Probabilistic Strategy Logic

Benjamin Aminof[1] , **Marta Kwiatkowska**[2] , **Bastien Maubert**[3] , **Aniello Murano**[3] , **Sasha Rubin**[3]
[1]JKU Linz and TU Wien, Austria
[2]University of Oxford, UK
[3]University of Naples "Federico II", Italy
aminof@forsyte.at, marta.kwiatkowska@cs.ox.ac.uk, bastien.maubert@gmail.com,
{murano, rubin}@unina.it

Abstract

We introduce *Probabilistic Strategy Logic*, an extension of Strategy Logic for stochastic systems. The logic has probabilistic terms that allow it to express many standard solution concepts, such as Nash equilibria in randomised strategies, as well as constraints on probabilities, such as independence. We study the model-checking problem for agents with perfect- and imperfect-recall. The former is undecidable, while the latter is decidable in space exponential in the system and triple-exponential in the formula. We identify a natural fragment of the logic, in which every temporal operator is immediately preceded by a probabilistic operator, and show that it is decidable in space exponential in the system and the formula, and double-exponential in the nesting depth of the probabilistic terms. Taking a fixed nesting depth, this gives a fragment that still captures many standard solution concepts, and is decidable in exponential space.

1 Introduction

There are a number of logical formalisms for expressing properties of strategic behaviour multi-agent systems, including Alternating-time Temporal Logics (ATL/ATL*) [Alur *et al.*, 2002], variants such as ATL with strategy contexts [Laroussinie and Markey, 2015], Strategy Logic (SL) [Chatterjee *et al.*, 2010; Mogavero *et al.*, 2014], and extensions of SL, e.g., allowing imperfect information [Belardinelli *et al.*, 2017; Cermák *et al.*, 2018]. The general trend of these logics is to capture increasingly sophisticated game-theoretic concepts, such as winning strategies, Nash equilibria, secure equilibria, in systems where agents have linear-temporal goals (e.g., safety, achievement, liveness).

In order to capture settings that combine probability and multiple agents, such logics have been extended, typically, by adding an operator that allows one to express the probability that an agent's goal holds. The main drawback of all existing strategic logics with probabilistic aspects falls into one of two camps: either, it is widely believed they cannot express classic solution concepts such as Nash equilibria, or they can express some classic solution concepts, but do so via dedicated operators for each solution concept, and thus are not a general purpose logic. In this work we define a logic that does not suffer these drawbacks. In particular, we define *Probabilistic Strategy Logic (*PSL*)* and study its model-checking problem.

The syntax of PSL is based on SL, but has additional arithmetic terms that allow one to compare the probabilities of formulas holding. The models of PSL are multi-agent stochastic transition systems (in contrast, the models of SL are deterministic). These transition systems capture many fundamental stochastic models in which agents have temporal goals, such as Markov chains and Markov decision processes with linear-temporal goals [Courcoubetis and Yannakakis, 1995].

We study the model-checking problem for two classic types of agents [Fagin *et al.*, 1995]: perfect-recall (who use *memoryful strategies*), and imperfect-recall (who use *memoryless strategies*, also called *Markovian strategies* or *policies*). We observe that the model-checking problem for perfect-recall agents is undecidable. On the other hand, we prove that the model-checking problem for imperfect-recall agents is decidable in EXPSPACE in the system and 3EXPSPACE in the formula. We also define a natural fragment, which we call the *vanilla* fragment, in which every temporal operator is immediately preceded by a probabilistic operator, and show that its model-checking problem is decidable in EXPSPACE in the system and the formula and 2EXPSPACE in the nesting depth of the probabilistic terms (we show that this depth is equal to 1 for many natural formulas).

To prove these results we reduce to the first-order theory of real arithmetic, known to be decidable in EXPSPACE [Ben-Or *et al.*, 1986]. Interestingly, the precise complexity of real arithmetic is a longstanding open problem [Berman, 1980]. Thus, any improvement in algorithms for real-arithmetic, also apply to model-checking PSL. On the other hand, we also prove that real-arithmetic is polynomial-reducible to the complexity of vanilla PSL with no nesting of probabilistic terms and a single agent using memoryless strategies. Thus, PSL and real-arithmetic are intimately connected logics.

