A Characterization of Mixed-Strategy Nash Equilibria in PCTL Augmented with a Cost Quantifier

Pedro Arturo Góngora and David A. Rosenblueth

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
Universidad Nacional Autónoma de México
A.P. 20-726, C.P. 01000, México D.F., México
pedro.gongora@gmail.com, drosenbl@servidor.unam.mx

Abstract. The game-theoretic approach to multi-agent systems has been incorporated into the model-checking agenda by using temporal and dynamic logic to characterize notions such as Nash equilibria. Recent efforts concentrate on pure-strategy games, where intelligent agents act deterministically guided by utility functions. We build upon this tradition by incorporating stochastic actions. First, we present an extension of the Probabilistic Computation-Tree Logic (PCTL) to quantify and compare expected costs. Next, we give a discrete-time Markov chain codification for mixed-strategy games. Finally, we characterize mixed-strategy Nash equilibria.

1 Introduction

As a decision theory for multi-agent settings, game theory is undoubtedly in the interest of Computer Science and Artificial Intelligence. Recent works have incorporated this interest into the model-checking agenda, characterizing various game-theoretic notions in temporal and dynamic logic (cf. [3,5,9]). These works concentrate on pure-strategy games, where intelligent agents act deterministically guided by utility functions. The focus has been on characterizing notions such as Nash equilibria, Pareto optimality, and dominating/dominated strategies. In this paper, we build upon this tradition by incorporating stochastic actions, focusing on the characterization of mixed-strategy Nash equilibria for finite strategic games.

Previous works include, but are not limited to, characterizations of Nash equilibria. In [3] the author gives a characterization of backward induction predictions (i.e., Nash equilibria for extensive-form games) using a branching-time logic. In [5] the authors proceed in a similar vein, but using Propositional Dynamic Logic (PDL). Another similar approach is in [9], where the authors introduce Alternating-Time Temporal Logic (ATL) augmented with a counterfactual operator. This extension to ATL allows us to express properties such as "*if player* 1 *committed to strategy* a*, then* φ *would follow*". Counterfactual reasoning is then used to characterize Nash equilibria for strategic-form games. Further works emphasize other game-theoretic notions, such as automated mechanism-design (cf.

J. Dix, M. Fisher, and P. Novák (Eds.): CLIMA X, LNAI 6214, pp. 158–177, 2010.

[8,10]). None of these previous works handle mixed strategies. In [2], the authors make a quantitative analysis of a bargaining game, but do not provide a characterization of Nash equilibria. In [6], the author provides a characterization of Nash equilibria using a multi-valued temporal logic.

We start from Probabilistic Computation-Tree Logic (PCTL, [4]) augmented with costs as our underlying framework and proceed as follows. First, we present an extension of PCTL for quantifying the values in the expected-cost formulas (e.g., in $\mathcal{E}_{\bowtie x}[\varphi]$, x might be existentially or universally quantified). Next, we give a discrete-time Markov chain codification of a finite strategic game. The codification consists in unfolding the outcomes of a game, under a mixed-strategy profile, into a treelike structure that models the possibilities of action for each agent. Finally, we give a simple formula of the extended logic characterizing Nash equilibria under our codification.

The rest of the paper is organized as follows. Section 2 is devoted to presenting all the definitions used from game theory. In Sect. 3 we introduce discrete-time Markov chains and PCTL with costs. In Sect. 4 we present the cost-quantifier extension to PCTL and discuss its model checking. In Sect. 5 we present the game codification on Markov chains, a characterization of a Nash equilibrium, and prove its correctness. We finish with some final thoughts and a discussion of future and related work.

2 Strategic Games

Game theory studies the interaction between rational agents. Here, rationality is directly related to the maximization of utility. A game is just a formal description of that interaction. We will deal with games in which the sets of possible actions are those of individual players, sometimes called non-cooperative. For brevity, we will refer to non-cooperative games simply as games.

Of the two formalizations for games, strategic and extensive games, we will use the former. There exist several concepts of solution for games of which, arguably, the most widely known is that of Nash equilibrium. Broadly speaking, a Nash equilibrium is characterized by the decisions made by all players of a game, such that no player can increase her/his payoff by taking another action, assuming that every other player will stick to her/his decision.

This section is based on the first chapters of [7], to where we refer the reader for a more thorough discussion.

Definition 1 (Finite Strategic Game). *A finite strategic game is a structure:*

$$G = \langle N, \{A_i\}_{i \in N}, \{u_i\}_{i \in N} \rangle$$

where $N = \{1, \ldots, n\}$ *is a finite set of* n *agents,* A_i *is a finite set of the pure strategies of agent* i*,* $u_i : A \to \mathbb{R}$ *is the payoff or utility function of agent* i*, and* $A = \times_{i \in N} A_i$ *is the set of all pure-strategy profiles of* G*.*

Example 1 (Bach or Stravinsky). Consider the game known as *Bach or Stravinsky* (BoS) for players 1 and 2. The players wish to decide which concert to go to, Bach or Stravinsky. Player 1 prefers twice as much Bach, while player 2 prefers twice as much Stravinsky. Both players prefer to go to either concert over disagreement. Each player makes her/his choice independently of the other but accounting that preferences are common knowledge among them. Two-player finite strategic games can be described using payoff matrices. The matrix shown in Fig. 1 defines the utility functions for BoS, e.g., $u_1(B_1, B_2) = 2$, $u_2(B_1, B_2) = 1$.

	B_2	S_2
B_1	$2, 1$	$0, 0$
S_1	$0, 0$	$1, 2$

Fig. 1. Payoff matrix for the strategic game BoS

We use the following notational conventions. We use Latin letters a and a' to range over the set A of strategy profiles. If a is a strategy profile, we use a_i to refer to the strategy of agent i specified in a. Also, as a notational abuse, we denote with a_{-i} the strategy profile which specifies the strategies of every agent but i, such that if $a_i \in A_i$, then $(a_{-i}, a_i) \in A$. We also assume that A_i sets are pairwise disjoint and, when it is clear, we will identify a strategy profile $a \in A$ with another n-tuple a' iff they contain exactly the same elements regardless of the order.

Definition 2 (Best-Response Strategy and Nash Equilibrium). *Given a finite strategic game $G = \langle N, \{A_i\}_{i\in N}, \{u_i\}_{i\in N}\rangle$, we say that a strategy a_i is a best-response to strategy profile a iff $u_i(a_{-i}, a_i) \geq u_i(a_{-i}, a'_i)$ for each $a'_i \in A_i$. We say that a strategy profile a is a Nash Equilibrium of G iff every strategy a_i such that $a = (a_{-i}, a_i)$ is a best-response to a itself.*

Consider the previous definition and the matrix in Fig. 1. We can easily verify that both strategy profiles (B_1, B_2) and (S_1, S_2), are Nash equilibria of BoS (Example 1).

Definition 3 (Mixed Extension of a Game). *Let $\Delta(B)$ be the set of all probability distributions over the finite set B. For any finite strategic game:*

$$G = \langle N, \{A_i\}_{i\in N}, \{u_i\}_{i\in N}\rangle$$

we define its mixed extension as the structure:

$$\widehat{G} = \langle N, \{\Delta(A_i)\}_{i\in N}, \{U_i\}_{i\in N}\rangle$$

where $\Delta(A_i)$ is the set of all the mixed-strategies of player i, $U_i : \widehat{A} \to \mathbb{R}$ is the mathematical expectation of utility with respect to the probability measure induced by a mixed-strategy profile, and $\widehat{A} = \times_{i\in N}\Delta(A_i)$ is the set of all mixed-strategy profiles of $\widehat{G}$.

