{ac} : \mathrm{St} \times \mathrm{Ag} \to 2^{\mathrm{Ac}}$ defined as follows:

$$\mathsf{dc}(s) \triangleq \{\delta \in \mathrm{Dc} : s \in \mathsf{dom}(\mathsf{tr}(\delta))\};$$

$$\mathsf{ag}(s) \triangleq \{a \in \mathrm{Ag} : \exists \delta \in \mathsf{dc}(s) \, . \, a \in \mathsf{dom}(\delta)\};$$

$$\mathsf{ac}(s, a) \triangleq \{\delta(a) \in \mathrm{Ac} : \delta \in \mathsf{dc}(s) \ \wedge \ a \in \mathsf{dom}(\delta)\}, \text{ for all } a \in \mathrm{Ag}.$$

These functions can be easily lifted to the set of histories as follows: $\mathsf{dc} : \mathrm{Hst} \to 2^{\mathrm{Dc}}$ with $\mathsf{dc}(\rho) \triangleq \mathsf{dc}(\mathsf{lst}(\rho))$, $\mathsf{ag} : \mathrm{Hst} \to 2^{\mathrm{Ag}}$ with $\mathsf{ag}(\rho) \triangleq \mathsf{ag}(\mathsf{lst}(\rho))$, and $\mathsf{ac} : \mathrm{Hst} \times \mathrm{Ag} \to 2^{\mathrm{Ac}}$ with $\mathsf{ac}(\rho, a) \triangleq \mathsf{dc}(\mathsf{lst}(\rho), a)$.

A decision $\delta \in \mathrm{Dc}$ is *coherent w.r.t.* a state $s \in \mathrm{St}$ (*s-coherent*, for short), if $\mathsf{ag}(s) \subseteq \mathsf{dom}(\delta)$ and $\delta(a) \in \mathsf{ac}(s, a)$, for all $a \in \mathsf{ag}(s)$. By $\mathrm{Dc}(s) \subseteq \mathrm{Dc}$, we denote the set of all s-coherent decisions.

A *strategy* is a partial function $\sigma \in \mathrm{Str} \triangleq \mathrm{Hst} \rightharpoonup \mathrm{Ac}$ prescribing, whenever defined, which action has to be performed for a certain history of the current outcome. Roughly speaking, it is a generic conditional plan which specifies *"what to do"* but not *"who will do it"*. Indeed, a given strategy can be used by more than one agent at the same time. We say that σ is *coherent w.r.t.* an agent $a \in \mathrm{Ag}$ (*a-coherent*, for short) if, in each possible evolution of the game, either a is not influential or the action that σ prescribes is available to a. Formally, for each history $\rho \in \mathrm{Hst}$, it holds that either $a \notin \mathsf{ag}(\rho)$ or $\rho \in \mathsf{dom}(\sigma)$ and $\sigma(\rho) \in \mathsf{ac}(\rho, a)$. By $\mathrm{Str}(a) \subseteq \mathrm{Str}$ we denote the set of a-coherent strategies. Moreover, $\mathrm{Str}(\mathrm{A}) \triangleq \bigcap_{a \in \mathrm{A}} \mathrm{Str}(a)$ indicates the set of strategies that are coherent with all agents in $\mathrm{A} \subseteq \mathrm{Ag}$.

A *profile* is a function $\xi \in \mathrm{Prf} \triangleq \mathrm{Ag} \rightarrow_a \mathrm{Str}(a)$ specifying a unique behavior for each agent $a \in \mathrm{Ag}$ by associating it with an *a-coherent* strategy $\xi(a) \in \mathrm{Str}(a)$. Given a profile ξ, to identify which action an agent $a \in \mathrm{Ag}$ has chosen to perform on a history $\rho \in \mathrm{Hst}$, we first extract the corresponding strategy $\xi(a)$ and then we determinate the action $\xi(a)(\rho)$, whenever defined. To identify, instead, the whole decision on ρ, we apply the flipping operator to ξ. We get so a function $\widehat{\xi} : \mathrm{Hst} \rightarrow \mathrm{Dc}$ such that $\widehat{\xi}(\rho)(a) = \xi(a)(\rho)$, which maps each history to the planned decision.

A path $\pi \in \mathrm{Pth}$ is a *play w.r.t.* a profile $\xi \in \mathrm{Prf}$ (*ξ-play*, for short) iff, for all $i \in [0, |\pi|[$, there exists a decision $\delta \in \mathsf{dc}((\pi)_i)$ such that $\delta \subseteq \widehat{\xi}((\pi)_{\leq i})$ and $((\pi)_i, (\pi)_{i+1}) \in \mathsf{tr}(\delta)$, *i.e.* $(\pi)_{i+1}$ is one of the successors of $(\pi)_i$ induced by the decision $\widehat{\xi}((\pi)_{\leq i})$ prescribed by the profile ξ on the history $(\pi)_{\leq i}$.

Arenas describe generic mathematical structures, where the basilar game-theoretic notions of history, strategy, profile, and play can be defined. However, in several contexts, some constraints rule out how the function tr maps partial decisions to transitions between states. Here, as already observed, we require that arenas are deterministic. We do this by means of the following constraints:

1. there are no sink-states, *i.e.*, $\mathsf{dc}(s) \neq \varnothing$, for all $s \in \mathrm{St}$;
2. for all s-coherent decisions $\delta \in \mathrm{Dc}(s)$, there exists a set of agents $\mathrm{A} \subseteq \mathsf{ag}(s)$ such that $\delta_{\restriction \mathrm{A}} \in \mathsf{dc}(s)$;
3. each decision induces a partial function among states, *i.e.* $\mathsf{tr}(\delta) \in \mathrm{St} \rightharpoonup \mathrm{St}$, for all $\delta \in \mathrm{Dc}$;
4. there are no different but indistinguishable active decisions in a given state $s \in \mathrm{St}$, *i.e.*, for all $\delta_1, \delta_2 \in \mathsf{dc}(s)$ with $\delta_1 \neq \delta_2$, there exist $a \in \mathsf{dom}(\delta_1) \cap \mathsf{dom}(\delta_2)$ such that $\delta_1(a) \neq \delta_2(a)$.

Given a state $s \in \mathrm{St}$, the determinism of the arena ensures that there exists exactly one ξ-play π starting in s, *i.e.*, $\mathsf{fst}(\pi) = s$. Such a play is called *(ξ, s)-play*. For this reason, we use the *play function* $\mathsf{play} : \mathrm{Prf} \times \mathrm{St} \rightarrow \mathrm{Pth}$ to identify, for each profile $\xi \in \mathrm{Prf}$ and state $s \in \mathrm{St}$, the corresponding *(ξ, s)-play* $\mathsf{play}(\xi, s)$.

Example 3.1.1. *As a running example, consider the arena* $\mathcal{A} = \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathtt{tr} \rangle$ *depicted in Figure 3.1, where* $\mathrm{Ag} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$*,* $\mathrm{Ac} = \{0, 1, 2\}$*, and* $\mathrm{St} = \{\mathtt{s}_0, \mathtt{s}_1, \mathtt{s}_2, \mathtt{s}_3, \mathtt{s}_4, \mathtt{s}_5, \mathtt{s}_6, \mathtt{s}_7\}$*. Note that agent* $\mathtt{a}$ *is active in all states, agent* $\mathtt{b}$ *only in* $\mathtt{s}_0$*, and agent* $\mathtt{c}$ *in* $\mathtt{s}_2$*,* $\mathtt{s}_3$*, and* $\mathtt{s}_4$*. Moreover, we have that* $\mathtt{ac}(\mathtt{s}_0, \mathtt{a}) = \mathtt{ac}(\mathtt{s}_0, \mathtt{b}) = \mathtt{ac}(\mathtt{s}_2, \mathtt{a}) = \mathtt{ac}(\mathtt{s}_2, \mathtt{c}) = \{0, 1, 2\}$ *and* $\mathtt{ac}(\mathtt{s}_1, \mathtt{a}) = \mathtt{ac}(\mathtt{s}_1, \mathtt{c}) = \mathtt{ac}(\mathtt{s}_3, \mathtt{a}) = \mathtt{ac}(\mathtt{s}_3, \mathtt{c}) = \mathtt{ac}(\mathtt{s}_4, \mathtt{a}) = \mathtt{ac}(\mathtt{s}_4, \mathtt{c}) = \mathtt{ac}(\mathtt{s}_5, \mathtt{a}) = \mathtt{ac}(\mathtt{s}_6, \mathtt{a}) = \mathtt{ac}(\mathtt{s}_7, \mathtt{a}) = \{0, 1\}$.

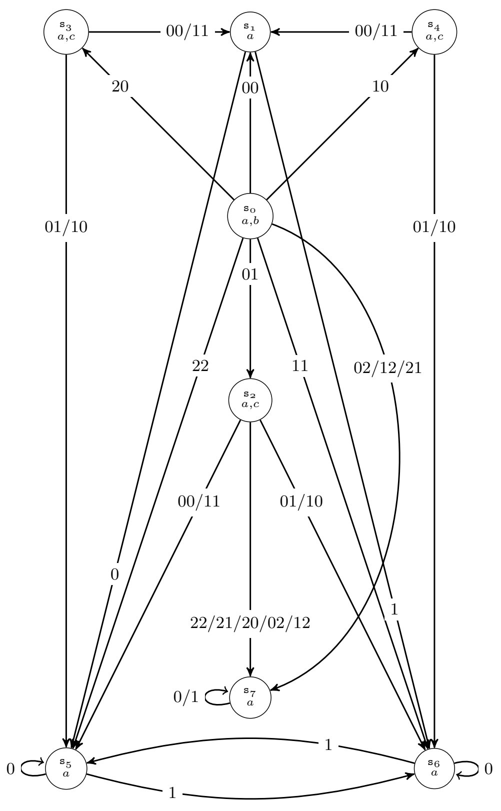

Figure 3.1: Arena $\mathcal{A}$. Note that the node of the arena are labeled with its name (in the upper part) and with the subset of player that are active on its (in the lower part).

3.1.2 Extensions

Besides its dynamics, a game also consists of private objectives or global goals, which typically require to check some properties over the possible plays. For instance, two relevant conditions in chess are whether a given play ends up in a checkmate or a stalemate, which determine a winner or a draw, respectively. Others situations in the same context are the claiming of the fifty-move rule or the threefold repetition. More in general, play properties may represent fairness, reachability, safety, or other conditions, usually expressed by means of some kind of *temporal language*, like LTL [Pnu77] or the linear μCALCULUS [Var88], which combine atomic propositions over states with suitable temporal operators. As game theorists usually do, our aim is to separate the strategic reasoning from the specification of such properties. Therefore, we represent them as generic Boolean predicates over paths, by introducing the concept of *extension*.

Definition 3.1.2 (Extension)**.** *An* extension *is a tuple* $\mathcal{E} \triangleq \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle \in \mathrm{Ex}(\mathrm{Ag}, \mathrm{Pr})$*, where* $\mathcal{A} \in \mathrm{Ar}(\mathrm{Ag})$ *is the underlying arena,* Pr *is the finite non-empty set of* predicates*, and* $\mathsf{pr} : \mathrm{Pr} \to 2^{\mathrm{Pth}}$ *is the* predicate function *mapping each predicate* $p \in \mathrm{Pr}$ *to the set of paths* $\mathsf{pr}(p) \subseteq \mathrm{Pth}$ *satisfying it. Finally,* $\mathrm{Ex}(\mathrm{Ag}, \mathrm{Pr})$ *denotes the class of all extensions over* Ag *and* Pr.

Intuitively, an extension classifies the paths trough a set of monadic predicates that describe all relevant temporal properties for the specific domain under analysis. Coming back to the previous analogy, the extension is the part of framework corresponding to the *model* in the classic context of modal logic.

An extension is *Borelian* (resp., *regular*) iff, for each predicate $p \in \mathrm{Pr}$, the induced language $\mathsf{pr}(p) \subseteq \mathrm{St}^{\omega}$ of infinite words over the states is Borelian (resp., regular) [PP04]. In particular, a predicate p is *open* (resp., *closed*) with witness $\mathrm{W} \subseteq \mathrm{Hst}$ if, for all path $\pi \in \mathrm{Pth}$, it holds that $\pi \in \mathsf{pr}(p)$ iff there is an index $i \in \mathbb{N}$ (resp., for all indexes $i \in \mathbb{N}$) such that $(\pi)_{\leq i} \in \mathrm{W}$.

Example 3.1.2. *As an example, consider the extension* $\mathcal{E} \triangleq \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$*, where* $\mathcal{A}$ *is the arena of Figure 3.1,* $\mathrm{Pr} = \{\mathrm{p}_U, \mathrm{p}_V\}$*, the open predicate* p_U *has witness* U *containing those histories that passing in* s_1*,* s_5*, or* s_6*, and the closed predicate* p_V *has witness* V *containing the histories ending in* s_7.

3.1.3 Schemas

Considering again the chess analysis, one can note that we have defined the chessboard together with the legal moves, *i.e.*, the arena. We have also introduced all relevant properties, such as checkmate or stalemate, which have to be checked during a play, *i.e.*, the extension. However, we have not yet completely described the entire game. Indeed, an initial configuration needs to be specified and, more important, it is necessary to indicate which behavior we expect from the players. In particular, being chess a zero-sum game, the two participant must be adversary. To do this in general, we introduce the concept of *schema*, which abstractly describes the possible roles and the allowed interactions of the agents, by means of a relation between an extension, with an associated initial state, and some solution concepts.

Definition 3.1.3 (Schema). *A* schema *over the sets of agents* Ag *and predicates* Pr *is a tuple* $\mathcal{S} \triangleq \langle \mathrm{Cn}, \models \rangle \in \mathrm{Sc}(\mathrm{Ag}, \mathrm{Pr})$*, where* Cn *is the non-empty set of* solution concepts *and* $\models \subseteq \mathrm{Ex}(\mathrm{Ag}, \mathrm{Pr}) \times_{\mathcal{E}} \mathrm{St}_{\mathcal{E}} \times \mathrm{Cn}$ *is the* schema relation *describing which solution concepts* $\varphi \in \mathrm{Cn}$ *are fulfilled on an extension* $\mathcal{E} \in \mathrm{Ex}(\mathrm{Ag}, \mathrm{Pr})$ *starting from a given state* $s \in \mathrm{St}_{\mathcal{E}}$*, in symbols* $\mathcal{E}, s \models \varphi$*. Finally,* $\mathrm{Sc}(\mathrm{Ag}, \mathrm{Pr})$ *denotes the class of all schemas over* Ag *and* Pr*.*

Informally, a schema is a mean to identify the global properties that a game satisfies, by summarizing in a prescribed suitable way the behaviors the agents exhibit during all possible plays.

3.1.4 Games

By summing up the basic definitions of arena, extension, and schema, we can now describe the formalization of the concept of *game*.

Definition 3.1.4 (Game). *Let* $\mathcal{S} \in \mathrm{Sc}(\mathrm{Ag}, \mathrm{Pr})$ *be a schema over the sets of agents* Ag *and predicates* Pr*. Then, a* game w.r.t. $\mathcal{S}$ *is a tuple* $\eth \triangleq \langle \mathcal{E}, s_I, \varphi \rangle \in \mathrm{Gm}(\mathcal{S})$*, where* $\mathcal{E} \in \mathrm{Ex}(\mathrm{Ag}, \mathrm{Pr})$ *is the underlying extension,* $s_I \in \mathrm{St}_{\mathcal{E}}$ *is the designated initial state, and* $\varphi \in \mathrm{Cn}$ *is the prescribed solution concept. Finally,* $\mathrm{Gm}(\mathcal{S})$ *denotes the class of all games over* $\mathcal{S}$*.*

To conclude with the exposition of game-theoretic framework, we need to introduce the fundamental concept of *fulfillment*, which prescribe how to determine the outcome of the game.

Definition 3.1.5 (Fulfillment). *A game* $\eth = \langle \mathcal{E}, s_I, \varphi \rangle \in \mathrm{Gm}(\mathcal{S})$ w.r.t. *a schema* $\mathcal{S} = \langle \mathrm{Cn}, \models \rangle \in \mathrm{Sc}(\mathrm{Ag}, \mathrm{Pr})$ *over the sets of agents* Ag *and predicates* Pr *is* fulfilled *iff* $\mathcal{E}, s_I \models \varphi$*. The* fulfillment problem *is to decide whether* $\eth$ *is fulfilled.*

Intuitively, a game is fulfilled if all involved agents can play together, on the board described by the arena contained into the extension $\mathcal{E}$ starting from the initial position s_I, in such a way to verify the solution concept φ. Observe that this problem is an abstract generalization of the classic *model checking* of a language against a structure, here represented via φ and $(\mathcal{E}, s_I)$, respectively.

3.2 Graded Strategy Logic

As observed in [MMS14], a schema can be seen as an interface that allows to describe the notion of fulfillment of a game in a completely abstract way. Obviously, we need a formal language to implement such an interface and the more expressive it is the wider the class of strategic reasonings we can grasp is as well. To this aim, we introduce *Graded Strategy Logic* (GSL, for short), an extension of a particular version of *Strategy Logic* (SL, for short) [MMV10, MMS14] that allows to reason about the number of strategies that an agent may exploit in order to satisfy a given temporal goal.

3.2.1 Syntax

GSL extends SL by replacing the two classic *strategy quantifiers* $\langle\langle x\rangle\rangle$ and $[[x]]$, where x belongs to a countable set Vr of variables, with their graded version $\langle\langle x\geq g\rangle\rangle$ and $[[x<g]]$, where the finite number $g\in\mathbb{N}$ denotes the corresponding *degree*. Intuitively, these quantifiers are read as *"there exist at least g strategies"* and *"all but less than g strategies"*. Moreover, GSL syntax comprises a set Pr of predicates to expresses properties over paths, a binding operator to link strategies to agents, and Boolean connectives.

Definition 3.2.1 (GSL Syntax)**.** GSL formulas *are built inductively by means of the following context-free grammar, where* $a\in\mathrm{Ag}$*,* $p\in\mathrm{Pr}$*,* $x\in\mathrm{Vr}$*, and* $g\in\mathbb{N}$*:*

$$\varphi := p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \varphi\vee\varphi \mid \langle\langle x\geq g\rangle\rangle\varphi \mid [[x<g]]\varphi \mid (a,x)\varphi.$$

As usual, to provide the semantics of a predicative logic, it is necessary to define the concept of free and bound *placeholders* of a formula. As for SL, since strategies can be associated to both agents and variables, we need the set of *free agents/variables* $\mathsf{free}(\varphi)$ as the subset of $\mathrm{Ag}\cup\mathrm{Vr}$ containing *(i)* all agents a for which there is no binding (a,x) before the occurrence of a predicate p and *(ii)* all variables x for which there is a binding (a,x) but no quantification $\langle\langle x\geq g\rangle\rangle$ or $[[x<g]]$.

Definition 3.2.2 (GSL Free Agents/Variables)**.** *The set of* free agents/variables *of a* GSL *formula is given by the function* $\mathsf{free}:\mathrm{GSL}\rightarrow 2^{\mathrm{Ag}\cup\mathrm{Vr}}$ *defined as follows:*

1. $\mathsf{free}(p) \triangleq \mathrm{Ag}$, *where* $p \in \mathrm{Pr}$*;*
2. $\mathsf{free}(\neg\varphi) \triangleq \mathsf{free}(\varphi)$*;*
3. $\mathsf{free}(\varphi_1 \vee \varphi_2) \triangleq \mathsf{free}(\varphi_1) \cup \mathsf{free}(\varphi_2)$*;*
4. $\mathsf{free}(\varphi_1 \wedge \varphi_2) \triangleq \mathsf{free}(\varphi_1) \cup \mathsf{free}(\varphi_2)$*;*
5. $\mathsf{free}(\langle\langle x \geq g \rangle\rangle\varphi) \triangleq \mathsf{free}(\varphi) \setminus \{x\}$*;*
6. $\mathsf{free}([[x < g]]\varphi) \triangleq \mathsf{free}(\varphi) \setminus \{x\}$*;*
7. $\mathsf{free}((a, x)\varphi) \triangleq \mathsf{free}(\varphi)$*, if* $a \notin \mathsf{free}(\varphi)$*, where* $a \in \mathrm{Ag}$ *and* $x \in \mathrm{Vr}$*;*
8. $\mathsf{free}((a, x)\varphi) \triangleq (\mathsf{free}(\varphi) \setminus \{a\}) \cup \{x\}$*, if* $a \in \mathsf{free}(\varphi)$*, where* $a \in \mathrm{Ag}$ *and* $x \in \mathrm{Vr}$.

A formula φ without free agents (resp., variables), *i.e.*, with $\mathsf{free}(\varphi) \cap \mathrm{Ag} = \emptyset$ (resp., $\mathsf{free}(\varphi) \cap \mathrm{Vr} = \emptyset$)), is named *agent-closed* (resp., *variable-closed*). A *sentence* is a both agent- and variable-closed formula.

Since a variable x may be bound to more than a single agent at the time, we also need the subset $\mathsf{shr}(\varphi, x)$ of Ag containing those agents for which a binding (a, x) occurs in φ.

Definition 3.2.3 (GSL Shared Variables)**.** *The set of* shared variables *of a* GSL *formula with respect to variable is given by the function* $\mathsf{shr} : \mathrm{GSL} \times \mathrm{Vr} \to 2^{\mathrm{Ag}}$ *defined as follows:*

1. $\mathsf{shr}(p, x) \triangleq \emptyset$*, where* $p \in \mathrm{Pr}$*;*
2. $\mathsf{shr}(\neg\varphi, x) \triangleq \mathsf{shr}(\varphi, x)$*;*
3. $\mathsf{shr}(\varphi_1 \vee \varphi_2, x) \triangleq \mathsf{shr}(\varphi_1, x) \cup \mathsf{shr}(\varphi_2, x)$*;*
4. $\mathsf{shr}(\varphi_1 \wedge \varphi_2, x) \triangleq \mathsf{shr}(\varphi_1, x) \cup \mathsf{shr}(\varphi_2, x)$*;*
5. $\mathsf{shr}(\langle\langle x \geq g \rangle\rangle\varphi, x) \triangleq \mathsf{shr}(\varphi, x)$*;*
6. $\mathsf{shr}([[x < g]]\varphi, x) \triangleq \mathsf{shr}(\varphi, x)$*;*
7. $\mathsf{shr}((a, y)\varphi, x) \triangleq \mathsf{shr}(\varphi, x)$*, if* $a \notin \mathsf{free}(\varphi)$ *or* $y \neq x$*, where* $a \in \mathrm{Ag}$ *and* $y \in \mathrm{Vr}$*;*
8. $\mathsf{shr}((a, x)\varphi, x) \triangleq \mathsf{shr}(\varphi, x) \cup \{a\}$*, if* $a \in \mathsf{free}(\varphi)$*, where* $a \in \mathrm{Ag}$*;*

3.2.2 Semantics

Similarly to SL, the interpretation of a GSL formula requires a valuation for its free placeholders. This is done via *assignment*, *i.e.*, a partial function $\chi \in \mathrm{Asg} \triangleq (\mathrm{Vr} \cup \mathrm{Ag}) \rightharpoonup \mathrm{Str}$ mapping variables and agents to strategies. An assignment χ is *complete* iff it is defined on all agents, *i.e.*, $\chi(a) \in \mathrm{Str}(a)$, for all $a \in \mathrm{Ag} \subseteq \mathsf{dom}(\chi)$. In this case, it directly identifies the profile $\chi_{\restriction \mathrm{Ag}}$ given by the restriction of χ to Ag. In addition, $\chi[e \mapsto \sigma]$, with $e \in \mathrm{Vr} \cup \mathrm{Ag}$ and $\sigma \in \mathrm{Str}$, is the assignment defined on $\mathsf{dom}(\chi[e \mapsto \sigma]) \triangleq \mathsf{dom}(\chi) \cup \{e\}$ that differs from χ only on the fact that e is associated with σ. Formally, $\chi[e \mapsto \sigma](e) = \sigma$ and $\chi[e \mapsto \sigma](e') = \chi(e')$, for all $e' \in \mathsf{dom}(\chi) \setminus \{e\}$. Finally, given a formula φ, we say that χ is φ*-coherent* iff *(i)* $\mathsf{free}(\varphi) \subseteq \mathsf{dom}(\chi)$, *(ii)* $\chi(a) \in \mathrm{Str}(a)$, for all $a \in \mathsf{dom}(\chi) \cap \mathrm{Ag}$, and *(iii)* $\chi(x) \in \mathrm{Str}(a)$, for all $x \in \mathsf{dom}(\chi) \cap \mathrm{Vr}$ and $a \in \mathsf{shr}(\varphi, x)$.

We now define the semantics of a GSL formula φ *w.r.t.* an extension $\mathcal{E}$, one of its states s, and a φ-coherent assignment χ. In particular, we write $\mathcal{E}, \chi, s \models \varphi$ to indicate that φ holds at s in $\mathcal{E}$ under χ. The semantics of formulas involving predicates, Boolean connectives, and agent bindings are defined as in SL. The definition of graded strategy quantifiers, instead, makes use of a generic equivalence relation $\equiv^{\varphi}_{\mathcal{E},s}$ on assignments that depends on the extension, the state, and the formula under exam. This equivalence is used to reasonably count the number of strategies that satisfy a formula starting from a given state, *w.r.t.* an *a priori* fixed criterion. Observe that we use a relation on assignments instead of a more direct one on strategies, since the classification may also depend on the context determined by the strategies previously quantified. In Chapter 4, we will come back on the properties the equivalence has to satisfy in order to be used in the semantics of GSL.

Definition 3.2.4 (GSL Semantics)**.** *Let $\mathcal{E}$ be an extension, $s \in \mathrm{St}$ one of its states, and φ a* GSL *formula. Then, for all φ-coherent assignments $\chi \in \mathrm{Asg}$, the relation $\mathcal{E}, \chi, s \models \varphi$ is inductively defined on the structure of φ as follows.*

1. *For every $p \in \mathrm{Pr}$, it holds that $\mathcal{E}, \chi, s \models p$ iff* $\mathsf{play}(\chi_{\restriction \mathrm{Ag}}, s) \in \mathsf{pr}(p)$.

2. *For all formulas φ, φ_1 and φ_2, it holds that:*

   (a) $\mathcal{E}, \chi, s \models \neg\varphi$ iff $\mathcal{E}, \chi, s \not\models \varphi$;

(b) $\mathcal{E}, \chi, s \models \varphi_1 \wedge \varphi_2$ *iff* $\mathcal{E}, \chi, s \models \varphi_1$ *and* $\mathcal{E}, \chi, s \models \varphi_2$*;*

(c) $\mathcal{E}, \chi, s \models \varphi_1 \vee \varphi_2$ *iff* $\mathcal{E}, \chi, s \models \varphi_1$ *or* $\mathcal{E}, \chi, s \models \varphi_2$.

3. *For each* $x \in \mathrm{Vr}$*,* $g \in \mathbb{N}$*, and* $\varphi \in \mathrm{GSL}$*, it holds that:*

   (a) $\mathcal{E}, \chi, s \models \langle\langle x \geq g \rangle\rangle \varphi$ *iff* $|(\{\chi[x \mapsto \sigma] : \sigma \in \varphi[\mathcal{E}, \chi, s](x)\}/\equiv^{\varphi}_{\mathcal{E},s})| \geq g$*;*

   (b) $\mathcal{E}, \chi, s \models [[x < g]] \varphi$ *iff* $|(\{\chi[x \mapsto \sigma] : \sigma \in \neg\varphi[\mathcal{E}, \chi, s](x)\}/\equiv^{\neg\varphi}_{\mathcal{E},s})| < g$*;*

   where $\eta[\mathcal{E}, \chi, s](x) \triangleq \{\sigma \in \mathrm{Str}(\mathsf{shr}(\eta, x)) : \mathcal{E}, \chi[x \mapsto \sigma], s \models \eta\}$ *is the set of* $\mathsf{shr}(\eta, x)$*-coherent strategies that, being assigned to* x *in* χ*, satisfy* η*.*

4. *For each* $a \in \mathrm{Ag}$*,* $x \in \mathrm{Vr}$*, and* $\varphi \in \mathrm{GSL}$*, it holds that:*

   $\mathcal{E}, \chi, s \models (a, x)\varphi$ *iff* $\mathcal{E}, \chi[a \mapsto \chi(x)], s \models \varphi$*.*

Intuitively, by using the graded existential quantifier $\langle\langle x \geq g \rangle\rangle \varphi$, we can count how many different equivalence classes *w.r.t.* $\equiv^{\varphi}_{\mathcal{E},s}$ there are over the set of assignments $\{\chi[x \mapsto \sigma] : \sigma \in \varphi[\mathcal{E}, \chi, s](x)\}$ extending χ that satisfy φ. The universal quantifier $[[x \geq g]]\varphi$ is simply the dual of $\langle\langle x \geq g \rangle\rangle \varphi$ and it allows to count how many classes *w.r.t.* $\equiv^{\neg\varphi}_{\mathcal{E},s}$ there are over the set of assignments $\{\chi[x \mapsto \sigma] : \sigma \in \neg\varphi[\mathcal{E}, \chi, s](x)\}$ extending χ that do not satisfy φ. It is important to note that, all GSL formulas with degree 1 are SL formulas, and *vice versa*. Note that the satisfaction of a sentence φ does not depend on assignments, hence we omit them and write $\mathcal{E}, s \models \varphi$.