2 Probabilistic Strategy Logic (PSL)

In this section we define *Probabilistic Strategy Logic (*PSL*)*. The syntax is inspired by Strategy Logic (SL) [Mogavero *et al.*, 2014] and the semantics are finite-state multi-agent stochastic transition-systems.

2.1 PSL Syntax

Fix finite non-empty sets of *atoms* AP, *agents* Ag, and *strategy variables* Var. The syntax of PSL is defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \tau \leq \tau$$
$$\tau ::= c \mid \tau^{-1} \mid \tau - \tau \mid \tau + \tau \mid \tau \times \tau \mid \mathbb{P}_\beta(\psi)$$
$$\psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi$$

where $p \in AP$, $x \in Var$, $c \in \mathbb{Q}$ and $\beta : Ag \to Var$.

Functions $\beta : Ag \to Var$ are called *bindings*; intuitively, they tell an agent which strategy to use. Formulas φ are called *history formulas* (they will be interpreted over histories). Formulas τ are called *arithmetic terms*, and the ones of the form $\mathbb{P}_\beta(\psi)$ are called *probabilistic terms*. The formulas ψ are called *path formulas* (they will be interpreted over paths). Every history formula is also a path formula. Note that every linear-temporal logic (LTL) formula [Pnueli, 1977] is a path formula (i.e., one that does not mention quantification or arithmetic terms). We introduce the usual shorthands for Boolean and arithmetic operations, e.g., $\varphi \to \varphi'$ is shorthand for $\neg\varphi \vee \varphi'$, and $\mathbb{P}_\beta(\psi) = c$ is shorthand for $(\mathbb{P}_\beta(\psi) \leq c) \wedge (c \leq \mathbb{P}_\beta(\psi))$. Write $[n]$ for the set $\{1, 2, \cdots, n\}$, and assume $Ag = [n]$ for some $n \in \mathbb{N}$.

Semantics (intuition). Formulas of PSL are interpreted over stochastic transition systems whose transitions are determined by the simultaneous actions of agents. A strategy is a rule that tells an agent, in every situation, what action to do with what probability. In general, this decision is based on the history of the evolution of the system (also known as randomised history-dependent strategies in the stochastic games literature, and similar to behavioural strategies in extensive-form games in the game-theory literature). In PSL, there are first-order variables x that vary over agent strategies; they are quantified in state formulas of the form $\exists x.\varphi$. The temporal operators are those of LTL, i.e., the path formula $\mathsf{X}\,\psi$ is read "ψ holds in the next step", and $\psi_1\,\mathsf{U}\,\psi_2$ is read "ψ_1 holds until ψ_2 holds". The meaning of the term $\mathbb{P}_\beta(\psi)$ is this: if agent i uses the strategy $\beta(i)$ (for each $i \in Ag$), then $\mathbb{P}_\beta(\psi)$ is the probability that a random path in the system satisfies ψ. Note that arithmetic terms are polynomials over the constants c and over the probabilistic terms $\mathbb{P}_\beta(\psi)$; in particular, there are no arithmetic variables in our logic.

2.2 PSL Examples

PSL can express central game-theoretic notions about probabilistic agents, e.g., domination (strict, weak, very weak), ϵ best-response, ϵ Nash-equilibria, Pareto optimality, social-welfare maximising equilbrium, k-resilience and t-immunity, cf. [Halpern, 2008]. Here, agents' goals can be given in LTL. This declarative specification of goals (in contrast to the classic case of having agents maximise the expected value of a reward function [Russell and Norvig, 1995]) is in line with the literature on stochastic games with ω-regular goals [Chatterjee and Henzinger, 2012]. Here are some examples.

Suppose there are n agents, and let $\psi_1, \cdots, \psi_k$ be LTL formulas representing agent goals (in general, these can be arbitrary path formulas of PSL). Let player i receive reward $c_{i,j} \in \mathbb{Q}$ if ψ_j holds on the outcome of the game. Thus, in an outcome π of the game, player i's utility is $\sum_{j:\pi\models\psi_j} c_{i,j}$. A player's payoff is defined to be its expected utility.