We use Greek letters α and α' to range over $\widehat{A}$. All other notational conventions for pure-strategy games are used as well for their mixed extensions. As α_i is a probability distribution over A_i, we use $\alpha_i(a_i)$ to denote the probability assigned by α_i to the event that pure strategy a_i is selected. For a mixed strategy α_i, the set of elements of A_i to which α_i assigns probability greater than 0 is called the *support* of α_i. We denote by $supp(\alpha_i)$ the subset of A_i whose elements are in the support of mixed strategy α_i. We say that *a mixed strategy α_i degenerates to a pure strategy a_i* iff it assigns probability 1 to the event a_i (i.e., $\alpha_i(a_i) = 1$). Finally, we say that mixed-strategy profile α is a Nash equilibrium of a game G if it is a Nash equilibrium of its mixed extension $\widehat{G}$.

The expected utility under some mixed-strategy profile is the mean value of such a utility. For some mixed-strategy profile α and player i the utility function is determined by:

$$U_i(\alpha) = \sum_{a \in A} p_\alpha(a) u_i(a)$$
$$p_\alpha(a) = \prod_{j \in N} \alpha_j(a_j)$$

The following theorem provides a useful characterization of Nash equilibria. See Lemma 33.2 in [7, p. 33] for a similar characterization and a proof for the *if* direction.

Theorem 1. *Given any finite strategic game $G = \langle N, \{A_i\}_{i \in N}, \{u_i\}_{i \in N} \rangle$, a mixed-strategy profile $\alpha \in \widehat{A}$ is a Nash equilibrium of G iff the following two conditions hold for each player $i \in N$:*

1. *The equality $U_i(\alpha_{-i}, a_i) = U_i(\alpha_{-i}, a'_i)$ holds for each (degenerate strategy) a_i and a'_i in $supp(\alpha_i)$.*
2. *The inequality $U_i(\alpha) \geq U_i(\alpha_{-i}, a_i)$ holds for each (degenerate strategy) a_i in $A_i - supp(\alpha_i)$.*

Consider again the matrix in Fig. 1. We can use Theorem 1 to verify that the mixed-strategy profile $\alpha = \left(\left(\frac{2}{3}, \frac{1}{3}\right), \left(\frac{1}{3}, \frac{2}{3}\right)\right)$ is a Nash equilibrium for BoS. For example, for player 1, we replace α_1 with one of the degenerate mixed-strategies that assigns probability 1 to B_1 or S_1, and compare the expected utility in both cases. For $B_1 = (1, 0)$ and $S_1 = (0, 1)$ we have: $U_1\left(B_1, \left(\frac{1}{3}, \frac{2}{3}\right)\right) = U_1\left(S_1, \left(\frac{1}{3}, \frac{2}{3}\right)\right) = \frac{2}{3}$. We can follow the same procedure for player 2 to conclude that α is a Nash equilibrium for BoS.

3 Markov Chains and PCTL

PCTL formulas describe qualitative and quantitative properties of probabilistic systems, sometimes modeled as Markov chains. These formulas address properties such as "*the probability of getting p satisfied is at least one half*", or "*the expected cost (or reward) of getting p satisfied is at most 10*". This section has the purpose of introducing Markov chains and PCTL. We first introduce Markov chains, that will serve as the semantic model for PCTL formulas. Next, we introduce PCTL syntax and satisfaction. For details on the material presented in this section, we refer the reader to the original paper [4], and also to the book [1].

Definition 4 (Discrete-Time Markov Chain). *A Discrete-Time Markov Chain (DTMC) is a structure:*

$$M = \langle S, s_{init}, \mathbf{P}, \mathbf{C}, \mathit{AtProp}, \ell \rangle$$

where S is a finite set whose elements are called states, s_{init} is a distinguished element of S which is called the initial state, $\mathbf{P} : S \times S \to [0,1]$ is a transition probability function, such that for any state $s \in S$, $\sum_{s' \in S} \mathbf{P}(s, s') = 1$, $\mathbf{C} : S \to [0, \infty)$ is a cost function, AtProp is a set of countably many atomic propositions, and $\ell : S \to 2^{AtProp}$ is a labelling function that marks each state in S with a subset of AtProp.

$Post_M(s) = \{s' \mid \mathbf{P}(s, s') > 0\}$ is the set of states which are possible to visit from s in one step. A *path* of a DTMC M is a possibly infinite sequence of states $\pi = s_0 s_1 \cdots$ such that for any s_i and s_{i+1}, $\mathbf{P}(s_i, s_{i+1}) > 0$. A path is finite if the sequence is finite. We denote by $Paths_M$ the set of all infinite paths of M, and by $Paths_M^{\text{fin}}$ the set of all finite paths of M. Given a path $\pi = s_0 s_1 \cdots s_i \cdots$, we use $\pi[i] = s_i$ to refer to the ith element of π, and $\pi[0, i]$ to refer to the finite prefix $s_0 \cdots s_i$ of π. The set $Paths_M(s) = \{\pi \mid \pi \in Paths_M \text{ and } \pi[0] = s\}$ denotes the set of all infinite paths of M beginning with s. Similarly, the set $Paths_M^{\text{fin}}(s) = \{\pi \mid \pi \in Paths_M^{\text{fin}} \text{ and } \pi[0] = s\}$ denotes the set of all finite paths of M beginning with s.

For any finite path π, the *cylinder set* of π is the set $Cyl(\pi) = \{\pi' \in Paths_M \mid \pi' \text{ has the prefix } \pi\}$. The *probability measure* Pr_s associated with a DTMC M and state s is that of the smallest σ-algebra Σ_s that contains all the cylinder sets $Cyl(\pi)$, for $\pi \in Paths_M^{\text{fin}}(s)$. For finite paths $\pi = s_0 \cdots s_n$, the probability of π is defined as $\mathbf{P}(\pi) = \prod_{i<n} \mathbf{P}(s_i, s_{i+1})$. The probability of $Cyl(\pi)$ under Pr_s is determined by $Pr_s(Cyl(\pi)) = \mathbf{P}(\pi)$. Let $\{C_i\}_{i \in I}$ be a collection of pairwise disjoint cylinder sets for some countable index I. The probability of the countable union $\bigcup_{i \in I} C_i$ is determined by $Pr_s\left(\bigcup_{i \in I} C_i\right) = \sum_{i \in I} Pr_s(C_i)$.

The application $\mathbf{C}(s)$ for some s in DTMC M denotes the cost (or reward, depending on the model in consideration) gained at *leaving* state s. Then, for any finite $\pi = s_0 \cdots s_n$ in $Paths_M^{\text{fin}}$ the *cumulative cost of* π is defined by $Cost_M(\pi) = \sum_{0 \le i < n} \mathbf{C}(s_i)$. Note that the cost of leaving the last state of a path is not in the sum, and that for paths consisting of a single state s, $Cost_M(s) = 0$.

For an infinite path $\pi \in Paths_M(s)$ and $A \subseteq S$, we define the *cumulative cost of reaching a state in* A as:

$$Cost_M(\pi, A) = \begin{cases} Cost_M(\pi[0, n]) & \text{if } \exists n \ge 0 : \pi[n] \in A \land \forall 0 \le i < n : \pi[i] \notin A \\ \infty & \text{otherwise} \end{cases}$$

For some state s and $A \subseteq S$, we define the set $\{s \models \mathcal{F}A\}$ of all finite paths $\pi = s_0 \cdots s_n$, such that $s_0 = s$, $s_n \in A$ and $\forall 0 \le i < n : s_i \notin A$. Note that the set $\{s \models \mathcal{F}A\}$ is measurable, therefore $Pr_s(\{s \models \mathcal{F}A\})$ is the *probability of reaching a state in A from s*. We now define the *expected cumulative cost of reaching a state in A from s* as:

$$ExpCost_M(s, A) = \begin{cases} \sum_{\pi \in \{s \models \mathcal{F}A\}} \mathbf{P}(\pi) Cost_M(\pi) & \text{if } Pr_s(\{s \models \mathcal{F}A\}) = 1 \\ \infty & \text{otherwise} \end{cases}$$

Definition 5 (PCTL Well-formed Formulas). *The set of well-formed formulas φ of PCTL for some countable set of atomic propositions AtProp is defined as the set generated by the following BNF grammar:*

$$\varphi ::= \top \mid p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \mathcal{P}_{\bowtie a}[\tau] \mid \mathcal{E}_{\bowtie c}[\varphi]$$
$$\tau ::= \mathcal{X}\varphi \mid \varphi\, \mathcal{U}\, \varphi$$

where $p \in AtProp$, $a \in [0,1]$, $c \in [0, \infty)$ and $\bowtie \in \{<, >, \leq, \geq\}$.