In order to complete the description of the semantics, we now give the classic notions of *model* and *satisfiability* of an GSL sentence. Let φ be a GSL formula and $\mathcal{E}$ an extension having $\mathcal{A}$ as an underlying arena with a set of states St comprising s_0 as an initial state. Then, $\mathcal{E}$ is a *model* of an GSL sentence φ, in symbols $\mathcal{E} \models \varphi$, if $\mathcal{E}, \emptyset, s_0 \models \varphi$. In general, we also say that $\mathcal{E}$ is a model for φ on a generic $s \in \mathrm{St}$, in symbols $\mathcal{E}, s \models \varphi$, if $\mathcal{E}, \emptyset, s \models \varphi$. An GSL sentence φ is *satisfiable* if there is a model for it.

To complete with the semantic properties of GSL, we give the concepts of *implication* and *equivalence* between GSL formulas.

Definition 3.2.5 (GSL *Implication and Equivalence*)**.** *For all* GSL *formulas* φ_1 *and* φ_2 *with* $\mathsf{free}(\varphi_1) = \mathsf{free}(\varphi_2)$*, we say that* φ_1 implies φ_2*, in symbols* $\varphi_1 \Rightarrow \varphi_2$*, if, for all extensions* $\mathcal{E}$*,* *states s, and $\chi \in \mathrm{Asg}(\mathsf{free}(\varphi_1), s)$, it holds that if $\mathcal{E}, \chi, s \models \varphi_1$ then $\mathcal{E}, \chi, s \models \varphi_2$. Consequently, we say that φ_1 is* equivalent *to φ_2, in symbols $\varphi_1 \equiv \varphi_2$, if both $\varphi_1 \Rightarrow \varphi_2$ and $\varphi_2 \Rightarrow \varphi_1$ hold.*

Once the sets of agents and predicates are fixed, GSL induces a schema $\mathcal{S}_{\mathrm{GSL}}$, where the set of solution concepts is GSL itself, *i.e.*, its set of sentences, and the schema relation is given by the definition of its semantics. A *Graded Strategy Game* is a game *w.r.t.* $\mathcal{S}_{\mathrm{GSL}}$, whose set is denoted by $\mathrm{GSG} \triangleq \mathrm{Gm}(\mathcal{S}_{\mathrm{GSL}})$. In what follows, we also consider some constraints on the number of agents/variables or the type of arenas and predicates. More specifically, TB indicates that we restrict to turn-based arenas, and kVAR (resp., kAG) bounds the maximal number of variables (resp., agents) in a formula to k. For example, GSG[1G, TB, 2AG/VAR] is the fragment of GSG restricted to GSL[1G] solution concepts, turn-based arenas, and with at most two agents and variables.

3.2.3 Binding Fragments

We now consider some fragments of GSL that allow to formalize interesting game properties not expressible in other logics. A *quantification prefix* over a set $\mathrm{V} \subseteq \mathrm{Vr}$ of variables is a finite word $\wp \in \{\langle\langle x \geq g \rangle\rangle, [[x < g]] : x \in \mathrm{V} \wedge g \in \mathbb{N}\}^{|\mathrm{V}|}$ of length $|\mathrm{V}|$ such that each variable $x \in \mathrm{V}$ occurs just once in $\wp$. A *binding prefix* over a set of variables V is a word $\flat \in \{(a, x) : a \in \mathrm{Ag} \wedge x \in \mathrm{V}\}^{|\mathrm{Ag}|}$ such that each agent $a \in \mathrm{Ag}$ occurs exactly once in $\flat$. By Bn we indicate the set of all binding prefixes. With $\mathrm{Qn}(\mathrm{V})$ we indicate the set of quantification prefixes over V. We now have all tools to define the syntactic fragments we want to analyze, which we name, respectively, *Boolean-Goal*, *Conjunctive-Goal*, *Disjunctive-Goal*, and *One-Goal Graded Strategy Logic* (GSL[BG], GSL[CG], GSL[DG], and GSL[1G] for short). For *goal* we mean an GSL agent-closed formula of the kind $\flat\varphi$, with $\mathrm{Asg} \subseteq \mathsf{free}(\varphi)$, being $\flat \in \mathrm{Bn}(\mathrm{Vr})$ a binding prefix. The idea GSL[BG] is that, the formulas, after quantification prefix, may have more goals related with boolean operators. The GSL[CG] (resp., GSL[DG]) *w.r.t.* GSL[BG], after quantification prefix, may have more goals but related only with conjunctive operator (resp. disjunctive operator). Finally, GSL[1G] forces the use of a different quantification prefix for each single goal in the formula.

Definition 3.2.6 (Binding fragments)**.** GSL[BG] formulas *are defined by the following context-free grammar:*

$$\varphi := \wp\phi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi,$$
$$\phi := \flat\psi \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi,$$
$$\psi := p \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi.$$

where $\wp \in \mathrm{Qn}(\mathsf{free}(\psi))$*,* $\flat \in \mathrm{Bn}$*, and* $p \in \mathrm{Pr}$*. Moreover, the fragments obtained by replacing the second row with* $\phi := \flat\psi \mid \phi \wedge \phi$*,* $\phi := \flat\psi \mid \phi \vee \phi$*, of* $\phi := \flat\psi$*, are called* GSL[CG], GSL[DG]*, and* GSL[1G] *respectively.*

In the sequel, we also consider some constraints on the number of agents/variables or the type of arenas and predicates. More specifically, TB indicates that we restrict to turn-based arenas, and kVAR (resp. kAG) bounds the maximal number of variables (resp. agents) in a formula to k.

4

Strategy Equivalence

Our definition of GSL semantics makes use of an arbitrary equivalence relation on assignments. This choice introduce flexibility in its description, since one can come up with different semantics by opportunely choosing different equivalences. In this Chapter, we focus on a particular relation whose key feature is to classify as equivalent all assignments that reflect the same "*strategic reasoning*", although they may have completely different structures. Just to get intuition about what we mean, consider for example two assignments and the corresponding involved strategies. Assume now that all such strategies differ only on histories never met by a play because of a specific combination of agent actions. Clearly, these assignments induce the same agent behaviors, which means to reflect the same strategic reasoning. Therefore, it is natural to set them as equivalent, as we do.

The remaining part of this Chapter is organized as follows. First, in Section 4.1, we introduce two basic properties of equivalence on strategies, namely *syntax independent* and *satisfiability matching*. In Section 4.2, we deal with the equivalence relation related to open and closed predicates. Then, in Section 4.3, we extend the study to the strategy operators. In Section 4.4, we give two consistency properties on the Boolean connectives $\wedge$ and $\vee$. Finally, it is worth observing that it is possible to derive the equivalences for predicates belonging to any level of the Borel hierarchy, by opportunely using conjunctions and disjunctions of equivalences for predicates of lower levels.

4.1 Elementary Requirements

In this section, we introduce the property of *syntax independence* and *satisfiability matching*, that are two basic concept that our equivalence relation must have.

Suppose we have two equivalent formulas φ_1 and φ_2. We need to require that, whenever two assignments are equivalent *w.r.t.* φ_1, they are equivalent *w.r.t.* φ_2, as well. The formal definitions of this concept follows.

Definition 4.1.1 (Syntax Independence). *An equivalence relation on assignments $\equiv^{\cdot}_{\mathcal{E},s}$ is* syntax independent *iff, for any pairs of equivalent formulas φ_1 and φ_2 w.r.t $\equiv$ and φ_1-coherent assignments χ_1,$\chi_2 \in \mathrm{Asg}$ we have that $\chi_1 \equiv^{\varphi_1}_{\mathcal{E},s} \chi_2$ iff $\chi_1 \equiv^{\varphi_2}_{\mathcal{E},s} \chi_2$.*

Consider two assignments χ_1 and χ_2. The equivalence relation is established in such a way that if χ_1 and χ_2 are equivalent with respect to a formula φ, either both satisfy φ or none of them does it. We called this properties *satisfiability matching* and the formal definition follows.

Definition 4.1.2 (Satisfiability Matching). *An equivalence relation on assignments $\equiv^{\cdot}_{\mathcal{E},s}$ is* satisfiability matching *if, for any φ and φ-coherent assignments χ_1, χ_2 , we have that if $\chi_1 \equiv^{\varphi}_{\mathcal{E},s} \chi_2$ then either $\mathcal{E}, \chi_1, s \models \varphi$ and $\mathcal{E}, \chi_2, s \models \varphi$ or $\mathcal{E}, \chi_1, s \not\models \varphi$ and $\mathcal{E}, \chi_2, s \not\models \varphi$.*

4.2 Play Requirements

We now formalize the equivalence relation for the basic case of atomic properties over paths. In particular, Definition 4.2.1 handles open predicates and, as a particular case, the reachability properties, while Definition 4.2.2 considers closed predicates and, consequently, the safety properties. As it will be clear in the following, the two definitions are symmetric, due to the intrinsic nature of the involved concepts.

Before disclosing the two formalizations, we would like first to give an intuition on how the relation acts on a generic predicate p. Our aim is to evaluate the equivalence of two assignments

χ_1 and χ_2 *w.r.t.* their agreement on its verification. Specifically, in case both χ_1 and χ_2 do not satisfy p, it is reasonable to think that they are equivalent as they induce two plays that, although possibly different, are irrelevant for p, *i.e.*, none of them belongs to the set of paths satisfying it. Conversely, if one satisfies the predicate, while the other does not, in accordance with the idea to set as equivalent those assignments that reflect the same strategic reasoning, χ_1 and χ_2 are intended to be not equivalent.

It is left to analyze the case in which both assignments satisfy p. If the latter is an open predicate, the plays π_1 and π_2 induced by χ_1 and χ_2 must have two possibly different prefixes belonging to the witness set U of p. Having said that, it is reasonable to evaluate their equivalence by restricting the attention to the common prefix $\rho = \mathsf{prf}(\pi_1, \pi_2)$. Indeed, if also ρ has a prefix that belongs to U, this can be used as a witness for both the memberships $\pi_1, \pi_2 \in \mathsf{pr}(p)$. In such a case, we set χ_1 and χ_2 to be equivalent. Otherwise, there are necessarily different witnesses for the two memberships and, so, we set the corresponding assignments as non-equivalent.

Definition 4.2.1 (Open-Play Consistency). *An equivalence relation on assignments $\equiv^{\cdot}_{\mathcal{E},s}$ is* open-play consistent *if, for any open predicate p with witness $\mathrm{U} \subseteq \mathrm{Hst}$ and p-coherent assignments $\chi_1, \chi_2 \in \mathrm{Asg}$, we have that if $\mathcal{E}, \chi_1, s \not\models p$ and $\mathcal{E}, \chi_2, s \not\models p$ then $\chi_1 \equiv^{p}_{\mathcal{E},s} \chi_2$, otherwise, $\chi_1 \equiv^{p}_{\mathcal{E},s} \chi_2$ iff there is a history $\rho \in \mathrm{Hst}$ with $\rho \leq \mathsf{prf}(\mathsf{play}(\chi_{1\restriction \mathrm{Ag}}, s), \mathsf{play}(\chi_{2\restriction \mathrm{Ag}}, s))$ such that $\rho \in \mathrm{U}$.*

Example 4.2.1. *Consider the extension $\mathcal{E}$ related to the arena $\mathcal{A}$ depicted in Figure 3.1, the state $\mathsf{s}_0 \in \mathrm{St}$, and the open predicate p_U. Moreover, let $\chi_1, \chi_2, \chi_3 \in \mathrm{Asg}(\{\mathsf{a}, \mathsf{b}, \mathsf{c}\})$ be three assignments on which we want to check the equivalence. In particular, assume that each χ_i associates a strategy σ_i^{α} with the agent $\alpha \in \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ as defined in the following: $\sigma^{\mathsf{a}}_{1/2/3}(\mathsf{s}_0) \triangleq 2$, $\sigma^{\mathsf{a}}_{1/3}(\rho) = \sigma^{\mathsf{b}}_{1/2/3}(\rho') = \sigma^{\mathsf{c}}_{1}(\rho') \triangleq 0$, and $\sigma^{\mathsf{a}}_{2}(\rho) = \sigma^{\mathsf{c}}_{2/3}(\rho') \triangleq 1$, for all $\rho \in \mathrm{Hst} \setminus \{\mathsf{s}_0\}$ and $\rho' \in \mathrm{Hst}$. Now, it is easy to see that $\chi_1 \equiv^{\mathsf{p}_U}_{\mathcal{E},\mathsf{s}} \chi_2$ but $\chi_1 \not\equiv^{\mathsf{p}_U}_{\mathcal{E},\mathsf{s}} \chi_3$. Indeed, χ_1, χ_2, and χ_3 induce the plays $\pi_1 = \mathsf{s}_0\mathsf{s}_3\mathsf{s}_1\mathsf{s}_5{}^{\omega}$, $\pi_2 = \mathsf{s}_0\mathsf{s}_3\mathsf{s}_1(\mathsf{s}_6\mathsf{s}_5)^{\omega}$, and $\pi_3 = \mathsf{s}_0\mathsf{s}_3\mathsf{s}_5{}^{\omega}$, respectively, where $\rho_{12} = \mathsf{s}_0\mathsf{s}_3\mathsf{s}_1 = \mathsf{prf}(\pi_1, \pi_2)$ and $\rho_{13} = \mathsf{s}_0\mathsf{s}_3 = \mathsf{prf}(\pi_1, \pi_3)$ are the corresponding common histories. Thus, ρ_{12} belongs to the witness U of p_U, while ρ_{13} does not.*

We now give the intuition behind the equivalence of two assignments χ_1 and χ_2 *w.r.t.* a closed predicate p. First recall that, if both of them satisfy p, all prefixes of the induced plays π_1 and π_2

must belong to the witness V of p. Now, as for open predicates, consider their common prefix $\rho = \mathsf{prf}(\pi_1, \pi_2)$. It is obvious that, if all histories extending ρ belong to V, we can use ρ itself as a witness for the two memberships $\pi_1, \pi_2 \in \mathsf{pr}(p)$, since there is no way to violate the property described by p, once a play reaches ρ. In this case, we set χ_1 and χ_2 to be equivalent. Otherwise, the reason why both the assignments satisfy the predicate necessarily reside in the part the two plays differ, so, we set them as non-equivalent.

Definition 4.2.2 (Closed-Play Consistency)**.** *An equivalence relation on assignments $\equiv^{\cdot}_{\mathcal{E},s}$ is* closed-play consistent *if, for any closed predicate p with witness* $\mathrm{V} \subseteq \mathrm{Hst}$ *and p-coherent assignments* $\chi_1, \chi_2 \in \mathrm{Asg}$, *we have that if $\mathcal{E}, \chi_1, s \not\models p$ and $\mathcal{E}, \chi_2, s \not\models p$ then $\chi_1 \equiv^{p}_{\mathcal{E},s} \chi_2$, otherwise, $\chi_1 \equiv^{p}_{\mathcal{E},s} \chi_2$ iff, for each history $\rho \in$* Hst *with* $\mathsf{prf}(\mathsf{play}(\chi_{1\restriction \mathrm{Ag}}, s), \mathsf{play}(\chi_{2\restriction \mathrm{Ag}}, s)) \leq \rho$, *it holds that $\rho \in$* V.

Example 4.2.2. *Consider again the extension $\mathcal{E}$, the state $\mathtt{s}_0$, and the closed predicate $\mathtt{p}_V$. Moreover, let $\chi_1, \chi_2, \chi_3 \in$* $\mathrm{Asg}(\{\mathtt{a}, \mathtt{b}, \mathtt{c}\})$ *be three assignments on which we want to check the equivalence. In particular, assume that each χ_i associates a strategy σ_i^α with the agent $\alpha \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ as defined in the following: $\sigma^{\mathtt{a}}_{1/2/3}(\rho) \triangleq 0$, $\sigma^{\mathtt{a}}_{2/3}(\mathtt{s}_0\mathtt{s}_2) = \sigma^{\mathtt{a}}_{2/3}(\mathtt{s}_0\mathtt{s}_7) = \sigma^{\mathtt{b}}_{1/2}(\rho') = \sigma^{\mathtt{c}}_{1}(\rho') \triangleq 1$, and $\sigma^{\mathtt{a}}_{1}(\mathtt{s}_0\mathtt{s}_2) = \sigma^{\mathtt{a}}_{1}(\mathtt{s}_0\mathtt{s}_7) = \sigma^{\mathtt{b}}_{3}(\rho') = \sigma^{\mathtt{c}}_{2/3}(\rho') \triangleq 2$, for all $\rho \in \mathrm{Hst} \setminus \{\mathtt{s}_0\mathtt{s}_2, \mathtt{s}_0\mathtt{s}_7\}$ and $\rho' \in$* Hst. *Now, it holds that $\chi_1 \equiv^{\mathtt{p}_V}_{\mathcal{E},\mathtt{s}} \chi_2$ but $\chi_1 \not\equiv^{\mathtt{p}_V}_{\mathcal{E},\mathtt{s}} \chi_3$. Indeed, χ_1 and χ_2 induce exactly the same play $\pi_{12} = \mathtt{s}_0\mathtt{s}_2{\mathtt{s}_7}^\omega$. On the contrary, χ_3 induces the play $\pi_3 = \mathtt{s}_0{\mathtt{s}_7}^\omega$. Thus, the common history $\rho = \mathtt{s}_0 = \mathsf{prf}(\pi_{12}, \pi_3)$ has an extension that does not belong to* V.

4.3 Strategy Requirements

At this point, we can study the formalization of the equivalence relation for the strategy constructs. First, in Definition 4.3.1, we address the binding operator that results to be the most intuitive case, as it just involves a redefinition of the assignments under analysis. Then, in Definition 4.3.2 and 4.3.3, we move to the existential and universal strategy quantifiers that represent the core of our counting technique.

The semantics of a formula $\varphi = (a, x)\eta$ intuitively asserts that φ holds under an assignment χ once the inner part η is satisfied by associating the agent a to the strategy $\chi(x)$. Therefore, the equivalence of two assignments χ_1 and χ_2 *w.r.t.* φ necessarily depends on that of their extensions on a *w.r.t.* η.

Definition 4.3.1 (Binding Consistency)**.** *An equivalence relation on assignments* $\equiv^{\cdot}_{\mathcal{E},s}$ *is* binding consistent *if, for any given formula* $\varphi = (a, x)\eta$ *and* φ*-coherent assignments* $\chi_1, \chi_2 \in \mathrm{Asg}$*, we have that* $\chi_1 \equiv^{\varphi}_{\mathcal{E},s} \chi_2$ *iff* $\chi_1[a \mapsto \chi_1(x)] \equiv^{\eta}_{\mathcal{E},s} \chi_2[a \mapsto \chi_2(x)]$*.*

Example 4.3.1. *To gain insight on the above definition consider the formulas* $\varphi_1 = \flat_1 \mathtt{p}_U$ *and* $\varphi_2 = \flat_2 \mathtt{p}_U$*, where* $\flat_1 \triangleq (\mathtt{a}, \mathtt{x})(\mathtt{b}, \mathtt{y})(\mathtt{c}, \mathtt{z})$*,* $\flat_2 \triangleq (\mathtt{a}, \mathtt{x})(\mathtt{b}, \mathtt{y})(\mathtt{c}, \mathtt{x})$*, and* $\mathtt{p}_U$ *is the open predicate of the extension* $\mathcal{E}$*. Moreover, let* $\chi_1, \chi_2 \in \mathrm{Asg}(\{\mathtt{x}, \mathtt{y}, \mathtt{z}\})$ *be two assignments that associate, respectively, the strategies* σ^{α}_1 *and* σ^{α}_2 *to the variable* $\alpha \in \{\mathtt{x}, \mathtt{y}, \mathtt{z}\}$ *as defined in the following:* $\sigma^{1/2}_{\mathtt{x}}(\mathtt{s}_0) \triangleq 2$*,* $\sigma^{2}_{\mathtt{x}}(\rho) = \sigma^{1/2}_{\mathtt{y}}(\rho') \triangleq 0$*, and* $\sigma^{1}_{\mathtt{x}}(\rho) = \sigma^{1/2}_{\mathtt{z}}(\rho') \triangleq 1$*, for all* $\rho \in \mathrm{Hst} \setminus \{\mathtt{s}_0\}$ *and* $\rho' \in \mathrm{Hst}$*. Now, it is easy to see that* $\chi_1 \equiv^{\varphi_2}_{\mathcal{E},\mathtt{s}} \chi_2$ *but* $\chi_1 \not\equiv^{\varphi_1}_{\mathcal{E},\mathtt{s}} \chi_2$*. Indeed,* $\chi_1 \circ \flat_1$*,* $\chi_2 \circ \flat_1$*,* $\chi_1 \circ \flat_2$*, and* $\chi_2 \circ \flat_2$ *induce the plays* $\pi^1_1 = \mathtt{s}_0\mathtt{s}_3\mathtt{s}_1(\mathtt{s}_6\mathtt{s}_5)^{\omega}$*,* $\pi^1_2 = \mathtt{s}_0\mathtt{s}_3{\mathtt{s}_5}^{\omega}$*,* $\pi^2_1 = \mathtt{s}_0\mathtt{s}_3\mathtt{s}_1(\mathtt{s}_6\mathtt{s}_5)^{\omega}$*,* $\pi^2_2 = \mathtt{s}_0\mathtt{s}_3\mathtt{s}_1{\mathtt{s}_5}^{\omega}$*, respectively, where* $\rho_1 = \mathtt{s}_0\mathtt{s}_3 = \mathsf{prf}(\pi^1_1, \pi^1_2)$ *and* $\rho_2 = \mathtt{s}_0\mathtt{s}_3\mathtt{s}_1 = \mathsf{prf}(\pi^2_1, \pi^2_2)$ *are the corresponding common histories. Thus,* ρ_1 *belongs to the witness* U *of* $\mathtt{p}_U$*, while* ρ_2 *does not.*

Before proceeding with the analysis of the remaining logic constructs, it is important to make an observation about the dual nature of the existential and universal quantifiers *w.r.t.* the counting over strategies. We do this by exploiting the classic game-semantics metaphor originally proposed for first-order logic by Lorenzen and Hintikka, where the choice of an existential variable is done by a player called $\exists$ and that of the universal ones by its opponent $\forall$. Consider a formula $\langle\langle x_1 \geq g_1 \rangle\rangle [[x_2 < g_2]] \eta$, having $\langle\langle \mathtt{y}_1 \geq h_1 \rangle\rangle \eta_1$ and $[[\mathtt{y}_2 < h_2]] \eta_2$ as two subformulas. When player $\exists$ chooses the h_1 different strategies $\mathtt{y}_1$ to satisfy η_1, it has also to maximize the number of strategies x_1 verifying $[[x_2 < g_2]]\eta$ to be sure that the constraint $\geq g_1$ is not violated. At the same time, player $\forall$ tries to do exactly the opposite while choosing the h_2 different strategies y_2 not satisfying η_2, *i.e.*, to maximize the number of strategies x_2 falsifying η in order to violate the constraint $< g_2$.

With this observation in mind, we can now deal with the relation for the existential quantifier. Two assignments χ_1 and χ_2 are equivalent *w.r.t.* a formula $\varphi = \langle\langle x \geq g \rangle\rangle \eta$ if player $\exists$ is not able to find a strategy σ among those satisfying η, to associate with the variable x, that allows the corresponding extensions of χ_1 and χ_2 on x to induce different behaviors *w.r.t.* η.

Definition 4.3.2 (Existential Consistency)**.** *An equivalence relation on assignments $\equiv^{\cdot}_{\mathcal{E},s}$ is* existentially consistent *if, for any given formula $\varphi = \langle\langle x \geq g \rangle\rangle \eta$ and φ-coherent assignments $\chi_1, \chi_2 \in \mathrm{Asg}$, we have that $\chi_1 \equiv^{\varphi}_{\mathcal{E},s} \chi_2$ iff, for each strategy $\sigma \in \eta[\mathcal{E}, \chi_1, s](x) \cup \eta[\mathcal{E}, \chi_2, s](x)$, it holds that $\chi_1[x \mapsto \sigma] \equiv^{\eta}_{\mathcal{E},s} \chi_2[x \mapsto \sigma]$.*

Example 4.3.2. *Again, to see an application on the above, consider a formula $\varphi = \langle\langle \mathtt{z} \geq 2 \rangle\rangle \flat \mathrm{p}_U$, with $\flat = (\mathtt{a}, \mathtt{x})(\mathtt{b}, \mathtt{y})(\mathtt{c}, \mathtt{z})$, p_U is the open predicate of the extension $\mathcal{E}$. Moreover, let $\chi_1, \chi_2, \chi_3 \in \mathrm{Asg}(\{\mathtt{x}, \mathtt{y}\})$ be three assignments that associate, respectively, the strategies σ^{α}_{i} to the variable $\alpha \in \{\mathtt{x}, \mathtt{y}\}$ as defined in the following: $\sigma^{\mathtt{x}}_{1}(\rho) = \sigma^{\mathtt{x}}_{3}(\rho') = \sigma^{\mathtt{y}}_{1/2/3}(\mathrm{s}_0) \triangleq 0$, $\sigma^{\mathtt{x}}_{2}(\rho) = \sigma^{\mathtt{x}}_{3}(\mathrm{s}_0\mathrm{s}_3) \triangleq 1$, and $\sigma^{\mathtt{x}}_{i}(\mathrm{s}_0) \triangleq 2$, for all $\rho \in \mathrm{Hst}(\mathrm{s}_0) \setminus \{\mathrm{s}_0\}$ and $\rho' \in \mathrm{Hst}(\mathrm{s}_0) \setminus \{\mathrm{s}_0, \mathrm{s}_0\mathrm{s}_3\}$. Under these assignments we show that χ_2 and χ_3 are existential consistent while χ_1 and χ_2 are not.*

We start analyzing the existential consistency for χ_1 and χ_2. First observe that $\chi_1 \equiv^{\varphi}_{\mathcal{E},\mathrm{s}} \chi_2$ iff all strategies σ that satisfy $\flat \mathrm{p}_1$ on χ_1 or χ_2 are such that χ_1 and χ_2 restricted to σ are equivalent with respect to $\flat \mathrm{p}_1$. Moreover the set of these strategies for $\mathtt{z}$ is $\{\sigma^{\mathtt{z}}_{0}, \sigma^{\mathtt{z}}_{1}\}$, where $\sigma^{\mathtt{z}}_{0}(\rho) = 0$ and $\sigma^{\mathtt{z}}_{1}(\rho) = 1$, with $\rho \in \mathrm{Hst}(\mathrm{s}_0)$. First, let us analyze the equivalence relation under the strategy $\sigma^{\mathtt{z}}_{0}$ assigned to variable $\mathtt{z}$. We have that $\chi_1 \circ \flat$ and $\chi_2 \circ \flat$ induce the plays $\pi^{0}_{1} = \mathrm{s}_0\mathrm{s}_3\mathrm{s}_1\mathrm{s}_5^{\omega}$ and $\pi^{0}_{2} = \mathrm{s}_0\mathrm{s}_3(\mathrm{s}_5\mathrm{s}_6)^{\omega}$, respectively, where $\rho^{0}_{12} = \mathrm{s}_0\mathrm{s}_3 = \mathsf{prf}(\pi_1, \pi_2)$ is the corresponding common histories. Thus, ρ^{0}_{12} does not belong to the witness U of p_U. So, χ_1 and χ_2 are not existential consistent as required. It remains to analyze the existential consistency for χ_2 and χ_3. As before we want to check that all strategies σ that satisfy $\flat \mathrm{p}_1$ on χ_2 or χ_3 are such that χ_2 and χ_3 restricted to σ are equivalent with respect $\flat \mathrm{p}_1$. Moreover, the set of these strategies are as before. We start analyzing the equivalence relation under the strategy $\sigma^{\mathtt{z}}_{0}$ assigned to $\mathtt{z}$. We have that $\chi_3 \circ \flat$ induce the play $\pi^{0}_{3} = \mathrm{s}_0\mathrm{s}_3\mathrm{s}_5^{\omega}$, and $\rho^{0}_{23} = \mathrm{s}_0\mathrm{s}_3\mathrm{s}_5 = \mathsf{prf}(\pi_2, \pi_3)$ is the corresponding common histories. Thus, ρ^{0}_{23} belongs to the witness U of p_U. Now, we apply the same reasoning for σ_1. We get that $\chi_2 \circ \flat$ and $\chi_3 \circ \flat$ induce the plays $\pi^{1}_{2} = \mathrm{s}_0\mathrm{s}_3\mathrm{s}_1(\mathrm{s}_6\mathrm{s}_5)^{\omega}$ and $\pi^{1}_{3} = \mathrm{s}_0\mathrm{s}_3\mathrm{s}_1\mathrm{s}_5^{\omega}$, respectively, where $\rho^{1}_{23} = \mathrm{s}_0\mathrm{s}_3\mathrm{s}_1 = \mathsf{prf}(\pi_2, \pi_3)$ is the corresponding common histories. Thus, ρ^{1}_{23} belongs to the witness U of p_U. So, χ_2 and χ_3 are existential consistent as required and the ends the example.

