# A Generic Constructive Solution for Concurrent Games with Expressive Constraints on Strategies

Sophie Pinchinat*

Computer Sciences Laboratory of RSISE
The Australian National University, Canberra

**Abstract.** The emerging technology of interacting systems calls for new formalisms to ensure their reliability. Concurrent games are paradigmatic abstract models for which several logics have been studied. However, the existing formalisms show certain limitations in face of the range of strategy properties required to address intuitive situations. We propose a generic solution to specify expressive constraints on strategies in concurrent games. Our formalism naturally extends alternating-time logics while being highly flexible to combine constraints. Our approach is constructive and can synthesize many types of complex strategies, via automata-theoretic techniques.

## 1 Introduction

Computer-system design currently relies on complex assemblages of *interacting components* which communicate and share resources in order to achieve services. The combinatorics of such systems is so enormous that the development of adequate formal methods to ensure their reliability has become a major challenge. In this context, games are paradigmatic for providing expressive mathematical models of interactive systems, reflecting their operational semantics and offering adequate reasoning tools. In order to reason formally about interactive models, it is necessary to devise appropriate specification languages in which the desirable behaviors of the system can be stated; once the properties are formulated, methods for automated verification and synthesis can be employed to support the design process.

In the past decade, extensions of state-transition based models, such as *Concurrent Game Structures* [AHK02] which extend Kripke structures, have raised considerable interest in virtue of offering mathematical settings to address formal analysis of complex systems. At the same time, alternating-time logics such as ATL, ATL$^*$, AMC and GL [AHK02] have been proposed as a natural extension of standard temporal logics to the multiplayer setting. Noticeable theoretical and practical results exist for these logics, such as effective decision procedures with reasonable cost for ATL [vD03, KP04, GvD06, SF06, LMO07], and implementations [HKQ98, AdAdS$^+$06]. However, it should be made clear that alternating time logics show certain limitations in face of the range of strategy properties required to address intuitive situations. For example, communication protocols often require to consider fairness assumptions, which

* This research was supported by the Marie Curie Scientific Project MASLOG 021669 (FP6-2004-Mobility-6) and Université de Rennes 1.

enable to exclude some undesirable computations of the system. When such assumptions are not expressible in the logic, non trivial efforts are needed to impose the constraints directly in the models [AHK02]. By this type of approach, only very dedicated kind of analysis can be performed, and often, a minor addition of new constraints compels the user to re-design its problem from scratch. Hence, there is a need for a formalism where constraints on strategies can be combined. Other examples of limitations can be borrowed in solution concepts for non-zero sum games. As far as we are concerned, uniqueness of a Nash equilibrium [Cha05], or dominance of strategies [Ber07] cannot be expressed in any respect, because the formalisms do not have strategies as main objects.

In this paper, we propose a generic constructive solution to analyze the strategies of concurrent games. Our formalism is tuned to specify at the same time the strategies, their properties (e.g. fairness), and their objectives in a unified framework. Expressive constraints can henceforth be formulated; for example all the limitations discussed above are overcome. Moreover, all the concepts for nonzero sum games considered by [CHP07] can be captured, since we can express Strategy Logic (SL) in our formalism; notice that SL is limited to turned-based arenas, whereas we also consider general concurrent game structures, and SL is not powerful enough to express, e.g. it cannot express the Alternating Mu-Calulus [AHK02]. We start from the logic $D_\mu$, a traditional propositional mu-calculus [Koz83], augmented with *decision modalities*. By the semantics of the logic, the game is unfold into an infinite tree. The purpose of decision modalities is to specify particular monadic predicates over the nodes of the computation tree of the game, to establish a one-to-one correspondence between these particular predicates and the strategies of sets of players; the outcome of a strategy is the sub-tree whose nodes form the predicate, and still is a concurrent game, but with less players.

Objectives of strategies can be any $\omega$-regular property. In essence, strategies together with their objectives have an *assume-guarantee* flavor: by assuming that a certain strategy is adopted, we guarantee some temporal property of its outcome. From this point of view, we operate on the model (by applying the strategy) and leave the property intact, as done in [CHP07]. In order to decrease the intricacy of the problem, we propose a powerful although simple mechanism to operate on the logic side while leaving the models intact; it is called *relativization*. The benefit is to transform the complex assume-guarantee statement into a mere temporal statement about the model.

Following the original idea of [RP03] for controller synthesis problems, we define the logic $\text{Q}D_\mu$ a monadic second order extension of $D_\mu$, but where fix-points and quantifiers can arbitrarily interleave. Our calculus is then equipped to quantify over strategies, as we show, in a highly expressive manner. In particular, it subsumes alternating time logics, while being amenable to automata constructions, hence to an effective procedure to synthesize the strategies.

The paper is organized as follows: we present the models in Section 2, and the logic in Section 3. *Strategies* and *outcomes* are defined in Section 4, followed by the *relativization* principle. Section 5 is dedicated to significant examples of logical specifications. Section 6 describes automata constructions for $\text{Q}D_\mu$. We complete the work by the embedding of alternating-time logics into our system (Section 7), and a note on a customized automata construction for these logics.

## 2 The Models

In the following, we assume an infinite countable ordered set $\boldsymbol{P} = \{p, p', \ldots\}$ of players, and an infinite set of *atomic propositions* Prop $= \{Q, Q_1, Q_2, \ldots\}$. Finite sets $C$ of players are *coalitions*. For any integer $i \geq 1$, let $[i]$ denote the set $\{1, \ldots, i\}$.

A *Concurrent Game Structure* (CGS) over $\Lambda$ and $M$ is a tuple $\mathcal{S} = \langle \Pi, S, \Lambda, \lambda, M, \delta \rangle$, where:

- $\Pi \subseteq \boldsymbol{P}$ is a non-empty finite set of *players*, whose cardinal is denoted by $n$; we may represent the ordered set $\Pi$ by the natural numbers $1, \ldots, n$.
- $S$ is a set of *states*, with typical elements of $S$ written $s, s', \ldots$.
- $M$ is a set of *moves*. Each $j \in M$ is a possible *move* available in each state to each player $p \in \Pi$. A *decision vector* is a tuple $x = \langle j_1, \ldots, j_n \rangle \in M^n$, where $j_p$ is a move of $p \in \Pi$. The value $Card(M)^n$ is the *branching degree* of $\mathcal{S}$.
- $\delta : S \times M^n \to S$ is the *transition function*: given a state $s \in S$ and decision vector $\langle j_1, \ldots, j_n \rangle$, the game moves to the state $\delta(s, x)$. Each $\delta(s, x)$ where $x \in M^n$ is a *successor* of $s$, and successors of $s$ form the set $Succ(s)$.
- $\Lambda \subseteq$ Prop, and $\lambda : \Lambda \to 2^S$ labels states by propositions. A state $s$ is labelled by $Q \in \Lambda$ whenever $s \in \lambda(Q)$. We let $\lambda(\mathcal{Q}) := \bigcap_{Q \in \mathcal{Q}} \lambda(Q)$, for any set of propositions $\mathcal{Q} \subseteq \Lambda$.

Comparing this with the original definition of [AHK02], we may notice the following:

1. We use the same set of moves for all players, independently of the current state.
2. Each player moves independently of the others.
3. Players make concurrent choices in each state.

However, the proposed models are expressive enough to capture the essential features of concurrent games, as we can actually simulate any concurrent game: In general, each player in a current state $s$ has a set $M_s^p$ of moves. We can simulate this situation with a unique set of moves $M$ by renaming the moves in $M_s^p$ and by qualifying some states *dummy*; the logical statements need being interpreted on the relevant part of the models, namely on computations which do not encounter dummy states. Hence Points 1 and 2 are not restrictive. Since we can restrict players' set of moves from a given state, enforcing all but one player to have a single choice simulates turned-based games; this sorts out Point 3. Notice that asynchronous games are also captured: following [AHK02], we designate a particular player *scheduler* which in every state selects one of the players; the latter then determines the next state. Now, the crucial assumption that the scheduler fairly selects the players can be expressed in the logic, as opposed to [AHK02] where fairness is defined in the models.