PCTL formulas describe properties of the infinite computations of a probabilistic system. We can study two classes of formulas: path or temporal formulas and state formulas. Path formulas inherit their meaning from LTL. $\mathcal{X}\varphi$ is satisfied by paths in which the next state satisfies φ. $\varphi\, \mathcal{U}\, \psi$ is satisfied by paths where there exists a future or present state that satisfies ψ, while all the previous states satisfy φ. State formulas inherit their meanings from CTL. The formula $\top$ is satisfied by every DTMC at every state. The formulas $\neg\varphi$, for negation, and $(\varphi \wedge \psi)$, for conjunction, have their usual meanings. The CTL path quantifiers are replaced with the operator $\mathcal{P}$. A formula $\mathcal{P}_{\bowtie a}[\tau]$ means that the probability of the temporal formula τ being satisfied is $\bowtie a$. $\mathcal{E}_{\bowtie c}[\varphi]$ is satisfied at states where the expected cost of reaching another state where φ is satisfied is $\bowtie c$.

The other connectives from the propositional logic are defined as usual:

$$\begin{aligned} \bot &= \neg\top \\ (\varphi \vee \psi) &= \neg(\neg\varphi \wedge \neg\psi) \\ (\varphi \rightarrow \psi) &= (\neg\varphi \vee \psi) \\ (\varphi \leftrightarrow \psi) &= ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)) \end{aligned}$$

where $\bot$ is not satisfied by any DTMC at any state, $(\varphi \vee \psi)$ is a disjunction, $(\varphi \rightarrow \psi)$ is a material implication and $(\varphi \leftrightarrow \psi)$ is a biconditional.

We also define the following derived formulas:

$$\begin{aligned} \mathcal{P}_{\bowtie a}[\mathcal{F}\varphi] &= \mathcal{P}_{\bowtie a}[\top\, \mathcal{U}\, \varphi] \\ \mathcal{P}_{\bowtie a}[\mathcal{G}\varphi] &= \mathcal{P}_{\overline{\bowtie} 1-a}[\mathcal{F}\neg\varphi] \\ \mathcal{P}_{=a}[\tau] &= (\mathcal{P}_{\geq a}[\tau] \wedge \mathcal{P}_{\leq a}[\tau]) \\ \mathcal{E}_{=a}[\varphi] &= (\mathcal{E}_{\geq a}[\varphi] \wedge \mathcal{E}_{\leq a}[\varphi]) \end{aligned}$$

where $\overline{<} = >$, $\overline{>} = <$, $\overline{\leq} = \geq$ and $\overline{\geq} = \leq$. The derived path formulas also inherit their meanings from LTL. $\mathcal{F}\varphi$ is satisfied by paths where there exists a future or present state that satisfies φ. $\mathcal{G}\varphi$ is satisfied by paths where φ is satisfied at every state of the path.

Definition 6 (PCTL Satisfaction). *Let $M = \langle S, s_{init}, \mathbf{P}, \mathbf{C}, AtProp, \ell \rangle$ be a DTMC. The satisfaction relation $\models$ between pairs (M, s) with $s \in S$ and well-formed formulas with atomic propositions in AtProp is defined as the smallest relation such that:*

$$\begin{aligned}
&(M,s) \models \top \\
&(M,s) \models p && \Leftrightarrow p \in \ell(s)\ (p \in AtProp) \\
&(M,s) \models \neg\varphi && \Leftrightarrow (M,s) \not\models \varphi \\
&(M,s) \models (\varphi \wedge \psi) && \Leftrightarrow (M,s) \models \varphi \text{ and } (M,s) \models \psi \\
&(M,s) \models \mathcal{P}_{\bowtie a}[\tau] && \Leftrightarrow p_s(\tau) \bowtie a \\
&(M,s) \models \mathcal{E}_{\bowtie c}[\varphi] && \Leftrightarrow e_s(\varphi) \bowtie c
\end{aligned}$$

where the functions $p_s(\tau)$ and $e_s(\varphi)$ are the following:

$$\begin{aligned}
p_s(\tau) &= Pr_s(\{\pi \in Paths_M(s) \mid \pi \models \tau\}) \\
e_s(\varphi) &= ExpCost_M(s, \{s' \mid (M, s') \models \varphi\})
\end{aligned}$$

Pr_s is the probability measure described before and the relation $\models$ between paths in $Paths_M$ and temporal formulas is defined as:

$$\begin{aligned}
&\pi \models \mathcal{X}\varphi && \Leftrightarrow \pi[1] \models \varphi \\
&\pi \models \varphi\,\mathcal{U}\,\psi && \Leftrightarrow \exists n \geq 0 : \forall i < n : \pi[i] \models \varphi \wedge \pi[n] \models \psi
\end{aligned}$$

If there is some φ such that $(M, s_{init}) \models \varphi$, then we say that φ is initially satisfied, and write $M \models \varphi$.

Note that the set $\{\pi \in Paths_M(s) \mid \pi \models \tau\}$ is a measurable set. The case $\tau = \mathcal{X}\varphi$ is straightforward. When $\tau = \varphi\,\mathcal{U}\,\psi$, the set coincides with the countable union of cylinder sets $Cyl(\pi')$, for finite prefix π' of π such that only its last state s_n satisfies ψ, and all its previous states s_i satisfy φ.

Example 2 (Simple protocol). Let M be the DTMC depicted in Fig. 2. This example models a simple protocol for sending a message through an unreliable channel [1]. After sending the message, a failure may occur with probability 0.1. In such a case, the protocol only dictates to try again. Consider the following:

- There are infinitely many possible paths from the initial state s_0 to the state s_3 (representing that the message is delivered). For example:

$$\begin{aligned}
\pi_0 &= s_0 s_1 s_3 \\
\pi_1 &= s_0 s_1 s_2 s_0 s_1 s_3 \\
\pi_2 &= s_0 s_1 s_2 s_0 s_1 s_2 s_0 s_1 s_3
\end{aligned}$$

- Each prefix spans a cylinder set. Therefore, it is possible to measure its probability. For example:

$$\begin{aligned}
Pr_{s_0}(Cyl(\pi_0)) &= 0.9 \\
Pr_{s_0}(Cyl(\pi_1)) &= 0.1 \cdot 0.9 \\
Pr_{s_0}(Cyl(\pi_2)) &= (0.1)^2 \cdot 0.9
\end{aligned}$$

– The probability of delivering the message, $p_{s_0}(\mathcal{F}\,delivered)$, is the (infinite) sum of the probabilities of each cylinder:

$$p_{s_0}(\mathcal{F}\,delivered) = 0.9 + (0.1)^1 \cdot 0.9 + (0.1)^2 \cdot 0.9 + \cdots = 1$$

This fact is expressed in PCTL as follows:

$$(M, s_0) \models \mathcal{P}_{=1}[\mathcal{F}\,delivered]$$

– For this model, the cumulative cost of a path where the message is eventually delivered (i.e., the path reaches s_3) counts the number of tries. For example:

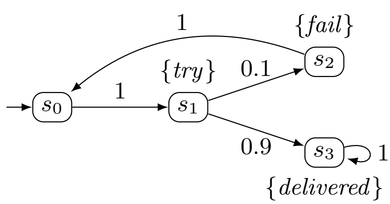
$$Cost_M(\pi_0) = 1$$
$$Cost_M(\pi_1) = 2$$
$$Cost_M(\pi_2) = 3$$