We conclude by dealing with the relation for the universal quantifier. Two assignments χ_1 and χ_2 are equivalent *w.r.t.* a formula $\varphi = [[x < g]]\eta$ if the following holds: for each strategy σ_1 player $\forall$ chooses among those satisfying η under χ_1, there is a strategy σ_2 this player can choose among those satisfying η under χ_2 and *vice versa*, such that, once the two strategies are associated to the variable x, they make the corresponding extensions of χ_1 and χ_2 equivalent *w.r.t.* η. This means that the parts of the arena that are reachable when the two assignments are fixed contain exactly the same information *w.r.t.* the verification of the inner formula.

Definition 4.3.3 (Universal Consistency)**.** *An equivalence relation on assignments* $\equiv^{\cdot}_{\mathcal{E},s}$ *is* universally consistent *if, for any given formula* $\varphi = [\![x < g]\!]\eta$ *and* φ*-coherent assignments* $\chi_1, \chi_2 \in \mathrm{Asg}$*, we have that* $\chi_1 \equiv^{\varphi}_{\mathcal{E},s} \chi_2$ *iff, for each* $i \in \{1, 2\}$ *and strategy* $\sigma_i \in \eta[\mathcal{E}, \chi_i, s](x)$*, there is a strategy* $\sigma_{3-i} \in \eta[\mathcal{E}, \chi_{3-i}, s](x)$ *such that* $\chi_1[x \mapsto \sigma_1] \equiv^{\eta}_{\mathcal{E},s} \chi_2[x \mapsto \sigma_2]$.

Example 4.3.3. *To see an application on the above, consider formula* $\varphi = [\![\mathtt{y} < 2]\!]\flat\mathtt{p}_U$*, where* $\flat = (\mathtt{a}, \mathtt{x})(\mathtt{b}, \mathtt{y})(\mathtt{c}, \mathtt{x})$ *and* $\mathtt{p}_U$ *is the open predicate of the extension* $\mathcal{E}$*. Moreover, let* $\chi_1, \chi_2, \chi_3 \in \mathrm{Asg}(\{\mathtt{x}\})$ *be three assignments that associate, respectively, the strategies* σ^{α}_{i} *to the variable* $\alpha \in \{\mathtt{x}\}$ *as defined in the following:* $\sigma^{\mathtt{x}}_{1}(\rho') = \sigma^{\mathtt{x}}_{3}(\rho) \triangleq 0$, $\sigma^{\mathtt{x}}_{2}(\rho') \triangleq 1$, *and* $\sigma^{\mathtt{x}}_{1/2}(\mathtt{s}_0) \triangleq 2$*, for all* $\rho \in \mathrm{Hst}(\mathtt{s}_0)$*, and* $\rho' \in \mathrm{Hst}(\mathtt{s}_0) \setminus \{\mathtt{s}_0\}$. *Under these assignments we show that* χ_1 *and* χ_2 *are universal consistent while* χ_2 *and* χ_3 *are not.*

Let us start analyzing the universal consistency for χ_1 *and* χ_2*. Observe that,* $\chi_1 \equiv^{\varphi}_{\mathcal{E},\mathtt{s}} \chi_2$ *iff for every strategy that satisfies* $\flat\mathtt{p}_U$ *on* χ_1 *(resp.,* χ_2*) there is a strategy that satisfies* $\flat\mathtt{p}_U$ *on* χ_2 *(resp.,* χ_1*) and* χ_1 *and* χ_2 *respectively restricted to such strategies are equivalent over* $\flat\mathtt{p}_U$*. So, the set of strategy that satisfy* $\flat\mathtt{p}_U$ *is the same for* χ_1 *and* χ_2*, and it is composed by* $\sigma^{\mathtt{y}}_{0}$ *and* $\sigma^{\mathtt{y}}_{2}$*, where* $\sigma^{\mathtt{y}}_{0}(\mathtt{s}_0) = 0$ *and* $\sigma^{\mathtt{y}}_{2}(\mathtt{s}_0) = 2$*. Now, we verify the equivalence for* $\sigma^{\mathtt{y}}_{0}$*. We have that* $\chi_1 \circ \flat$ *and* $\chi_2 \circ \flat$ *induce the plays* $\pi^{0}_{1} = \mathtt{s}_0\mathtt{s}_3\mathtt{s}_1\mathtt{s}_5^{\omega}$*, and* $\pi^{0}_{2} = \mathtt{s}_0\mathtt{s}_3\mathtt{s}_1(\mathtt{s}_6\mathtt{s}_5)^{\omega}$*, respectively, where* $\rho^{0}_{12} = \mathtt{s}_0\mathtt{s}_3\mathtt{s}_1 = \mathsf{prf}(\pi^{0}_{1}, \pi^{0}_{2})$ *is the corresponding common histories. Thus, the common prefix belong to the witness* U *of* $\mathtt{p}_U$*. Now, let us analyze the statement with represent to* $\sigma^{\mathtt{y}}_{2}$*. We have that* $\chi_1 \circ \flat$ *and* $\chi_2 \circ \flat$ *induce the plays* $\pi^{2}_{1} = \mathtt{s}_0\mathtt{s}_5^{\omega}$ *, and* $\pi^{2}_{2} = \mathtt{s}_0(\mathtt{s}_5\mathtt{s}_6)^{\omega}$*, respectively, where* $\rho^{2}_{12} = \mathtt{s}_0\mathtt{s}_5 = \mathsf{prf}(\pi^{2}_{1}, \pi^{2}_{2})$ *is the corresponding common histories. Thus, the common prefix belong to the witness* U *of* $\mathtt{p}_U$*. In conclusion, we have that* χ_1 *and* χ_2 *are universal consistent. We now analyze the universal consistency regarding* χ_2 *and* χ_3*. By applying a similar reasoning as above, we have that the set of strategy that satisfy* $\flat\mathtt{p}_U$ *for* χ_2 *is the same of the above, while the set for* χ_3 *is composed by* $\sigma^{\mathtt{y}}_{0}$ *and* $\sigma^{\mathtt{y}}_{1}$*, where* $\sigma^{\mathtt{y}}_{0}(\mathtt{s}_0) = 0$ *and* $\sigma^{\mathtt{y}}_{1}(\mathtt{s}_0) = 1$*. We start with* $\sigma^{\mathtt{y}}_{0}$ *for both assignments. We have that* $\chi_3 \circ \flat$ *induce the play* $\pi^{0}_{3} = \mathtt{s}_0\mathtt{s}_1\mathtt{s}_5^{\omega}$*. So,* $\rho^{0}_{23} = \mathtt{s}_0 = \mathsf{prf}(\pi_2, \pi_3)$ *is the* *corresponding common histories. Thus, ρ^{0}_{23} does not belong to the witness* U *of* p_U*. Now, we verify with σ^{y}_{1} for χ_3. By applying the same semantics reasoning, we have that $\chi_3 \circ \flat$ induce the play $\pi^{1}_{3} = \mathrm{s}_0\mathrm{s}_2\mathrm{s}^{\omega}_{5}$. So, $\rho^{1}_{23} = \mathrm{s}_0 = \mathsf{prf}(\pi_2, \pi_3)$ is the corresponding common histories. Thus, ρ^{1}_{23} does not belong to the witness* U *of* p_U*. So χ_2 and χ_3 are not universal consistent. Directly from this we have that χ_1 and χ_3 are not universal consistent.*

4.4 Boolean Requirements

At this point, we can reason about properties that an equivalence has to satisfy *w.r.t.* the positive Boolean formulas. Specifically we give properties that must hold *w.r.t.* conjunction and disjunction.

Suppose we have two GSL formulas φ_1 and φ_2. A request property is that both numbers of assignments that are equivalent *w.r.t.* φ_1 and φ_2 are not less than those ones that are equivalent *w.r.t.* their conjunction. Hence, we need that assignments equivalent *w.r.t.* both φ_1 and φ_2 are equivalent *w.r.t.* $\varphi_1 \wedge \varphi_2$ too, otherwise, each equivalence class for φ_1 and φ_2 may provide more than one equivalence class for $\varphi_1 \wedge \varphi_2$ allowing the latter formula to have more assignments. Moreover, we would like that, among the assignments that are equivalent for φ_1 (resp., φ_2), the number of those equivalent for φ_2 (resp., φ_1) is equal to those for $\varphi_1 \wedge \varphi_2$. Hence, we need that assignments equivalent *w.r.t.* $\varphi_1 \wedge \varphi_2$ are also equivalent *w.r.t.* both φ_1 and φ_2.

Definition 4.4.1 (Conjunctive Consistency)**.** *An equivalence relation on assignments $\equiv^{\cdot}_{\mathcal{E},s}$ is* conjunctive consistent *if, for any given formula $\varphi = \varphi_1 \wedge \varphi_2$ and φ-coherent assignments $\chi_1, \chi_2 \in$* Asg*, we have that $\chi_1 \equiv^{\varphi_1 \wedge \varphi_2}_{\mathcal{E},s} \chi_2$ iff $\chi_1 \equiv^{\varphi_1}_{\mathcal{E},s} \chi_2$ and $\chi_1 \equiv^{\varphi_2}_{\mathcal{E},s} \chi_2$.*

Example 4.4.1. *Consider the extension $\mathcal{E}$ related to the arena $\mathcal{A}_{DG}$ depicted in Figure 4.1, the state $\mathtt{s}_0 \in$ St, and a formula $\varphi = \flat_1 \mathtt{p}_U \wedge \flat_2 \mathtt{p}_V$, with $\flat_1 = (\mathtt{a},\mathtt{y})(\mathtt{b},\mathtt{x})(\mathtt{c},\mathtt{z})$, $\flat_2 = (\mathtt{a},\mathtt{x})(\mathtt{b},\mathtt{z})(\mathtt{c},\mathtt{y})$. $\mathrm{Pr} = \{\mathtt{p}_U, \mathtt{p}_V\}$, the open predicate $\mathtt{p}_U$ has witness U containing those histories ending in $\mathtt{s}_2$ or $\mathtt{s}_4$, and the closed predicate $\mathtt{p}_V$ has witness V containing the histories ending in $\mathtt{s}_3$ or $\mathtt{s}_4$. Moreover, let $\chi_1, \chi_2, \chi_3 \in \mathrm{Asg}(\{\mathtt{x},\mathtt{y},\mathtt{z}\})$ be three assignments on which we want to check the equivalence. In particular, assume that each χ_i associates a strategy σ_i^{α} with the variable $\alpha \in \{\mathtt{x},\mathtt{y},\mathtt{z}\}$ as defined in the following: $\sigma^{\mathtt{x}}_{1/2/3}(\rho) = \sigma^{\mathtt{y}}_{3}(\rho) = \sigma^{\mathtt{z}}_{1/3}(\rho) \triangleq 0$, $\sigma^{\mathtt{y}}_{1/2}(\rho) = \sigma^{\mathtt{z}}_{2}(\rho) \triangleq 1$, for all $\rho \in$ Hst. Now, it holds that $\chi_1 \equiv^{\varphi}_{\mathcal{E},\mathtt{s}_0} \chi_2$ but $\chi_2 \not\equiv^{\varphi}_{\mathcal{E},\mathtt{s}_0} \chi_3$.*

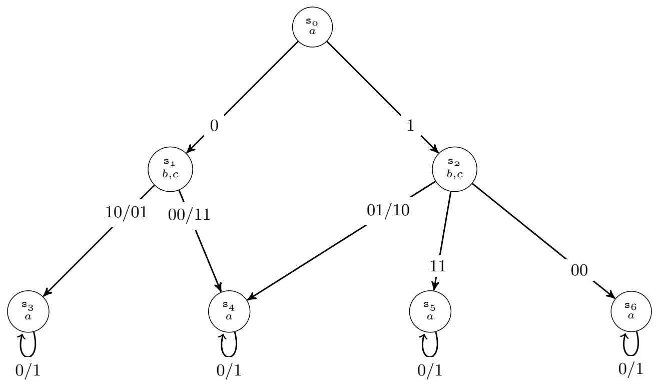

Figure 4.1: Arena $\mathcal{A}_{DG}$. Note that the node of the arena are labeled with its name (in the upper part) and with the subset of player that are active on its (in the lower part).

First observe that, $\chi_1 \equiv^{\varphi}_{\mathcal{E},\mathtt{s}} \chi_2$ iff $\chi_1 \equiv^{\flat_1 \mathtt{p}_U}_{\mathcal{E},\mathtt{s}} \chi_2 \wedge \chi_1 \equiv^{\flat_2 \mathtt{p}_V}_{\mathcal{E},\mathtt{s}} \chi_2$. Now, $\chi_1 \circ \flat_1$, $\chi_2 \circ \flat_1$, $\chi_1 \circ \flat_2$, and $\chi_2 \circ \flat_2$ induce the plays $\pi^1_1 = \mathtt{s}_0\mathtt{s}_2{\mathtt{s}_6}^{\omega}$, $\pi^1_2 = \mathtt{s}_0\mathtt{s}_2{\mathtt{s}_4}^{\omega}$, $\pi^2_1 = \mathtt{s}_0\mathtt{s}_1{\mathtt{s}_3}^{\omega}$, $\pi^2_2 = \mathtt{s}_0\mathtt{s}_1{\mathtt{s}_4}^{\omega}$, respectively, where $\rho_1 = \mathtt{s}_0\mathtt{s}_2 = \mathsf{prf}(\pi^1_1, \pi^1_2)$ and $\rho_2 = \mathtt{s}_0\mathtt{s}_1 = \mathsf{prf}(\pi^2_1, \pi^2_2)$ are the corresponding common histories. Thus, ρ_1 belongs to the witness U of $\mathtt{p}_U$, and ρ_2 has all extension that belong to the witness V of $\mathtt{p}_V$.

It remains to analyze the conjunctive consistency for χ_2 and χ_3. By applying the same semantics reasoning of the above, we have that $\chi_3 \circ \flat_1$ and $\chi_3 \circ \flat_2$ induce the plays $\pi^1_3 = \mathtt{s}_0\mathtt{s}_1{\mathtt{s}_6}^{\omega}$, $\pi^2_3 = \mathtt{s}_0\mathtt{s}_1{\mathtt{s}_4}^{\omega}$, respectively, where $\rho'_1 = \mathtt{s}_0 = \mathsf{prf}(\pi^1_2, \pi^1_3)$ and $\rho'_2 = \mathtt{s}_0\mathtt{s}_1 = \mathsf{prf}(\pi^2_2, \pi^2_3)$ are the corresponding common histories. Thus, ρ'_1 does not belong to the witness U of $\mathtt{p}_U$ then χ_2 and χ_3 are not conjunctive consistent.

Now, we discuss the equivalence relation *w.r.t.* the disjunction.

Definition 4.4.2 (Disjunctive Consistency). *An equivalence relation on assignments $\equiv^{\cdot}_{\mathcal{E},s}$ is* disjunctive consistent *if, for any given formula $\varphi = \varphi_1 \vee \varphi_2$ and φ-coherent assignments $\chi_1, \chi_2 \in$ Asg, we have that $\chi_1 \equiv^{\varphi_1 \vee \varphi_2}_{\mathcal{E},s} \chi_2$ iff the followings holds:*

1. *$\mathcal{E}, \chi_1, s \models \varphi_i$ iff $\mathcal{E}, \chi_2, s \models \varphi_i$* $\quad \forall i \in \{1, 2\}$

2. *If $\mathcal{E}, \chi_1, s \models \varphi_i$ then $\chi_1 \equiv^{\varphi_i}_{\mathcal{E},s} \chi_2$* $\quad \forall i \in \{1, 2\}$

Example 4.4.2. *Consider the extension $\mathcal{E}$ related to the arena $\mathcal{A}_{CG}$ depicted in Figure 4.2, the state $\mathtt{s}_0 \in$ St, and a formula $\varphi = \flat_1 \mathtt{p}_V \vee \flat_2 \mathtt{p}_U$, with $\flat_1 = (\mathtt{a}, \mathtt{x})(\mathtt{b}, \mathtt{y})(\mathtt{c}, \mathtt{z})$, $\flat_2 = (\mathtt{a}, \mathtt{y})(\mathtt{b}, \mathtt{x})(\mathtt{c}, \mathtt{z})$. Pr $= \{\mathtt{p}_U, \mathtt{p}_V\}$, the open predicate $\mathtt{p}_U$ has witness U containing those histories ending in $\mathtt{s}_1$, $\mathtt{s}_4$, or $\mathtt{s}_5$ and the closed predicate $\mathtt{p}_V$ has witness V containing the histories ending in $\mathtt{s}_1$. Moreover, let $\chi_1, \chi_2, \chi_3 \in$ Asg$(\{\mathtt{x}, \mathtt{y}, \mathtt{z}\})$ be three assignments on which we want to check the equivalence. In particular, assume that each χ_i associates a strategy σ^{α}_i with the variable $\alpha \in \{\mathtt{x}, \mathtt{y}, \mathtt{z}\}$ as defined in the following: $\sigma^{\mathtt{x}}_2(\rho') = \sigma^{\mathtt{y}}_3(\rho') = \sigma^{\mathtt{z}}_3(\rho) \triangleq 0$, $\sigma^{\mathtt{x}}_{1/3}(\rho) = \sigma^{\mathtt{x}}_2(\mathtt{s}_0) = \sigma^{\mathtt{y}}_{1/2}(\rho) = \sigma^{\mathtt{y}}_3(\mathtt{s}_0) = \sigma^{\mathtt{z}}_{1/2}(\rho) \triangleq 1$, for all $\rho \in$ Hst, $\rho' \in$ Hst $\setminus \{\mathtt{s}_0\}$. Under these assignments we show that χ_1 and χ_2 are disjunctive consistent while χ_2 and χ_3 are not. First observe that, $\chi_1 \equiv^{\varphi}_{\mathcal{E},\mathtt{s}} \chi_2$ iff the followings holds:*

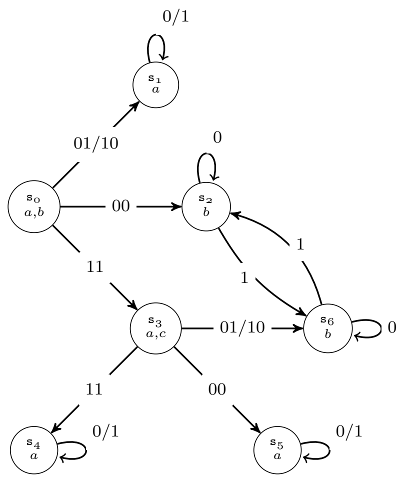

Figure 4.2: Arena $\mathcal{A}_{CG}$. Note that the node of the arena are labeled with its name (in the upper part) and with the subset of player that are active on its (in the lower part).

1. *$\mathcal{E}, \chi_1, \mathtt{s} \models \flat_i \mathtt{p}_i$ iff $\mathcal{E}, \chi_2, \mathtt{s} \models \flat_i \mathtt{p}_i$* $\quad \forall \mathtt{i} \in \{1, 2\}$;

2. *If $\mathcal{E}, \chi_1, \mathtt{s} \models \flat_i \mathtt{p}_i$ then $\chi_1 \equiv^{\flat_i \mathtt{p}_i}_{\mathcal{E},\mathtt{s}} \chi_2$* $\quad \forall \mathtt{i} \in \{1, 2\}$.

We have that, $\chi_1 \circ \flat_1$, $\chi_2 \circ \flat_1$, $\chi_1 \circ \flat_2$, and $\chi_2 \circ \flat_2$ induce the plays $\pi_1^1 = \mathrm{s}_0\mathrm{s}_3{\mathrm{s}_4}^\omega$, $\pi_2^1 = \mathrm{s}_0\mathrm{s}_3{\mathrm{s}_6}^\omega$, $\pi_1^2 = \mathrm{s}_0\mathrm{s}_3{\mathrm{s}_4}^\omega$, $\pi_2^2 = \mathrm{s}_0\mathrm{s}_3{\mathrm{s}_4}^\omega$, respectively, where $\rho_1 = \mathrm{s}_0\mathrm{s}_3 = \mathsf{prf}(\pi_1^1, \pi_2^1)$ and $\rho_2 = \mathrm{s}_0\mathrm{s}_3\mathrm{s}_4 = \mathsf{prf}(\pi_1^2, \pi_2^2)$ are the corresponding common histories. Now, π_1^1 and π_2^1 are not belongs to $\mathsf{pr}(\mathrm{p}_V)$ whereas π_1^2 and π_2^2 are belongs to $\mathsf{pr}(\mathrm{p}_U)$, also ρ_2 belongs to the witness U *of* p_U. *Therefore, χ_1 and χ_2 are disjunctive consistent as required.*

Now, we analyze the disjunctive consistency for χ_2 and χ_3. By applying the same semantics reasoning of the above, we have that $\chi_3 \circ \flat_1$ and $\chi_3 \circ \flat_2$ induce the plays $\pi_3^1 = \mathrm{s}_0\mathrm{s}_3{\mathrm{s}_6}^\omega$, $\pi_3^2 = \mathrm{s}_0\mathrm{s}_3{\mathrm{s}_5}^\omega$, respectively, where $\rho_1' = \mathrm{s}_0\mathrm{s}_3{\mathrm{s}_6}^\omega = \mathsf{prf}(\pi_2^1, \pi_3^1)$ and $\rho_2' = \mathrm{s}_0\mathrm{s}_3 = \mathsf{prf}(\pi_2^2, \pi_3^2)$ are the corresponding common histories. Now, π_2^1 and π_3^1 are not belongs to $\mathsf{pr}(\mathrm{p}_V)$ whereas π_2^2 and π_3^2 are belongs to $\mathsf{pr}(\mathrm{p}_U)$, but ρ_2' is not belong to the witness U *of* p_U. *Therefore, χ_2 and χ_3 are not disjunctive consistent as required and this concludes the example.*

4.5 Example

In this Section we illustrate four examples of the semantics application, and consequently of equivalence relation, on composed formulas, so as to analyze the interactions. In Subsection 4.5.1 we analyze the model in the Figure 3.1 with two formulas in GSL[1G] and comparing the obtained results. In Subsection 4.5.2 we use the model in Figure 4.2 to analyze a formula in GSL[CG]. In Subsection 4.5.3 we give an example of a formula in GSL[DG] on the model in the Figure 4.1. For clarity, in this last example we show two tables that contain a terminal state in each cell, there will be a cell for each possible combination of strategies on the players involved. Finally, in Subsection 4.5.4 we give a concrete example *w.r.t.* the scheduling problem.

4.5.1 Example in GSL[1G]

To make practice with the equivalence for the strategy constructs, consider again the arena $\mathcal{A}$ depicted in Figure 3.1 and the associated extension $\mathcal{E}$ having the open predicate p_U. Moreover, let $\varphi = \wp\flat\mathrm{p}_U$ be a GSL[1G] formula with $\wp = \langle\langle \mathrm{x} \geq g_1 \rangle\rangle [[\mathrm{y} < g_2]] \langle\langle \mathrm{z} \geq g_3 \rangle\rangle$ and $\flat = (\mathrm{a}, \mathrm{x})(\mathrm{b}, \mathrm{y})(\mathrm{c}, \mathrm{z})$. Then, it can be showed that $\mathcal{E}, \mathrm{s}_0 \models \varphi$ with $(g_1, g_2, g_3) = (6, 3, 2)$, while $\mathcal{E}, \mathrm{s}_0 \not\models \varphi$ with $(g_1, g_2, g_3) = (1, 2, 2)$. Thus, there exist six strategies σ_i^{x} not equivalent associated to the variable

$\mathtt{x}$, where $\sigma_0^{\mathtt{x}}(\rho) = \sigma_3^{\mathtt{x}}(\mathtt{s}_0) = \sigma_4^{\mathtt{x}}(\rho') = \sigma_5^{\mathtt{x}}(\rho') \triangleq 0$, $\sigma_1^{\mathtt{x}}(\rho) = \sigma_2^{\mathtt{x}}(\rho') = \sigma_3^{\mathtt{x}}(\rho') = \sigma_4^{\mathtt{x}}(\mathtt{s}_0) \triangleq 1$, and $\sigma_2^{\mathtt{x}}(\mathtt{s}_0) = \sigma_5^{\mathtt{x}}(\mathtt{s}_0) \triangleq 2$ for all $\rho \in \mathrm{Hst}(\mathtt{s}_0)$ and $\rho' \in \mathrm{Hst}(\mathtt{s}_0) \setminus \{\mathtt{s}_0\}$. The variable $\mathtt{y}$ has three strategies $\sigma_j^{\mathtt{y}}$ not equivalent where $\sigma_0^{\mathtt{y}}(\rho) = 0$, $\sigma_1^{\mathtt{y}}(\rho) = 1$, and $\sigma_2^{\mathtt{y}}(\rho) = 2$, for all $\rho \in \mathrm{Hst}(\mathtt{s}_0)$. Finally, the variable $\mathtt{z}$ has two strategies $\sigma_k^{\mathtt{z}}$ not equivalent, where $\sigma_0^{\mathtt{z}}(\rho) = 0$ and $\sigma_1^{\mathtt{z}}(\rho) = 1$, for all $\rho \in \mathrm{Hst}(\mathtt{s}_0)$. It is not hard to see that the above strategies, once assigned to the variable x, result to be (along their corresponding assignment) not-equivalent among them.

Below we illustrate the strategy equivalence on assignments $\sigma_2^{\mathtt{x}}$ and $\sigma_5^{\mathtt{x}}$. Let $\varphi' = \langle\langle \mathtt{z} \geq 2 \rangle\rangle (\mathtt{a}, \mathtt{x})(\mathtt{b}, \mathtt{y})(\mathtt{c}, \mathtt{z})\mathtt{p}$. We have that $\chi_1[\mathtt{x} \mapsto \sigma_2^{\mathtt{x}}] \equiv_{\mathcal{E},\mathtt{s}_0}^{[\![\mathtt{y}<3]\!]\varphi'} \chi_2[\mathtt{x} \mapsto \sigma_5^{\mathtt{x}}]$ iff for every strategy that satisfies φ' on χ_1 (resp., χ_2) there is a strategy that satisfies φ' on χ_2 (resp., χ_1) and χ_1 and χ_2 respectively restricted to such strategies are equivalent over φ'. So, the set of strategy that satisfy φ' is the same for χ_1 and χ_2, and it is composed by $\sigma_0^{\mathtt{y}}$ and $\sigma_2^{\mathtt{y}}$. Now, we verify the equivalence for $\sigma_0^{\mathtt{y}}$ on χ_1 and χ_2. Thus, for the existential consistency we have that $\chi_1 \equiv_{\mathcal{E},\mathtt{s}}^{\varphi'} \chi_2$ iff all strategies σ that satisfy $\flat \mathtt{p}_U$ on χ_1 or χ_2 are such that χ_1 and χ_2 restricted to σ are equivalent with respect to $\flat \mathtt{p}_1$. Moreover the set of these strategies for $\mathtt{z}$ is $\{\sigma_0^{\mathtt{z}}, \sigma_1^{\mathtt{z}}\}$. For the strategy $\sigma_0^{\mathtt{z}}$ of $\mathtt{z}$, the assignments are not equivalent. In fact, we have two different paths that are $\mathtt{s}_0\mathtt{s}_3\mathtt{s}_1{\mathtt{s}_5}^{\omega}$ and $\mathtt{s}_0\mathtt{s}_3(\mathtt{s}_5\mathtt{s}_6)^{\omega}$, respectively. Now, we verify the equivalence for $\sigma_0^{\mathtt{y}}$ on χ_1 and $\sigma_2^{\mathtt{y}}$ on χ_2. Thus, the set of strategies for $\mathtt{z}$ is that above. For the strategy $\sigma_0^{\mathtt{z}}$ of $\mathtt{z}$, the assignments are not equivalent. In fact, we have two different paths that are $\mathtt{s}_0\mathtt{s}_3\mathtt{s}_1{\mathtt{s}_5}^{\omega}$ and $\mathtt{s}_0(\mathtt{s}_5\mathtt{s}_6)^{\omega}$, respectively. So, $\sigma_2^{\mathtt{x}}$ and $\sigma_5^{\mathtt{x}}$ are not equivalent.