Notation. Let $\overline{x} = (x_1, \cdots, x_n)$ be a tuple of n strategy variables. To express that agent i uses strategy x_i we use bindings. Define the binding $\beta : Ag \to Var$ by $\beta(i) = x_i$ for $i \in Ag$. We write $\beta[i \mapsto y]$ to denote the binding that agrees with β on $j \neq i$, and that maps i to variable y.

Example 1 (Expected payoff)**.** The arithmetic term $\mathbb{E}_{i,\beta}(\overline{x})$ defined as $\sum_{j\in[k]} c_{i,j} \times \mathbb{P}_\beta(\psi_j)$ is the expected payoff for agent i given that agent l plays strategy x_l.

We now show how to express that a strategy is a best-response. These are stable in the sense that a player, knowing the other players' strategies, has no incentive to switch.

Example 2 (Best response)**.** Define the PSL formula $\mathrm{BR}_{i,\beta}(\overline{x})$ to be $\forall y\,\big(\mathbb{E}_{i,\beta[i\mapsto y]} \leq \mathbb{E}_{i,\beta}\big)$. This formula is read "for every strategy y, the expected payoff for agent i if she plays y (and agent $j \neq i$ plays x_j) is no larger than the expected payoff for agent i if she plays x_i (and agent $j \neq i$ plays x_j)". In other words, this expresses that agent i playing x_i is a best-response to the other agents playing x_j (for $j \neq i$). Note that replacing $\leq \mathbb{E}_{i,\beta}$ by $\leq \mathbb{E}_{i,\beta} + \epsilon$ (for non-negative rational ϵ) defines ϵ *best-response*.

If every player is playing a best-response strategy, the strategy profile is called a *Nash equilibrium*. This is the foundational solution concept in Game Theory.

Example 3 (Nash equilibrium)**.** Define the PSL formula $\mathrm{NE}_\beta(\overline{x})$ to be $\bigwedge_{i\in[1,n]} \mathrm{BR}_{i,\beta}(\overline{x})$. This formula is read "every agent i is playing a best response (given that agent j plays x_j)", i.e., that $\overline{x}$ is a Nash equilibrium. Moreover, $\mathrm{NE}_\beta(\overline{x}) \wedge \bigwedge_{i\in Ag} l_i \leq \mathbb{E}_{i,\beta}(\overline{x}) \leq r_i$ expresses an equilbrium where the expected payoff for agent i falls into a given interval $[l_i, r_i]$ [Ummels and Wojtczak, 2011].

Example 4 (Polynomial expressions)**.** The syntax can express polynomial constraints on probabilities. For instance, the formula $\mathbb{P}_\beta(\psi_1 \wedge \psi_2) = \mathbb{P}_\beta(\psi_1) \times \mathbb{P}_\beta(\psi_2)$ expresses that the events ψ_1 and ψ_2 (under the binding β) are independent.

Example 5 (Social-welfare maximisation)**.** The following expresses that $\overline{x}$ is a Nash equilibrium that maximises social welfare, where $\beta'(i) = x'_i$ for $i \in Ag$: $\mathrm{NE}_\beta(\overline{x}) \wedge \Big[\forall\overline{x}'.\mathrm{NE}_{\beta'}(\overline{x}') \to \sum_{i\in[n]} \mathbb{E}_{i,\beta'}(\overline{x}') \leq \sum_{i\in[n]} \mathbb{E}_{i,\beta}(\overline{x})\Big]$.

Example 6 (Rational Synthesis)**.** Recent studies of synthesis for rational agents have proposed asking for the existence of a strategy for the system (modeled as agent 1), to enforce a path formula ψ against all of the strategies of the rational environment (modeled as the agents $2, 3, \cdots, n$) that form a Nash equilibrium [Fisman *et al.*, 2010; Kupferman *et al.*, 2016]. In a stochastic system, one might ask that the path formula ψ should hold with probability one. This is expressed by the PSL formula $\exists x_1.\forall x_2\forall x_3 \ldots \forall x_n.\,\Big(\Big[\bigwedge_{i\in[2,n]} \mathrm{BR}_{i,\beta}(\overline{x})\Big] \to \mathbb{P}_\beta(\psi) = 1\Big)$. A variant formula can be written to express that ψ holds with probability one against *some* equilibrium.