– The expected cost of reaching s_3 is calculated by the following sum:

$$e_{s_0}(delivered) = 1 \cdot 0.9 + 2 \cdot (0.1)^1 \cdot 0.9 + 3 \cdot (0.1)^2 \cdot 0.9 + \cdots = 1\frac{1}{9}$$

This value represents the average number of tries for the message to be delivered. Also, this fact is expressed in PCTL as follows:

$$(M, s_0) \models \mathcal{E}_{=1\frac{1}{9}}[delivered]$$



s	$\mathbf{C}(s)$
s_0	1
s_1	0
s_2	0
s_3	0

Fig. 2. Sending a message through an unreliable channel

If a DTMC is reduced to a Kripke structure, the PCTL formula $\mathcal{P}_{>0}[\tau]$ is equivalent to the CTL formula $\exists\tau$. On the contrary, the PCTL formula $\mathcal{P}_{=1}[\tau]$ *is not equivalent* to the CTL formula $\forall\tau$. See the example above: there is an infinite path never reaching a *delivered*-state, although $\mathcal{P}_{=1}[\mathcal{F}\,delivered]$ holds.

Given a DTMC M, a state s of M and a PCTL formula φ, the problem of deciding whether $(M, s) \models \varphi$ is called the *PCTL model-checking problem*. The basic algorithm for solving the model-checking problem consists in recursively computing the set $Sat(\varphi) = \{s \in S \mid (M, s) \models \varphi\}$. The computation of Sat for atomic formulas is given by the labelling function ℓ. Only basic set operations are needed for computing Sat for formulas with basic logical connectives. The computation of *Sat* for formulas $\mathcal{P}_{\bowtie}[\tau]$ and $\mathcal{E}_{\bowtie}[\varphi]$ involves the calculation of reachability probabilities and expected costs for every state. These tasks can be reduced to the problem of finding a solution to a system of linear equations. We cannot give a detailed explanation of these algorithms here; for the details we refer the reader to [4,1].

4 A Cost Quantifier for PCTL

In this section, we present the language of Cost-Quantified PCTL (CQ-PCTL). CQ-PCTL extends its ancestor by adding the possibility to quantify the values of the expected cost operator. The model-checking algorithm, however, is limited to formulas satisfying a syntactic constraint: *the occurrence of quantified variables cannot be nested*. We first define the syntax of the modified language, followed by the algorithm for model checking.

The syntax of CQ-PCTL is almost the same as that of PCTL. We modify the definition of expected cost formulas and add an extra clause to the grammar defining the syntax of PCTL formulas.

Definition 7 (CQ-PCTL Well-formed Formulas). *For some countable set of atomic propositions AtProp and some set Var of countably many variable names, the set of the well-formed formulas φ of CQ-PCTL is defined as the set generated by the following BNF grammar:*

$$\varphi ::= \top \mid p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \mathcal{P}_{\bowtie a}[\tau] \mid \mathcal{E}_{\bowtie c}[\varphi] \mid \exists x.\varphi$$
$$\tau ::= \mathcal{X}\varphi \mid \varphi\, \mathcal{U}\, \varphi$$

where $p \in AtProp$, $a \in [0,1]$, $c \in ([0,\infty) \cup Var)$, $x \in Var$ and $\bowtie \in \{<,>,\leq,\geq\}$.

From the basic syntax we can derive the universal quantifier:

$$\forall x.\varphi = \neg\exists x.\neg\varphi$$

Also, we say that *a variable x occurs free in φ* if x does not occur under the scope of an existential or universal quantifier; otherwise we say that x is *bound*. For a formula φ, we say that it has no *nested variables* if for any subformula $\mathcal{E}_{\bowtie x}[\psi]$ of φ: (i) the set of free variables of ψ contains at most x and (ii) the set of bound variables of ψ is empty. A formula with no free variables is called a *sentence*.

Remark 1. In the rest of this paper we will assume that formulas are sentences without nested variables.

We now define the satisfaction relation for CQ-PCTL. Because of remark 1, it is sufficient to incorporate a new clause to the PCTL satisfaction definition for treatment of the new existential formulas.

Definition 8 (CQ-PCTL Satisfaction). *The satisfaction relation is defined as follows for the new formulas:*

$$(M, s) \models \exists x.\varphi \Leftrightarrow \textit{there exists } c \in [0, \infty) \textit{ such that } (M, s) \models \varphi[x := c]$$

where $\varphi[x := c]$ *is the syntactic substitution replacing all the free occurrences of the variable* x *in* φ *by the non-negative real* c*. The satisfaction for the rest of the formulas is defined as for PCTL.*

Before presenting the model-checking algorithm for CQ-PCTL, it is necessary to introduce a transformation for the subformulas of $\exists x.\varphi$ so as to eliminate negative formulas. This is done by transforming φ into its *Positive Normal Form* (PNF) [1].

Definition 9 (Positive Normal Form). *A formula* φ *is non-negative iff* $\varphi \neq \neg\varphi'$ *for some* φ'*. Also, we say that* φ *is in Positive Normal Form if* φ*, and all of its subformulas, excepting atomic propositions and the constants* $\top$ *and* $\bot$*, are non-negative.*

Note that it is possible to transform every formula into another equivalent formula in PNF. This can be done by (i) introducing the constant $\bot$, the disjunction, and the universal quantifier into the base syntax; (ii) applying De Morgan's and double negation Laws; and (iii) applying the following additional equivalences:

$$\neg\mathcal{P}_{\bowtie a}[\tau] \Leftrightarrow \mathcal{P}_{\neg\bowtie a}[\tau] \tag{1}$$

$$\neg\mathcal{E}_{\bowtie c}[\varphi] \Leftrightarrow \mathcal{E}_{\neg\bowtie c}[\varphi] \tag{2}$$

where $\neg\!< \,=\, \geq$, $\neg\!> \,=\, \leq$, $\neg\!\leq \,=\, >$ and $\neg\!\geq \,=\, <$.[1] Also, we will use $PNF(\varphi)$ to denote a PNF formula equivalent to φ.

The readers familiar with CTL may notice that the release operator $\mathcal{R}$ (dual of $\mathcal{U}$) is not included in the basic language. The $\mathcal{R}$ operator is necessary for defining PNF for CTL, but not for (CQ-)PCTL. The reason is that in (1) the negation is absorbed by the predicate $\bowtie a$. Leaving out the $\mathcal{R}$ operator does not alter the expressiveness of (CQ-)PCTL in PNF. Despite this fact, the CQ-PCTL model-checking algorithm must take into account these implicit negations.

For their shared formulas, the model-checking algorithm for CQ-PCTL is essentially the same as for PCTL. In the rest of this section we will only present the method for calculating the set $Sat(\exists x.\varphi)$ for the new quantified formulas.

The algorithm for computing $Sat(\exists x.\varphi)$ consists of two steps. The first step computes a set $I(\exists x.\varphi)$ of intervals. These intervals are constraints that a value c assigned to x must satisfy for $\varphi[x := c]$ being satisfied at some state in S. The second step consists of several attempts to compute $Sat(\varphi[x := c])$, each attempt using a value for c taken from an interval obtained beforehand.

[1] Note that $\neg\bowtie$ negates $\bowtie$, while the $\overline{\bowtie}$ notation from Sect. 3 indicates inverting the direction of $\bowtie$.

The application $I(\exists x.\varphi) = i(x,\varphi)$ of Def. 10 below builds a set containing intervals of real numbers. The values c in these intervals may cause $\varphi[x := c]$ to be satisfied. Moreover, this set is constructed in such a way that if there is a satisfying c (i.e., $\varphi[x := c]$ is satisfiable at some state), then there is an interval A such that $c \in A \in i(x,\varphi)$. In such a case, it is also important that the interval contains only satisfying values (Theorem 2), for we have to choose just one of the possibly infinitely many values in the interval.