Given $s \in S$, $p \in \Pi$, and $j \in M$, we let $Succ_j(s, p) \subseteq Succ(s)$ be the set of successors of $s$ which can be enforced by the move $j$ of player $p$; formally, it is the set of states of the form $\delta(s, \langle j_1, \ldots, j_n \rangle)$ with $j_p = j$. Consider the classic two-player game *Paper, Rock, and Scissors (PRS)* as depicted in Figure 1: the possible moves of each player range over $M = \{P, R, S\}$ for "paper", "rock" and "scissors" respectively. The propositions 1-Win and 2-Win indicate who is the winner in the current configuration; let us ignore proposition $Q$ for the moment. In this game, $Succ_S(s_0, 2)$

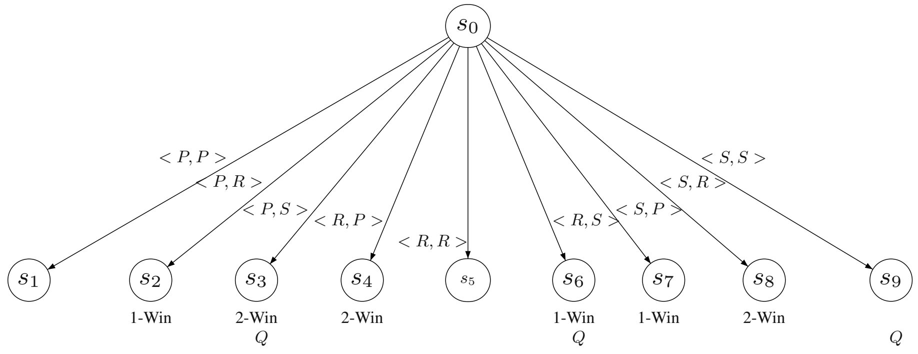

**Fig. 1.** The Paper, Rock, and Scissors game

$= \{s_3, s_6, s_9\}$ is the set of successors of $s_0$ player 2 can enforce by playing "$S$". We let $SuccSets(s,p) \subseteq 2^{Succ(s)}$ be the set of all $Succ_j(s,p)$ where $j \in M$, composed of sets of successors of $s$ which can be enforced by some move of player $p$. In PRS, $SuccSets(s_0, 2) = \{\{s_1, s_4, s_7\}, \{s_2, s_5, s_8\}, \{s_3, s_6, s_9\}\}$. Since a player $p \notin \Pi$ cannot influence the game, we take the convention that $SuccSets(s,p) = \{Succ(s)\}$. Given a coalition $C$ and $s \in S$, a $C$-move from $s$ is an element of $\cap_{p \in C} SuccSets(s,p)$; it is a subset of $Succ(s)$ which elements result from fixing a particular move for each player in $C$. In PRS, a $\{1,2\}$-move from $s_0$ is $\{s_i\}$, for some $1 \leq i \leq 9$.

## 3 The Logical Framework

We propose a generalization of [RP03] which is twofold: we enrich the propositional mu-calculus [Koz83] by allowing *decision modalities*, and we consider its monadic second order extension by allowing quantifications over atomic propositions, even under the scope of fix-points operators. We first present the propositional mu-calculus with decision modalities; the second order extension follows in this section.

The logic $D_\mu$ is the mu-calculus with *decision modalities* (formulas $\diamond_p \mathcal{Q}$). Given a set Prop of atomic propositions, an infinite set $\boldsymbol{P}$, and a set of variables $\text{Var} = \{Z, Y, \ldots\}$, the syntax of $D_\mu$ is:

$$Q \mid \diamond_p \mathcal{Q} \mid \top \mid \neg\beta \mid \beta_1 \vee \beta_2 \mid \mathbf{EX}\,\beta \mid Z \mid \mu Z.\beta(Z)$$

where $Q \in \text{Prop}$, $p \in \boldsymbol{P}$, $\mathcal{Q} \subseteq \text{Prop}$, and $\beta, \beta_1, \beta_2$ are $D_\mu$ formulas. Fix-point formulas ($\mu Z.\beta(Z)$) are such that any occurrence of $Z \in \text{Var}$ in $\beta(Z)$ occurs under an even number of negation symbols $\neg$. A *sentence* is a formula where any occurrence of a variable $Z$ occurs under the scope of a $\mu Z$ operator. The set of formulas which do not contain any decision modality correspond to the traditional propositional mu-calculus, whence the standard notations $\perp$, $\mathbf{AX}\,\beta$, $\beta_1 \wedge \beta_2$, $\beta_1 \Rightarrow \beta_2$, and $\nu Z.\beta(Z)$ for respectively $\neg\top$, $\neg\mathbf{EX}\,\neg\alpha$, $\neg(\neg\beta_1 \vee \neg\beta_2)$, $\neg\beta_1 \vee \beta_2$, and $\neg\mu Z.\neg\beta(\neg Z)$. Moreover, given $\beta \in D_\mu$, we freely use the concise CTL-like notation $\mathbf{AG}\,(\beta)$ for $\nu Z.(\beta \wedge \mathbf{AX}\,Z)$, which expresses that $\beta$ is globally true in the future, and $\mathbf{EF}\,(\beta)$ for $\neg\mathbf{AG}\,(\neg\beta)$.

As for the traditional mu-calculus, a formula $\beta \in D_\mu$ is interpreted in a CGS $\mathcal{S} = \langle \Pi, S, \Lambda, \lambda, M, \delta \rangle$ supplied with a valuation $\mathrm{val} : \mathrm{Var} \to 2^S$. Its semantics $[\![ \beta ]\!]_{\mathcal{S}}^{\mathrm{val}}$ is a subset of $S$, defined by induction over the structure of $\beta$. The following is very standard as the mu-calculus operators semantics:

$$\begin{aligned}
[\![ Q ]\!]_{\mathcal{S}}^{\mathrm{val}} &= \{s \in S \mid s \in \lambda(Q)\} \\
[\![ \top ]\!]_{\mathcal{S}}^{\mathrm{val}} &= S \\
[\![ \neg\beta ]\!]_{\mathcal{S}}^{\mathrm{val}} &= S \setminus [\![ \beta ]\!]_{\mathcal{S}}^{\mathrm{val}} \\
[\![ \beta_1 \vee \beta_2 ]\!]_{\mathcal{S}}^{\mathrm{val}} &= [\![ \beta_1 ]\!]_{\mathcal{S}}^{\mathrm{val}} \cup [\![ \beta_2 ]\!]_{\mathcal{S}}^{\mathrm{val}} \\
[\![ Z ]\!]_{\mathcal{S}}^{\mathrm{val}} &= \mathrm{val}(Z) \\
[\![ \mathbf{EX}\,\beta ]\!]_{\mathcal{S}}^{\mathrm{val}} &= \{s \in S \mid \exists s' \in \mathit{Succ}(s) \ \wedge\ s' \in [\![ \beta ]\!]_{\mathcal{S}}^{\mathrm{val}}\} \\
[\![ \mu Z.\beta(Z) ]\!]_{\mathcal{S}}^{\mathrm{val}} &= \cap\{S' \subseteq S \mid [\![ \beta(Z) ]\!]_{\mathcal{S}}^{\mathrm{val}(S'/Z)} \subseteq S'\}
\end{aligned}$$

Classically, as a valuation val does not influence the semantics of a sentence $\beta \in D_\mu$, we then simply write $[\![ \beta ]\!]_{\mathcal{S}}$.

We now focus on decision modalities which are essential to our logic:

$$[\![ \diamond_p \mathcal{Q} ]\!]_{\mathcal{S}}^{\mathrm{val}} = \{s \in S \mid \mathit{Succ}(s) \cap \lambda(\mathcal{Q}) \in \mathit{SuccSets}(s,p)\}$$

By definition, $s \in [\![ \diamond_p \mathcal{Q} ]\!]_{\mathcal{S}}$ whenever the set $\mathit{Succ}(s) \cap \{\lambda(Q) \mid Q \in \mathcal{Q}\}$ can alternatively be characterized as a move of player $p$, namely as $\mathit{Succ}_j(s,p)$ for some move $j \in M$. Notice that the semantics of $\diamond_p \mathcal{Q}$ is well defined even if $p \notin \Pi$, since in this case $\mathit{SuccSets}(s,p)$ equals $\{\mathit{Succ}(s)\}$ by convention. In most of our examples, the set $\mathcal{Q}$ has a single element $Q$, so we simply write $\diamond_p Q$ for $\diamond_p\{Q\}$.