Differently, the formula φ it is not satisfied with $g_1 = 3$, $g_2 = 2$, and $g_3 = 2$, since there exist three strategies for $\mathtt{x}$ but, excluding a strategy for $\mathtt{y}$, there are not two strategies for $\mathtt{z}$.

Now, we analyze the formula $\psi = \wp' \flat' \mathtt{p}_U$ where $\wp' = \langle\langle \mathtt{x} \geq g_4 \rangle\rangle [\![\mathtt{y} < g_5]\!]$ and $\flat' = (\mathtt{a}, \mathtt{x})(\mathtt{b}, \mathtt{y})(\mathtt{c}, \mathtt{x})$. We have that ψ is satisfied with $g_4 = 3$ and $g_5 = 2$ but it is not satisfied for $g_4 = 6$ and $g_5 = 3$, since the agents $\mathtt{a}$ and $\mathtt{c}$ accomplish the same strategy as they are binding at the same variable $\mathtt{x}$ and then there are not six strategies that one not equivalent for the agent $\mathtt{a}$. The checking of semantics proceeds as above.

4.5.2 Example in $\mathrm{GSL}[\mathrm{CG}]$

Consider the arena $\mathcal{A}_{CG}$ depicted in Figure 4.2 associated to the extension $\mathcal{E} \triangleq \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$ where there are an open predicate p_U and a closed predicate p_V. $\mathrm{Pr} = \{\mathrm{p}_U, \mathrm{p}_V\}$, the predicate

p_U has witness U containing those histories ending in s_2, s_4, s_5, or s_6 and the predicate p_V has witness V containing the histories ending in s_2 or s_6. Moreover, let $\varphi = \wp(\flat_1 \mathrm{p}_U \wedge \flat_2 \mathrm{p}_V)$ be a GSL[CG] formula, where $\wp = \langle\langle \mathtt{x} \geq g_1 \rangle\rangle [[\mathtt{y} < g_2]][[\mathtt{z} < g_3]]$, $\flat_1 = (\mathtt{a}, \mathtt{x})(\mathtt{b}, \mathtt{y})(\mathtt{c}, \mathtt{z})$, and $\flat_2 = (\mathtt{a}, \mathtt{y})(\mathtt{b}, \mathtt{x})(\mathtt{c}, \mathtt{z})$. Thus, there exist three strategies $\sigma_i^{\mathtt{x}}$ not equivalent associated to the variable x, where $\sigma_0^{\mathtt{x}}(\rho) = \sigma_2^{\mathtt{x}}(\rho') \triangleq 0$, $\sigma_1^{\mathtt{x}}(\rho) = \sigma_2^{\mathtt{x}}(\mathrm{s}_0) \triangleq 1$, for all $\rho \in \mathrm{Hst}(\mathrm{s}_0)$ and $\rho' \in \mathrm{Hst}(\mathrm{s}_0) \setminus \{\mathrm{s}_0\}$. The variable y has three strategies $\sigma_j^{\mathtt{y}}$ where $\sigma_0^{\mathtt{y}}(\rho) = \sigma_2^{\mathtt{y}}(\rho') \triangleq 0$, $\sigma_1^{\mathtt{y}}(\rho) = \sigma_2^{\mathtt{y}}(\mathrm{s}_0) \triangleq 1$, for all $\rho \in \mathrm{Hst}(\mathrm{s}_0)$ and $\rho' \in \mathrm{Hst}(\mathrm{s}_0) \setminus \{\mathrm{s}_0\}$. Finally, the variable z has two strategies $\sigma_k^{\mathtt{z}}$ not equivalent, where $\sigma_0^{\mathtt{z}}(\rho) \triangleq 0$ and $\sigma_1^{\mathtt{z}}(\rho) \triangleq 1$, for all $\rho \in \mathrm{Hst}(\mathrm{s}_0)$. It is not hard to see that the above strategies, once assigned to the variable x, result to be (along their corresponding assignment) not-equivalent among them. Now, we evaluate the equivalences on σ_1^x and σ_2^x associated to χ_1 and χ_2, respectively. First, we used, for simplicity, a three new formulas φ', φ'', and φ''' that represent $[[\mathtt{y} < g_2]][[\mathtt{z} < g_3]]\flat_1 \mathrm{p}_U \wedge \flat_2 \mathrm{p}_V$, $[[\mathtt{z} < g_3]]\flat_1 \mathrm{p}_U \wedge \flat_2 \mathrm{p}_V$, and $\flat_1 \mathrm{p}_U \wedge \flat_2 \mathrm{p}_V$, respectively. We have that, $\chi_1 \equiv_{\mathcal{E},\mathrm{s}_0}^{\varphi'} \chi_2$ iff for every strategy that satisfies φ'' on χ_1 (resp., χ_2) there is a strategy that satisfies φ'' on χ_2 (resp., χ_1) and χ_1 and χ_2 respectively restricted to such strategies are equivalent over φ''. So, the set of strategy that satisfy φ'' is the same for χ_1 and χ_2, and it is composed by $\sigma_1^{\mathtt{y}}$ and $\sigma_2^{\mathtt{y}}$. Now, we verify the equivalence for $\sigma_2^{\mathtt{y}}$ for both assignments. $\chi_1 \equiv_{\mathcal{E},\mathrm{s}}^{\varphi''} \chi_2$ iff for every strategy that satisfies φ''' on χ_1 (resp., χ_2) there is a strategy that satisfies φ''' on χ_2 (resp., χ_1) and χ_1 and χ_2 respectively restricted to such strategies are equivalent over φ''. So, the set of strategy that satisfy φ''' is the same for χ_1 and χ_2, and it is composed by $\sigma_1^{\mathtt{z}}$. For the strategy $\sigma_1^{\mathtt{z}}$, we have that $\chi_1 \equiv_{\mathcal{E},\mathrm{s}}^{\varphi'''} \chi_2$ iff $\chi_1 \equiv_{\mathcal{E},\mathrm{s}}^{\flat_1 \mathrm{p}_U} \chi_2 \wedge \chi_1 \equiv_{\mathcal{E},\mathrm{s}}^{\flat_2 \mathrm{p}_V} \chi_2'$. We have that $\chi_1 \circ \flat_1$ and $\chi_2 \circ \flat_1$ induce the plays $\pi_1 = \mathrm{s}_0\mathrm{s}_3\mathrm{s}_4{}^{\omega}$, and $\pi_2 = \mathrm{s}_0\mathrm{s}_3\mathrm{s}_6{}^{\omega}$, respectively, where $\rho_{12} = \mathrm{s}_0\mathrm{s}_3 = \mathsf{prf}(\pi_1, \pi_2)$ is the corresponding common histories. Hence, the common prefix $\mathrm{s}_0\mathrm{s}_3$ are not belongs to $\mathrm{U}_{\mathrm{p}_U}$. Now, we verify the equivalence for $\sigma_2^{\mathtt{y}}$ to χ_1 and $\sigma_1^{\mathtt{y}}$ to χ_2. $\chi_1 \equiv_{\mathcal{E},\mathrm{s}}^{\varphi''} \chi_2$ iff for every strategy that satisfies φ''' on χ_1 (resp., χ_2) there is a strategy that satisfies φ''' on χ_2 (resp., χ_1) and χ_1 and χ_2 respectively restricted to such strategies are equivalent over φ''. So, the set of strategy that satisfy φ''' is composed by $\sigma_0^{\mathtt{z}}$ for χ_2. We have that $\chi_1 \equiv_{\mathcal{E},\mathrm{s}}^{\varphi'''} \chi_2$ iff $\chi_1 \equiv_{\mathcal{E},\mathrm{s}}^{\flat_1 \mathrm{p}_U} \chi_2 \wedge \chi_1 \equiv_{\mathcal{E},\mathrm{s}}^{\flat_2 \mathrm{p}_V} \chi_2'$. We have that $\chi_2 \circ \flat_1$ induce the play $\pi_2 = \mathrm{s}_0\mathrm{s}_3\mathrm{s}_5{}^{\omega}$, where $\rho_{12} = \mathrm{s}_0\mathrm{s}_3 = \mathsf{prf}(\pi_1, \pi_2)$ is the corresponding common histories. Hence, the common prefix $\mathrm{s}_0\mathrm{s}_3$ are not belongs to $\mathrm{U}_{\mathrm{p}_U}$. So, χ_1 and χ_2 are not equivalent. Fixed a strategy σ_0^x for x, we evaluate the equivalences on σ_1^y and σ_2^y associated to χ_1

and χ_2, respectively. First, we used, for simplicity, a two new formulas φ' and φ'' that represent $\langle\langle \mathtt{z} \geq g_3 \rangle\rangle \flat_1 \neg \mathrm{p}_U \vee \flat_2 \neg \mathrm{p}_V$, $\flat_1 \neg \mathrm{p}_U \vee \flat_2 \neg \mathrm{p}_V$, respectively. We have that $\chi_1 \equiv^{\varphi'}_{\mathcal{E},\mathrm{s}_0} \chi_2$ iff all strategies σ that satisfy φ'' on χ_1 or χ_2 are such that χ_1 and χ_2 restricted to σ are equivalent with respect to φ''. Since the strategy for z is irrelevant for the two assignments is easy to see that χ_1 and χ_2 are equivalent. Then, it can be showed that $\mathcal{E}, \mathrm{s}_0 \models \varphi$ with $(g_1, g_2, g_3) = (3, 2, 2)$, while $\mathcal{E}, \mathrm{s}_0 \not\models \varphi$ with $(g_1, g_2, g_3) = (1, 1, 1)$.

4.5.3 Example in GSL[DG]

Consider the arena $\mathcal{A}_{DG}$ depicted in Figure 4.1 associated to the extension $\mathcal{E} \triangleq \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$ where there is a closed predicate p_V and an open predicate p_U. Moreover, let $\varphi = \wp(\flat_1 \mathrm{p}_V \vee \flat_2 \mathrm{p}_U)$ be a disjunctive GSL formula, where $\wp = [[\mathtt{x} < g_1]] \langle\langle \mathtt{y} \geq g_2 \rangle\rangle [[\mathtt{z} < g_3]]$, $\flat_1 = (\mathtt{a}, \mathtt{y})(\mathtt{b}, \mathtt{x})(\mathtt{c}, \mathtt{z})$, and $\flat_2 = (\mathtt{a}, \mathtt{x})(\mathtt{b}, \mathtt{z})(\mathtt{c}, \mathtt{y})$. $\mathrm{Pr} = \{\mathrm{p}_U, \mathrm{p}_V\}$, the predicate p_U has witness U containing those histories ending in s_5, or s_6 and the predicate p_V has witness V containing the histories ending in s_3. Thus, there exist eight strategies $\sigma^{\mathtt{x}}_i$ not equivalent associated to the variable x, where $\sigma^{\mathtt{x}}_0(\rho) = \sigma^{\mathtt{x}}_1(\rho'') = \sigma^{\mathtt{x}}_2(\rho''') = \sigma^{\mathtt{x}}_3(\mathrm{s}_0) = \sigma^{\mathtt{x}}_4(\rho') = \sigma^{\mathtt{x}}_5(\mathrm{s}_0\mathrm{s}_1) = \sigma^{\mathtt{x}}_6(\mathrm{s}_0\mathrm{s}_2) \triangleq 0$, $\sigma^{\mathtt{x}}_1(\mathrm{s}_0\mathrm{s}_2) = \sigma^{\mathtt{x}}_2(\mathrm{s}_0\mathrm{s}_1) = \sigma^{\mathtt{x}}_3(\rho') = \sigma^{\mathtt{x}}_4(\mathrm{s}_0) = \sigma^{\mathtt{x}}_5(\rho''') = \sigma^{\mathtt{x}}_6(\rho'') = \sigma^{\mathtt{x}}_7(\rho) \triangleq 1$, for all $\rho \in \mathrm{Hst}(\mathrm{s}_0)$, $\rho' \in \mathrm{Hst}(\mathrm{s}_0) \setminus \{\mathrm{s}_0\}$, $\rho'' \in \mathrm{Hst}(\mathrm{s}_0) \setminus \{\mathrm{s}_0\mathrm{s}_2\}$, and $\rho''' \in \mathrm{Hst}(\mathrm{s}_0) \setminus \{\mathrm{s}_0\mathrm{s}_1\}$. The variable y has four strategies $\sigma^{\mathtt{y}}_j$ where $\sigma^{\mathtt{y}}_0(\rho) = \sigma^{\mathtt{y}}_1(\mathrm{s}_0) = \sigma^{\mathtt{y}}_2(\rho') \triangleq 0$, $\sigma^{\mathtt{y}}_1(\rho') = \sigma^{\mathtt{y}}_2(\mathrm{s}_0) = \sigma^{\mathtt{y}}_3(\rho) \triangleq 1$ for all $\rho \in \mathrm{Hst}(\mathrm{s}_0)$ and $\rho' \in \mathrm{Hst}(\mathrm{s}_0) \setminus \{\mathrm{s}_0\}$. Finally, the variable z has two strategies $\sigma^{\mathtt{z}}_k$ not equivalent, where $\sigma^{\mathtt{z}}_0(\rho) \triangleq 0$ and $\sigma^{\mathtt{z}}_1(\rho) \triangleq 1$, for all $\rho \in \mathrm{Hst}(\mathrm{s}_0)$. It is not hard to see that the above strategies, once assigned to the variable x, result to be (along their corresponding assignment) not-equivalent among them.

In the Table 4.1 and Table 4.2 we find the last state of each play induced by chosen strategies from three agents. Now, if x chose one of the strategies σ^x_0, σ^x_1, σ^x_2, or σ^x_3, for satisfy the formula y must choose or σ^y_0 or σ^y_1, whereas z must choose σ^z_0 for σ^x_2 or σ^x_3 while must choose σ^z_1 for σ^x_0 or σ^x_1. Whereas, if x choose one of the strategies σ^x_4, σ^x_5, σ^x_6, or σ^x_7 for satisfy the formula y may choose any strategies, whereas z must choose σ^z_0 or σ^z_1 with respect to the choice of y. Based on the above, we have that the formula φ is satisfied with $(g_1, g_2, g_3) = (1, 1, 2)$, $(g_1, g_2, g_3) = (3, 1, 1)$, and $(g_1, g_2, g_3) = (2, 4, 2)$, which we will call φ_1, φ_2, and φ_3, respectively. We show this in the following. For φ_1 is trivial, just look at the table. In order to verify φ_2 we must prove that the

		σ_x^0	σ_x^1	σ_x^2	σ_x^3	σ_x^4	σ_x^5	σ_x^6	σ_x^7
σ_y^0	σ_z^0	s_4	s_4	s_3	s_3	s_4	s_4	s_3	s_3
σ_y^1	σ_z^0	s_4	s_4	s_3	s_3	s_4	s_4	s_3	s_3
σ_y^2	σ_z^0	s_6	s_4	s_6	s_4	s_6	s_4	s_6	s_4
σ_y^3	σ_z^0	s_6	s_4	s_6	s_4	s_6	s_4	s_6	s_4
σ_y^0	σ_z^1	s_3	s_3	s_4	s_4	s_3	s_3	s_4	s_4
σ_y^1	σ_z^1	s_3	s_3	s_4	s_4	s_3	s_3	s_4	s_4
σ_y^2	σ_z^1	s_4	s_5	s_4	s_5	s_4	s_5	s_4	s_5
σ_y^3	σ_z^1	s_4	s_5	s_4	s_5	s_4	s_5	s_4	s_5

Table 4.1: Table of the assignments with binding $\flat_1$

		σ_x^0	σ_x^1	σ_x^2	σ_x^3	σ_x^4	σ_x^5	σ_x^6	σ_x^7
σ_y^0	σ_z^0	s_4	s_4	s_4	s_4	s_6	s_6	s_6	s_6
σ_y^1	σ_z^0	s_3	s_3	s_3	s_3	s_4	s_4	s_4	s_4
σ_y^2	σ_z^0	s_4	s_4	s_4	s_4	s_6	s_6	s_6	s_6
σ_y^3	σ_z^0	s_3	s_3	s_3	s_3	s_4	s_4	s_4	s_4
σ_y^0	σ_z^1	s_3	s_3	s_3	s_3	s_4	s_4	s_4	s_4
σ_y^1	σ_z^1	s_4	s_4	s_4	s_4	s_5	s_5	s_5	s_5
σ_y^2	σ_z^1	s_3	s_3	s_3	s_3	s_4	s_4	s_4	s_4
σ_y^3	σ_z^1	s_4	s_4	s_4	s_4	s_5	s_5	s_5	s_5

Table 4.2: Table of the assignments with binding $\flat_2$

strategies σ_0^x, σ_1^x, σ_2^x, σ_3^x assigned to χ_1, χ_2, χ_3, and χ_4, respectively, are pairwise not equivalent.

First, we used, for simplicity, a three new formulas φ', φ'', and φ''' that represent $[[\mathtt{y} < 1]]\langle\langle \mathtt{z} \geq 1\rangle\rangle \flat_1 \neg \mathtt{p}_V \wedge \flat_2 \neg \mathtt{p}_U$, $\langle\langle \mathtt{z} \geq 1\rangle\rangle \flat_1 \neg \mathtt{p}_V \wedge \flat_2 \neg \mathtt{p}_U$, and $\flat_1 \neg \mathtt{p}_V \wedge \flat_2 \neg \mathtt{p}_U$, respectively. For generic $\mathtt{i}$ and $\mathtt{j}$, we have that $\chi_i \equiv^{\varphi'}_{\mathcal{E},\mathtt{s}_0} \chi_j$ iff for every strategy that satisfies φ'' on χ_i (resp., χ_j) there is a strategy that satisfies φ' on χ_j (resp., χ_i) and χ_i and χ_j respectively restricted to such strategies are equivalent over φ''. Now, we prove that (for strategy that satisfy the above restriction) $\chi_i \equiv^{\varphi''}_{\mathcal{E},\mathtt{s}_0} \chi_j$ iff all strategies σ that satisfy φ''' on χ_1 or χ_2 are such that χ_1 and χ_2 restricted to σ are equivalent with respect to φ'''. Finally, by conjunctive consistency we have that $\chi_i \equiv^{\varphi'''}_{\mathcal{E},\mathtt{s}_0} \chi_j$ iff $\chi_i \equiv^{\flat_1 \neg \mathtt{p}_V}_{\mathcal{E},\mathtt{s}_0} \chi_j$ and $\chi_i \equiv^{\flat_2 \neg \mathtt{p}_U}_{\mathcal{E},\mathtt{s}_0} \chi_j$ From the tables it is easy to note that χ_1 and χ_2 are equivalent and χ_3 and χ_4 are equivalent, but χ_1 and χ_3 are not equivalent. Now, we show that χ_1 and χ_3 are not equivalent. By apply the equivalence relation on universal quantifier, we have that the set of strategies for variable y that verify the sub-formula φ'' is $\{\sigma_0^y, \sigma_1^y, \sigma_2^y, \sigma_3^y\}$ for both assignments. For χ_1 restricted for the strategy σ_0^y there are not exist the strategy σ_k^y on χ_3, that satisfy φ', such that $\chi_1 \equiv_{\mathcal{E},\mathtt{s}_0}^{\varphi'} \chi_3$. In fact, for χ_3 restricted on σ_0^y we have that $\chi_1 \equiv_{\mathcal{E},\mathtt{s}_0}^{\varphi'} \chi_3$ iff all strategies σ_k^z that satisfy φ''' on χ_1 or χ_3 are such that χ_1 and χ_3 restricted to σ_k^z are equivalent with respect to φ'''. For $\mathtt{z} = \sigma_0^z$ we have that, $\chi_1 \circ \flat_1$, $\chi_2 \circ \flat_1$, $\chi_1 \circ \flat_2$, and $\chi_2 \circ \flat_2$ induce the plays $\pi_1^1 = \mathtt{s}_0\mathtt{s}_1\mathtt{s}_4{}^\omega$, $\pi_2^1 = \mathtt{s}_0\mathtt{s}_1\mathtt{s}_3{}^\omega$, $\pi_1^2 = \mathtt{s}_0\mathtt{s}_1\mathtt{s}_4{}^\omega$, $\pi_2^2 = \mathtt{s}_0\mathtt{s}_1\mathtt{s}_4{}^\omega$, respectively, where $\rho_1 = \mathtt{s}_0\mathtt{s}_1 = \mathsf{prf}(\pi_1^1, \pi_2^1)$ and $\rho_2 = \mathtt{s}_0\mathtt{s}_1\mathtt{s}_4{}^\omega = \mathsf{prf}(\pi_1^2, \pi_2^2)$ are the corresponding common histories. Thus, ρ_2 belongs to the witness U of $\neg\mathtt{p}_U$, while ρ_1 does not belongs to the witness V of $\neg\mathtt{p}_V$. Similar reasoning is carried out to other strategy.

In order to verify φ_3 we must prove that the strategies σ_0^x, σ_1^x, σ_2^x, σ_3^x assigned to χ_1, χ_2, χ_3, and χ_4, respectively, are pairwise not equivalent. From the tables it is easy to note that the four assignments are equivalent. Now, we show that χ_1 and χ_3 are equivalent. By apply the equivalence relation on universal quantifier, we have that the set of strategies for variable y that verify the sub-formula φ'' are $\{\sigma_2^y, \sigma_3^y\}$ for both assignments. From the tables it is easy to note that χ_1 and χ_3 are equivalent. In fact, these strategies induce the same plays! At this point, in order to verify φ_3 given a strategy for x, between those not excluded (for example σ_4^x), we must prove that the strategies σ_0^y, σ_1^y, σ_2^y, σ_3^y assigned to χ_1, χ_2, χ_3, and χ_4, respectively, are pairwise not equivalent. First, we used, for simplicity, a two new formulas φ' and φ'' that represent $[\![\mathtt{z} < 2]\!]\flat_1\mathtt{p}_V \vee \flat_2\mathtt{p}_U$, $\flat_1\mathtt{p}_V \vee \flat_2\mathtt{p}_U$, respectively. For generic i and j, we have that $\chi_i \equiv_{\mathcal{E},\mathtt{s}_0}^{\varphi'} \chi_j$ iff for every strategy that satisfies φ'' on χ_i (resp., χ_j) there is a strategy that satisfies φ'' on χ_j (resp., χ_i) and χ_i and χ_j respectively restricted to such strategies are equivalent over φ''. By apply the equivalence relation on universal quantifier, we have that the set of strategies for variable z that verify the sub-formula φ'' are $\{\sigma_0^z, \sigma_1^z\}$ for χ_1, $\{\sigma_1^z\}$ for χ_2 and χ_4, and $\{\sigma_0^z\}$ for χ_3. We define with χ_1^1 and χ_1^2 the two assignments on σ_0^z and σ_1^z, respectively. By the disjunctive consistency we have that $\chi_i \equiv_{\mathcal{E},\mathtt{s}_0}^{\varphi''} \chi_j$ iff the followings holds:

- $\mathcal{E}, \chi_i, \mathtt{s}_0 \models \flat_k\mathtt{p}_h$ iff $\mathcal{E}, \chi_j, \mathtt{s}_0 \models \flat_k\mathtt{p}_h$ $\quad \forall \mathtt{k} \in \{1,2\}$ and $\mathtt{h} \in \{U, V\}$

- If $\mathcal{E}, \chi_i, \mathtt{s}_0 \models \flat_k\mathtt{p}_h$ iff $\chi_i \equiv_{\mathcal{E},\mathtt{s}_0}^{\flat_k\mathtt{p}_h} \chi_j$ $\quad \forall \mathtt{k} \in \{1,2\}$ and $\mathtt{h} \in \{U, V\}$

For $\mathtt{i} = 1$, $\mathtt{j} = 2$, and $\mathtt{k} = 1$ we have that $\chi_1^1 \circ \flat_1$, $\chi_2 \circ \flat_1$, induce the plays $\pi_1^1 = \mathtt{s}_0\mathtt{s}_1\mathtt{s}_4{}^\omega$, $\pi_2 = \mathtt{s}_0\mathtt{s}_1\mathtt{s}_3{}^\omega$, respectively. Now, π_1^1 is not belong to $\mathsf{pr}(\mathsf{p}_V)$ whereas π_2 belongs to $\mathsf{pr}(\mathsf{p}_V)$. Therefore, χ_1 and χ_2 are not equivalent.

For $\mathtt{i} = 1$, $\mathtt{j} = 3$, and $\mathtt{k} = 1$ we have that $\chi_1^2 \circ \flat_1$, $\chi_3 \circ \flat_1$, induce the plays $\pi_1^2 = \mathtt{s}_0\mathtt{s}_1\mathtt{s}_3{}^\omega$, $\pi_3 = \mathtt{s}_0\mathtt{s}_2\mathtt{s}_6{}^\omega$, respectively. Now, π_1^2 belongs to $\mathsf{pr}(\mathsf{p}_V)$ whereas π_3 is not. Therefore, χ_1 and χ_3 are not equivalent.

For $\mathtt{i} = 1$, $\mathtt{j} = 4$, and $\mathtt{k} = 1$ we have that $\chi_4 \circ \flat_1$, induces the play $\pi_4 = \mathtt{s}_0\mathtt{s}_2\mathtt{s}_4{}^\omega$. Now, π_1^2 belongs to $\mathsf{pr}(\mathsf{p}_V)$ whereas π_4 is not. Therefore, χ_1 and χ_4 are not equivalent.

For $\mathtt{i} = 2$, $\mathtt{j} = 3$, and $\mathtt{k} = 1$ we say that π_2 belongs to $\mathsf{pr}(\mathsf{p}_V)$ whereas π_3 is not. Therefore, χ_2 and χ_3 are not equivalent.

For $\mathtt{i} = 2$, $\mathtt{j} = 4$, and $\mathtt{k} = 1$ we say that π_2 belongs to $\mathsf{pr}(\mathsf{p}_V)$ whereas π_4 is not. Therefore, χ_2 and χ_4 are not equivalent.

For $\mathtt{i} = 3$, $\mathtt{j} = 4$, and $\mathtt{k} = 1$ we say that π_3 and π_4 are not belong to $\mathsf{pr}(\mathsf{p}_V)$. So, we must analyze for $\mathtt{k} = 2$, we have that $\chi_3 \circ \flat_2$, $\chi_4 \circ \flat_2$, induce the plays $\pi_3 = \mathtt{s}_0\mathtt{s}_2\mathtt{s}_6{}^\omega$, $\pi_4 = \mathtt{s}_0\mathtt{s}_2\mathtt{s}_5{}^\omega$, respectively. Now, π_3 and π_4 belong to $\mathsf{pr}(\mathsf{p}_V)$. But, the common histories $\mathtt{s}_0\mathtt{s}_2$ is not belong to $\mathrm{U}_{\mathsf{p}_U}$. So, χ_3 and χ_4 are not equivalent and this concludes the example.