Definition 10 (Set $I(\exists x.\varphi)$). *Given a DTMC $M = \langle S, S_{init}, \mathbf{P}, \mathbf{C}, AtProp, \ell\rangle$ and a CQ-PCTL existential formula in PNF $\exists x.\varphi$, the set $I(\exists x.\varphi) = i(x,\varphi)$ of intervals of non-negative reals is inductively constructed by the following definition:*

$$\begin{aligned}
i(x,l) &= \{[0,\infty)\} \qquad (\textit{where } l \in AtProp \cup \{\top,\bot\})\\
i(x,\neg p) &= \{[0,\infty)\} \qquad (\textit{where } p \in AtProp)\\
i(x,(\psi \vee \psi')) &= i(x,\psi) \cup i(x,\psi')\\
i(x,(\psi \wedge \psi')) &= \{A \cap B \mid A \in i(x,\psi), B \in i(x,\psi')\}\\
i(x,\mathcal{E}_{\bowtie x}[\psi]) &= \{i(s,x,\mathcal{E}_{\bowtie x}[\psi]) \mid s \in S\}\\
i(x,\mathcal{E}_{\bowtie a}[\psi]) &= \|(i(x,\psi)) \cup \|(i(x,PNF(\neg\psi)))\\
i(x,\mathcal{P}_{\bowtie a}[\mathcal{X}\psi]) &= \|(i(x,\psi)) \cup \|(i(x,PNF(\neg\psi)))\\
i(x,\mathcal{P}_{\bowtie a}[\psi\,\mathcal{U}\,\psi']) &= \{A \cap B \mid A \in \|(i(x,\psi)), B \in \|(i(x,\psi')\})\\
&\quad \cup \{A \cap B \mid A \in \|(i(x,PNF(\neg\psi))), B \in \|(i(x,PNF(\neg\psi'))\})
\end{aligned}$$

where $i(s,x,\mathcal{E}_{\bowtie x}[\psi]) = \{r \in [0,\infty) \mid e_s(\psi) \bowtie r\}$ and $\|(\mathcal{I}) = \left\{\bigcap X \mid X \in 2^{\mathcal{I}}\right\}$ for $\mathcal{I}$ a set of intervals and $\bigcap\emptyset = [0,\infty)$.

The set $i(x,\varphi)$ is constructed inductively. At the basis of the induction there are the atoms and the formulas $\mathcal{E}_{\bowtie x}\psi$. The atoms do not pose any constraints on the values assignable to x. For $\mathcal{E}_{\bowtie x}\psi$, the computation of the bounds for the required intervals is straightforward using the PCTL model-checking algorithm. For disjunctions, the set interval may be in the union of the sets calculated for both disjuncts. The case of conjunction is more complicated: if there is a satisfying c, then c must be at the same time in one interval calculated for each one of the conjuncts. For the formulas $\mathcal{E}_{\bowtie a}[\psi]$ (resp. $\mathcal{P}_{\bowtie a}[\tau]$), a similar reasoning to that for the conjunctions is made. If there is a c such that $\psi[x := c]$ (resp. $\tau[x := c]$) satisfies the given predicate at each state of some subset of S, then c may need to be contained in several of the intervals calculated for the immediate subformulas of ψ (resp. τ).

Note that, because (1) and (2), the formulas $\mathcal{E}_{\bowtie a}[\psi]$ and $\mathcal{P}_{\bowtie a}[\tau]$ may represent an implicit negation contained in their predicate $\bowtie a$. For this reason, the algorithm must search for values that may satisfy the complementary paths when the immediate subformulas are negated.

Also, observe that the intersection $\bigcap\emptyset$ in Def. 10 is not the same as $\emptyset \cap \emptyset$. On the one hand, $\bigcap\emptyset$ represent a constraint posed by the empty set of states. On the other hand, $\emptyset \cap \emptyset$ is the intersection of the empty set with itself.

Example 3 (Interval computation). Let M be the DTMC depicted in the left panel of Fig. 3. Using the PCTL model-checking algorithm it is possible to compute the expected costs of reaching s_3 and s_4 (characterized by atoms p and q, resp.). These values are shown in the table in the right panel of the same figure. Consider the following examples:

– Given the values in the table we can compute the following sets:

$$i(x, \mathcal{E}_{\geq x}[p]) = \{[0, 10], [0, 5], [0, 0], [0, \infty)\}$$
$$i(x, \mathcal{E}_{\leq x}[q]) = \{[20, \infty), [15, \infty), [10, \infty), [0, \infty)\}$$

The intervals in these sets give a direct solution to the question of whether $(M, s) \models \exists x.\mathcal{E}_{\geq x}[p]$ or $(M, s) \models \exists x.\mathcal{E}_{\leq x}[q]$ hold for some s.

– Let $\varphi = \mathcal{P}_{=0.2}[\mathcal{E}_{\geq x}[p]\, \mathcal{U}\, \mathcal{E}_{\leq x}[q]]$ (for simplicity, assume that $\mathcal{P}_{=a}[\tau]$ is primitive in the basic language). The only assignable value to x making $(M, s_0) \models \exists x.\varphi$ to hold is 10. This value makes the path $\pi = s_0 s_2 s_3 \ldots$ satisfy the until subformula. The first two states of π satisfy the left part of the until, whereas the third state of π satisfies the right part. The satisfying value is computed as the intersection $[10, 10] = [0, 10] \cap [10, \infty)$ where:

$$[0, 10] \in \|(i(x, \mathcal{E}_{\geq x}[p])) \quad \text{(because of } s_0 \text{ and } s_2\text{)}$$
$$[10, \infty) \in \|(i(x, \mathcal{E}_{\leq x}[q])) \quad \text{(because of } s_3\text{)}$$

– Consider the problem of whether $(M, s_0) \models \exists x.\neg\varphi$ holds. Transforming to PNF we have: $PNF(\neg\varphi) = \mathcal{P}_{\neq 0.2}[\mathcal{E}_{\geq x}[p]\ \mathcal{U}\ \mathcal{E}_{\leq x}[q]]$. The previous solution makes the probability of the satisfying path equal to 0.2, but the problem ask for other complementary values of x. Following this reasoning, the algorithm tries to calculate intervals for complementary paths satisfying $\mathcal{E}_{<x}[p]\, \mathcal{U}\, \mathcal{E}_{>x}[q]$. One solution is the interval $(20, \infty) = [0, \infty) \cap (20, \infty)$ such that:

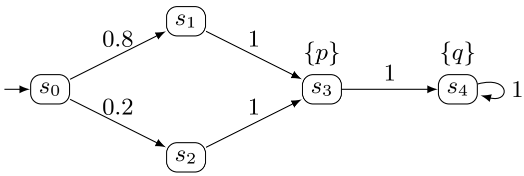
$$[0, \infty) \in \|(i(x, \mathcal{E}_{<x}[p])) \quad \text{(because of the empty set)}$$
$$(20, \infty) \in \|(i(x, \mathcal{E}_{>x}[q])) \quad \text{(because of } s_0\text{)}$$

A value c from this interval never satisfies the original until, thus $p_{s_0}(\mathcal{E}_{\geq c}[p]\, \mathcal{U}\, \mathcal{E}_{\leq c}[q]) = 0 \neq 0.2$, satisfying the predicate.

The following theorem states the property necessary for using the set $I(\exists x.\varphi)$ in the model-checking algorithm.