In the example of the PRS game, Figure 1, $s_0 \notin [\![\diamond_1(\text{1-Win})]\!]_{\mathcal{S}}$ because the predicate 1-Win does not match a set of successors enforced by a move of player 1; as a matter of fact, player 1 does not have a winning strategy, neither does player 2 for symmetrical reasons. On the contrary $s_0 \in [\![ \diamond_1 Q ]\!]_{\mathcal{S}}$, since $Q$ matches $\mathit{Succ}_{\mathcal{S}}(s_0, 2)$. When the game is infinite, eg by repeating the game PRS, it can be unfolded as an infinite tree, the nodes of which are histories of an ongoing play. Assume given a predicate $Q$ on the tree nodes such that $\diamond_p Q$ is invariant in the computation tree, that is such that $\mathbf{AG}\ (\diamond_p Q)$ holds in the root. Then any computation inside $Q$ corresponds to a play for a fixed strategy of player $p$, namely the one designated by $Q$, and the sub-tree formed by these computations is the outcome of this very strategy. Combining decision modalities for several players characterizes coalition moves: for instance, when a formula like $\diamond_{p_1} Q_1 \wedge \diamond_{p_2} Q_2 \wedge \diamond_{p_3} Q_3$ holds, then the set of successors which satisfy $Q_1 \wedge Q_2 \wedge Q_3$ corresponds to some move of the coalition $\{p_1, p_2, p_3\}$. By extension, if each predicate $Q_i$ designates a strategy of $p_i$, the sub-tree whose computations remain inside $Q_1 \wedge Q_2 \wedge Q_3$ is the outcome of the coalition strategy.

The logic is extended to the monadic second order to capture strategies as main objects of the logic: stating that there exists a predicate $Q$ such $\mathbf{AG}\ (\diamond_p Q)$ holds expresses the existence of a strategy. This extension of the logic is written $\mathrm{Q}D_\mu$, for "quantified $D_\mu$"; its syntax is as for $D_\mu$ but with quantifications over sets of atomic propositions. The syntax of $\mathrm{Q}D_\mu$ is:

$$Q \mid \diamond_p\ \mathcal{Q} \mid \top \mid \neg\alpha \mid \alpha_1 \vee \alpha_2 \mid \mathbf{EX}\,\alpha \mid Z \mid \mu Z.\alpha(Z) \mid \exists \mathcal{Q}.\alpha$$

We write $\forall \mathcal{Q}.\alpha$ for $\neg\exists \mathcal{Q}.\neg\alpha$.

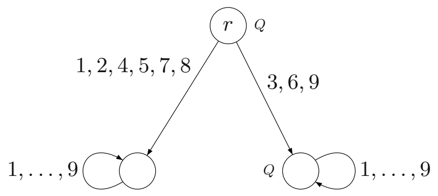

**Fig. 2.** A $Q$-labeling $(\mathcal{E}, r)$ of degree 9

The semantics of $\mathrm{Q}D_\mu$ generalizes the one of $D_\mu$: the cases of $Q$, $\diamond_p \mathcal{Q}$, $\top$, $\neg\alpha$, $\alpha_1 \vee \alpha_2$, $\mathbf{EX}\,\alpha$, $Z$, and $\mu Z.\alpha(Z)$ are dealt inductively. The semantics of quantification follows the proposal of [RP03]: the mechanism to define new predicates $Q \in \mathcal{Q}$ on a game structure relies on a composition of the structure with a Kripke structure over $\mathcal{Q}$, called a *labeling*.

**Definition 1 ($\mathcal{Q}$-labelings).** *Given* $\mathcal{Q} \subseteq \mathrm{Prop}$ *and* $m \geq 1$*, a* $\mathcal{Q}$-labeling *(or a* labeling *over* $\mathcal{Q}$*) is a pair* $(\mathcal{E}, r)$ *where* $\mathcal{E} = \langle E, \mathcal{Q}, \gamma, [m], \delta' \rangle$ *is a (one player) CGS structure over* $\mathcal{Q}$ *and* $[m]$*, and* $r \in E$ *is its root. It is a Krikpe structure.*

We compose labelings and CGS's with the same branching degree. We suppose fixed once for all a principle to bijectively relate any set of the form $M^n$ to the set $[Card(M)^n]$ (recall it is $\{1, \ldots, Card(M)^n\}$); for example one can use the coding proposed by [GvD06]. In the following, let us qualify *canonical* a bijection from $M^n$ to $[Card(M)^n]$ whenever it is based on this principle.

Now assume given a rooted CGS $(\mathcal{S}, s)$ with $n$ players over $\Lambda$ and $M$, and a labeling $(\mathcal{E}, r)$ over $\mathcal{Q}$ and $[Card(M)^n]$, where $\mathcal{E} = \langle E, \mathcal{Q}, \gamma, [Card(M)^n], \delta' \rangle$; denote by $\tau$ the canonical bijection from $M^n$ to $[Card(M)^n]$. The *labeling of* $(\mathcal{S}, s)$ *by* $(\mathcal{E}, r)$ is the synchronous product of the two structures, where $x$-transitions in $\mathcal{S}$ are synchronized with the $\tau(x)$-transitions in $\mathcal{E}$. Formally,

$$(\mathcal{S}, s) \times (\mathcal{E}, r) = \langle \Pi, S \times E, \Lambda \cup \mathcal{Q}, (\lambda \times \gamma), \delta\text{"} \rangle$$

is the CGS over $\Lambda \cup \mathcal{Q}$ and $M$ rooted at $(s, r)$, where:

- $(\lambda \times \gamma)(Q) = \lambda(Q) \cup \gamma(Q)$, for each $Q \in \Lambda \cup \mathcal{Q}$ with the convention that if $Q \notin \Lambda$ (or $\notin \mathcal{Q}$) then $\lambda(Q)$ (respectively $\gamma(Q)$) is the empty set, and
- $\delta\text{"}((s_1, e_1), x) = (s_2, e_2)$ whenever $\delta(s_1, x) = s_2$ and $\delta'(e_1, \tau(x)) = e_2$.

In the following, composition of a structure with a labeling implicitly assumes that their branching degrees match. Figure 1 shows a (regular) labeling $(\mathcal{E}, r)$ over $Q$ and $[3^2]$, and the labeling of the game structure PRS by $(\mathcal{E}, r)$, with the convention that $\tau(\langle P, P \rangle) = 1$, $\tau(\langle P, R \rangle) = 2$, ..., $\tau(\langle S, R \rangle) = 8$, and $\tau(\langle S, S \rangle) = 9$. The result is depicted in Figure 1.

Notice that since propositions of compound states accumulate, and because $\mathcal{E}$ and $\mathcal{S}$ have the same branching degree, $(\mathcal{S}, s) \times (\mathcal{E}, r)$ is bisimilar to $(\mathcal{S}, s)$ in the usual sense, if we restrict to propositions that are not in $\mathcal{Q}$. In particular if $\mathcal{Q}$ is empty, $(\mathcal{S}, s) \times (\mathcal{E}, r)$

is bisimilar to $(\mathcal{S}, s)$. The composition of $\mathcal{S}$ and labelings is tedious but it only aims at formalizing the means to decoration nodes of the computation tree by propositions; in particular, when $\mathcal{E}$ is a finite state $\mathcal{Q}$-labeling, the predicates $Q$ are regularly placed on the computation tree of the game structure.

We have now the material to define the meaning of quantifiers: $s \in [\![ \exists \mathcal{Q}.\alpha ]\!]^{\mathrm{val}}_{\mathcal{S}}$ if and only if there exists a $\mathcal{Q}$-labeling $(\mathcal{E}, r)$ such that $(s, r) \in [\![ \alpha ]\!]^{\mathrm{val}'}_{(\mathcal{S},s)\times(\mathcal{E},r)}$, where $\mathrm{val}'(Z) = \mathrm{val}(Z) \times E$.