4.5.4 Scheduler

Consider the arena $\mathcal{A}_S$ depicted in Figure 4.3 associated to the extension $\mathcal{E} \triangleq \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$. It represents a model of a simple *scheduler system* in which two *processes*, $\mathtt{P}_1$ and $\mathtt{P}_2$, can require the access to a *shared resource*, like a processor, and an *arbiter* `W` is used to solve all conflicts that may arise when contending requests are made. The processes use four actions to interact with the system: `i` for idle, `r` for request, `f` for free/release, and `a` for abandon/relinquish. The first means that the process does not want to change the current situation in which entire system reside. The second is used to request the resource, when this is not yet owned, while the third releases it, when this is not needed anymore. Finally, the last is asserted by a process that, although it has requested the resource, did not obtain it and so it decided to relinquish the request. The whole scheduler system can reside in the following six states: `I`, `1`, `2`, `1/2`, `2/1` and `W`. The idle state `I` indicates that none of the process owns the resource. The state k, with $k \in \{1, 2\}$, indicates that process

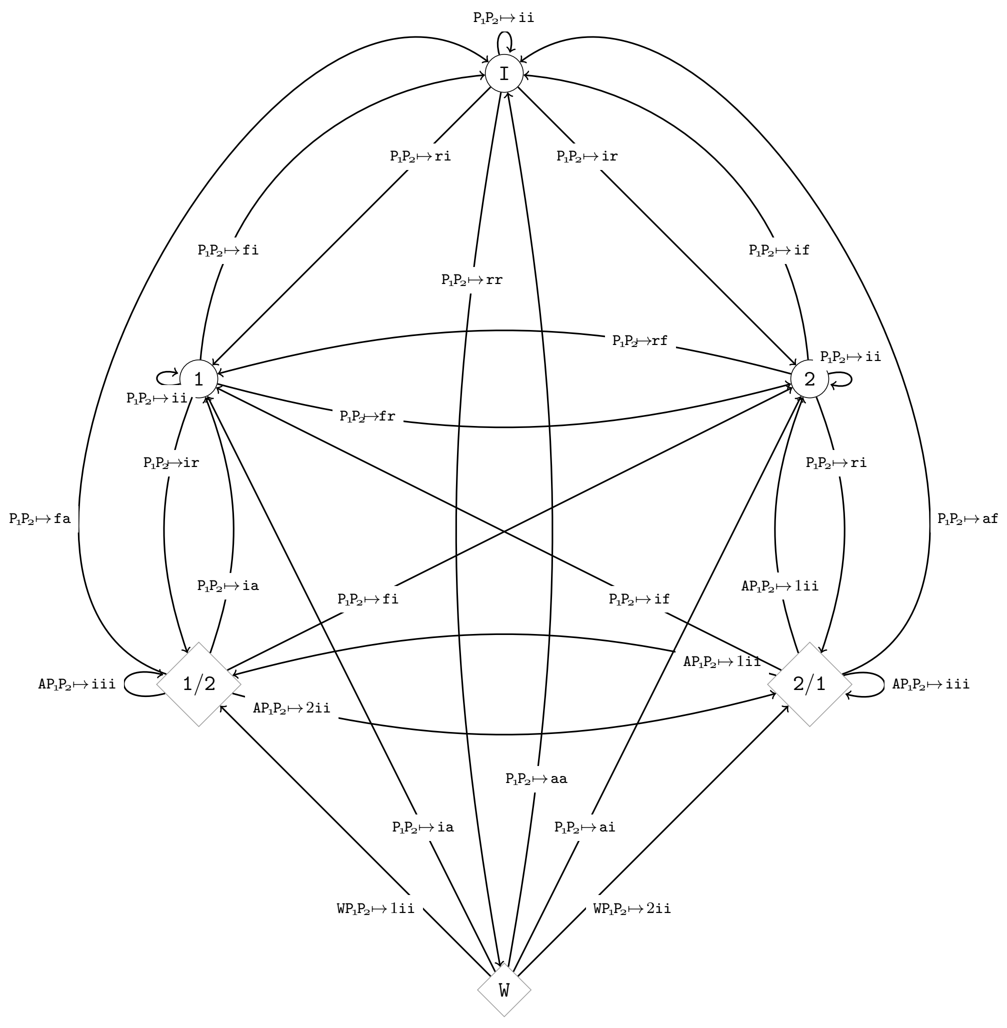

Figure 4.3: Scheduler Arena $\mathcal{A}_S$.

$\mathtt{P}_k$ is using the resource. The $\mathtt{1/2}$ (resp. $\mathtt{2/1}$) states indicates that the process $\mathtt{P}_1$(resp., $\mathtt{P}_2$) has the resource while the process $\mathtt{P}_2$ (resp., $\mathtt{P}_1$) requires it. Finally, the arbitrage state $\mathtt{W}$ represents the situation in which an action from the arbiter is required in order to solve a conflict between contending requests. For readability reasons, a decision is graphically represented by $\mathtt{a} \rightarrow \mathtt{c}$, where $\mathtt{a}$ is a sequence of agents and $\mathtt{c}$ is a sequence of corresponding actions. For example $\mathtt{P}_1\mathtt{P}_2 \rightarrow \mathtt{ir}$ indicates that agents $\mathtt{P}_1$ and $\mathtt{P}_2$ take actions $\mathtt{i}$ and $\mathtt{r}$, respectively.

Formally, $\mathcal{A}_S$ is the arena $\langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathtt{tr}\rangle$, where all components but the transition function are set as follows: $\mathrm{Ag} = \{\mathtt{W}, \mathtt{P}_1, \mathtt{P}_2\}$, $\mathrm{Ac} = \{\mathtt{i}, \mathtt{r}, \mathtt{f}, \mathtt{a}, \mathtt{1}, \mathtt{2}\}$, and $\mathrm{St} = \{\mathtt{I}, \mathtt{1}, \mathtt{2}, \mathtt{1/2}, \mathtt{2/1}, \mathtt{W}\}$. For the sake of simplicity, the transition are depicted in Figure 4.3 and is not listed here textually.

In addition, we have that $\mathtt{ag}(\mathtt{s}) = \{\mathtt{P}_1, \mathtt{P}_2\}$, for all $s \in \{\mathtt{I}, \mathtt{1}, \mathtt{2}\}$, and $\mathtt{ag}(s) = \{\mathtt{P}_1, \mathtt{P}_2, \mathtt{A}\}$ for all $s \in \{\mathtt{1/2}, \mathtt{2/1}, \mathtt{W}\}$. Moreover, for all $k \in \{1, 2\}$ we have that $\mathtt{ac}(\mathtt{I}, \mathtt{P}_k) = \{\mathtt{i}, \mathtt{r}\}$, $\mathtt{ac}(k, \mathtt{P}_k) = \{\mathtt{i}, \mathtt{f}\}$, $\mathtt{ac}(k, \mathtt{P}_{3-k}) = \{\mathtt{i}, \mathtt{r}\}$, $\mathtt{ac}(W, \mathtt{A}) = \{\mathtt{1}, \mathtt{2}\}$, $\mathtt{ac}(W, \mathtt{P}_k) = \{\mathtt{i}, \mathtt{a}\}$, $\mathtt{ac}(k/(3-k), \mathtt{A}) = \{\mathtt{i}, 3-k\}$, $\mathtt{ac}(k/(3-k), \mathtt{P}_k) = \{\mathtt{i}, \mathtt{f}\}$, $\mathtt{ac}(k/(3-k), \mathtt{P}_{3-k}) = \{\mathtt{i}, \mathtt{a}\}$.

Consider the formula $\varphi = \wp\flat\mathtt{p}$, with $\wp = \langle\!\langle \mathtt{x} \geq g_1 \rangle\!\rangle [\![\mathtt{y} < g_2]\!] [\![\mathtt{z} < g_3]\!]$ and $\flat = (\mathtt{A}, \mathtt{x})(\mathtt{P}_1, \mathtt{y})$ $(\mathtt{P}_2, \mathtt{z})$, where $\mathtt{p}$ is the open predicate that represents the situation in which one of the two processes has obtained the resource. Formally, $\mathtt{p}$ has witness $\mathrm{U} = \mathtt{I}^+ \cdot (\mathtt{1}+\mathtt{2}) + (\mathtt{I}^+ \cdot \mathtt{W})^+ \cdot (\mathtt{1}+\mathtt{2}+\mathtt{1/2}+\mathtt{2/1})$. Then, it can be showed that $\mathcal{E}, \mathtt{I} \models \varphi$ with $(g_1, g_2, g_3) = (\mathtt{k}, 1, 2)$ for all $\mathtt{k} \geq 0$, while $\mathcal{E}, \mathtt{I} \not\models \varphi$ with $(g_1, g_2, g_3) = (1, 1, 1)$.

To verify that the formula $\langle\!\langle \mathtt{x} \geq 1 \rangle\!\rangle [\![\mathtt{y} < 1]\!] [\![\mathtt{z} < 1]\!] \flat\mathtt{p}$ is not satisfied, we verify that the negated formula $[\![\mathtt{x} < 1]\!] \langle\!\langle \mathtt{y} \geq 1 \rangle\!\rangle \langle\!\langle \mathtt{z} \geq 1 \rangle\!\rangle \flat\neg\mathtt{p}$ is satisfied. Actually, we verify the even stronger formula $\langle\!\langle \mathtt{y} \geq k \rangle\!\rangle \langle\!\langle \mathtt{z} \geq 1 \rangle\!\rangle [\![\mathtt{x} < 1]\!] \flat\neg\mathtt{p}$, for all $\mathtt{k} \geq 0$. Now, the closed predicate $\neg\mathtt{p}$ has witness $\mathrm{V} = (\mathtt{I}^+ \cdot \mathtt{W})^* \cdot \mathtt{I}^* \setminus \{\epsilon\}$. For all $\pi \in \mathsf{pr}(\neg\mathtt{p})$, there exists a strategy $\sigma_{\neg\mathtt{p}}$ such that, for all $\mathtt{j} \in \mathbb{N}$ it holds:

$$\sigma_{\neg\mathtt{p}}(\pi_{\leq \mathtt{j}}) \triangleq \begin{cases} i, & \text{if } \pi_{\mathtt{j}} = \mathtt{I} \text{ and } \pi_{\mathtt{j}+1} = \mathtt{I}; \\ r, & \text{if } \pi_{\mathtt{j}} = \mathtt{I} \text{ and } \pi_{\mathtt{j}+1} = \mathtt{W}; \\ a, & \text{otherwise.} \end{cases}$$

Two strategies $\sigma_{\neg\mathtt{p}}(\pi_1)$ and $\sigma_{\neg\mathtt{p}}(\pi_2)$, for $\pi_1, \pi_2 \in \mathsf{pr}(\neg\mathtt{p})$ with $\pi_1 \neq \pi_2$ are evidently non equivalent with respect to $\langle\!\langle \mathtt{z} \geq 1 \rangle\!\rangle [\![\mathtt{x} < 1]\!] \flat\neg\mathtt{p}$. Indeed the agent $\mathtt{P}_2$ to satisfy the formula must choose the same strategy of agent $\mathtt{P_1}$. In this case, the strategy for the arbiter $\mathtt{A}$ is irrelevant. In fact, for all $(\neg\mathtt{p})$-coherent assignments $\chi_\pi \in \mathrm{Asg}$ such that $\chi_\pi(\mathtt{y}) = \chi_\pi(\mathtt{z}) = \sigma_{\neg\mathtt{p}}(\pi)$ for $\pi \in \mathsf{pr}(\neg\mathtt{p})$ we have that $\mathsf{play}(\chi_\pi[\mathtt{x} \mapsto \sigma] \circ \flat, \mathtt{I}) = \pi$, for all $\sigma \in \mathrm{Str}$. Now, it is clear to see that $\chi_{\pi_1} \not\equiv_{\mathcal{E},\mathtt{I}}^{[[\mathtt{x}<1]]\flat\neg\mathtt{p}} \chi_{\pi_2}$ since there surely exists an history $\rho \in \mathrm{Hst}$ with prefix $\mathsf{prf}(\pi_1, \pi_2) \leq \rho$ such that $\rho \not\in \mathrm{V}$. As observed above, agent $\mathtt{P_2}$ necessarily needs to use the same strategy of $\mathtt{P_1}$ to satisfy the formula. Therefore, the formula $\langle\langle\mathtt{y} \geq k\rangle\rangle\langle\langle\mathtt{z} \geq 2\rangle\rangle[[\mathtt{x} < 1]]\flat\neg\mathtt{p}$ is not satisfied, because there are not two strategy for the agent $\mathtt{P_2}$ once that $\mathtt{P_1}$ has chose is one. The last reasoning leads us to claim that the formula $\langle\langle\mathtt{x} \geq \mathtt{k}\rangle\rangle[[\mathtt{y} < 1]][[\mathtt{z} < 2]]$ is satisfied on $\mathcal{E}$, for all $\mathtt{k} \geq 0$. Indeed, consider the infinite set on strategies $\{\sigma_{\mathtt{i}} : \forall i \in \mathbb{N}\}$ defined as follows, for all $\rho \in \mathrm{Hst}$:

$$\sigma_{\mathtt{i}}(\rho) \triangleq \begin{cases} 2, & \text{if } \rho \in (\mathtt{I}^\star \cdot \mathtt{W})^{\mathtt{i}}; \\ 1, & \text{otherwise.} \end{cases}$$

It easy observe that $\mathcal{E}, \chi_{\mathtt{i}}, \mathtt{I} \models [[\mathtt{y} < 1]][[\mathtt{z} < 2]]\flat\mathtt{p}$ where $\chi_i(\mathtt{x}) = \sigma_{\mathtt{i}}$, for all $\mathtt{i} \in \mathbb{N}$. Indeed, the unique strategy that $\mathtt{P_2}$ can use to verify $\neg\mathtt{p}$ can be excluded due to the degree 2 of the universal quantifier $[[\mathtt{z} < 2]]$. Moreover, we can be observe that χ_i and χ_j are not equivalent *w.r.t.* $[[\mathtt{y} < 1]][[\mathtt{z} < 2]]\flat\mathtt{p}$, for all $\mathtt{i}, \mathtt{j} \in \mathbb{N}$ and $\mathtt{i} \neq \mathtt{j}$.

5 Game Type Conversion

In this Chapter, we show how to transform a game in a simpler equivalent one. This is done in order to solve efficiently the proposed game question. First, in Section 5.1, we introduce the conversion from GSG[1G, kVAR] to GSG[1G, TB,kAG/VAR]. For this conversion we need to build a new arena and, consequently, a new extension. We divide this in three-step process. The first phase, called normalization, reduces the number of agents to the number of variables. The second phase, called minimization, reduces the space of the actions. The third and final step performs the construction of turn-based game. Second, in Section 5.2, we define the conversion from GSG[BG] to SG[BG]. For this conversion, the arena remains the same, but the extension is enriched with new predicates. Regarding the solutions concepts the transformation makes use of a specifics two functions. The first on takes can of a syntactic transformation. The other one is used to enforce the equivalence among assignments holding for the input target. It is important to observe that starting with a GSG[1G], the algorithm return in general a SG[BG]. It is for this reason that we introduce an ad hoc translation regarding GSG[1G].

5.1 From GSG[1G, kVAR] To GSG[1G, TB, kAG/VAR]

In this Section, we show how to transform a game from concurrent to turn-based games. It is worth recalling that similar reductions have been also used to solve questions related to GATL in [FNP10] and the one-goal fragment of SL in [MMS14]. However, none of them can be used for GSL[1G]. The main reason resides in the fact that in both mentioned cases, the reduction always results in a two-player game, where the two players represent a collapsing of all existential and universal modalities, respectively. Conversely, in GSL[1G] we need to maintain a multi-player setting in the construction. This is due to the fact that GSL[1G] uses graded strategies (in addition to SL) and an equivalence relation that allows for a more powerful methodology (than the one used in GATL) to count strategies. In particular, regarding the latter, it is worth recalling that in GATL strategies are grouped *w.r.t.* the actions taken by the agents, while in GSL[1G] we consider instead the underling strategic reasoning. Thus, we introduce an ad hoc transformation of the concurrent game under exam into a multi-player turn-based one, which has the peculiarity of retaining the same number of variables, but can collapse equivalent actions. More precisely, starting with a game having k variables, we end in a game with k agents and k variables. The proposed conversion is divided in three parts. The first, called *normalization*, regards the elimination of the binding. The second, named *minimization*, is the elimination of redundant actions. Finally, the third is the real transformation of the game in a turn-based one.

5.1.1 Normalization

In this Subsection, we introduce the concept of normalized arena *w.r.t.* a given binding. The aim is to show how to turn an arena $\mathcal{A}$ in a new one $\mathcal{A}^{\bullet}$ in which all agents associated with the same variable are merged into a single player. Basically, by applying the normalization we restrict our attention to the part of $\mathcal{A}$ that is effectively involved in the verification check of the formula *w.r.t.* a binding $\flat$. From a technical point of view, the normalization consists of two steps. The first transforms the set of variables into the set of agents; this means that all bindings become identities of the kind (x, x). The second involves the transition function, which is augmented in order to associate decisions to the right agent (via the binding).

Construction 5.1.1 (Arena Normalization)**.** *From an arena* $\mathcal{A}=\langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr} \rangle$ *and a binding prefix* $\flat \in \mathrm{Bn}(\mathrm{Ag})$*, we build the* normalized arena $\mathcal{A}^{\bullet} \triangleq \langle \mathrm{Ag}^{\bullet}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr}^{\bullet} \rangle$*, where the new agents in* $\mathrm{Ag}^{\bullet} \triangleq \mathsf{rng}(\flat)$ *are all variables bound by* $\flat$ *and the new transition function* $\mathsf{tr}^{\bullet}(\delta^{\bullet})(s) \triangleq \mathsf{tr}(\delta^{\bullet} \circ \flat)(s)$ *simply maps a state* $s \in \mathrm{St}$ *and a new active decision* $\delta^{\bullet} \in \mathsf{dc}^{\bullet}(s) \triangleq \{\delta^{\bullet} \in \mathrm{Dc}^{\bullet} : \delta^{\bullet} \circ \flat \in \mathsf{dc}(s)\}$ *into the successor* $\mathsf{tr}(\delta^{\bullet} \circ \flat)(s)$ *of* s *following the original decision* $\delta^{\bullet} \circ \flat \in \mathrm{Dc}$.

To normalize the game, we simply need to normalize its arena, as well as change the underling solution concept as now agents and variables names coincide. Intuitively, the new solution concept differs from the original one for its bindings, which now are all identities.

Construction 5.1.2 (Game Normalization)**.** *From a* GSG[1G, kVAR] $\partial = \langle \mathcal{E}, s_I, \varphi \rangle$ *with extension* $\mathcal{E} = \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$ *and solution concept* $\varphi = \wp \flat \psi$*, we build the* normalized GSG[1G, kAG/VAR] $\partial^{\bullet} \triangleq \langle \mathcal{E}^{\bullet}, s_I, \varphi^{\bullet} \rangle$ *with extension* $\mathcal{E}^{\bullet} \triangleq \langle \mathcal{A}^{\bullet}, \mathrm{Pr}, \mathsf{pr} \rangle$ *and solution concept* $\varphi^{\bullet} \triangleq \wp \prod_{x \in \mathsf{rng}(\flat)} (x, x) \psi$*, where the arena* $\mathcal{A}^{\bullet}$ *is built as in Construction 5.1.1.*

From the structure of the formula, one can easily prove by induction the following statement.

Theorem 5.1.1 (Normalization)**.** *For each* GSG[1G, kVAR] ∂*, there is a normalized* GSG[1G, kAG/VAR] $\partial^{\bullet}$ *such that* ∂ *is fulfilled iff* $\partial^{\bullet}$ *is.*

Example 5.1.1. *Consider the game* $\eth = \langle \mathcal{E}, \mathsf{s}_0, \varphi \rangle$ *with extension* $\mathcal{E} = \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$*, the state* $\mathsf{s}_0 \in \mathrm{St}$*, and solution concept* $\varphi = \wp\flat\mathrm{p}_U$*, where* $\wp = \langle\!\langle \mathtt{x} \geq 3 \rangle\!\rangle [\![\mathtt{y} < 2]\!]$ *and* $\flat = (\mathtt{a}, \mathtt{x})(\mathtt{b}, \mathtt{y})(\mathtt{c}, \mathtt{x})$*. Moreover, let* $\mathcal{A} = \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr} \rangle$ *the related arena depicted in Figure 3.1. We want to build a normalized game* $\eth^\bullet$ *from* $\eth$*. First, the new arena is* $\mathcal{A}^\bullet \triangleq \langle \mathrm{Ag}^\bullet, \mathrm{Ac}, \mathrm{St}, \mathsf{tr}^\bullet \rangle$*, where the set of new agents is* $\mathrm{Ag}^\bullet \triangleq \{\mathtt{x}, \mathtt{y}\}$ *and, for the sake of simplicity, the transition are depicted in Figure 5.1 and will not listed here textually. Finally, the* normalized GSG[1G, kAG/VAR] $\eth^\bullet \triangleq \langle \mathcal{E}^\bullet, \mathsf{s}_0, \varphi^\bullet \rangle$ *is composed by extension* $\mathcal{E}^\bullet \triangleq \langle \mathcal{A}^\bullet, \mathrm{Pr}, \mathsf{pr} \rangle$ *and solution concept* $\varphi^\bullet \triangleq \langle\!\langle \mathtt{x} \geq 3 \rangle\!\rangle [\![\mathtt{y} < 2]\!] (\mathtt{x}, \mathtt{x})(\mathtt{y}, \mathtt{y})\mathrm{p}_U$.

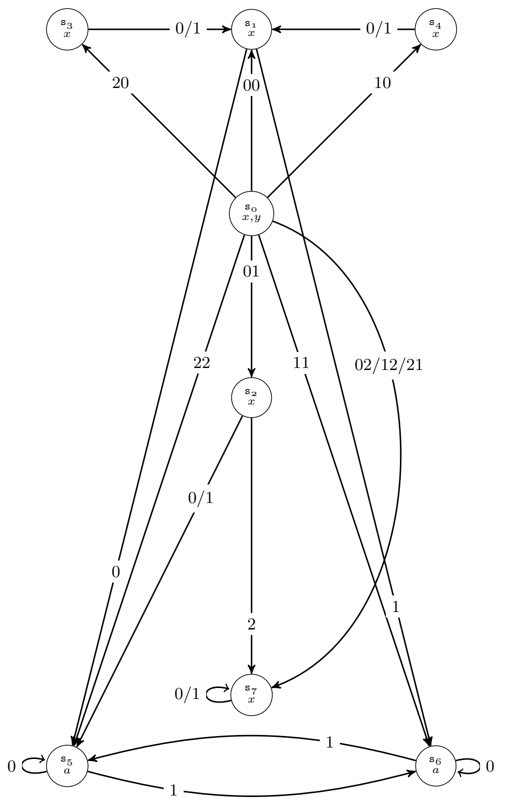

Figure 5.1: Normalized Arena $\mathcal{A}^\bullet$ built on Arena $\mathcal{A}$.

5.1.2 Minimization

As for previous considerations, actions involving the same strategic reasoning need to be merged together. We accomplish this by constructing a new arena that maintains just one representative for each class of equivalence actions. The formal construction follows.

Given an arena $\mathcal{A} = \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr} \rangle$, one of its states $s \in \mathrm{St}$, and a quantification prefix $\wp \in \mathrm{Qn}(\mathsf{ag}(s))$, we can define an equivalence relation $\delta_1 \equiv_s^\wp \delta_2$ between decisions $\delta_1, \delta_2 \in \mathrm{Dc}$

with $\mathsf{ag}(s) \setminus \mathsf{vr}(\wp) \subseteq \mathsf{dom}(\delta_1), \mathsf{dom}(\delta_2)$ that locally mimics the behavior of the one between assignments previously discussed. Intuitively, it allows to determine whether two different moves of a set of agents are actually mutually substitutable *w.r.t.* the strategy quantification of interest. Formally, we have that:

1) for the empty quantification prefix ϵ, it holds that $\delta_1 \equiv_s^{\epsilon} \delta_2$ iff $\mathsf{tr}(\delta_1)(s) = \mathsf{tr}(\delta_2)(s)$;

2) $\delta_1 \equiv_s^{\langle\langle a \geq g \rangle\rangle \wp} \delta_2$ iff, for all active actions $c \in \mathsf{ac}(s, a)$, it holds that $\delta_1[a \mapsto c] \equiv_s^{\wp} \delta_2[a \mapsto c]$;

3) $\delta_1 \equiv_s^{[[a < g]] \wp} \delta_2$ iff, for all indexes $i \in \{1, 2\}$ and active actions $c_i \in \mathsf{ac}(s, a)$, there exists an active action $c_{3-i} \in \mathsf{ac}(s, a)$ such that $\delta_1[a \mapsto c_1] \equiv_s^{\wp} \delta_2[a \mapsto c_2]$.

At this point, we can introduce an equivalence relation between the active actions $c_1, c_2 \in \mathsf{ac}(s, a)$ of an agent $a \in \mathsf{ag}(s)$, once a partial decision $\delta \in \mathrm{Dc}$ with $\{a' \in \mathsf{ag}(s) : a' <_{\wp} a\} \subseteq \mathsf{dom}(\delta)$ of the agents already quantified is given. Formally, $c_1 \equiv_{s,\delta}^{\langle\langle a \geq g \rangle\rangle \wp} c_2$ iff $\delta[a \mapsto c_1] \equiv_s^{\wp} \delta[a \mapsto c_2]$ and $c_1 \equiv_{s,\delta}^{[[a < g]] \wp} c_2$ iff $\delta[a \mapsto c_1] \equiv_s^{\overline{\wp}} \delta[a \mapsto c_2]$, where $\overline{\wp}$ represent the dual prefix of $\wp$. Intuitively, the two actions c_1, c_2 are equivalent *w.r.t.* δ iff agent a can use indifferently one of the two to extend δ, without changing the set of successor of s it can force to reach.

We can now introduce the concept of minimization of an arena, in which the behavior of each agent is restricted in such a way that he can only choose the representative element from each class of equivalence actions. Before moving to the formal definition, as an additional notation we introduce $\wp_{\geq a}$ that takes in input a prefix of quantification $\wp$ and returns the fragment of prefix starting from a.

Construction 5.1.3 (Arena Minimization)**.** *From an arena* $\mathcal{A} = \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr} \rangle$ *normalized* w.r.t. *a binding prefix* $\flat \in \mathrm{Bn}(\mathrm{Ag})$ *and a quantification prefix* $\wp \in \mathrm{Qn}(\mathsf{rng}(\flat))$*, we build the* minimized arena $\mathcal{A}^{\blacklozenge} \triangleq \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr}^{\blacklozenge} \rangle$*, where the new transition function* $\mathsf{tr}^{\blacklozenge}$ *is defined as follows. First, assume* $\Lambda(s, \delta, a) \subseteq \mathsf{ac}(s, a)$ *to be a subset of active actions for the agent* $a \in \mathsf{ag}(s)$ *on the state* $s \in \mathrm{St}$ *such that, for each* $c \in \mathsf{ac}(s, a)$*, there is exactly one* $c' \in \Lambda(s, \delta, a)$ *with* $c \equiv_{s,\delta}^{\wp_{\geq a}} c'$*, where the quantification prefix* $\wp_{\geq a}$ *is obtained as suffix from* a *onward of* $\wp$*. Intuitively,* $\Lambda(s, \delta, a)$ *is one of the minimal sets of actions needed by the agent* a *in order to preserve the essential structure of the arena. At this point, let* $\mathsf{dc}^{\blacklozenge}(s) \triangleq \{\delta \in \mathsf{dc}(s) : \forall a \in \mathsf{dom}(\delta) \,.\, \delta(a) \in \Lambda(s, \delta_{\restriction \{a' \in \mathsf{dom}(\delta) : a' <_{\wp} a\}}, a)\}$

to be the set of active decisions having only values among those ones previously chosen. Finally, for each state $s \in \mathrm{St}$ *and decision* $\delta \in \mathrm{Dc}$*, assume* $\mathsf{tr}^{\blacklozenge}(\delta)(s) \triangleq \mathsf{tr}(\delta)(s)$*, if* $\delta \in \mathsf{dc}^{\blacklozenge}(s)$*, and* $\mathsf{tr}^{\blacklozenge}(\delta) \triangleq \emptyset$*, otherwise.*

The minimization of the game involves just the arena as only the actions can change (while states and agents remain the same). The formal definition follows.

Construction 5.1.4 (Game Minimization)**.** *From a normalized* GSG[1G] $\partial = \langle \mathcal{E}, s_I, \varphi \rangle$ *with extension* $\mathcal{E} = \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$ *and solution concept* $\varphi = \wp\flat\psi$*, we build the* minimized GSG[1G] $\partial^{\blacklozenge} \triangleq \langle \mathcal{E}^{\blacklozenge}, s_I, \varphi \rangle$ *with extension* $\mathcal{E}^{\blacklozenge} \triangleq \langle \mathcal{A}^{\blacklozenge}, \mathrm{Pr}, \mathsf{pr} \rangle$*, where the arena* $\mathcal{A}^{\blacklozenge}$ *is built as in Construction 5.1.3.*

Again, by induction on the formula, one can prove the following result.