2.3 PSL semantics

PSL formulas are interpreted over multi-agent stochastic transition systems. Here are the definitions.

Distributions. For a finite non-empty set X let $Dist(X)$ denote the set of *distributions* over X, i.e., functions $d : X \to [0,1]$ such that $\sum_{x \in X} d(x) = 1$. Write $x \in d$ for $d(x) > 0$. A *point distribution* is one for which $d(x) = 1$ for some element $x \in X$. If d_i is a distribution over X_i, then, writing $X = \prod_i X_i$, the *product distribution* is the distribution $d : X \to [0,1]$ defined by $d(x) = \prod_i d_i(x_i)$.

Systems. A *multi-agent stochastic transition system* (or simply *system*) G is a tuple (Ac, St, Tr, Lab) where

- Ac is a finite non-empty set of *actions*,
- St is a finite non-empty set of *states*,
- $Tr : St \times Ac^{Ag} \to Dist(St)$ is a *transition function*,
- $Lab : St \to 2^{AP}$ is a *labelling function*.

We say that G is *deterministic* (instead of stochastic) if every $Tr(v,a)$ is a point distribution.

Multi-agent stochastic transition systems can model games in extensive form, repeated one-stage games, Markov chains (no agent), Markov decision processes (one agent), decentralised Markov decision processes (multiple agents), and stochastic games (however, instead of agents being rewarded at states, in our setting agents are rewarded if given path formulas hold). Such systems have been used to model multi-agent path-planning with unreliable actuators, human in the loop UAV mission planning, autonomous urban driving, randomised communication- and security-protocols, energy management systems, etc. [Kwiatkowska *et al.*, 2012; Chen *et al.*, 2013b; Svorenová and Kwiatkowska, 2016].

Markov chains. A *Markov chain* M is a tuple (Q, p) where Q is a set of states and $p \in Dist(Q \times Q)$ is a distribution. The values $p(s,t)$ are called *transition probabilities* of M. A Markov chain M and a state $q \in Q$ induce a canonical probability space on the set of infinite paths starting in q [Kemeny and Snell, 1976].

Actions and paths. A *joint-action* a is an element of Ac^{Ag}. A *path* π is a non-empty sequence $v_1 v_2 \cdots$ of states such that there exists a sequence $a_1 a_2 \cdots$ of joint-actions such that $v_{i+1} \in Tr(v_i, a_i)$ for every i. We write $len(\pi)$ for the length of π, and $\pi_{\leq i}$ for the prefix of π of length i (for $i \leq len(\pi)$). Finite paths are called *histories*, and the set of all histories is denoted $Hist$. Write $last(h)$ for the last state of a history h.

Strategies. A *strategy* is a function $\sigma : Hist \to Dist(Ac)$. Let Σ denote the set of all strategies. A *strategy profile* is a tuple ρ of strategies, one for each agent. We write ρ_i for the strategy of agent i. We overload notation and let ρ also denote the function $Hist \to Dist(Ac^{Ag})$ that maps h to the product distribution of the $\rho_i(h)$'s. *I.e., a strategy profile assigns, for each history h, a distribution over the set of joint-actions.*

Probability space on outcomes. An *outcome* of a strategy profile ρ and a history h is a (finite or infinite) path π that starts with h and is extended by ρ, i.e., i) $\pi_1 = h$, and ii) for every $k \geq len(h)$ there exists $a_k \in \rho(\pi_{\leq k})$ such that $\pi_{k+1} \in Tr(\pi_k, a_k)$. The set of outcomes of a strategy profile ρ and history h is denoted $out(\rho, h)$.