Remark that formulas of $\mathrm{Q}D_\mu$ have the same truth value if we unravel the structure $\mathcal{S}$. Besides, the semantics of quantified formulas is a lot more intuitive on the computation tree: $\exists \mathcal{Q}.\alpha$ holds if there is a way to assign the propositions of $\mathcal{Q}$ to the nodes of the computation tree so that $\alpha$ holds.

## 4 Strategies and Outcomes

In this section, we assume a fixed CGS $\mathcal{S} = \langle \Pi, S, \Lambda, \lambda, M, \delta \rangle$.

We revisit the central concepts of *strategies* and *outcomes* which underlies the semantics of all logics for CGS's: as already explained in Section 3, giving a strategy of player $p$ is equivalent to labeling the structure by some proposition $Q$ where the property $\diamond_p Q$ is invariant. Since invariance is definable in the logic, we obtain the following definition for strategies:

**Definition 2 (Strategies).** *Given a coalition $C \subseteq \Pi$, and a set $\{Q_p \,|\, p \in C\} \subseteq \mathrm{Prop}$, a $C$-strategy from $s$ designated by $\{Q_p \,|\, p \in C\}$ is a labeling $(\mathcal{E}, r)$ of $(\mathcal{S}, s)$ over $\{Q_p \,|\, p \in C\}$, such that*

$$(s, r) \in [\![ \mathbf{AG}\, (\bigwedge_{p \in C} \diamond_p Q_p) ]\!]_{(\mathcal{S},s)\times(\mathcal{E},r)} \qquad (1)$$

For each $C$-strategy $(\mathcal{E}, r)$ from $s$ designated by a set $\mathcal{Q}_C = \{Q_p \,|\, p \in C\}$, where $\mathcal{E} = \langle E, \mathcal{Q}, \gamma, [m], \delta' \rangle$, we define its *outcome* as the structure obtained by forgetting all states $(s', e)$ which are not chosen by the coalition $C$, hence not in the predicate $\bigwedge_{p \in C} Q_p$, and by forgetting the players of $C$ as their moves are fixed by the strategy. Formally, assuming $C \neq \emptyset$, we define $\mathrm{OUT}(\mathcal{Q}_C, \mathcal{S}, s) = \langle \Pi \setminus C, (S \times E) \cap (\lambda \times \gamma)(\mathcal{Q}), \Lambda \cup \mathcal{Q}, M, \delta" \rangle$, with $\delta"((s_1, e_1), y) = \delta((s_1, e_1), y')$ where $y'$ is the decision vector of players in $\Pi$ obtained by completing the decision vector $y$ of players in $\Pi \setminus C$ by the moves of the players in $C$ recommended by the $C$-strategy. If $C = \emptyset$ then as expected $\mathrm{OUT}(\mathcal{Q}_\emptyset, \mathcal{S}, s) = (\mathcal{S}, s)$.

**Lemma 1.** *$\mathrm{OUT}(\mathcal{Q}_C, \mathcal{S}, s)$ is a CGS (rooted at $s$) over the set of players $\Pi \setminus C$.*

Our definition of outcome is sensible as the set of maximal paths in $\mathrm{OUT}(\mathcal{Q}_C, \mathcal{S}, s)$ coincides with the original definition of 'outcome' in the sense of [AHK02]. However, because our notion retains the structure of a game, contrary to the original definition, we can state any logical statements anew.

In the following, and when it is clear from the context, we simply say "$\mathcal{Q}_C$-strategy" for "$C$-strategy designated by $\mathcal{Q}_C$", and we write $Q_p$ for $\mathcal{Q}_{\{p\}}$. Also, we concisely write $\mathcal{Q}$ for $\bigwedge_{Q \in \mathcal{Q}} Q$ and define $\bigwedge_{Q \in \emptyset} Q$ as $\top$.

We present now a simple mechanism called the *relativization* which transforms a formula by propagating downward in the formula a set of atomic propositions.

**Definition 3 ($\mathcal{Q}$-Relativization of a formula).** *For $\mathcal{Q} \subseteq \text{Prop}$, the $\mathcal{Q}$-*relativization *is a mapping $(\cdot|\mathcal{Q}) : \text{Q}D_\mu \to \text{Q}D_\mu$ defined by induction:*

$$\begin{array}{rcc}
(Q|\mathcal{Q}) = Q & (\top|\mathcal{Q}) = \top & (Z|\mathcal{Q}) = Z \\
(\neg\alpha|\mathcal{Q}) = \neg(\alpha|\mathcal{Q}) & (\alpha_1 \vee \alpha_2|\mathcal{Q}) = (\alpha_1|\mathcal{Q}) \vee (\alpha_2|\mathcal{Q}) & \\
(\mu Z.\alpha(Z)|\mathcal{Q}) = \mu Z.(\alpha(Z)|\mathcal{Q}) & (\exists \mathcal{Q}'.\alpha|\mathcal{Q}) = \exists \mathcal{Q}'.(\alpha|\mathcal{Q}) & \\
(\diamond_p Q|\mathcal{Q}) = \diamond_p(\mathcal{Q} \wedge Q) & (\mathbf{EX}\,\alpha|\mathcal{Q}) = \mathbf{EX}\,[\bigwedge_{Q\in\mathcal{Q}} Q \wedge (\alpha|\mathcal{Q})] &
\end{array}$$

From the above definition, we immediately obtain the equivalences:

$$(\alpha|\emptyset) \equiv \alpha \quad \text{and} \quad (\alpha|\mathcal{Q} \cup \{Q\}) \equiv ((\alpha|\mathcal{Q})|Q). \tag{2}$$

Regarding properties brought about by strategies, Theorem 1 below shows that we can either operate on the model, by considering the outcome and examine its property, or else operate on the formula, by considering the relativization and interpret it on the structure:

**Theorem 1.** *Given a rooted CGS $(\mathcal{S}, s)$, a coalition $C$, and a $\mathcal{Q}_C$-strategy $(\mathcal{E}, r)$ from $s$, we have: for any $\alpha \in \text{Q}D_\mu$, and any valuation $\text{val} : \text{Var} \to 2^S$:*

$$[\![ (\alpha|\mathcal{Q}_C) ]\!]^{\text{val}'}_{(\mathcal{S},s)\times(\mathcal{E},r)} = [\![ \alpha ]\!]^{\text{val}'}_{\text{OUT}(\mathcal{Q}_C,\mathcal{S},s)}$$

*where* $\text{val}'(Z) = \text{val}(Z) \times E$.

*Proof.* The proof of Theorem 1 is conducted by a double induction on the set $C$ and on the structure of $\alpha$. The case $C = \emptyset$ is trivial and independent of $\alpha$, since $(\alpha|\mathcal{Q}_C)$ is $\alpha$ by (2), on the one hand, and $(\mathcal{S}, s) \times (\mathcal{E}, r)$ and $\text{OUT}(\mathcal{Q}_C, \mathcal{S}, s)$ are isomorphic to $(\mathcal{S}, s)$, on the other hand. Assume now $C = C' \cup \{p\}$, with $c \notin C$. The $\mathcal{Q}_C$-strategy $(\mathcal{E}, r)$ can be decomposed into $(\mathcal{E}', r') \times (\mathcal{E}_p, r_p)$, where $(\mathcal{E}', r')$ is a $\mathcal{Q}_{C'}$-strategy and $(\mathcal{E}_p, r_p)$ is a $Q_p$-strategy; let us write $(\mathcal{S}', r')$ for $(\mathcal{S}, s) \times (\mathcal{E}', r')$. By (2):

$$[\![ (\alpha|\mathcal{Q}_C) ]\!]^{\text{val}'}_{(\mathcal{S},s)\times(\mathcal{E},r)} = [\![ ((\alpha|\mathcal{Q}_{C'})|Q_p) ]\!]^{\text{val}'}_{(\mathcal{S}',s')\times(\mathcal{E}_p,r_p)} \tag{3}$$

**Lemma 2.** *For any rooted CGS $(\mathcal{S}', s')$, any $\{p\}$-strategy $(\mathcal{E}, r)$ designated by $Q$, any $\alpha \in \text{Q}D_\mu$, and any valuation $\text{val} : \text{Var} \to 2^S$, $[\![ (\alpha|Q) ]\!]^{\text{val}'}_{(\mathcal{S}',s')\times(\mathcal{E},r)} = [\![ \alpha ]\!]^{\text{val}'}_{\text{OUT}(Q,\mathcal{S},s)}$ where $\text{val}'(Z) = \text{val}(Z) \times E$.*

The proof of this lemma is based on a simple induction over the formulas, in the spirit of [RP03]. Informally, remark first that the relativization is inductively defined for all formulas but those of the form $\mathbf{EX}\,\alpha$. The inductive cases of the lemma follow this line. Regarding statements like $\mathbf{EX}\,\alpha$, the lemma simply expresses that a successor exists in the prune structure $\text{OUT}(Q, \mathcal{S}, s)$ if and only if it already existed in the complete structure and it was labeled by $Q$.