Theorem 5.1.2 (Minimization)**.** *For each* GSG[1G, kVAR] ∂*, there is a minimized* GSG[1G, kAG/VAR] $\partial^{\blacklozenge}$ *such that* ∂ *is fulfilled iff* $\partial^{\blacklozenge}$ *is.*

Example 5.1.2. *Consider the normalized game $\Game^\bullet$ of the Example 5.1.1 with extension $\mathcal{E}^\bullet$, the state* $\mathtt{s}_0 \in \mathrm{St}$*, and solution concept* $\varphi^\bullet = \wp\flat\mathrm{p}_U$*, where* $\wp = \langle\!\langle \mathtt{x} \geq 3 \rangle\!\rangle [\![\mathtt{y} < 2]\!]$ *and* $\flat = (\mathtt{x}, \mathtt{x})(\mathtt{y}, \mathtt{y})$*. Moreover, let $\mathcal{A}^\bullet$ the related arena depicted in Figure 5.1. We want to build a minimized game $\Game^\blacklozenge$ from $\Game^\bullet$. The new arena is* $\mathcal{A}^\blacklozenge \triangleq \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr}^\blacklozenge \rangle$*, where the new transition function* $\mathsf{tr}^\blacklozenge$ *is depicted in Figure 5.2. For give an intuition, we analyze the equivalence relation between the actions* $[\mathtt{x} \mapsto 0], [\mathtt{x} \mapsto 1] \in \mathsf{ac}(\mathtt{s}_3, \mathtt{x})$ *of the agent* $\mathtt{x}$*. So, we have that* $[\mathtt{x} \mapsto 0] \equiv^{[\![\mathtt{y}<2]\!]}_{\mathtt{s}_3,\delta} [\mathtt{x} \mapsto 1]$ *iff* $\delta[\mathtt{x} \mapsto 0] \equiv^{\wp}_{\mathtt{s}_3} \delta[\mathtt{x} \mapsto 0]$ *iff, for all indexes* $i \in \{1, 2\}$ *and active actions* $\mathtt{c}_i \in \mathsf{ac}(\mathtt{s}_3, \mathtt{y})$*, there exists an active action* $\mathtt{c}_{3-i} \in \mathsf{ac}(\mathtt{s}_3, \mathtt{y})$ *such that* $\delta_1[\mathtt{y} \mapsto \mathtt{c}_1] \equiv^{\epsilon}_{\mathtt{s}_3} \delta_2[\mathtt{y} \mapsto \mathtt{c}_2]$*. Since,* $\mathsf{ac}(\mathtt{s}_3, \mathtt{y}) = \emptyset$ *then* $\delta_1[\mathtt{y} \mapsto] \equiv^{\epsilon}_{\mathtt{s}_3} \delta_2[\mathtt{y} \mapsto]$ *iff* $\mathsf{tr}(\delta_1)(\mathtt{s}_3) = \mathsf{tr}(\delta_2)(\mathtt{s}_3)$*. We have that,* $\mathsf{tr}(\delta_1)(\mathtt{s}_3) = \mathsf{tr}(\delta_2)(\mathtt{s}_3) = \mathtt{s}_1$*, so the actions* $[\mathtt{x} \mapsto 0], [\mathtt{x} \mapsto 1]$ *are equivalent. Now, the* minimized GSG[1G] $\Game^\blacklozenge \triangleq \langle \mathcal{E}^\blacklozenge, s_I, \varphi \rangle$ *differ of normalized* GSG[1G, kAG/VAR] *only by* $\mathcal{A}^\blacklozenge$*.*

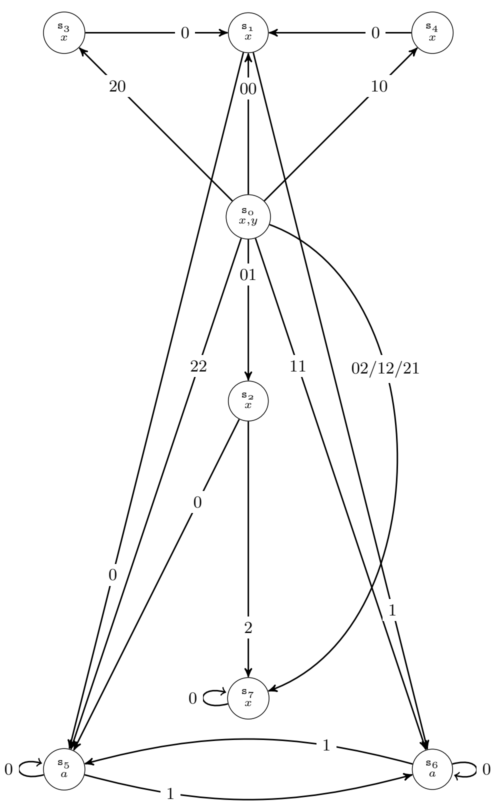

Figure 5.2: Minimized Arena $\mathcal{A}^\blacklozenge$ built on Normalized Arena $\mathcal{A}^\bullet$.

5.1.3 Conversion

Finally, we describe the conversion of concurrent games into turn-based ones. As anticipated before, differently from other cases that apply a similar transformation, the game we obtain is one with k agents and k variables, where k is the number of variables of the starting game. Additionally, our construction makes use of the concepts of minimization and equivalence between actions, by removing the ones that induce equivalent paths. The intuitive idea of our reduction is to replace each state in the concurrent game with a finite tree whose height depends on the number of agents in input. Also, we enrich each state of the new game with extra information regarding the corresponding state in the concurrent game, the index of the operator in the prefix of quantifications we are sitting on, and the sequence of actions taken by the agents along a play. The formal definition follows.

Construction 5.1.5 (Arena Conversion)**.** *From an arena $\mathcal{A} = \langle \text{Ag}, \text{Ac}, \text{St}, \mathsf{tr} \rangle$ minimized* w.r.t. *a binding prefix $\flat \in \text{Bn}(\text{Ag})$ and a quantification prefix $\wp \in \text{Qn}(\mathsf{rng}(\flat))$, we build the new arena $\mathcal{A}^\star \triangleq \langle \text{Ag}, \text{Ac}, \text{St}^\star, \mathsf{tr}^\star \rangle$, where the new set of states $\text{St}^\star$ and the new transition function $\mathsf{tr}^\star$ are defined as follows. Given a state $s \in \text{St}$, we denote by $\wp^s$ the quantification prefix obtained from $\wp$ by simply deleting all variables not in $\mathsf{ag}(s)$ and by $\text{Vr}(\wp^s)$ the corresponding set of variables. The state space has to maintain the information about the position in $\mathcal{A}$ together with the index of the first variable that has still to be evaluated and the values already associated to the previous variables. To do this, we set $\text{St}^\star \triangleq \text{St} \times_s [0, |\mathsf{ag}(s)|] \times_i (\text{Vr}(\wp^s_{<i}) \to \text{Ac})$. Observe that, when a play is in a state $(s, |\mathsf{ag}(s)|, \delta)$, all quantifications are already resolved and it is time to evaluate the corresponding decision δ. Before proceeding with the definition of the transition function, it is helpful to identify which are the active agents and decisions for each possible state. Formally, for all $(s, i, \delta) \in \text{St}^\star$, we have that $\mathsf{ag}((s, i, \delta)) \triangleq \{\mathsf{vr}(\wp^s_i)\}$ and $\mathsf{dc}^\star((s, i, \delta)) \triangleq \{\mathsf{vr}(\wp^s_i) \mapsto c : c \in \mathsf{ac}(s, \mathsf{vr}(\wp^s_i))\}$, if $i < |\wp^s|$, and $\mathsf{ag}((s, i, \delta)) \triangleq \emptyset$ and $\mathsf{dc}^\star((s, i, \delta)) \triangleq \{\varnothing\}$, otherwise. The transition function is defined as follows. For each new state (s, i, δ) with $i < |\wp^s|$ and new decision $\mathsf{vr}(\wp^s_i) \mapsto c$, we simply need to increase the counter i and embed $\mathsf{vr}(\wp^s_i) \mapsto c$ into δ. Formally, we set $\mathsf{tr}^\star(\mathsf{vr}(\wp^s_i) \mapsto c)((s, i, \delta)) \triangleq (s, i+1, \delta[\mathsf{vr}(\wp^s_i) \mapsto c])$. For a new state $(s, |\mathsf{ag}(s)|, \delta)$, instead, we just introduce a transition to the state $(s', 0, \varnothing)$, where s' is the successor of s in the arena $\mathcal{A}$ following the decision δ. Formally, we have $\mathsf{tr}^\star(\varnothing)((s, |\mathsf{ag}(s)|, \delta)) \triangleq (\mathsf{tr}(\delta)(s), 0, \varnothing)$.*

Due to the specific setting of the introduced $\text{GSG}[\text{1G}]$, the game transformation also affects the extensions and in particular its function on predicates. Clearly, one can link a path in the new arena $\mathcal{A}^\star$ with one from the input arena by simply dropping the additional information introduced to manage quantifications. Hence, a play in the arena $\mathcal{A}^\star$ belongs to a function predicate if and only if the corresponding play in arena $\mathcal{A}$ belongs to the same function.

Construction 5.1.6 (Extension Conversion)**.** *From an extension $\mathcal{E} = \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$ minimized* w.r.t. *a binding prefix $\flat \in \mathrm{Bn}(\mathrm{Ag})$ and a quantification prefix $\wp \in \mathrm{Qn}(\mathsf{rng}(\flat))$, we build the new extension $\mathcal{E}^\star \triangleq \langle \mathcal{A}^\star, \mathrm{Pr}, \mathsf{pr}^\star \rangle$, where the new arena $\mathcal{A}^\star$ is built as in Construction 5.1.5 and the new predicate function $\mathsf{pr}^\star$ is such that, for all predicates $p \in \mathrm{Pr}$ and new paths $\pi^\star \in \mathrm{Pth}^\star$, it holds that $\pi^\star \in \mathsf{pr}^\star(p)$ iff there exist an original path $\pi \in \mathsf{pr}(p)$ and a strictly increasing map $\gamma : \mathbb{N} \to \mathbb{N}$ with $\gamma(0) = 0$ such that $(\pi^\star)_{\gamma(i)+j} = ((\pi)_i, j, \zeta)$ for some valuation $\zeta \in \mathrm{Vr}(\wp^{(\pi)_i}_{<j}) \to \mathrm{Ac}$, for all $i \in \mathbb{N}$ and $j \in [0, \gamma(i+1) - \gamma(i)[$.*

Finally, the conversion of the entire game follows from the conversion of the underlying extension and by providing an updated initial state (as the state space is converted as well).

Construction 5.1.7 (Game Conversion)**.** *From a minimized* GSG[1G] $\partial = \langle \mathcal{E}, s_I, \varphi \rangle$ *with solution concept $\varphi = \wp\flat\psi$, we build the* GSG[1G, TB] $\partial^\star \triangleq \langle \mathcal{E}^\star, s_I^\star, \varphi \rangle$ *with initial state $s_I^\star \triangleq (s_I, 0, \varnothing)$, where the extension $\mathcal{E}^\star$ is built as in Construction 5.2.1.*

By means of an opportune generalization of the classical proof of transformation of a concurrent game into a turn-based one, the following result can be derived.

Theorem 5.1.3 (Concurrent/Turn-Based Conversion)**.** *For each* GSG[1G, kVAR] ∂ *of order n, there is a* GSG[1G, TB, kAG/VAR] $\partial^\star$ *of order $n \cdot \mathrm{O}(2^k)$ such that ∂ is fulfilled iff $\partial^\star$ is.*

Example 5.1.3. *Consider the minimized game $\partial^\blacklozenge$ of the Example 5.1.2 with extension $\mathcal{E}^\blacklozenge$, the state $\mathrm{s}_0 \in \mathrm{St}$, and solution concept $\varphi^\blacklozenge = \wp\flat\mathrm{p}_U$, where $\wp = \langle\langle \mathrm{x} \geq 3 \rangle\rangle [[\mathrm{y} < 2]]$ and $\flat = (\mathrm{x}, \mathrm{x})(\mathrm{y}, \mathrm{y})$. Moreover, let $\mathcal{A}^\blacklozenge$ the related arena depicted in Figure 5.2. We want to build a turn-based game $\partial^\star$ from $\partial^\blacklozenge$. The new arena is $\mathcal{A}^\star \triangleq \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}^\star, \mathsf{tr}^\star \rangle$, where the new set of states $\mathrm{St}^\star$ and the new transition function $\mathsf{tr}^\star$ are depicted in Figure 5.3. The new extension $\mathcal{E}^\star \triangleq \langle \mathcal{A}^\star, \mathrm{Pr}, \mathsf{pr}^\star \rangle$, has a new predicate function $\mathsf{pr}^\star$. For given an intuition , we analyze the history $\mathrm{s}_0\mathrm{s}_1 \in \mathsf{pr}(\mathrm{p}_U)$. The corresponding history in $\mathsf{pr}^\star$ is $(\mathrm{s}_0, 0,)(\mathrm{s}_0, 1, 0)(\mathrm{s}_0, 2, 00)(\mathrm{s}_1, 0,)$. The* GSG[1G, TB] *is $\partial^\star \triangleq \langle \mathcal{E}^\star, s_I^\star, \varphi \rangle$ with initial state $\mathrm{s}^\star \triangleq (\mathrm{s}, 0, \varnothing)$.*

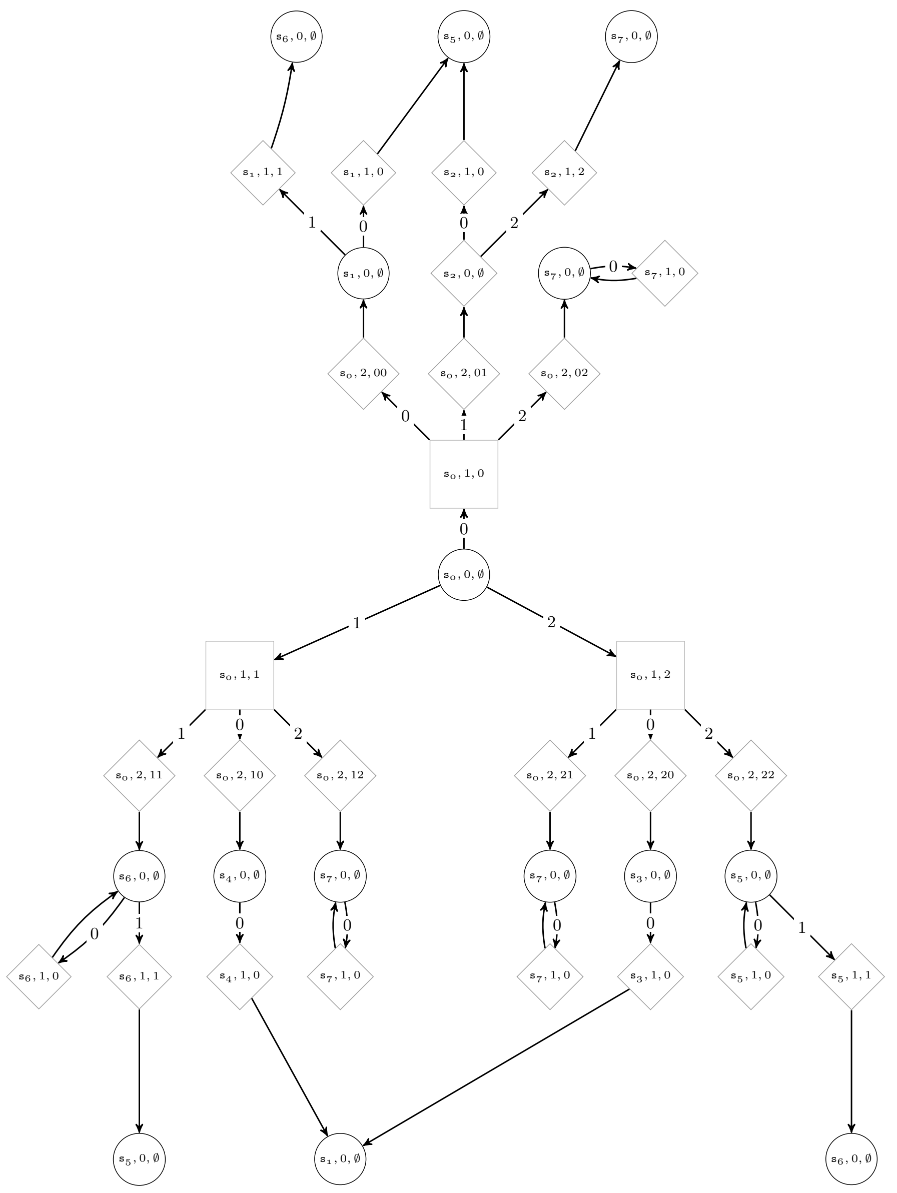

Figure 5.3: Turn-based Arena $\mathcal{A}^\star$ built on Minimized Arena $\mathcal{A}^\blacklozenge$. Note that, in the circle node is active the agent x, in the square node is active the agent y, and the diamond node represent the transition state. For see understanding on the arena, we duplicate some nodes and here the state without exit edges are just copies of the nodes in the same name.

5.2 From GSG[BG] To SG[BG]

In this Section, we show how to transform a graded strategy game in strategy game with solution concept in boolean fragment. Before starting with the construction, we introduce a constraint on the set of equivalence classes. The equivalence relation $\equiv^p_{\mathcal{E},s}$ is required to be of a finite order, *i.e.* $|(\mathrm{Asg}/{\equiv^p_{\mathcal{E},s}})| < \infty$. This is simply due to the fact that we associate to each equivalence class a predicate, whose total number of is required to be finite. Assume $(\mathrm{Asg}/{\equiv^p_{\mathcal{E},s}}) = \{C^p_1, ..., C^p_k\}$, where each C^p_i represents a set of assignments that are part of the same equivalence class. Suppose now we associate a new predicate to each class, so $C^p_i \rightarrow q^p_i$. Having said that, now we start with the construction. Since that the arena does not change, we show the construction of an extension.

Construction 5.2.1 (Extension Conversion)**.** *From an extension $\mathcal{E} = \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$, we build the new extension $\mathcal{E}^\star \triangleq \langle \mathcal{A}, \mathrm{Pr}^\star, \mathsf{pr}^\star \rangle$, where the new set of predicates is $\mathrm{Pr}^\star = \mathrm{Pr} \cup \{q^p_i : p \in \mathrm{Pr}\}$ and the new predicate function $\mathsf{pr}^\star$ is defined as $\mathsf{pr}(p)$ if $p \in \mathrm{Pr}$ and C^p_i otherwise.*

The conversion of the entire game follows from the conversion of the underlying extension and by change the solution concept from GSL to SL[BG].

Construction 5.2.2 (Game Conversion)**.** *From a GSG[BG] $\eth = \langle \mathcal{E}, s_I, \varphi \rangle$ with extension $\mathcal{E} = \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$ and solution concept φ, we build a SG[BG] $\eth^\star \triangleq \langle \mathcal{E}^\star, s, \varphi^\star \rangle$ with extension $\mathcal{E}^\star \triangleq \langle \mathcal{A}, \mathrm{Pr}^\star, \mathsf{pr}^\star \rangle$. To construct the solution concept $\varphi^\star$ we introduce a transformation function $\mathsf{t} \triangleq \mathrm{Cn} \times \mathsf{z} \rightarrow \mathrm{Cn}$, where z is a function used to rename variables. The function t is defined inductively as follows. The base case involves the transformation of predicates, simply the predicate remains the same.*

$$\mathsf{t}(p, \mathsf{z}) = p$$

For the Boolean operators, the function t bring out the operation and recursively calls itself without the operation. So for the transformation $\neg\varphi$ is denied the transformation of φ. For $\varphi_1 \wedge \varphi_2$ (resp.,

$\varphi_1 \vee \varphi_2$) simply performs two transformations in and (resp., or) of φ_1 and φ_2.

$$\mathsf{t}(\neg\varphi, \mathsf{z}) = \neg\mathsf{t}(\varphi, \mathsf{z})$$
$$\mathsf{t}(\varphi_1 \wedge \varphi_2, \mathsf{z}) = \mathsf{t}(\varphi_1, \mathsf{z}) \wedge \mathsf{t}(\varphi_2, \mathsf{z})$$
$$\mathsf{t}(\varphi_1 \vee \varphi_2, \mathsf{z}) = \mathsf{t}(\varphi_1, \mathsf{z}) \vee \mathsf{t}(\varphi_2, \mathsf{z})$$

As with Boolean operators, the binding is carried out associating the agent a a renaming of the variable by the function z.

$$\mathsf{t}((a, x)\varphi, \mathsf{z}) = (a, \mathsf{z}(x))\mathsf{t}(\varphi, \mathsf{z})$$

For existential operator we introduce a new function $neqv \triangleq \mathsf{z} \times \mathsf{z} \to \mathbb{N}$. The function $neqv$ compares two assignments and returns 1 if they are not equivalent, and 0 if they are equivalent. Suppose that the degree of existential operator is g, then the transformation will replace this with g existential operators. Through the function $neqv$ will evaluate whether all assignments, composed of the existential variables, are pairwise not equivalent. Finally, for each of the assignments recursively invoke the function t.

$$\mathsf{t}(\langle\langle x\rangle\rangle g\varphi, \mathsf{z}) = \prod_{1\leq k\leq g} \langle\langle x_k\rangle\rangle \, (\bigwedge_{1\leq i<j\leq g} neqv_{\varphi}(\mathsf{z}[x \mapsto x_i], \mathsf{z}[x \mapsto x_j])) \wedge (\bigwedge_{1\leq i\leq g} \mathsf{t}(\varphi, \mathsf{z}[x \mapsto x_i]))$$

For the universal operator directly we deny the formula and we apply that we have analyzed previously on existential operator.

$$\mathsf{t}([\![x < g]\!]\varphi, \mathsf{z}) = \mathsf{t}(\neg\langle\langle x \geq g\rangle\rangle\neg\varphi, \mathsf{z})$$

After the description of the function t we can now analyze in detail the function $neqv$ in its entirety. All items that meet from now on will be constructed by referring to the definitions of Chapter 4. We now illustrate how first step, the result of the function $neqv$ on predicate p. The function returns 0, if there is a predicate q_i^p that is satisfied by both z_1 and z_2. Indeed, if z_1 and z_2 satisfy the same predicate that means they belong to the same equivalence class, then they are equivalent. Conversely, the function returns 1 if there is not a predicate q_i^p that is satisfied by both z_1 and z_2, then z_1 and z_2 are not equivalent.

$$neqv_p(\mathsf{z}_1, \mathsf{z}_2) = \bigwedge_{1\leq i\leq k} \neg(\prod_{a\in \mathrm{Ag}} (a, \mathsf{z}_1(a))q_i^p \wedge \prod_{a\in \mathrm{Ag}} (a, \mathsf{z}_2(a))q_i^p)$$

Analyzing the Definition 4.4.1, the $neqv$ of conjunctive formulas, it will be exactly the or of the $neqv$ individual formulas.

$$neqv_{\varphi_1 \wedge \varphi_2}(\mathsf{z}_1, \mathsf{z}_2) = neqv_{\varphi_1}(\mathsf{z}_1, \mathsf{z}_2) \vee neqv_{\varphi_2}(\mathsf{z}_1, \mathsf{z}_2)$$

For the disjunctive form, we analyze the Definition 4.4.2. In the definition of equivalence we have 2 points in conjunction, we analyze them separately for the function $neqv$. Point 1 for $neqv$, we must verify that for at least one of the two formulas, z_1 (resp., z_2) satisfies and z_2 (resp., z_1) does not satisfy. Point 2 is checked by requiring that for at least one formula satisfied by z_1 and z_2, the two assignments are not equivalent. The result of the $neqv$ of a disjunctive formula consists of four disjoint, the first two verify point 1, while the following two disjoint verify point 2.

$$\begin{aligned} neqv_{\varphi_1 \vee \varphi_2}(\mathsf{z}_1, \mathsf{z}_2) = {} & (\mathsf{t}(\varphi_1, \mathsf{z}_1) \leftrightarrow \neg \mathsf{t}(\varphi_1, \mathsf{z}_2)) \vee (\mathsf{t}(\varphi_2, \mathsf{z}_1) \leftrightarrow \neg \mathsf{t}(\varphi_2, \mathsf{z}_2)) \vee \\ & \vee \; (\mathsf{t}(\varphi_1, \mathsf{z}_1) \wedge neqv_{\varphi_1}(\mathsf{z}_1, \mathsf{z}_2)) \vee (\mathsf{t}(\varphi_2, \mathsf{z}_2) \wedge neqv_{\varphi_2}(\mathsf{z}_1, \mathsf{z}_2)) \end{aligned}$$

The binding operator in the $neqv$ is trivially, a new $neqv$ in which we associate the agent at renaming of the variable.

$$neqv_{(a,x)\varphi}(\mathsf{z}_1, \mathsf{z}_2) = neqv_{\varphi}(\mathsf{z}_1[a \mapsto \mathsf{z}_1(x)], \mathsf{z}_2[a \mapsto \mathsf{z}_2(x)])$$

From the Definition 4.3.2 is easy to note that the $neqv$ for an existential quantifier with any degree g, consists in finding a strategy x for which the two assignments z_1 and z_2 are not equivalent and at least one satisfy φ.

$$neqv_{\langle\langle x \geq g \rangle\rangle \varphi}(\mathsf{z}_1, \mathsf{z}_2) = \langle\langle x \rangle\rangle ((\mathsf{t}(\varphi, \mathsf{z}_1[x \mapsto x]) \vee \mathsf{t}(\varphi, \mathsf{z}_2[x \mapsto x])) \wedge neqv_{\varphi}(\mathsf{z}_1[x \mapsto x], \mathsf{z}_2[x \mapsto x]))$$

From the Definition 4.3.3 we know that two assignments z_1 and z_2 are equivalent if for each extension of z_1 which satisfies the formula, there is an extension of z_2 that satisfies the formula and is equivalent to z_1 and vice versa. Our function $neqv$ will consist of two disjoint then each of these will check if there is an extension of z_1 (resp., z_2) that for any extension of z_2 (resp., z_1) imply that

z_1 *and* z_2 *are not equivalent.*

$$\begin{aligned} neqv_{[\![x<g]\!]\varphi}(\mathsf{z}_1, \mathsf{z}_2) = (\langle\!\langle x' \rangle\!\rangle(\mathsf{t}(\varphi, \mathsf{z}_1[x \mapsto x'])\wedge \\ \wedge [\![x'']\!](\mathsf{t}(\varphi, \mathsf{z}_2[x \mapsto x'']) \rightarrow neqv_{\varphi}(\mathsf{z}_1[x \mapsto x'], \mathsf{z}_2[x \mapsto x'']))) \\ \vee (\langle\!\langle x' \rangle\!\rangle(\mathsf{t}(\varphi, \mathsf{z}_2[x \mapsto x'])\wedge \\ \wedge [\![x'']\!](\mathsf{t}(\varphi, \mathsf{z}_1[x \mapsto x'']) \rightarrow neqv_{\varphi}(\mathsf{z}_2[x \mapsto x'], \mathsf{z}_1[x \mapsto x'']))) \end{aligned}$$

By the above conversion, the following result can be showed.

Theorem 5.2.1 (GSG[BG]/SG[BG] Conversion)**.** *For each* GSG[BG] ∂ *of order* n*, with solution concept* φ*, there is a* SG[BG] $\partial^\star$ *of order* n *with solution concept* $\varphi^\star$ *where* $|\varphi^\star| = \mathrm{O}(2^{|\varphi|})$ *such that* ∂ *is fulfilled iff* $\partial^\star$ *is.*

By induction it is possible to prove that the transformation function t is exponentially. That is it transform the formula φ into a formula $\varphi^\star$ whose length is $\mathrm{O}(2^{|\varphi|})$. It is to easy note that, the function t when apply to the existential or universal quantifier course itself twice, ones to direct function t and the other through to $neqv$ function.