Theorem 2. *Let M be a DTMC, s a state of M, and $\exists x.\varphi$ a CQ-PCTL formula in PNF. Then, for all $c \in [0, \infty)$ the following two conditions hold:*

1. *If $(M, s) \models \varphi[x := c]$, then there exists $A \in i(x, \varphi)$ such that $c \in A$ and for all $c' \in A$, $(M, s) \models \varphi[x := c']$*
2. *If $(M, s) \not\models \varphi[x := c]$, then there exists $A \in i(x, PNF(\neg\varphi))$ such that $c \in A$ and for all $c' \in A$, $(M, s) \models PNF(\neg\varphi)[x := c']$.*



s	$\mathbf{C}(s)$	$e_s(p)$	$e_s(q)$
s_0	4	10	20
s_1	5	5	15
s_2	10	10	20
s_3	11	0	10
s_4	0	∞	0

Fig. 3. DTMC model for Example 3 and some expected costs

Theorem 2 suggests the last step of the algorithm. Given a CQ-PCTL formula in PNF $\exists x.\varphi$, we build the set $Sat(\exists x.\varphi)$ as follows:

$$Sat(\exists x.\varphi) = \bigcup_{A \in I(\exists x.\varphi)} \{Sat(\varphi[x := c]) \mid c \in A\}$$

Note that Theorem 2 also implies that it suffices to choose a single c from each interval A.

The basic algorithm presented here can be easily extended to the case where the values of $\mathcal{P}_{\bowtie a}$ formulas are also quantified. Also, it is possible to extend the results of this section to the general case of formulas, not only sentences, by enriching the models with variable-interpretation functions. For clarity, however, we restrict ourselves to sentences.

Some nesting constraints (remark 1) can be weakened, as long as there are no circular dependencies between the quantified variables. Nonetheless, the restriction for arbitrary nesting cannot be lifted. For example, consider the following simple sentence:

$$\exists x.\,(\varphi[x] \wedge \exists y.(\psi[x, y]))$$

where $\varphi[x]$ is a formula with free occurrences of x and $\psi[x, y]$ is a formula with free occurrences of both x and y. In this example, the problems of finding suitable values for x and y may be mutually dependent. For arbitrary nesting levels the problem may be even more complicated.

5 Model-Checking Games for Nash Equilibria

In this section, we show how to construct a DTMC $M_{G,\alpha}$ for a finite strategic game G and its mixed-strategy α. Although the construction is for strategic-form games, it is based on extensive forms.

Extensive-form games differ from strategic-form ones in that the sequentiality of the actions is important. An extensive game can be described by a tree structure. In a game tree each node represents the turn of only one player, and for each possible action, such a tree has one arc to another player's turn. In a strategic game it is assumed that each agent executes her/his action independently from and without knowing the other players' actions. To model this in an extensive game, states are grouped in such a manner that they represent the next player's uncertainty about previous actions (see Fig. 4 for an extensive form of BoS; dotted lines group player 1 moves as a single state, as player 2 does not know which action has been taken).

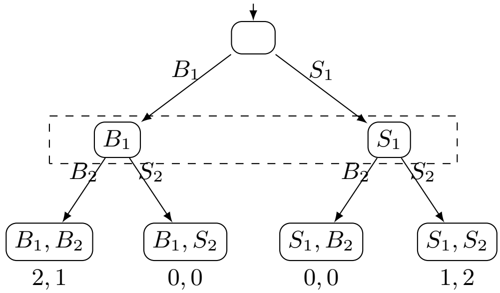


Fig. 4. An extensive form of BoS; utilities are shown under the leaf nodes

Given the game and the mixed-strategy profile, in our codification we build a structure similar to an extensive-form tree. In the built structure each arc, except the arcs leaving the root, is labelled with the probability that the mixed-strategy profile assigns to that particular action. As we cannot group states in a DTMC, we build one subtree for each player and each pure strategy. Each one of these subtrees models the situation where player i chooses some strategy a_i, but the other players follow the mixed-strategy.

By proceeding in this manner, each leaf node corresponds to one strategy profile of the strategic-form game. Consequently, each leaf node is associated with its utility via the cost function $\mathbf{C}$. As the cost function models the cost of *leaving* the state, we need to add a fictitious absorbing node below the leafs, representing the ending of the game.

Figure 5 illustrates one of the subtrees described above. Note that there is exactly one path from $s_{(i,a_i)}$ to the ending state, and going through each strategy profile. The arcs of such a path are the probabilities assigned by the mixed-strategy profile to that action. Hence, the expected cost coincides with the expected utility. We can therefore use a cost-quantified formula to compare expected costs and verify if Theorem 1 is applicable.

Definition 11 (DTMC Game Model). *For any game:*

$$G = \langle N, \{A_i\}_{i\in N}, \{u_i\}_{i\in N}\rangle$$

and a mixed-strategy profile α of its mixed extension $\widehat{G}$, we define the DTMC $M_{G,\alpha}$ as the structure:

$$M_{G,\alpha} = \langle S, s_{init}, \mathbf{P}, \mathbf{C}, AtProp, \ell\rangle$$

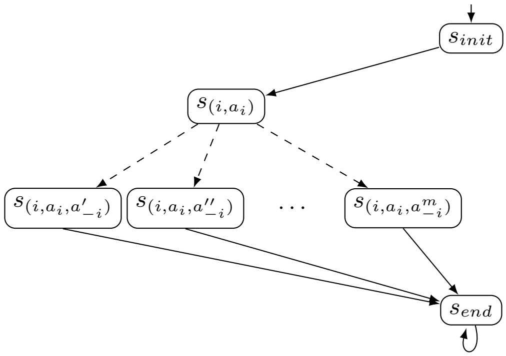


Fig. 5. After player i chooses strategy a_i other players make their own decisions, thus creating various strategy profiles

where the set of states is:

$$S = \{s_{init}\} \cup \{s_{end}\} \cup \{s_x\}_{x \in Idx}$$

Idx is the following index set:

$$Idx = \bigcup_{\substack{i \in N \\ a_i \in A_i}} \{(i, a_i), (i, a_i, a_{j_1}), \ldots, (i, a_i, a_{j_1}, \ldots, a_{j_m}) \\ \mid j_k \in N - \{i\}, j_k < j_{k+1}, \text{ and } (a_i, a_{j_1}, \ldots, a_{j_m}) \in A\}$$

The probability transition function is defined by cases:

$$\begin{aligned}
\mathbf{P}(s_{init}, s_{(i,a_i)}) &= 1/n && \text{for } i \in N, a_i \in A_i, n = |\bigcup_{j \in N} A_j| \\
\mathbf{P}(s_{(x)}, s_{(x,a_j)}) &= \alpha_j(a_j) && \text{for } j \in N, x \in Idx \\
\mathbf{P}(s_{(i,a)}, s_{end}) &= 1 && \text{for } i \in N, a \in A \\
\mathbf{P}(s_{end}, s_{end}) &= 1 && \\
\mathbf{P}(s, s') &= 0 && \text{otherwise}
\end{aligned}$$

The cost function is defined as follows:

$$\begin{aligned}
\mathbf{C}(s_{(i,a)}) &= u_i(a) && \text{for } a \in A \\
\mathbf{C}(s) &= 0 && \text{otherwise}
\end{aligned}$$

Finally, the set of atomic propositions and the labelling function are the following:

$$
\begin{aligned}
AtProp &= \{end\} \cup \bigcup_{i \in N} A_i \\
\ell(s_{end}) &= \{end\} \\
\ell(s_{(i,a_i)}) &= \{a_i\} && \text{for } i \in N, a_i \in A_i \\
\ell(s) &= \emptyset && \text{otherwise}
\end{aligned}
$$

Remark 2. The cost function of a DTMC requires non-negative values. We thus assume that games' utility functions also assign non-negative values only. If this is not the case, it is possible to add a constant sufficiently large to every value returned by the u_i functions, in order to make them non-negative. The addition of such a constant does no affect any result, as we only compare the mean values of utilities.