By Lemma 2, the right hand side of (3) is equal to $[\![ (\alpha|\mathcal{Q}_{C'}) ]\!]^{\text{val}'}_{\text{OUT}(Q,\mathcal{S}',s')}$. Since $\text{OUT}(Q, \mathcal{S}', s')$ and $\text{OUT}(Q, \mathcal{S}, s) \times (\mathcal{E}', r')$ are isomorphic, it is also equal to $[\![ (\alpha \mid \mathcal{Q}_{C'}) ]\!]^{\text{val}'}_{\text{OUT}(Q,\mathcal{S},s)\times(\mathcal{E}',r')}$ which by induction hypothesis coincides with $[\![ \alpha ]\!]^{\text{val}'}_{\text{OUT}(\mathcal{Q}_{C'},\text{OUT}(Q,\mathcal{S},s),(s,r_p))}$. By definition of the outcomes, we have:

**Lemma 3.** *Given two distinct coalitions $C_1, C_2$, and any two $\mathcal{Q}_{C_i}$-strategies $(\mathcal{E}_i, r_i)$ ($i \in \{1,2\}$), $\text{OUT}(\mathcal{Q}_{C_1 \cup C_2}, \mathcal{S}, s)$ and $\text{OUT}(\mathcal{Q}_{C_1}, \text{OUT}(\mathcal{Q}_{C_2}, \mathcal{S}, s), (s, r_2))$ are isomorphic.*

Applying Lemma 3 to the term $[\![\, \alpha \,]\!]^{\text{val}'}_{\text{OUT}(\mathcal{Q}_{C'}, \text{OUT}(Q, \mathcal{S}, s), (s, r_p))}$ yields $[\![\, \alpha \,]\!]^{\text{val}'}_{\text{OUT}(\mathcal{Q}_C, \mathcal{S}, s)}$, which concludes the proof of Theorem 1.

**Corollary 1.** *Given a CGS $\mathcal{S}$, a coalition $C$, and a sentence $\alpha \in \text{Q}D_\mu$. Consider a set of fresh atomic propositions $\mathcal{Q}_C = \{Q_p \mid p \in C\}$ indexed over $C$. The formula*

$$\exists \mathcal{Q}_C.[\mathbf{AG}\,(\bigwedge_{p \in C} \diamond_p Q_p) \wedge (\alpha | \mathcal{Q}_C)]$$

*characterizes the states from which there exists a $C$-outcome of $(\mathcal{S}, s)$ satisfying $\alpha$.*

*Proof.* By definition, there exists a $\mathcal{Q}_C$-labeling from $s$, $(\mathcal{E}, r)$ such that $(s, r)$ is a model of $[\mathbf{AG}\,(\bigwedge_{p \in C} \diamond_p Q_p) \wedge (\alpha | \mathcal{Q}_C)]$. Therefore,

$$(s, r) \in [\![\, \mathbf{AG}\,(\bigwedge_{p \in C} \diamond_p Q_p) \,]\!]_{(\mathcal{S}, s) \times (\mathcal{E}, r)}, \text{ and} \tag{4}$$

$$(s, r) \in [\![\, (\alpha | \mathcal{Q}_C) \,]\!]_{(\mathcal{S}, s) \times (\mathcal{E}, r)}. \tag{5}$$

By (4), $(\mathcal{E}, r)$ is a $\mathcal{Q}_C$-strategy. By Theorem 1, (5) is equivalent to $s \in [\![\alpha]\!]_{\text{OUT}(\mathcal{Q}_C, \mathcal{S}, s)}$, which concludes. For the reciprocal, simply unroll the reasoning backward.

We make strategies become the main objects of our logic: given a coalition $C$, and a set $\mathcal{Q}_C = \{Q_p \mid p \in C\}$ of propositions, we define the following dual expressions:

$$\hat{\exists} \mathcal{Q}_C.\alpha \stackrel{\text{def}}{=} \exists \mathcal{Q}_C.[\mathbf{AG}\,(\bigwedge_{p \in C} \diamond_p Q_p) \wedge \alpha] \qquad \hat{\forall} \mathcal{Q}_C.\alpha \stackrel{\text{def}}{=} \forall \mathcal{Q}_C.[\mathbf{AG}\,(\bigwedge_{p \in C} \diamond_p Q_p) \Rightarrow \alpha]$$

By Corollary 1, $\hat{\exists} \mathcal{Q}_C.(\alpha | \mathcal{Q}_C)$ expresses the existence of a $C$-strategy which enforces $\alpha$. As widely demonstrated in the next section, statements of the form $(\alpha | \mathcal{Q}_C)$ can be combined, and associated with other types of statements (see (6) and (7)). Moreover the property $\alpha$ itself can incorporate specifications about other strategies, hence expressing commitment (see (8)).

## 5 Expressiveness Issues

This section reveals the high expressiveness of the formalism. We present three significant properties we can express in our formalism, but, we believe, in none of the other logics developed for concurrent games so far. Let us simply write $(\alpha \mid Q_1 \wedge Q_2)$ for $(\alpha \mid \{Q_1, Q_2\})$, $(Q = Q')$ for $\mathbf{AG}\,(Q \Leftrightarrow Q')$ to denote equality of predicates, and $(Q \neq Q')$ for $\neg(Q = Q')$.

1. *Unique Nash equilibrium in $\omega$-regular games.* Given a two-player game, and an $\omega$-regular objective $\beta$ [Cha05], the existence of a Nash equilibrium can be stated by $\hat{\exists} Q_1.\hat{\exists} Q_2.Equil(\beta, Q_1, Q_2)$, where

$$Equil(\beta, Q_1, Q_2) \overset{\text{def}}{=} \begin{cases} (\beta | Q_1 \wedge Q_2) \\ \wedge \hat{\forall} Q_2'.(Q_2 \neq Q_2') \Rightarrow (\neg\beta | Q_1 \wedge Q_2') \\ \wedge \hat{\forall} Q_1'.(Q_1 \neq Q_1') \Rightarrow (\neg\beta | Q_1' \wedge Q_2) \end{cases}$$

   Uniqueness of the Nash equilibrium is specified by:

$$\hat{\exists} Q_1.\hat{\exists} Q_2.Equil(\beta, Q_1, Q_2) \wedge Unique(Equil(\beta, Q_1, Q_2), Q_1, Q_2) \qquad (6)$$

   where $Unique(\alpha, Q_1, Q_2) = \hat{\forall} Q_1'.\hat{\forall} Q_2'.[\alpha \Rightarrow (Q_1 = Q_1') \wedge (Q_2 = Q_2')]$.
2. *Dominance of strategies.* For instance, a strategy $Q$ weakly dominates another strategy $Q'$ with respect to a goal $\beta$ [Ber07] whenever (7) holds.

$$\hat{\forall} R.[(\beta | Q' \wedge R) \Rightarrow (\beta | Q' \wedge R)] \wedge \hat{\exists} R.[(\beta | Q \wedge R) \wedge (\neg\beta | Q' \wedge R)] \qquad (7)$$