Example 5.2.1. *Consider the extension* $\mathcal{E}$ *related to the arena* $\mathcal{A}$ *depicted in Figure 3.1, the state* $\mathrm{s}_0 \in \mathrm{St}$*, and the formula* $\varphi = \langle\!\langle x \geq 3 \rangle\!\rangle [\![y < 2]\!] \langle\!\langle z \geq 1 \rangle\!\rangle (a, x)(b, y)(c, z) p_U$*. We want to build a* SG[BG] $\partial^\star$ *from* GSG[BG] ∂*. First, we compute the set* $(\mathrm{Asg}/\equiv^{p_U}_{\mathcal{E},\mathrm{s}_0})$*. To do this, we define the following sets of path:* $\mathrm{P}^{p_U}_1 \triangleq \{\pi \in \mathrm{Pth} : \mathsf{s}_0\mathsf{s}_3\mathsf{s}_5 < \pi\}$, $\mathrm{P}^{p_U}_2 \triangleq \{\pi \in \mathrm{Pth} : \mathsf{s}_0\mathsf{s}_3\mathsf{s}_1 < \pi\}$, $\mathrm{P}^{p_U}_3 \triangleq \{\pi \in \mathrm{Pth} : \mathsf{s}_0\mathsf{s}_1 < \pi\}$, $\mathrm{P}^{p_U}_4 \triangleq \{\pi \in \mathrm{Pth} : \mathsf{s}_0\mathsf{s}_2\mathsf{s}_5 < \pi\}$, $\mathrm{P}^{p_U}_5 \triangleq \{\pi \in \mathrm{Pth} : \mathsf{s}_0\mathsf{s}_2\mathsf{s}_6 < \pi\}$, $\mathrm{P}^{p_U}_6 \triangleq \{\pi \in \mathrm{Pth} : \mathsf{s}_0\mathsf{s}_4\mathsf{s}_6 < \pi\}$, $\mathrm{P}^{p_U}_7 \triangleq \{\pi \in \mathrm{Pth} : \mathsf{s}_0\mathsf{s}_4\mathsf{s}_1 < \pi\}$, $\mathrm{P}^{p_U}_8 \triangleq \{\pi \in \mathrm{Pth} : \mathsf{s}_0\mathsf{s}_7 < \pi \vee \mathsf{s}_0\mathsf{s}_2\mathsf{s}_7 < \pi\}$, $\mathrm{P}^{p_U}_9 \triangleq \{\pi \in \mathrm{Pth} : \mathsf{s}_0\mathsf{s}_5 < \pi\}$, *and* $\mathrm{P}^{p_U}_{10} \triangleq \{\pi \in \mathrm{Pth} : \mathsf{s}_0\mathsf{s}_6 < \pi\}$*. Now, we can define the set of equivalence classes with respect* p_U*:* $(\mathrm{Asg}/\equiv^{p_U}_{\mathcal{E},\mathrm{s}_0}) \triangleq \{\mathrm{C}^{p_U}_1, \mathrm{C}^{p_U}_2, \mathrm{C}^{p_U}_3, \mathrm{C}^{p_U}_4, \mathrm{C}^{p_U}_5, \mathrm{C}^{p_U}_6, \mathrm{C}^{p_U}_7, \mathrm{C}^{p_U}_8, \mathrm{C}^{p_U}_9, \mathrm{C}^{p_U}_{10}\}$*, where* $\mathrm{C}^{p_U}_i \triangleq \{\chi \in \mathrm{Asg} : \mathrm{Ag} \subseteq \mathsf{dom}(\chi) \wedge \mathsf{play}(\chi, \mathsf{s}_0) \in \mathrm{P}^{p_U}_i\}$*. It is evident that the set* $(\mathrm{Asg}/\equiv^{p_U}_{\mathcal{E},\mathrm{s}_0})$ *is finite.*

We now define the sets of path for predicate $\neg p_U$*:* $\mathrm{P}^{\neg p_U}_1 \triangleq \{\mathsf{s}_0\mathsf{s}_7\}$*,* $\mathrm{P}^{\neg p_U}_2 \triangleq \{\mathsf{s}_0\mathsf{s}_2\mathsf{s}_7\}$*, and* $\mathrm{P}^{\neg p_U}_3 \triangleq \mathrm{Pth} \setminus (\mathrm{P}^{\neg p_U}_1 \cup \mathrm{P}^{\neg p_U}_2)$*. The set of equivalence class with respect* $\neg p_U$ *is:* $(\mathrm{Asg}/\equiv^{\neg p_U}_{\mathcal{E},\mathrm{s}_0}) \triangleq \{C^{\neg p_U p_U}_1, C^{\neg p_U}_2, C^{\neg p_U}_3\}$*, where* $C^{\neg p_U}_i \triangleq \{\chi \in \mathrm{Asg} : \mathrm{Ag} \subseteq \mathsf{dom}(\chi) \wedge \mathsf{play}(\chi, \mathsf{s}_0) \in \mathrm{P}^{\neg p_U}_i\}$*. So also the set* $(\mathrm{Asg}/\equiv^{\neg p_U}_{\mathcal{E},\mathrm{s}_0})$ *is finite.*

Now we define a new target $\varphi^\star$ with respect φ. To do this we apply recursively the function t *on φ. For simplicity, we used a three sub-formulas φ_1, φ_2, and φ_3 that represent $[\![y < 2]\!]\langle\!\langle z \geq 1\rangle\!\rangle(a,x)(b,y)(c,z)p_U$, $\langle\!\langle z \geq 1\rangle\!\rangle(a,x)(b,y)(c,z)p_U$, and $(a,x)(b,y)(c,z)p_U$, respectively.*

$$\begin{aligned}\mathsf{t}(\langle\!\langle x \geq 3\rangle\!\rangle\varphi_1, \mathsf{z}) = {} & \langle\!\langle x_1\rangle\!\rangle\langle\!\langle x_2\rangle\!\rangle\langle\!\langle x_3\rangle\!\rangle(\mathit{neqv}_{\varphi_1}(\mathsf{z}[x \mapsto x_1], \mathsf{z}[x \mapsto x_2])\wedge \\ & \wedge \mathit{neqv}_{\varphi_1}(\mathsf{z}[x \mapsto x_2], \mathsf{z}[x \mapsto x_3]) \wedge \mathit{neqv}_{\varphi_1}(\mathsf{z}[x \mapsto x_1], \mathsf{z}[x \mapsto x_3]))\wedge \\ & \wedge (\mathsf{t}(\varphi_1, \mathsf{z}[x \mapsto x_1]) \wedge \mathsf{t}(\varphi_1, \mathsf{z}[x \mapsto x_2]) \wedge \mathsf{t}(\varphi_1, \mathsf{z}[x \mapsto x_3]))\end{aligned}$$

$$\mathsf{t}([\![y < 2]\!]\varphi_2, \mathsf{z}) = \mathsf{t}(\neg\langle\!\langle y \geq 2\rangle\!\rangle\neg\varphi_2, \mathsf{z})$$

$$\begin{aligned}\mathsf{t}(\neg\langle\!\langle y \geq 2\rangle\!\rangle\neg\varphi_2, \mathsf{z}) = {} & \neg(\langle\!\langle y_1\rangle\!\rangle\langle\!\langle y_2\rangle\!\rangle(\mathit{neqv}_{\neg\varphi_2}(\mathsf{z}[y \mapsto y_1], \mathsf{z}[y \mapsto y_2]))\wedge \\ & \wedge (\mathsf{t}(\neg\varphi_2, \mathsf{z}[y \mapsto y_1]) \wedge \mathsf{t}(\neg\varphi_2, \mathsf{z}[y \mapsto y_2]))\end{aligned}$$

$$\mathsf{t}(\neg\varphi_2, \mathsf{z}) = \neg\mathsf{t}(\varphi_2, \mathsf{z})$$

$$\mathsf{t}(\langle\!\langle z \geq 1\rangle\!\rangle\varphi_3, \mathsf{z}) = \langle\!\langle z_1\rangle\!\rangle\mathsf{t}(\varphi_3, \mathsf{z}[z \mapsto z_1])$$

$$\mathsf{t}((a,x)(b,y)(c,z)p_U, \mathsf{z}) = (a, \mathsf{z}(x))(b, \mathsf{z}(y))(c, \mathsf{z}(z))p_U$$

In the first execution of t *the function neqv is called. Below we illustrate the steps of the function neqv up to the base case,* i.e. *when applied to the predicate p_U.*

$$\begin{aligned}\mathit{neqv}_{[\![y<2]\!]\varphi_2}(\mathsf{z}_1, \mathsf{z}_2) = {} & (\langle\!\langle y'\rangle\!\rangle(\mathsf{t}(\varphi_2, \mathsf{z}_1[y \mapsto y'])\wedge \\ & \wedge [\![y'']\!](\mathsf{t}(\varphi_2, \mathsf{z}_2[y \mapsto y'']) \to \mathit{neqv}_{\varphi_2}(\mathsf{z}_1[y \mapsto y'], \mathsf{z}_2[y \mapsto y'']))))\vee \\ & \vee (\langle\!\langle y'\rangle\!\rangle(\mathsf{t}(\varphi_2, \mathsf{z}_2[y \mapsto y'])\wedge \\ & \wedge [\![y'']\!](\mathsf{t}(\varphi_2, \mathsf{z}_1[y \mapsto y'']) \to \mathit{neqv}_{\varphi_2}(\mathsf{z}_1[y \mapsto y''], \mathsf{z}_2[y \mapsto y']))))\end{aligned}$$

$$\mathit{neqv}_{\langle\!\langle z\geq1\rangle\!\rangle\varphi_3}(\mathsf{z}_1, \mathsf{z}_2) = \langle\!\langle z\rangle\!\rangle((\mathsf{t}(\varphi_3, \mathsf{z}_1[z \mapsto z]) \vee \mathsf{t}(\varphi_3, \mathsf{z}_2[z \mapsto z])) \wedge \mathit{neqv}_{\varphi_3}(\mathsf{z}_1[z \mapsto z], \mathsf{z}_2[z \mapsto z]))$$

$$neqv_{(a,x)(b,y)(c,z)p_U}(\mathsf{z}_1,\mathsf{z}_2) = neqv_{(b,y)(c,z)p_U}(\mathsf{z}_1[a \mapsto \mathsf{z}_1(x)], \mathsf{z}_2[a \mapsto \mathsf{z}_2(x)])$$

$$neqv_{(b,y)(c,z)p_U}(\mathsf{z}_1,\mathsf{z}_2) = neqv_{(c,z)p_U}(\mathsf{z}_1[b \mapsto \mathsf{z}_1(y)], \mathsf{z}_2[b \mapsto \mathsf{z}_2(y)])$$

$$neqv_{(c,z)p_U}(\mathsf{z}_1,\mathsf{z}_2) = neqv_{p_U}(\mathsf{z}_1[c \mapsto \mathsf{z}_1(z)], \mathsf{z}_2[c \mapsto \mathsf{z}_2(z)])$$

$$neqv_{p_U}(\mathsf{z}_1,\mathsf{z}_2) = \bigwedge_{1\leq k\leq 10} \neg((a,\mathsf{z}_1(a))(b,\mathsf{z}_1(b))(c,\mathsf{z}_1(c))q_k^{p_U}) \wedge ((a,\mathsf{z}_2(a))(b,\mathsf{z}_2(b))(c,\mathsf{z}_2(c))q_k^{p_U})$$

Another invocation of the function $neqv$ *is performed by* t *and in the following we illustrate the recursive results.*

$$\begin{aligned} neqv_{[\![z<1]\!]\neg\varphi_3}(\mathsf{z}_1,\mathsf{z}_2) = {} & (\langle\!\langle z'\rangle\!\rangle(\mathsf{t}(\neg\varphi_3, \mathsf{z}_1[z \mapsto z'])\wedge \\ & \wedge [\![z'']\!](\mathsf{t}(\neg\varphi_3, \mathsf{z}_2[z \mapsto z'']) \rightarrow neqv_{\neg\varphi_3}(\mathsf{z}_1[z \mapsto z'], \mathsf{z}_2[z \mapsto z''])))\vee \\ & \vee (\langle\!\langle z'\rangle\!\rangle(\mathsf{t}(\neg\varphi_3, \mathsf{z}_2[z \mapsto z'])\wedge \\ & \wedge [\![z'']\!](\mathsf{t}(\neg\varphi_3, \mathsf{z}_1[z \mapsto z'']) \rightarrow neqv_{\neg\varphi_3}(\mathsf{z}_1[z \mapsto z''], \mathsf{z}_2[z \mapsto z']))) \end{aligned}$$

$$neqv_{(a,x)(b,y)(c,z)\neg p_U}(\mathsf{z}_1,\mathsf{z}_2) = neqv_{(b,y)(c,z)\neg p_U}(\mathsf{z}_1[a \mapsto \mathsf{z}_1(x)], \mathsf{z}_2[a \mapsto \mathsf{z}_2(x)])$$

$$neqv_{(b,y)(c,z)\neg p_U}(\mathsf{z}_1,\mathsf{z}_2) = neqv_{(c,z)\neg p_U}(\mathsf{z}_1[b \mapsto \mathsf{z}_1(y)], \mathsf{z}_2[b \mapsto \mathsf{z}_2(y)])$$

$$neqv_{(c,z)\neg p_U}(\mathsf{z}_1,\mathsf{z}_2) = neqv_{\neg p_U}(\mathsf{z}_1[c \mapsto \mathsf{z}_1(z)], \mathsf{z}_2[c \mapsto \mathsf{z}_2(z)])$$

$$neqv_{\neg p_U}(\mathsf{z}_1,\mathsf{z}_2) = \bigwedge_{1\leq k\leq 3} \neg((a,\mathsf{z}_1(a))(b,\mathsf{z}_1(b))(c,\mathsf{z}_1(c))q_k^{\neg p_U}) \wedge ((a,\mathsf{z}_2(a))(b,\mathsf{z}_2(b))(c,\mathsf{z}_2(c))q_k^{\neg p_U})$$

Finally, the function $neqv$ *invokes the function* t. *Below we illustrate the steps of the recursive function* t.

$$\mathsf{t}(\neg(a,x)(b,y)(c,z)p_U, \mathsf{z}) = \neg((a,\mathsf{z}(x))(b,\mathsf{z}(y))(c,\mathsf{z}(z))p_U)$$

6 Determinacy and Fulfilling

In this Chapter, we define a reformulation of the determinacy for graded strategy game, with turn-based arena and one-goal solution concept, and describe the solution of the related fulfilling problem. First, in Section 6.1, we prove the determinacy for GSG[1G, TB] in the case of 2 variables and open and closed predicates. Then, in Section 6.2, we solve the fulfilling problem for GSG[1G, 2VAR] with reachability and safety predicates that is PTIME-COMPLETE in the size of both the arena and the solution concept.

6.1 Determinacy

In this section, we check for the determinacy property regarding GSG[1G, TB]. It is worth recalling that determinacy has been first introduced for Borelian turned-based games in [Mar75] and [Mar85] and the proof considered for that setting does not apply to to our concurrent setting for GSG[1G, TB]. To give an evidence of the differences between the two framework, it is useful to observe that in SG, a formula of the kind $\langle\langle x\rangle\rangle [[y]]\eta$ implies $[[y]]\langle\langle x\rangle\rangle\eta$ as well, while for concurrent GSG this is not necessary the case. The determinacy property we are interested in makes use of the following equivalent transformation over the formulas: let $\varphi = \wp\eta$ be a formula with a generic quantification prefix $\wp$, we can transform φ in an equivalent formula with $\wp'\eta$, where $\wp'$ is a prefix in the specific form a sequence of existential quantifications followed by one of universal quantifications. Moreover, the order of the quantifications in $\wp$ is preserved in $\wp'$. Here, we restrict our attention to the case of 2 variables and open and closed predicates only. In particular, we extend the classic Gale-Stewart Theorem [PP04], by exploiting a deep generalization of the technique used in [FNP10], which constituted of a fixed-point calculation over the number of winning strategies an agent can select against all but a fixed number of those of its opponent. Regarding [FNP10], we remind that their counting is restricted to the existential agent only.

To help the reader, we now give an outline of this section. First, we introduce the construction of the grading function, which is used to determine how many different strategies an agent can have *w.r.t.* a fixed number of strategies to avoid for the opponent. Then, we give two lemmas that provide the fundamental properties of grading function. Finally, we use the latter to prove the determinacy property.

Construction 6.1.1 (Grading Function). *Consider a turn-based two-agent extension $\mathcal{E}$ with $\mathrm{Ag} = \{a, \overline{a}\}$. Moreover, let $s \in \mathrm{St}$ be one of its states and $p \in \mathrm{Pr}$ an open or closed predicate with witness $\mathrm{W} \subseteq \mathrm{Hst}(s)$. If p is open, assume $\mathrm{X} \triangleq \{\rho \in \mathrm{Hst} : \exists \rho' \in \mathrm{Hst} \,.\, \rho' \leq \rho \wedge \rho' \in \mathrm{W}\}$ as the set of immediate winning histories, and $\mathrm{Y} \triangleq \{\rho \in \mathrm{Hst} \setminus \mathrm{X} : \forall \rho' \in \mathrm{Hst} . \rho \leq \rho' \Rightarrow \rho' \notin \mathrm{W}\}$ as the set of immediate loosing ones. Dually, if p is closed, define $\mathrm{Y} \triangleq \{\rho \in \mathrm{Hst} : \exists \rho' \in \mathrm{Hst} \,.\, \rho' \leq \rho \wedge \rho' \notin \mathrm{W}\}$ and $\mathrm{X} \triangleq \{\rho \in \mathrm{Hst} \setminus \mathrm{Y} : \forall \rho' \in \mathrm{Hst} . \rho \leq \rho' \Rightarrow \rho' \in \mathrm{W}\}$. It is not hard to see that, in case $s \in \mathrm{X} \cup \mathrm{Y}$, all strategy profiles starting in s are equivalent* w.r.t. *both p and $\neg p$. If $s \in \mathrm{Z} \triangleq \mathrm{Hst} \setminus (\mathrm{X} \cup \mathrm{Y})$, instead, we need to introduce a* grading function $\mathsf{G}_p^\alpha : \mathrm{Z} \to \Gamma$, *where $\Gamma \triangleq \mathbb{N} \to (\mathbb{N} \cup \{\omega\})$, that allows to determine how many different strategies a given agent $\alpha \in \mathrm{Ag}$ owns* w.r.t. *the predicate p. Informally, $\mathsf{G}_p^\alpha(\rho)(j)$ denotes the number of winning strategies α can put up against all but at most j strategies of its adversary $\overline{\alpha}$, once the current play starting from s has already reached the history $\rho \in \mathrm{Z}$. To formalize this function, we need to introduce the auxiliary set of α-histories $\mathrm{S} \triangleq \{\rho \in \mathrm{Z} : \exists \rho' \in \mathrm{Y} \,.\, \rho < \rho' \wedge \forall \rho'' \in \mathrm{Hst} \,.\, \rho \leq \rho'' < \rho' \Rightarrow \rho'' \in \mathrm{Hst}_\alpha\}$, where $\mathrm{Hst}_\alpha = \{\forall \rho \in \mathrm{Hst} : \mathsf{ag}(\mathsf{lst}(\rho)) = \{\alpha\}\}$, from which this agent has the possibility to "suicide", independently from the behavior of its opponent. With more details, α can autonomously extend an history $\rho \in \mathrm{S}$ into one $\rho' \in \mathrm{Y}$ that is surely loosing. Observe that there may be several suicide strategies, but all of them are equivalent* w.r.t. *the predicate p. Also, against them, all counter strategies of $\overline{\alpha}$ are equivalent* *as well. At this point, we introduce the functor* $\mathsf{F}_p^\alpha : (\mathrm{Z} \to \Gamma) \to (\mathrm{Z} \to \Gamma)$*, whose least fixpoint represents a function returning the maximum number of different strategies* α *can use against all but a given fixed number of counter strategies of* $\overline{\alpha}$*. Formally, we have that:*

$$\mathsf{F}_p^\alpha(\mathsf{f})(\rho)(i) \triangleq \begin{cases} \sum_{\rho' \in \mathsf{suc}(\rho) \cap \mathrm{Z}} \mathsf{f}(\rho')(0) + |\mathsf{suc}(\rho) \cap \mathrm{X}|, & \textit{if } \rho \in \mathrm{Hst}_\alpha \textit{ and } i = 0; \\ \sum_{\rho' \in \mathsf{suc}(\rho) \cap \mathrm{Z}} \mathsf{f}(\rho')(i), & \textit{if } \rho \in \mathrm{Hst}_\alpha \textit{ and } i > 0; \\ \sum_{\mathsf{c} \in \mathrm{C}(\rho)(i)} \prod_{\rho' \in \mathsf{dom}(\mathsf{c})} \mathsf{f}(\rho')(\mathsf{c}(\rho')), & \textit{otherwise}; \end{cases}$$

where $\mathsf{suc}(\rho) = \{\rho' \in \mathrm{Hst} : \exists s \in \mathrm{St}.\ \rho s = \rho'\}$ *and* $\mathsf{C}(\rho)(i) \subseteq (\mathsf{suc}(\rho) \cap \mathrm{Z}) \rightharpoonup \mathbb{N}$ *contains all partial functions* $\mathsf{c} \in \mathsf{C}(\rho)(i)$ *for which, on the histories not in their domains,* α *owns a suicide strategy,* i.e., $(\mathsf{suc}(\rho) \cap \mathrm{Z}) \setminus \mathsf{dom}(\mathsf{c}) \subseteq \mathrm{S}$*, and the sum of all values assumed by* c *plus the number of successor histories that are neither surely winning nor contained in the domain of* c *equals to* i*,* i.e., $i = \sum_{\rho' \in \mathsf{dom}(\mathsf{c})} \mathsf{c}(\rho') + |\mathsf{suc}(\rho) \setminus (\mathrm{X} \cup \mathsf{dom}(\mathsf{c}))|$*. Intuitively, the first item of the definition simply asserts that the number of strategies* $\mathsf{F}_p^\alpha(\mathsf{f})(\rho)(0)$ *that* α *has on the* α*-history* ρ*, without excluding any counter strategy, is the sum of the* $\mathsf{f}(\rho')(0)$ *strategies on the successor histories* $\rho' \in \mathrm{Z}$ *plus a single strategy for each successor history that is surely winning. Similarly, the second item takes into account the case in which we can avoid exactly* i *counter strategies. The last item, instead, computes the number of strategies for* α *on the* $\overline{\alpha}$*-history* ρ*. In particular, through the set* $\mathsf{C}(\rho)(i)$*, it first determines in how many ways it is possible to split the number* i *of counter strategies to avoid among all successor strategies of* ρ*. Then, for each of these splittings, it calculates the product of the corresponding numbers* $\mathsf{f}(\rho')(\mathsf{c}(\rho'))$ *of strategies for* α*. We can finally define the grading function* G_p^α *by means of the least fixpoint* $\mathsf{f}^\star = \mathsf{F}_p^\alpha(\mathsf{f}^\star)$ *of* F_p^α *as follows:* $\mathsf{G}_p^\alpha(\rho)(j) \triangleq \sum_{h=0}^{j} \mathsf{f}^\star(\rho)(h) + [\rho \in \mathrm{S} \wedge j \geq 1]$*. Intuitively,* $\mathsf{G}_p^\alpha(\rho)(j)$ *is the sum of the numbers* $\mathsf{f}^\star(\rho)(h)$ *of winning strategies* α *can exploit against all but exactly* h *strategies of its adversary* $\overline{\alpha}$*, for each* $h \in [0, j]$*. Moreover, if* $\rho \in \mathrm{S}$*, we need to add to this counting the suicide strategy that* α *can use once* $\overline{\alpha}$ *avoids to apply his unique counter strategy. This is formalized through the function* $[\eth]$ *that is evaluated to* 1*, if the condition* $\eth$ *is true, and to* 0*, otherwise.*

Thanks to the above construction, one can compute the maximum number of strategies that a player has against a fixed number of strategies to avoid for the opponent. Next lemma, whose statement can be proved constructively, precisely describe this fact, showing how the fulfilling of a game can be decided via the grading function. Observe that, by applying the normalization reasoning of Subsection 5.1.1, this construction extends to the setting of n agents with only 2 variables.

Lemma 6.1.1 (Grading Function). *Let $\eth = \langle \mathcal{E}, s_I, \varphi \rangle$ be a* $\mathrm{GSG}[1\textsc{g}, \textsc{tb}, 2\textsc{var}]$ *having solution concept $\varphi = \langle\langle x \geq i \rangle\rangle [[\overline{x} \leq j]] \flat p$, where the predicate $p \in \mathrm{Pr}$ is either open or closed. Moreover, let $\mathrm{X}, \mathrm{Y}, \mathrm{Z} \subseteq \mathrm{Hst}$ be the sets obtained in Construction 6.1.1. Then, $\eth$ is fulfilled iff one of the following three conditions hold:* (i) *$i = 1$, $j = 0$, and $s_I \in \mathrm{X}$;* (ii) *$i = 1$, $j = 1$, and $s_I \in \mathrm{Y}$;* (iii) *$i \leq \mathsf{G}_p^x(s_I)(j)$, and $s_I \in \mathrm{Z}$.*

A fundamental property of the grading function is its duality in the form described in the next lemma, which can be proved by induction on the recursive structure of G itself. To give an intuition, assume that an agent $\overline{x}$ has at most j strategies to satisfy the predicate $\neg p$ against at most i strategies to avoid for its adversary x. Then, it can be shown that the latter has more than i strategies to satisfy the predicate p against at most j strategies to avoid for the former.

Lemma 6.1.2 (Grading Duality). *For all histories $\rho \in \mathrm{Z}$ and indexes $i, j \in \mathbb{N}$, it holds that if $\mathsf{G}_{\neg p}^{\overline{x}}(\rho)(i) \leq j$ then $i < \mathsf{G}_p^x(\rho)(j)$.*

Summing up the above results, we can easily prove that the $\mathrm{GSG}[1\textsc{g}, \textsc{tb}, 2\textsc{var}]$ with open or closed predicates are determined, as stated in the following theorem. Indeed, suppose that $s_I \in \mathrm{Z}$ and $\mathcal{E}, s_I \models [[\overline{x} \leq j]] \langle\langle x \geq i \rangle\rangle \flat p$. Obviously, $\mathcal{E}, s_I \not\models \langle\langle \overline{x} \geq j+1 \rangle\rangle [[x \leq i-1]] \flat \neg p$. Consequently, by Lemma 6.1.1, we have that $\mathsf{G}_{\neg p}^{\overline{x}}(s_I)(i-1) \leq j$. Hence, by Lemma 6.1.2, it follows that $i \leq \mathsf{G}_p^x(s_I)(j)$. Finally, again by Lemma 6.1.1, we obtain that $\mathcal{E}, s_I \models \langle\langle x \geq i \rangle\rangle [[\overline{x} \leq j]] \flat p$, as required by the definition of determinacy.

Theorem 6.1.1 (Determinacy). *The* $\mathrm{GSG}[1\textsc{g}, \textsc{tb}, 2\textsc{var}]$ *over open or closed extensions are determined.*

Example 6.1.1. *Consider the extension $\mathcal{E}$ related to the arena $\mathcal{A}$ depicted in Figure 6.1, the state* $\mathrm{s}_0 \in \mathrm{St}$*, and the open predicate* p_U *and a formula* $\varphi = \langle\langle x \leq g_1 \rangle\rangle [\![y < g_2]\!] \flat p_U$*, with* $\flat = (a, x)(b, y)$*.* $\mathrm{Pr} = \{p_U\}$*, the open predicate* p_U *has witness* U *containing those histories that pass in* s_4 *or* s_7*, or* s_9*. Since* p_U *is open, the set of histories* X *is* $s_0 \cdot s_3 \cdot s_7^+ + s_0 \cdot (s_1 \cdot s_5)^+ \cdot s_8^+ \cdot s_9 \cdot (s_8 + s_9)^* + s_0 \cdot s_1 \cdot ((s_5 \cdot s_1)^* \cdot s_4)^+ \cdot (\epsilon + s_5 + s_5 \cdot (s_1 + s_8 \cdot (s_8 + s_9)^*))$*, while* $\mathrm{Y} \triangleq s_0 \cdot (s_2^+ \cdot s_6^* + s_3 \cdot s_6^+)$*. The set* Z *contains* $s_0 + s_0 \cdot s_3 + s_0 \cdot (s_1 \cdot s_5)^* \cdot s_1 + s_0 \cdot (s_1 \cdot s_5)^+ + s_0 \cdot (s_1 \cdot s_5)^+ \cdot s_8^+$*. Finally, the set of suicide strategies is* $s_0 + s_0 \cdot s_3$*.*

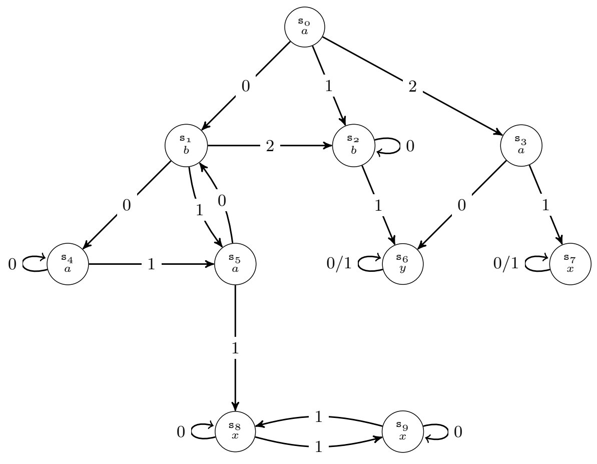

Figure 6.1: Turn-based Arena $\mathcal{A}$.