Example 4. (Model for BoS) The DTMC model M constructed for the game BoS and the mixed-strategy profile $\alpha = \left(\left(\frac{2}{3}, \frac{1}{3}\right), \left(\frac{1}{3}, \frac{2}{3}\right)\right)$ is depicted in Fig. 6. We can verify the following facts:

$$
\begin{aligned}
&(M, s_{(1,B_1)}) \models B_1 \wedge \mathcal{E}_{=\frac{2}{3}} end && (M, s_{(2,B_2)}) \models B_2 \wedge \mathcal{E}_{=\frac{2}{3}} end \\
&(M, s_{(1,S_1)}) \models S_1 \wedge \mathcal{E}_{=\frac{2}{3}} end && (M, s_{(2,S_2)}) \models S_2 \wedge \mathcal{E}_{=\frac{2}{3}} end
\end{aligned}
$$

For every player, all the pure strategies in the support of α yield the same payoff. Then, by Theorem 1 α is a Nash equilibrium. We can characterize this fact with a formula of CQ-PCTL:

$$
\begin{aligned}
(M, s_{init}) \models\ & \exists x. \left(\mathcal{P}_{>0}[\mathcal{X}(B_1 \wedge \mathcal{E}_{=x} end)] \wedge \mathcal{P}_{>0}[\mathcal{X}(S_1 \wedge \mathcal{E}_{=x} end)]\right) \\
& \wedge \exists x. \left(\mathcal{P}_{>0}[\mathcal{X}(B_2 \wedge \mathcal{E}_{=x} end)] \wedge \mathcal{P}_{>0}[\mathcal{X}(S_2 \wedge \mathcal{E}_{=x} end)]\right)
\end{aligned}
$$

The previous example shows how it is possible to characterize a mixed-strategy Nash equilibrium of a game with CQ-PCTL. Although it is not the case in BoS, by Theorem 1 we must verify that the expected cost is effectively a best response. This is achieved by verifying that the expected cost of deviating from the profile does not exceed that of the strategies in the support. The following definition captures this constraint.

Definition 12 (Mixed-strategy Nash Equilibria Characterization). *For a DTMC game model $M_{G,\alpha}$, the CQ-PCTL characterization of a mixed-strategy Nash equilibrium is the formula $NE_{G,\alpha}$ defined as follows:*

$$
\begin{aligned}
NE_{G,\alpha} &= \bigwedge_{i \in N} \exists x. \left(f_{supp(\alpha_i)} \wedge f_{\overline{supp}(\alpha_i)}\right) \\
f_{supp(\alpha_i)} &= \bigwedge_{a_i \in supp(\alpha_i)} \mathcal{P}_{>0}[\mathcal{X}(a_i \wedge \mathcal{E}_{=x} end)] \\
f_{\overline{supp}(\alpha_i)} &= \bigwedge_{a_i \in \overline{supp}(\alpha_i)} \mathcal{P}_{>0}[\mathcal{X}(a_i \wedge \mathcal{E}_{\leq x} end)]
\end{aligned}
$$

where $\overline{supp}(\alpha_i)$ denotes the complement of $supp(\alpha_i)$.

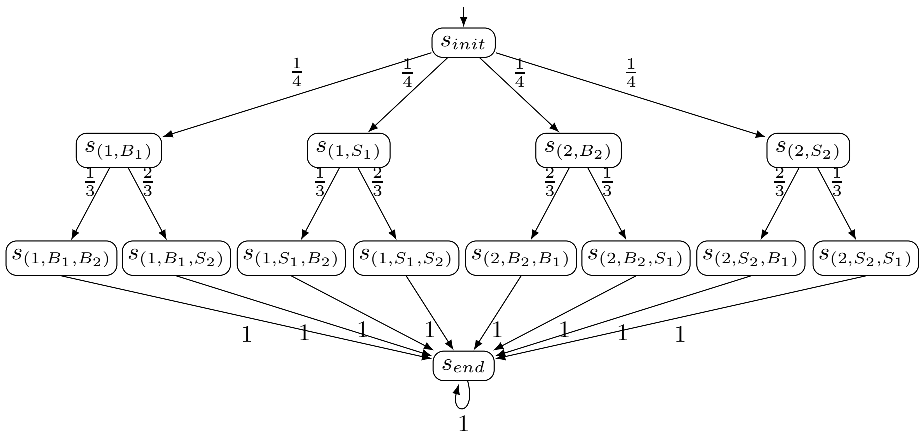


Fig. 6. DTMC for the game BoS and its mixed-strategy Nash equilibrium

Finally, we end this section stating a theorem asserting the correctness of the whole construction.

Theorem 3. *Let $M_{G,\alpha}$ be a DTMC game model. The mixed-strategy profile α is a Nash equilibrium of G if and only if $M_{G,\alpha} \models NE_{G,\alpha}$ holds.*

6 Conclusions

In this paper, we have addressed the problem of characterizing a mixed-strategy Nash equilibrium using PCTL enriched with an expected-cost quantifier: CQ-PCTL. Previous works include [3,5,9], where the authors give a characterization of pure-strategy Nash equilibria and other game-theoretic notions using temporal and dynamic logic. Our work also differs from [3,5] in that their characterization is validity-based, whereas our characterization is satisfaction-based, making our approach directly suitable for model-checking. In [2], the authors incorporate stochastic actions. They provide a model for a bargaining game (Rubinstein's alternating offers negotiation protocol, see [7]). With this model, the authors use PCTL formulas for making a quantitative analysis for several mixed strategies of the game. They, however, do not provide characterizations for Nash equilibria.

There are two general routes for future research: one dealing with CQ-PCTL and the other with its game-theoretic concepts.

As for the first route, recall that in Sect. 4 we presented an algorithm for model-checking a fragment of CQ-PCTL. The whole language includes formulas with nested variables. The nested variables introduce circular dependencies that our current algorithm cannot deal with. We do not know whether such an algorithm exists. As for the complexity of our algorithm, we do know that in the worst case it is exponential in the size of the formula. It is important to improve on this bound, if possible.

It would also be desirable, in the spirit of this work, to address other game solution concepts, such as evolutionary and correlated equilibria (cf. [7]). Beyond finite strategic games, it would be interesting to deal with other classes of games, like Bayesian and iterated games. Finally, further investigation would be necessary to determine if model-checking tools can be used to calculate solutions, besides characterizing them.

There is an implementation of the CQ-PCTL model checker and DTMC game construction of this paper written in the programming language Haskell. This implementation can be obtained by request to the authors.

Acknowledgments. We thank IIMAS and UNAM for their facilities. Pedro Arturo Góngora is sponsored by CONACyT. Finally, we also thank the anonymous referees for their comments.

References

1. Baier, C., Katoen, J.-P.: Principles of Model Checking. The MIT Press, Cambridge (2008)
2. Ballarini, P., Fisher, M., Wooldridge, M.: Automated game analysis via probabilistic model checking: a case study. In: Proceedings of the Third Workshop on Model Checking and Artificial Intelligence, pp. 125–137 (2006)
3. Bonanno, G.: Branching time, perfect information games, and backward induction. Games and Economic Behavior 36, 57–73 (2001)
4. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects of Computing 6, 102–111 (1994)
5. Harrenstein, P., van der Hoek, W., Meyer, J.-J., Witteveen, C.: On modal logic interpretations of games. In: Proceedings of the Fifteenth European Conference on Artificial Intelligence, pp. 28–32 (2002)
6. Jamroga, W.: A temporal logic for multi-agent mdp's. In: Proceedings of the AAMAS Workshop on Formal Models for Multi-Robot Systems, pp. 29–34 (2008)
7. Osborne, M.J., Rubinstein, A.: A Course in Game Theory. The MIT Press, Cambridge, Massachusetts (1994)
8. Pauly, M., Wooldridge, M.: Logic for mechanism design — a manifesto. In: In Proceedings of the 2003 Workshop on Game Theory and Decision Theory in Agent Systems, GTDT 2003 (2003)
9. van der Hoek, W., Jamroga, W., Wooldridge, M.: A logic for strategic reasoning. In: AAMAS 2005: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 157–164. ACM, New York (2005)
10. van der Hoek, W., Roberts, M., Wooldridge, M.: Social laws in alternating time: Effectiveness, feasibility, and synthesis. Synthese 156(1), 1–19 (2007)

A Proofs

Proof (Theorem 1). For the first part suppose that the equation $U_i(\alpha_{-i}, a_i) = U_i(\alpha_{-i}, a_i')$ does not hold for some i. Then either side must be greater than the other, but that contradicts the hypothesis of α being a Nash equilibrium, as i could increase his/her expected utility by assigning more probability to the pure strategy that increases his/her utility. The second part follows from the definition of Nash equilibria. The converse is direct: if both parts hold for each i, then it is impossible to increase some agent's utility by increasing the probability for some strategy (both parts show the worst-case probability of 1 for each strategy and agent), hence the profile is a best-response to itself. □

Proof (Theorem 2). We will show only the case when x occurs in φ. The proof is by induction on φ.