3. *Communication protocols.* By another reading of Corollary 1, a formula $\hat{\exists} \mathcal{Q}_C.(\alpha | \mathcal{Q}_C)$ states the existence of a $C$-outcome fixed once for all in which $\alpha$ is interpreted. If $\alpha$ contains a quantified sub-formula $\Delta \mathcal{Q}_{C'}.(\alpha' | \mathcal{Q}_{C'})$ ($\Delta \in \{\hat{\exists}, \hat{\forall}\}$), the statement $\alpha'$ is interpreted in $C'$-outcomes which lie "inside" the fixed $C$-outcome. Consider a system with two processors $a$ and $b$ which share a critical resource; we want to specify a protocol $mutex$ in charge of achieving the mutual exclusion. Consider the formula (8):

$$\hat{\exists} Q_{mutex}.(\mathbf{AG}\ (Exclusion \wedge \hat{\exists} Q_a.CritSec_a \wedge \hat{\exists} Q_b.CritSec_b) | Q_{mutex}) \qquad (8)$$

   where $Exclusion = \neg(CritSec_a \wedge CritSec_b)$, $CritSec_z = (\mathbf{AF}\, CritSec_z | Q_z)$. Protocol $mutex$ has a strategy to guarantee the safety property $\mathbf{AG}\ (Exclusion)$, on the one hand, and for each $z \in \{a, b\}$, to enable the recurrent liveness property $\mathbf{AG}\ (\hat{\exists} Q_z.(\mathbf{AF}\, CritSec_z\, | Q_z)$, on the other hand. Property $(\mathbf{AF}\, CritSec_z\, | Q_z)$ means that provided processor $z$ adopts policy $Q_{mutex}$, which consists e.g. in requiring the access to the critical resource and in maintaining this request, it will eventually access to critical section. The *commitment* of $mutex$ to the single strategy $Q_{mutex}$ entails fairness with respect to both processors, although not explicitly specified. Nevertheless, as explained in Section 7, solutions without commitment can also be specified.

   Many other examples of concepts in nonzero-sum games can be expressed in our setting, among which are all the proposals in [CHP07].

## 6 Automata Constructions

We assume that the reader is familiar with alternating parity tree automata (simply called *automata* when it is clear from the context), and with their relationship with the mu-calculus and parity games (we refer to [AN01], [KVW00], and [Wil01]).

Each formula of our calculus can be represented by alternating parity tree automata, thanks to a powerful construction which generalizes [RP03]. Remark that fix-points and quantifiers do not commute in general: consider the formulas $\alpha_\bot = \exists Q.\nu Z.(\mathbf{AX}\, Z \wedge Q \wedge \mathbf{EX}\, \neg Q)$ and $\alpha_\top = \nu Z.(\mathbf{AX}\, Z \wedge \exists Q.Q \wedge \mathbf{EX}\, \neg Q)$, interpreted on a single infinite path. Whereas the interpretation of $\alpha_\bot$ is the empty set, the one of $\alpha_\top$ is the entire set of states.

We start with an informal presentation of the construction's principles: Existential quantification corresponds to the projection, hence the need to handle non-deterministic automata [Rab69]; by the *Simulation Theorem* [MS95], every alternating automaton is equivalent to a non-deterministic automaton, and the procedure is effective with one exponential blow-up in the size of the automaton. Fix-point operators also have their counterpart on automata: by [AN01, Chapter 7, 7.2], automata can contain variables, we call them *extended automata*; their inputs are like $((\mathcal{S}, s), \mathrm{val})$, where $(\mathcal{S}, s)$ is as usual a model, and $\mathrm{val} : \mathrm{Var} \to 2^S$ is a valuation to interpret the variables, in the same line we interpret non-closed formulas. Extended automata have their own mu-calculus, and fix-point apply on them. Given an extented automaton $\mathcal{A}$, the extended automaton $\mu Z.\mathcal{A}$ can be defined in such a way that e.g. for an automaton $\mathcal{A}$ of a non-closed formula $\exists \mathcal{Q}.\alpha(Z)$, where $Z \in \mathrm{Var}$ is free in $\alpha(Z)$, the automaton $\mu Z.\mathcal{A}$ accepts the models of $\mu Z.(\exists \mathcal{Q}.\alpha(Z))$. Basically, the construction of Theorem 2 relies on three steps. (1) we build the automaton for $\alpha(Z)$; (2) by using the projection operation, we compute the automaton for $\exists \mathcal{Q}.\alpha(Z)$; (3) we build the automaton for $\mu Z.(\exists \mathcal{Q}.\alpha(Z))$. Notice that the automaton obtained for $\alpha(Z)$ may not be non-deterministic in general, either because e.g. $\alpha(Z)$ is of the form $\neg\alpha'(Z)$, or of the form $\alpha_1(Z) \wedge \alpha_2(Z)$. Preliminary to Step (2) we may therefore apply the Simulation Theorem (which by [AN01, Chapter 9] still applies to extended automata)entailing one exponential blow-up .

**Theorem 2.** *Let $m, n \geq 1$. For any $\alpha \in \mathrm{Q}D_\mu$, write $\kappa \in \mathbb{N}$ for the maximal number of nested quantifiers in $\alpha$. Then, there exists an alternating parity tree automaton $\mathcal{A}^k_\alpha$ with* $\max(\kappa, 0)$-EXPTIME$(|\alpha|)$ *states and* $\max(\kappa - 1, 0)$-EXPTIME$(|\alpha|)$ *priorities, which accepts exactly the models of $\alpha$ of branching degree k, where $k = m^n$, $m$ is the number of moves for each player, and $n$ is the number of players.*

Automata constructions established in Theorem 2 has many interesting corollaries: If we fix the maximal number $\kappa$ of $\hat{\exists}$ or $\hat{\forall}$ symbols in the formulas, the model-checking problem for $\mathrm{Q}D_\mu$ is $\kappa$-EXPTIME-complete (for a fixed branching degree of the structures); more precisely, it is $\kappa$-EXPTIME in the size of the formula, but polynomial in the size of the game structure $\mathcal{S}$. Indeed, for the upper bound, the proposed procedure amounts to solving "$\mathcal{S} \in L(\mathcal{A}^k_\alpha)$?", which in the light of [Jur98] for solving two-player parity games can be done with the announced complexity. For the lower bound, simply observe that $\mathrm{Q}D_\mu$ subsumes the proposal in [RP03]. As a consequence, the model-checking problem for $\mathrm{Q}D_\mu$ is non-elementary.

More interestingly, if coalition strategies solutions exist for a given existential $\mathrm{Q}D_\mu$ statements and some game structure, there always exists regular ones, that is describable by finite state machines. Indeed, while model-checking $\mathrm{Q}D_\mu$ formulas (using a classic parity game [Jur98]), any winning strategy for Player 0 in the parity game delivers adequate valuations of the quantified propositions; since parity games always have memoryless solutions, there always exist regular valuations of the propositions, yielding bounded memory solutions for coalition strategies.

## 7 Alternating Time Logics

We show that the alternating mu-calculus AMC and the "game logic" from [AHK02] GL are natural fragments of $\text{QD}_\mu$, as stated by Theorems 3 and 4 – we refer to [Pin07] for details; results for weaker logics such as ATL, Fair ATL, and ATL* follow from their natural embedding either into AMC or GL. As a corollary, automata constructions for alternating time logics can be derived from the procedure presented in Section 6; however, we briefly explain why these automata constructions can be significantly optimized.

For $\mathcal{Q} \subseteq \text{Prop}$, the *bounded $\mathcal{Q}$-relativization* $(\,\cdot\rfloor\mathcal{Q})$ is like the relativization (Definition 3), except that the downward propagation of propositions in the formulas terminates when a quantified sub-formula is encountered:

$$(\exists\mathcal{Q}'.\alpha'\rfloor\mathcal{Q}) = \exists\mathcal{Q}'.\alpha' \tag{9}$$

Relying on the bounded relativization, we define the modality $\hat{\exists}\mathcal{Q}_C(\,\cdot\rfloor\mathcal{Q}_C)$ which has the following semantics: $\hat{\exists}\mathcal{Q}_C(\alpha\rfloor\mathcal{Q}_C)$ states the existence a $C$-outcome where $\alpha$ holds, but where any further statement $\hat{\exists}\mathcal{Q}_{C'}.\alpha'$ is interpreted in the complete game structure, likewise the modalities of alternating time logics.