Now, we evaluate the results of function f *for each histories in* Z*. First, we set* $\mathsf{f}_0(\rho)(j) = 0$*,* $\forall \rho \in \mathrm{Hst}$ *and* $\forall j \geq 0$*. For all* $k > 0$*,* $i \geq 0$*, and the history* $s_0 s_3$*, we have that*

$$\mathsf{f}_k(s_0 s_3)(i) \triangleq \begin{cases} 0, & \textit{if } i > 0; \\ 1, & \textit{otherwise}. \end{cases}$$

For all $k > 0$*,* $i \geq 0$*, and* $\rho \in s_0 \cdot (s_1 \cdot s_5)^+ \cdot s_8^+$*, we have that*

$$\mathsf{f}_k(\rho)(i) \triangleq \begin{cases} 0, & \textit{if } i > 0; \\ k, & \textit{otherwise}. \end{cases}$$

For all $k > 0$*,* $i \geq 0$*, and* $\rho \in s_0 \cdot (s_1 \cdot s_5)^+$*, we have that*

$$\mathsf{f}_k(\rho)(i) \triangleq \begin{cases} 0, & \textit{if } k < (i \cdot 2) + 1; \\ k - ((i \cdot 2) + 1), & \textit{otherwise}. \end{cases}$$

For all $k > 0$, $i \geq 0$, and $\rho \in s_0 \cdot (s_1 \cdot s_5)^* \cdot s_1$, we have that

$$\mathsf{f}_k(\rho)(i) \triangleq \begin{cases} 0, & \textit{if } k < (i \cdot 2) \textit{ or } i = 0; \\ k - (i \cdot 2), & \textit{otherwise.} \end{cases}$$

For all $k > 0$, $i \geq 0$, and the history s_0, we have that

$$\mathsf{f}_k(s_0)(i) \triangleq \begin{cases} 0, & \textit{if } k < (i \cdot 2) + 1 \textit{ and } i > 0 \textit{ or } k < 2 \textit{ and } i = 0; \\ 1, & \textit{if } k \geq 2 \textit{ and } i = 0; \\ k - ((i \cdot 2) + 1), & \textit{otherwise.} \end{cases}$$

Now, we illustrate the results of fixpoint $\mathsf{f}^\star$. For all $i \geq 0$ and the history $s_0 s_3$, we have that

$$\mathsf{f}^\star(s_0 s_3)(i) \triangleq \begin{cases} 0, & \textit{if } i > 0; \\ 1, & \textit{otherwise.} \end{cases}$$

For all $i \geq 0$ and $\rho \in s_0 \cdot (s_1 \cdot s_5)^+ \cdot s_8^+$, we have that

$$\mathsf{f}^\star(\rho)(i) \triangleq \begin{cases} 0, & \textit{if } i > 0; \\ \omega, & \textit{otherwise.} \end{cases}$$

For all $i \geq 0$ and $\rho \in s_0 \cdot (s_1 \cdot s_5)^+$, we have that

$$\mathsf{f}^\star(\rho)(i) \triangleq \omega$$

For all $i \geq 0$ and $\rho \in s_0 \cdot (s_1 \cdot s_5)^* \cdot s_1$, we have that

$$\mathsf{f}^\star(\rho)(i) \triangleq \begin{cases} 0, & \textit{if } i = 0; \\ \omega, & \textit{otherwise.} \end{cases}$$

For all $k > 0$, $i \geq 0$, and history the s_0, we have that

$$\mathsf{f}^\star(s_0)(i) \triangleq \begin{cases} 1, & \textit{if } i = 0; \\ \omega, & \textit{otherwise.} \end{cases}$$

Finally, we evaluate the results of grading function. For all $j \geq 0$ and the history $s_0 s_3$, we have that

$$\mathsf{G}^{a}_{pU}(s_0 s_3)(j) \triangleq \begin{cases} 1, & \textit{if } j = 0; \\ 2, & \textit{otherwise}. \end{cases}$$

For all $j \geq 0$ *and* $\rho \in s_0 \cdot (s_1 \cdot s_5)^{+} \cdot s_8^{+}$, *we have that*

$$\mathsf{G}^{a}_{pU}(\rho)(j) \triangleq \omega$$

For all $\rho \in s_0 \cdot (s_1 \cdot s_5)^{+} \cup s_0 \cdot (s_1 \cdot s_5)^{*} \cdot s_1 \cup \{s_0\}$, *we have the same result of function* $\mathsf{f}^{\star}$, i.e., $\mathsf{G}^{a}_{pU}(\rho)(j) \triangleq \mathsf{f}^{\star}(\rho)(j)$ *for all* $j \geq 0$.

6.2 The Fulfilling Problem

We finally describe the solution of the fulfilling problem for GSG[1G, 2VAR] with reachability and safety predicates. Here, we just give the intuitive idea behind the polynomial-time decision procedure for the turn-based case, since having concurrent game can we transformed in turn-based one by means previous construction. From a high-level point of view, one may think that, since reachability and safety predicates are simple cases of open and closed ones, a slight variant of the fixpoint procedure previously described for the determinacy proof may work as a tool to determine the degrees with which a given GSL[1G, 2VAR] is satisfied. In particular, since these predicates are prefix independent, one can substitutes histories with states in the above formalization obtaining a completely equivalent procedure. Unfortunately, since the set Γ is infinite, the functional domain on which the functor F^{α}_{p} operates is infinite as well. Therefore, the naive calculation of the fixpoint may not terminate in a finite number of steps. To avoid this problem, we exploit a technical trick already used in [FNP10] to decide the model-checking problem for graded ATL. Let $\mathrm{X} \subseteq \mathrm{St}$ be the set of states the existential agents want to reach and $\mathrm{Y} \triangleq \{s \in \mathrm{St} : \mathcal{E}, s \models \mathtt{AG}\neg \mathrm{X}\}$ that containing those states from which it is impossible to reach X. Intuitively, X and Y represent the immediate winning and loosing positions of the game, respectively. Now, decompose the remaining state space $\mathrm{Z} \triangleq \mathrm{St} \setminus (\mathrm{X} \cup \mathrm{Y})$ into strong connected components. Then, starting with the constant function $\mathsf{f}_0 \in \mathrm{Z} \to \Gamma$ with $\mathsf{f}_0(s)(j) \triangleq 0$, for all $s \in \mathrm{Z}$ and $j \in \mathbb{N}$, apply the following procedure by induction on the classic partial order among the obtained components. Let $\mathrm{H} \subseteq \mathrm{Z}$ be a component not yet analyzed. If H is trivial, simply compute $\mathsf{f}_{k+1} \triangleq \mathsf{F}^{\alpha}_{p}(\mathsf{f}_k)$, where f_k is the function obtained from the previous iteration. Otherwise, let $\mathsf{f}' \triangleq (\mathsf{F}^{\alpha}_{p})^{n}(\mathsf{f}_k)$, where $n \in \mathbb{N}$ is the length of the longest simple path in H. Now, for all $s \in \mathrm{Z}$, define the new function f_{k+1} as follows. If $s \notin \mathrm{H}$ then $\mathsf{f}_{k+1}(s) \triangleq \mathsf{f}_k(s)$ else $\mathsf{f}_{k+1}(s)(j) \triangleq 0$, if $\mathsf{f}'(s)(j) = 0$, and $\mathsf{f}_{k+1}(s)(j) \triangleq \omega$, otherwise, for all $j \in \mathbb{N}$. Intuitively, if the existential agent has at least one strategy to reach X from a state s, once its opponent avoids j counter strategies, it is also able to construct an infinite number of non-equivalent winning strategies by using the closed paths in H passing through s. At this point, it is not hard to see that the following result holds, by assuming that each basic operation on numbers is performed in constant time.

Theorem 6.2.1 (Fulfilling Complexity). *The fulfilling problem of* $\mathrm{GSG}[1\mathrm{G}, 2\mathrm{VAR}]$ *for reachability or safety extensions is* PTIME-COMPLETE *in the size of both the arena and the solution concept.*

Example 6.2.1. *Consider the extension $\mathcal{E}$ related to the arena $\mathcal{A}$ depicted in Figure 6.1, the state* $\mathrm{s}_0 \in \mathrm{St}$, *and a formula $\varphi = \langle\langle x \leq g_1 \rangle\rangle [[y < g_2]] \flat p_U$, with $\flat = (a, x)(b, y)$ and p_U is an open predicate. The set of predicates is* $\mathrm{Pr} = \{p_U\}$, *where the predicate p_U has witness* U *containing those histories that pass in s_4 or s_7, or s_9. The set of states that the existential agent want to reach is* $\mathrm{X} \triangleq \{s_4, s_7, s_9\}$, *while the set of states from which it is impossible to reach* X *is* $\mathrm{Y} \triangleq \{s_2, s_6\}$. *The set of suicide is empty and the set* Z *contains* $\{s_0, s_1, s_3, s_5, s_8\}$. *There are four strong connected components* C_i, *where* $\mathrm{C}_1 = \{s_0\}$, $\mathrm{C}_2 = \{s_1, s_5\}$, $\mathrm{C}_3 = \{s_3\}$, $\mathrm{C}_4 = \{s_8\}$. *Now, we introduce a partial order $\prec$ on strong connected components. For all $i, j \in \mathbb{N}$, $\mathrm{C}_i \prec \mathrm{C}_j$ iff there exist a path π that from a state in C_j arrives to a state in C_i. So, we have that* $\mathrm{C}_2 \prec \mathrm{C}_1$, $\mathrm{C}_3 \prec \mathrm{C}_1$, $\mathrm{C}_4 \prec \mathrm{C}_2$, *and for transitivity relation* $\mathrm{C}_4 \prec \mathrm{C}_1$. *For apply the algorithm, we define an arbitrary total order $\prec'$, with respect the partial order describe above, in which* $\mathrm{C}_4 \prec' \mathrm{C}_3 \prec' \mathrm{C}_2 \prec' \mathrm{C}_1$. *First, we start analyze* C_4 *that is not a trivial component, so* $\mathsf{f}' \triangleq (\mathsf{F}^{a}_{p_U})^{1}(\mathsf{f}_0)$. *We have that* $\mathsf{F}^{a}_{p_U}(\mathsf{f}_0)(s_8)(0) \triangleq \sum_{\rho' \in \mathsf{suc}(s_8) \cap \mathrm{Z}} \mathsf{f}_0(\rho')(0) + |\mathsf{suc}(s_8) \cap \mathrm{X}|$, *where* $\mathsf{suc}(s_8) \cap \mathrm{Z} = \emptyset$ *then the summation is* 0, *while* $\mathsf{suc}(s_8) \cap \mathrm{X} = \{s_9\}$ *therefore* $\mathsf{f}'(s_8)(0) = 1$. *Since* $\mathsf{f}'(s_8)(0) \neq 0$ *then* $\mathsf{f}_1(s_8)(0) \triangleq \omega$. *Conversely,* $\mathsf{f}'(s_8)(j) = 0$, *with* $j \geq 1$. *Second, we analyze* C_3 *that a trivial component, so* $\mathsf{f}_2 \triangleq \mathsf{F}^{a}_{p_U}(\mathsf{f}_1)$. *We have that* $\mathsf{F}^{a}_{p_U}(\mathsf{f}_1)(s_3)(0) \triangleq \sum_{\rho' \in \mathsf{suc}(s_3) \cap \mathrm{Z}} \mathsf{f}_1(\rho')(0) + |\mathsf{suc}(s_3) \cap \mathrm{X}|$, *where* $\mathsf{suc}(s_3) \cap \mathrm{Z} = \emptyset$ *then the summation is* 0, *while* $\mathsf{suc}(s_3) \cap \mathrm{X} = \{s_7\}$ *therefore* $\mathsf{f}_2(s_3)(0) = 1$. *Conversely,* $\mathsf{F}^{a}_{p_U}(\mathsf{f}_1)(s_3)(j) = 0$, *with* $j \geq 1$. *Third, we analyze* C_2 *that is not a trivial component, so* $\mathsf{f}' \triangleq (\mathsf{F}^{a}_{p_U})^{2}(\mathsf{f}_2)$. *For the constraint on j, we have that* $j = \sum_{\rho' \in \mathsf{dom}(\mathsf{c})} \mathsf{c}(\rho') + |\mathsf{suc}(s_1) \setminus (\mathrm{X} \cup \mathsf{dom}(\mathsf{c}))|$, *where* $\mathsf{suc}(s_1) \setminus (\mathrm{X} \cup \mathsf{dom}(\mathsf{c})) = \{s_2\}$ *then* j *must have greater than* 0. *With* s_1 *and* $j \geq 1$, *we have that* $\mathsf{F}^a_{pU}(\mathsf{f}_2)(s_1)(j) \triangleq \sum_{\mathsf{c} \in \mathrm{C}(s_1)(j)} \prod_{\rho' \in \mathsf{dom}(\mathsf{c})} \mathsf{f}(\rho')(\mathsf{c}(\rho'))$, *where* $\mathsf{dom}(\mathsf{c}) = \{s_5\}$, *but* $\mathsf{f}_2(s_5)(\mathsf{c}(s_5)) = 0$ *for all* $\mathsf{c}(s_5) \in \mathbb{N}$. *So,* $\mathsf{f}'(s_1)(j) = 0$ *then* $\mathsf{f}_3(s_1)(j) = 0$. $\mathsf{F}^a_{pU}(\mathsf{f}_2)(s_5)(0) \triangleq \sum_{\rho' \in \mathsf{suc}(s_5) \cap \mathrm{Z}} \mathsf{f}_2(\rho')(0) + |\mathsf{suc}(s_5) \cap \mathrm{X}|$, *where* $\mathsf{suc}(s_5) \cap \mathrm{X} = \emptyset$, *while* $\mathsf{suc}(s_5) \cap \mathrm{Z} = \{s_1, s_8\}$. *Thus,* $\mathsf{f}_2(s_1)(0) = 0$ *and* $\mathsf{f}_2(s_8)(0) = \omega$ *then* $\mathsf{f}_3(s_5)(0) = \omega$. *Conversely,* $\mathsf{f}_3(s_5)(j) = 0$, *with* $j \geq 1$. *Now, we reapply the function for a second iteration. For* s_1 *and* $j = 1$, *we have that* $\mathsf{f}_3(s_5)(0) = \omega$ *then* $\mathsf{f}'(s_1)(1) = \omega$. *For* s_1 *and for all* $j > 1$, *we have that* $\mathsf{f}_3(s_5)(j-1) = 0$ *then* $\mathsf{f}'(s_1)(j) = 0$. *For* s_5 *the results remain the same of the first iteration. Finally, we analyze* C_1 *that a trivial component, so* $\mathsf{f}_4 \triangleq \mathsf{F}^a_{pU}(\mathsf{f}_3)$. *We have that* $\mathsf{F}^a_{pU}(\mathsf{f}_3)(s_0)(0) \triangleq \sum_{\rho' \in \mathsf{suc}(s_0) \cap \mathrm{Z}} \mathsf{f}_3(\rho')(0) + |\mathsf{suc}(s_0) \cap \mathrm{X}|$, *where* $\mathsf{suc}(s_0) \cap \mathrm{X} = \emptyset$, *while* $\mathsf{suc}(s_0) \cap \mathrm{Z} = \{s_1, s_3\}$. *Thus,* $\mathsf{f}_3(s_1)(0) = 0$ *and* $\mathsf{f}_3(s_3)(0) = 1$ *then* $\mathsf{f}_4(s_0)(0) = 1$. *For* $j = 1$, *we have that* $\mathsf{f}_3(s_1)(1) = \omega$ *and* $\mathsf{f}_3(s_3)(1) = 0$ *then* $\mathsf{f}_4(s_0)(1) = \omega$. *For* $j \geq 1$, *we have that* $\mathsf{f}_3(s_1)(j) = 0$ *and* $\mathsf{f}_3(s_3)(j) = 0$ *then* $\mathsf{f}_4(s_0)(j) = 0$. *Then,* $\mathcal{E}, \mathsf{s}_0 \models \varphi$ *with* $(g_1, g_2) = (1, 1)$ *or* $(g_1, g_2) = (\omega, 2)$.

7

Conclusion

In multi-agent strategic reasoning, SL has taken in recent years a key role as it allows to evaluate the strategies agents as first-order objects.From an expressive point of view, SL extends the temporal logic ATL*, which in turn extends CTL*. In CTL* one can specify whether there exist an execution or all execution of a system model satisfy a property. In ATL*, paths are given as an outcome of actions (strategies) taken by teams of players while playing existentially or universally. Furthermore, in SL strategies are abstract objects, in principle not associated to any player, and by means of a binding operators they can be associate to, reused or shared by specific agents. Also in SL strategies are taken to hold existentially or universally over the considered system plays.

The high power of SL has been shown to come at a price, as the model checking becomes non-elementary and the satisfiability problem becomes even undecidable. We recall that for ATL* both problems are just 2-EXPTIME-COMPLETE.

A deep study of SL has shown that this extra complexity resides in the fact that it admits "non-behavioral strategies", that is a choice an agent can make in a specific moment of the game may depend on choices made by other players in completely different (i.e., future or counterfactual) places. This has led to investigate several syntactic fragments of SL such as the one goal fragment

SG[1G] and the Boolean one SG[BG].

In particular, it has been shown that SG[1G] retains all theoretical properties of ATL* as well as the same complexity for the related decision problems. SG[BG], instead is not elementary, the satisfiability problem is not decidable and the exact complexity of the model checking question remains an open problem.

CTL*, ATL* and SL,along with all its fragments, have been extensively studied and applied in formal verification. Formal verification tools based on this formalisms have been also developed and shown to be useful in practice. Remarkably CTL* and ATL* have been investigated in several interesting extensions and among the others along graded modalities used in place of the classical existential and universal modalities. In CTL*, graded modalities allow to refine the number of paths under interest by counting the number of paths under interests. In ATL*, this framework is even more interesting as it allows to count strategies these is the notion of graded, that adds the ability to count strategies.

The extension of ATL* along with graded modalities suffers of all limitation inherented by the weakness of this logic with respect to SL. For example, one cannot express Nash equilibria or other important game-theoretical concepts such as stealing strategies. For this reason, in this thesis the extension of SL with graded modalities has been introduced and deeply investigated.

In particular, this has been done by using the more general framework of Strategy Game, recently introduced, for which Strategy Logic represents a particular case. In fact Strategy Game uses predicates to represent paths, in place of the CTL* formalism.

In summary, in this thesis we have introduced the following concepts:

- An extension of Strategy Logic along with graded modalities, called Graded Strategy Logic;
- An equivalence relation that classifies the strategies;
- A game-type conversion that given a concurrent game, with solution concept in GSG[1G], it can reduce us to a turn-based game, with solution concept in GSG[1G], in exponential time on number of variables;
- A game-type conversion that given a concurrent game, with solution concept in GSG[BG], it

can reduce us to a concurrent game, with solution concept in SG[BG], in exponential time on size of the solution concept;

- A grading function which verified that the concurrent game, with solution concept GSG[1G] and at most two variables, are determined;
- Finally, we have shown how in concurrent game, with solution concept GSG[1G] and at most two variables, the properties of safety or reachability is in PTIME-COMPLETE in the size of both the arena and the solution concept.

These results open to a number of interesting scenarios, in both practical and the theoretical fields. For example, one can check, in complex situations, if indeed there are multiple ways to solve a given game. In open system verification, one can use the graded strategy game to check whether a system is more secure than another by making a difference between how many strategies there are in both place. Of course, all these aspects can be implemented in a verification tool. In this regard, it is useful to remember that there is already a tool for verification of SL in [vLMM14]. So, should it not be difficult to extend this tool to graded modalities, this it would allow for the first time to count the strategies of the games.

Bibliography

[AHK02] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-Time Temporal Logic. *Journal of the ACM*, 49(5):672–713, 2002.

[BLMV06] P.A. Bonatti, C. Lutz, A. Murano, and M.Y. Vardi. The complexity of enriched μ-calculi. In *ICALP '06*, volume 4052 of *LNCS*, pages 540–551, 2006.

[BMM09] A. Bianco, F. Mogavero, and A. Murano. Graded Computation Tree Logic. In *Logic in Computer Science'09*, pages 342–351. IEEE Computer Society, 2009.

[BMM10] A. Bianco, F. Mogavero, and A. Murano. Graded Computation Tree Logic with Binary Coding. In *Computer Science Logic'10*, LNCS 6247, pages 125–139. Springer, 2010.

[BMM12] A. Bianco, F. Mogavero, and A. Murano. Graded Computation Tree Logic. *Transactions On Computational Logic*, 13(3):25:1–53, 2012.

[BP04] P.A. Bonatti and A. Peron. On the undecidability of logics with converse, nominals, recursion and counting. *Artificial Intelligence*, 158 : 1:75–96, 2004.

[BS06] J. Bradfield and C. Stirling. Modal μ-calculi. In Blackburn, Wolter, and van Benthem, editors, *Handbook of Modal Logic*, chapter 12, pages 722–756. Elsevier, 2006.

[CE81] E.M. Clarke and E.A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *Logic of Programs'81*, LNCS 131, pages 52–71. Springer, 1981.

[CHP07] K. Chatterjee, T.A. Henzinger, and N. Piterman. Strategy Logic. In *Concurrency Theory'07*, LNCS 4703, pages 59–73. Springer, 2007.

[CHP10] K. Chatterjee, T.A. Henzinger, and N. Piterman. Strategy Logic. *Information and Computation*, 208(6):677–693, 2010.

[EH85] E.A. Emerson and J.Y. Halpern. Decision Procedures and Expressiveness in the Temporal Logic of Branching Time. *Journal of Computer and System Science*, 30(1):1–24, 1985.

[EH86] E.A. Emerson and J.Y. Halpern. "Sometimes" and "Not Never" Revisited: On Branching Versus Linear Time. *Journal of the ACM*, 33(1):151–178, 1986.

[FHMV95] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge.* MIT Press, 1995.

[Fin72] K. Fine. In So Many Possible Worlds. *Notre Dame Journal of Formal Logic*, 13:516–520, 1972.

[FM07] A. Ferrante and A. Murano. Enriched Mu-Calculi Module Checking. In *Foundations of Software Science and Computational Structures'09*, LNCS 5504, pages 183–197. Springer, 2007.

[FMP08] A. Ferrante, A. Murano, and M. Parente. Enriched Mu-Calculi Module Checking. *Logical Methods in Computer Science*, 4(3):1–21, 2008.

[FNP09] A. Ferrante, M. Napoli, and M. Parente. Model Checking for Graded CTL. *Fundamenta Informaticae*, 96(3):323–339, 2009.

[FNP10] M. Faella, M. Napoli, and M. Parente. Graded Alternating-Time Temporal Logic. *Fundamenta Informaticae*, 105(1-2):189–210, 2010.

[GOR97] E. Grädel, M. Otto, and E. Rosen. Two-Variable Logic with Counting is Decidable. In *Logic in Computer Science'97*, pages 306–317. IEEE Computer Society, 1997.

[HB91] B. Hollunder and F. Baader. Qualifying Number Restrictions in Concept Languages. In *Knowledge Representation and Reasoning'91*, pages 335–346. Morgan Kaufmann, 1991.

[Hoa85] C.A.R. Hoare. Communicating sequential processes, 1985.

[HP85] D. Harel and A. Pnueli. *On the Development of Reactive Systems.* Springer, 1985.

[JvdH04] W. Jamroga and W. van der Hoek. Agents that Know How to Play. *Fundamenta Informaticae*, 63(2-3):185–219, 2004.

[JW95] D. Janin and I. Walukiewicz. Automata for the Modal μ-Calculus and Related Results. In *MFCS'95*, LNCS 969, pages 552–562. Springer-Verlag, 1995.

[Kel76] R.M. Keller. Formal Verification of Parallel Programs. *Communication of the ACM*, 19(7):371–384, 1976.

[Koz83] D. Kozen. Results on the Propositional muCalculus. *Theoretical Computer Science*, 27(3):333–354, 1983.

[KPV02] O. Kupferman, N. Piterman, and M.Y. Vardi. Pushdown specifications. In *LPAR*, pages 262–277, 2002.

[KSV02] O. Kupferman, U. Sattler, and M.Y. Vardi. The Complexity of the Graded muCalculus. In *Conference on Automated Deduction'02*, LNCS 2392, pages 423–437. Springer, 2002.

[KV97] O. Kupferman and M.Y. Vardi. Module checking revisited. In *CAV '96*, volume 1254 of *LNCS*, pages 36–47. Springer-Verlag, 1997.

[KVW00] O. Kupferman, M.Y. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *Journal of the ACM*, 47(2):312–360, 2000.

[KVW01] O. Kupferman, M.Y. Vardi, and P. Wolper. Module Checking. *Information and Computation*, 164(2):322–344, 2001.

[Lam80] L. Lamport. "Sometime" is Sometimes "Not Never": On the Temporal Logic of Programs. In *Principles of Programming Languages'80*, pages 174–185. Association for Computing Machinery, 1980.

[Lor10] E. Lorini. A Dynamic Logic of Agency II: Deterministic DLA, Coalition Logic, and Game Theory. *Journal of Logic, Language, and Information'*, 19(3):327–351, 2010.

[Mar75] A.D. Martin. Borel Determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.

[Mar85] A.D. Martin. A Purely Inductive Proof of Borel Determinacy. In *Symposia in Pure Mathematics'82*, Recursion Theory., pages 303–308. American Mathematical Society and Association for Symbolic Logic, 1985.

[MH84] S. Miyano and T. Hayashi. Alternating Finite Automata on ω-Words. *Theoretical Computer Science*, 32(3):321–330, 1984.

[MMS14] F. Mogavero, A. Murano, and L. Sauro. Strategy Games: A Renewed Framework. In *Autonomous Agents and MultiAgent Systems'14*, pages 869–876. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[MMV10] F. Mogavero, A. Murano, and M.Y. Vardi. Reasoning About Strategies. In *Foundations of Software Technology and Theoretical Computer Science'10*, LIPIcs 8, pages 133–144. Leibniz-Zentrum fuer Informatik, 2010.

[Pnu77] A. Pnueli. The Temporal Logic of Programs. In *Foundation of Computer Science'77*, pages 46–57. IEEE Computer Society, 1977.

[Pnu81] A. Pnueli. The Temporal Semantics of Concurrent Programs. *Theoretical Computer Science*, 13:45–60, 1981.

[PP04] D. Perrin and J. Pin. *Infinite Words.* Pure and Applied Mathematics. Elsevier, 2004.

[QS81] J.P. Queille and J. Sifakis. Specification and Verification of Concurrent Programs in Cesar. In *Symposium on Programming'81*, LNCS 137, pages 337–351. Springer, 1981.

[SV01] U. Sattler and M.Y. Vardi. The hybrid mu–calculus. In *IJCAR '01*, volume 2083 of *LNAI*, pages 76–91, 2001.

[Tob01] S. Tobies. PSpace Reasoning for Graded Modal Logics. *Journal of Logic and Computation*, 11(1):85–106, 2001.

[Var88] M.Y. Vardi. A Temporal Fixpoint Calculus. In *Principles of Programming Languages'88*, pages 250–259. Association for Computing Machinery, 1988.

[vE13] J. van Eijck. PDL as a Multi-Agent Strategy Logic. In *Theoretical Aspects of Rationality and Knowledge'13*, pages 206–215, 2013.

[vLMM14] P. Čermák, A. Lomuscio, F. Mogavero, and A. Murano. MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications. In *Computer Aided Verification'14*, LNCS 8559, pages 524–531. Springer, 2014.

[VW86] M.Y. Vardi and P. Wolper. An automata–theoretic approach to automatic program verification (preliminary report). pages 332–344, 1986.

[Wil99] T. Wilke. CTL+ is Exponentially More Succinct than CTL. In *FSTTCS'99*, pages 110–121. Springer-Verlag, 1999.

List of Figures

List of Tables