- Case $\varphi = \psi \vee \psi'$. Condition (1): the required interval A is in $i(x, \psi) \cup i(x, \psi')$. Condition (2): by the induction hypothesis we have corresponding intervals $A \in i(x, \mathit{PNF}(\neg\psi))$ and $B \in i(x, \mathit{PNF}(\neg\psi'))$. Therefore the required interval $A \cap B$ is in $i(x, \mathit{PNF}(\neg\psi) \wedge \mathit{PNF}(\neg\psi'))$.
- Case $\psi \wedge \psi'$. Condition (1): by the induction hypothesis we have corresponding intervals $A \in i(x, \psi)$ and $B \in i(x, \psi')$. Therefore the required interval $A \cap B$ is in $i(x, (\psi \wedge \psi'))$. Condition (2): by the induction hypothesis we have the corresponding intervals $A \in i(x, \mathit{PNF}(\neg\psi))$ and $B \in i(x, \mathit{PNF}(\neg\psi'))$. Therefore the required interval is in $i(x, \mathit{PNF}(\neg\psi)) \cup i(x, \mathit{PNF}(\neg\psi'))$.
- Case $\mathcal{E}_{\bowtie x}[\psi]$. Condition (1): direct by definition. Condition (2): also by definition and the equivalence $\neg\mathcal{E}_{\bowtie x} \Leftrightarrow \mathcal{E}_{\neg\bowtie x}$.
- Case $\mathcal{E}_{\bowtie a}[\psi]$ $(a \neq x)$. Condition (1): there are two subcases: (a) $e_s(\psi) \in [0, \infty)$ and (b) $e_s(\psi) = \infty$. (a) There is a path from s to a state in the nonempty set $\mathit{Sat}(\psi[x := c])$. By the induction hypothesis, for each state s_j in $\mathit{Sat}(\psi[x := c])$ there is a corresponding interval A_j. Then the required interval for $\mathcal{E}_{\bowtie a}[\psi]$ must be the intersection of some A_j intervals (contained in $\|(i(x, \psi)))$. (b) The set $\mathit{Sat}(\neg\psi[x := c])$ is nonempty. Again by the induction hypothesis, for each $s_j \in \mathit{Sat}(\neg\psi[x := c])$ there is a corresponding interval A_j (contained in $\|(i(x, \mathit{PNF}(\neg\psi))))$. Condition (2): holds by the equivalence $\neg\mathcal{E}_{\bowtie a} \Leftrightarrow \mathcal{E}_{\neg\bowtie a}$.
- Case $\mathcal{P}_{\bowtie a}[\mathcal{X}\psi]$. Condition (1): there are two possibilities: (a) $p_s(\mathcal{X}\psi[x := c]) \bowtie a$ holds when $\psi[x := c]$ is satisfiable at some states reachable from s in one step, and (b) $p_s(\mathcal{X}\psi) \bowtie a$ holds when $\psi[x := c]$ is not satisfiable at some states reachable from s in one step. For (a) the required interval is in $\|(i(x, \psi))$. For (b) the required interval is in $\|(i(x, \mathit{PNF}(\neg\psi)))$. Condition (2): holds by the equivalence $\neg\mathcal{P}_{\bowtie a} \Leftrightarrow \mathcal{P}_{\neg\bowtie a}$.
- Case $\mathcal{P}_{\bowtie a}[\psi\ \mathcal{U}\ \psi']$. Condition (1): once again, $p_s(\psi\ \mathcal{U}\ \psi') \bowtie a$ may hold when either the subformulas are satisfiable or not. The first possibility is included in $\{A \cap B \mid A \in \|(i(x, \psi)), B \in \|(i(x, \psi')\})$. The second and complementary possibility is included in $\{A \cap B \mid A \in \|(i(x, \mathit{PNF}(\neg\psi))), B \in \|(i(x, \mathit{PNF}(\neg\psi'))\})$. Condition (2): holds by the equivalence $\neg\mathcal{P}_{\bowtie a} \Leftrightarrow \mathcal{P}_{\neg\bowtie a}$.

□

Lemma 1. *Let $M_{G,\alpha}$ be a DTMC game model. For any player $i \in N$ and any strategy $a_i \in A_i$, the equation $U_i(a_i, \alpha_{-i}) = ExpCost_{M_{G,\alpha}}(s_{(i,a_i)}, s_{end})$ holds.*

Proof. Let $a = (a_i, a_{j_1}, \ldots, a_{j_m}) \in A$ be a profile such that its components follow the constraints of the index Idx. From the definitions of S and $\mathbf{P}$ we have that there is a unique path $\pi = s_{(i,a_i)} s_{(i,a_i,a_{j_1})} \cdots s_{(i,a_i,a_{j_1},\ldots,a_{j_m})} s_{\{end\}}$. For such a path, we have that:

$$\begin{aligned} Pr_{s_{(i,a_i)}}(\pi) &= \mathbf{P}(\pi) \\ &= \mathbf{P}(s_{(i,a_i)}, s_{(i,a_i,a_{j_1})}) \cdots \mathbf{P}(s_{(i,a_i,a_{j_1},\ldots,a_{j_m})}, s_{end}) \\ &= \prod_{j \in N} \alpha_j(a_j) \\ &= p_\alpha(a) \\ Cost_{M_{G,\alpha}}(\pi) &= \mathbf{C}(s_{(i,a_i)}) + \cdots + \mathbf{C}(s_{(i,a)}) \\ &= u_i(a) \end{aligned}$$

Moreover, the set of all such paths is equal to $P_{(i,a_i)} = \{s_{(i,a_i)} \models \mathcal{F}\{s_{end}\}\}$. Therefore:

$$\begin{aligned} ExpCost_{M_{G,\alpha}}(s_{(i,a_i)}, \{s_{end}\}) &= \sum_{\pi \in P_{(i,a_i)}} \mathbf{P}(\pi) Cost_{M_{G,\alpha}}(\pi) \\ &= \sum_{a \in A} p_\alpha(a) u_i(a) \\ &= U_i(\alpha) \end{aligned}$$

□

Proof (Theorem 3). We show the implication only in one direction (*if*); the proof for the converse is similar. Suppose as a contradiction that the consequent does not hold. Therefore, there must be some player $i \in N$ for which $\exists x. \left(f_{supp(\alpha_i)} \wedge f_{\overline{supp}(\alpha_i)}\right)$ is not initially satisfied. It follows by Lemma 1 that for any $a_i \in A_i$, if $u = U_i(a_i, \alpha_{-i})$, then $(M_{G,\alpha}, s_{(i,a_i)}) \models a_i \wedge \mathcal{E}_{=u} end$ holds (the first conjunct by def. of ℓ and the second conjunct by Lemma 1). Let $c = U_i(a_i, \alpha_{-i})$ for some $a_i \in supp(\alpha_i)$. Then, by the previous fact and Theorem 1, the formulas $f_{supp(\alpha_i)}[x := c]$ and $f_{\overline{supp}(\alpha_i)}[x := c]$ are both initially satisfied. A contradiction. □