### 7.1 The Alternating-Time $\mu$-Calculus

The syntax of AMC formulas is $Q \mid \top \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid Z \mid \mu Z.\varphi(Z) \mid \langle\!\langle C\rangle\!\rangle \bigcirc \varphi$ with $Q \in \text{Prop}$, $C \subseteq \boldsymbol{P}$, and where each $Z \in \text{Var}$ occurs under an even number of negation symbols $\neg$ in $\varphi(Z)$. These formulas are interpreted over CGS's supplied with a valuation $\text{val} : \text{Var} \to 2^S$. Given $\varphi \in$ AMC, its interpretation $\varphi^{\mathcal{S}}(\text{val}) \subseteq S$ is inductively defined by:

$$\begin{array}{ll} Q^{\mathcal{S}}(\text{val}) = \lambda(Q) & (\neg\varphi)^{\mathcal{S}}(\text{val}) = S \setminus \varphi^{\mathcal{S}}(\text{val}) \\ \top^{\mathcal{S}}(\text{val}) = S & Z^{\mathcal{S}}(\text{val}) = \text{val}(Z) \end{array}$$

$$(\varphi_1 \vee \varphi_2)^{\mathcal{S}}(\text{val}) = \varphi_1^{\mathcal{S}}(\text{val}) \cup \varphi_2^{\mathcal{S}}(\text{val})$$

$$(\mu Z.\varphi(Z))^{\mathcal{S}}(\text{val}) = \bigcap\{S' \subseteq S \mid \varphi(Z)^{\mathcal{S}}(\text{val}[S'/Z]) \subseteq S'\}$$

$(\langle\!\langle C\rangle\!\rangle \bigcirc \varphi)^{\mathcal{S}}(\text{val})$ is the set of states $s \in S$ such that there exists a $C$-move from $s$ contained in $\varphi^{\mathcal{S}}(\text{val})$.

We define the mapping $\widehat{\cdot} :$ AMC $\to \text{QD}_\mu$ inductively by: formulas like $Q$, $\top$ and $Z$ are left unchanged, formulas like $\neg\varphi$, $\varphi_1 \vee \varphi_2$, and $\mu Z.\varphi(Z)$ are dealt inductively, and we set

$$(\widehat{\langle\!\langle C\rangle\!\rangle \bigcirc} \varphi) = \hat{\exists}\mathcal{Q}_C.(\mathbf{AX}\,\widehat{\varphi}\rfloor\mathcal{Q}_C)$$

where $\mathcal{Q}_C = \{Q_p \mid p \in C\}$ is a set of fresh atomic propositions. Notice that the size of $\widehat{\varphi}$ is linear in the size of $\varphi$.

**Theorem 3.** *Given a CGS $\mathcal{S}$, $\varphi \in$ AMC, and a valuation* $\text{val} : \text{Var} \to 2^S$*, we have* $\varphi^{\mathcal{S}}(\text{val}) = [\![\,\widehat{\varphi}\,]\!]^{\text{val}}_{\mathcal{S}}$.

## 7.2 The Logic $GL$

Formulas of $GL$ are of three types (the two last types are inherited from CTL$^*$):

*State formulas* are of the form $Q$, $\top$, $\neg\varphi$, or $\varphi_1 \vee \varphi_2$ – where $\varphi$, $\varphi_1$, and $\varphi_2$ are state formulas –, and $\exists C.\theta$ – where $\theta$ is a tree formula –.
*Tree formulas* are of the form $\varphi$ – where $\varphi$ is a state formula –, $\neg\theta$, or $\theta_1 \vee \theta_2$ – where $\theta$, $\theta_1$, and $\theta_2$ are path formulas –, and $\mathbf{E}\,\psi$ – where $\psi$ is a path formula –.
*Path Formulas* are of the form $\theta$ – where $\theta$ is a tree formula –, $\neg\psi$, $\psi_1 \vee \psi_2$, $\bigcirc\psi$, or $\psi_1\,\mathbf{U}\,\psi_2$ – where $\psi$, $\psi_1$, and $\psi_2$ are path formulas –.

We simply sketch the semantics of GL, and we assume that the reader is familiar with CTL$^*$ (see [AHK02] for details). Let $\varphi$ be a state formula, and let $(\mathcal{S}, s)$ be a rooted CGS. $\mathcal{S}, s \models \varphi$, indicating that $s$ satisfies $\varphi$ in $\mathcal{S}$, is defined by induction over $\varphi$. We focus on formulas like $\exists C.\theta$ (the others are dealt inductively or follow the semantics of CTL$^*$): $\mathcal{S}, s \models \exists C.\theta$ whenever there exists a $C$-outcome $\text{OUT}(\mathcal{Q}_C, \mathcal{S}, s)$ which satisfies $\theta$. Now, $\theta$ is a tree formula which in CTL$^*$, up to (non propositional) state sub-formulas $\exists C'.\varphi'$ which must be interpreted back inside $\mathcal{S}$. Let $\varphi^{\mathcal{S}}$ denote the set $\{s \in S \mid \mathcal{S}, s \models \varphi\}$.

To lighten the translation of GL into Q$D_\mu$, we first establish a translation of GL into a second order extension of CTL$^*$ (with decision modalities), written QDCTL$^*$; it generalizes the proposal of [ES84] since quantifications may occur in sub-formulas. In QDCTL$^*$, we denote a tree formula by $\theta$ (it may contain quantifications) and a path formula by $\pi$, and we write $\mathbf{A}\,\pi$ for $\neg\mathbf{E}\,\neg\pi$, and $\mathbf{G}\theta$ for $\neg(\top\,\mathbf{U}\,\neg\theta)$.

We adapt the definition of the bounded relativization (Section 7) to the syntax of QDCTL$^*$. The relativization of a path formula is conditioned by the path quantifier which binds the formula, as exemplified by the two expressions:

$$(\mathbf{EX}\ \cdot\ |Q) = \mathbf{EX}\,[Q \wedge (\cdot\,|Q)] \qquad (\mathbf{AX}\ \cdot\ |Q) = \mathbf{EX}\,[Q \Rightarrow (\cdot\,|Q)]$$

In order to distinguish the two cases, we define two relativizations of path formulas $(\ \cdot\rfloor_\forall \mathcal{Q})$ and $(\ \cdot\rfloor_\exists \mathcal{Q})$, and set $(\theta\rfloor\mathcal{Q}) = (\theta\rfloor_\forall\mathcal{Q})$ for all tree formula $\theta$. Let $\Delta \in \{\exists, \forall\}$, and $\theta$, $\theta_1$, and $\theta_2$ be tree formulas:

– $(Q\rfloor_\Delta\mathcal{Q}) = Q$, $(\top\rfloor_\Delta\mathcal{Q}) = \top$, and $(\exists\mathcal{Q}'.\theta\rfloor_\Delta\mathcal{Q}) = \exists\mathcal{Q}'.\theta$.
– $(\neg\theta\rfloor_\Delta\mathcal{Q}) = \neg(\theta\rfloor_\Delta\mathcal{Q})$ and $(\theta_1 \vee \theta_2\rfloor_\Delta\mathcal{Q}) = (\theta_1\rfloor_\Delta\mathcal{Q}) \vee (\theta_2\rfloor_\Delta\mathcal{Q})$.
– $(\mathbf{E}\,\pi\rfloor_\forall\mathcal{Q}) = (\mathbf{E}\,\pi\rfloor_\exists\mathcal{Q}) = \mathbf{E}\,(\pi\rfloor_\exists\mathcal{Q})$, and $(\mathbf{A}\,\pi\rfloor_\forall\mathcal{Q}) = (\mathbf{A}\,\pi\rfloor_\exists\mathcal{Q}) = \mathbf{A}\,(\pi\rfloor_\forall\mathcal{Q})$.
– $(\pi_1\,\mathbf{U}\,\pi_2\rfloor_\forall\mathcal{Q}) = [\mathcal{Q}{\Rightarrow}(\pi_1\rfloor_\forall\mathcal{Q})]\,\mathbf{U}\,[\mathcal{Q}{\Rightarrow}(\pi_2\rfloor_\forall\mathcal{Q})]$.
– $(\pi_1\,\mathbf{U}\,\pi_2\rfloor_\exists\mathcal{Q}) = [\mathcal{Q} \wedge (\pi_1\rfloor_\exists\mathcal{Q})]\,\mathbf{U}\,[\mathcal{Q} \wedge (\pi_2\rfloor_\exists\mathcal{Q})]$.

(we set similar definitions for path formulas). It can be shown that this definition is consistent with the definition of Section 7. For example, consider the CTL$^*$ formula $\mathbf{EF}\,Q_1 \wedge \mathbf{EF}\,Q_2$ which is equivalent to mu-calculus formula $(\mu Z.\mathbf{EX}\,Z \vee Q_1) \wedge (\mu Z.\mathbf{EX}\,Z \vee Q_2)$. Their respective bounded $Q$-relativization $\mathbf{EF}\,(Q \wedge Q_1) \wedge \mathbf{EF}\,(Q \wedge Q_2)$ (computed according to above) and $(\mu Z.\mathbf{EX}\,(Q{\wedge}Z){\vee}Q_1){\wedge}(\mu Z.\mathbf{EX}\,(Q{\wedge}Z){\vee}Q_2)$ (computed according to Section 7) remain equivalent.

We define $\widehat{\cdot}$ : GL $\rightarrow$ QDCTL$^*$ by induction: atomic propositions and $\top$ are left unchanged; formulas like $\neg\varphi$, $\varphi_1 \vee \varphi_2$ are dealt inductively, and we define

$$\widehat{\exists C.\theta} = \hat{\exists}\mathcal{Q}_C.(\widehat{\theta}\rfloor\mathcal{Q}_C)$$

Clearly, the size of $\widehat{\varphi}$ is linear in the size of $\varphi$, for any $\varphi \in \text{GL}$. Also, since $\hat{\exists}\mathcal{Q}_C.\alpha \in$ Q$D_\mu$ is definable in QDCTL$^*$ provided $\alpha$ is, the co-domain of $\widehat{\cdot}$ is indeed QDCTL$^*$.

**Theorem 4.** *For any state formula $\varphi \in \text{GL}$, $\varphi^{\mathcal{S}} = [\![\, \widehat{\varphi} \,]\!]_{\mathcal{S}}$.*

By an easy adaptation of e.g. the procedure of [Dam94], statements in QDCTL$^*$ can be effectively expressed in Q$D_\mu$.

### 7.3 A Note on Automata Constructions for Alternating Time Logics

Although our translation $\widehat{\cdot}$ of AMC or GL into Q$D_\mu$ may generate an arbitrary large number of nested symbols $\hat{\exists}$, the corresponding automata nevertheless remain small, if their construction is carefully conducted; applying Theorem 2 is actually avoidable. Because formulas $\widehat{\varphi}$ are obtained by bounded relativizations of Q$D_\mu$ formulas, a quantified proposition never occurs in strict quantified sub-formulas. This observation enables us to construct automata in a top-down manner, as opposed to the bottom-up procedure of Theorem 2; due to lack of space, we refer the reader to [Pin07] for the proof details of these constructions, which incidentally match the tight bounds from [AHK02].

## Acknowledment

We are extremely grateful to Dietmar Berwanger and Lukasz Kaiser for helpful discussions to improve the first version of the paper. Finally, we thank the reviewers for their highly relevant comments.

## References

[AdAdS$^+$06] Adler, B.T., de Alfaro, L., da Silva, L.D., Faella, M., Legay, A., Raman, V., Roy, P.: Ticc: A tool for interface compatibility and composition. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 59–62. Springer, Heidelberg (2006)

[AHK02] Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. Journal of the ACM 49(5), 672–713 (2002)

[AN01] Arnold, A., Niwinski, D.: Rudiments of mu-calculus. North-Holland, Amsterdam (2001)

[Ber07] Berwanger, D.: Admissibility in infinite games. In: Thomas, W., Weil, P. (eds.) STACS 2007. LNCS, vol. 4393, Springer, Heidelberg (2007)

[Cha05] Chatterjee, K.: Two-player nonzero-sum omega -regular games. In: Abadi, M., de Alfaro, L. (eds.) CONCUR 2005. LNCS, vol. 3653, pp. 413–427. Springer, Heidelberg (2005)

[CHP07] Chatterjee, K., Henzinger, T.A., Piterman, N.: Strategy logic. In: Caires, L., Vasconcelos, V.T. (eds.) CONCUR 2007. LNCS, vol. 4703, pp. 59–73. Springer, Heidelberg (2007)

[Dam94] Dam, M.: CTL$\star$ and ECTL$\star$ as fragments of the modal $\mu$-calculus. Theoretical Computer Science 126(1), 77–96 (1994)

[ES84] Emerson, E.A., Sistla, A.P.: Deciding full branching time logic. Information and Control 61, 175–201 (1984)

[GvD06] Goranko, V., van Drimmelen, G.: Complete axiomatization and decidability of the alternating-time temporal logic. Theoretical Computer Science 353(1), 93–117 (2006)

[HKQ98] Henzinger, T.A., Kupferman, O., Qadeer, S.: From *re*-historic to *ost*-modern symbolic model checking. In: Vardi, M.Y., Hu, A.J. (eds.) CAV 1998. LNCS, vol. 1427, pp. 195–206. Springer, Heidelberg (1998)

[Jur98] Jurdziński, M.: Deciding the winner in parity games is in UP $\cap$ co-UP. Information Processing Letters 68(3), 119–124 (1998)

[Koz83] Kozen, D.: Results on the propositional $\mu$-calculus. Theoretical Computer Science 27(3), 333–354 (1983)

[KP04] Kacprzak, M., Penczek, W.: Unbounded model checking for alternating-time temporal logic. In: Kudenko, D., Kazakov, D., Alonso, E. (eds.) AAMAS 2004, pp. 646–653. IEEE Computer Society, Los Alamitos (2004)

[KVW00] Kupferman, O., Vardi, M.Y., Wolper, P.: An automata-theoretic approach to branching-time model checking. Journal of the ACM 47(2), 312–360 (2000)

[LMO07] Laroussinie, F., Markey, N., Oreiby, G.: On the expressiveness and complexity of ATL. In: Seidl, H. (ed.) FOSSACS 2007. LNCS, vol. 4423, pp. 243–257. Springer, Heidelberg (2007)

[MS95] Muller, D.E., Schupp, P.E.: Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. Theoretical Computer Science 141(1–2), 69–107 (1995)

[Pin07] Pinchinat, S.: A generic constructive solution for concurrent games with expressive constraints on strategies (full version) August 2007 IRISA Internal Publication 1861, INRIA Research Report (to appear)

[Rab69] Rabin, M.O.: Decidability of second-order theories and automata on infinite trees. Trans. Amer. Math. Soc. 141, 1–35 (1969)

[RP03] Riedweg, S., Pinchinat, S.: Quantified mu-calculus for control synthesis. In: Rovan, B., Vojtáš, P. (eds.) MFCS 2003. LNCS, vol. 2747, pp. 642–651. Springer, Heidelberg (2003)

[SF06] Schewe, S., Finkbeiner, B.: Satisfiability and finite model property for the alternating-time $\mu$-calculus. In: Ésik, Z. (ed.) CSL 2006. LNCS, vol. 4207, Springer, Heidelberg (2006)

[vD03] van Drimmelen, G.: Satisfiability in alternating-time temporal logic. In: LICS, pp. 208–217. IEEE Computer Society, Los Alamitos (2003)

[Wil01] Wilke. T.: Alternating tree automata, parity games, and modal $\mu$-calculus. Bull. Soc. Math. Belg. 8(2) (May 2001)