A given transition system G, strategy profile ρ, and history h induce an infinite-state Markov chain $G_{\rho,h}$ whose states are the histories in $out(\rho, h)$ and whose transition probabilities $p(h', h's')$ are defined as $\sum_a \rho(h')(a) \times Tr(last(h'), a)(s')$. The Markov chain $G_{\rho,h}$ induces a canonical probability space: its sample space can be identified with the set of infinite paths in $out(\rho, h)$, and its measure is denoted $\mu_{\rho,h}$.

Valuations and bindings. Recall (from the syntax) that a binding is a function $\beta : Ag \to Var$, i.e., it assigns variables to agents. A *valuation* is a partial function $\nu : Var \to \Sigma$, i.e., it assigns strategies to some variables. If the range of β is contained in the domain of ν, then composing them (left to right), we see that $\nu \circ \beta : Ag \to \Sigma$ is a joint strategy.

Free variables. A variable x is *free* in a PSL formula if it has a subformula of the form $\mathbb{P}_\beta(\psi)$ such that $x \in \beta(Ag)$ but $\mathbb{P}_\beta(\psi)$ is not in the scope of $\exists x$. A history formula with no free variables is called a *sentence*.

PSL Semantics.[1] Formulas of PSL are interpreted given a transition system G, a valuation $\nu : Var \to \Sigma$ whose domain contains the free variables of the formula, and either a history h (for history formulas), or an infinite path π and an index $i \in \mathbb{N}$ (for path formulas).

The semantics of history formulas is as follows:

- $G, \nu, h \models p$ iff $p \in Lab(last(h))$
- $G, \nu, h \models \neg\varphi$ iff $G, \nu, h \not\models \varphi$
- $G, \nu, h \models \varphi_1 \vee \varphi_2$ iff $\bigvee_j \ G, \nu, h \models \varphi_j$
- $G, \nu, h \models \exists x.\varphi$ iff $\exists \sigma \in \Sigma.\ G, \nu[x \mapsto \sigma], h \models \varphi$
- $G, \nu, h \models \tau_1 \leq \tau_2$ iff [2] $val_{\nu,h}(\tau_1) \leq val_{\nu,h}(\tau_2)$

where

- $val_{\nu,h}(c) = c$ and $val_{\nu,h}(\tau^{-1}) = (val_{\nu,h}(\tau))^{-1}$
- $val_{\nu,h}(\tau \oplus \tau') = val_{\nu,h}(\tau) \oplus val_{\nu,h}(\tau')$ for $\oplus \in \{-, +, \times\}$
- $val_{\nu,h}(\mathbb{P}_\beta(\psi)) = \mu_{\nu\circ\beta,h}(\{\pi : G, \nu, \pi, 1 \models \psi\})$.[3]

The semantics of the path formulas are defined as follows:

- $G, \nu, \pi, i \models \varphi$ iff $G, \nu, \pi_{\leq i} \models \varphi$
- $G, \nu, \pi, i \models \neg\psi$ iff $G, \nu, \pi, i \not\models \psi$
- $G, \nu, \pi, i \models \psi_1 \vee \psi_2$ iff $\bigvee_j \ G, \nu, \pi, i \models \psi_j$
- $G, \nu, \pi, i \models \mathsf{X}\,\psi$ iff $G, \nu, \pi, i+1 \models \psi$
- $G, \nu, \pi, i \models \psi_1 \,\mathsf{U}\, \psi_2$ iff $\exists k \geq i$ such that $G, \nu, \pi, k \models \psi_2$ and $\forall j \in [i, k).\ G, \nu, \pi, j \models \psi_1$

The *model-checking* problem is to decide, given a system G, a state $s \in St$, and a sentence φ, whether or not $G, s \models \varphi$.

---

[1] The standard semantics of SL shifts strategies during the evaluation of temporal operators [Mogavero *et al.*, 2014; Bouyer *et al.*, 2016]. Here instead, inspired by [Berthon *et al.*, 2017], we carry the accumulated history with us and do not shift strategies.

[2] To avoid the anomaly of division by zero, we let $G, \nu, h \not\models \tau_1 \leq \tau_2$ if τ_1 or τ_2 contain a subterm τ^{-1} for which $val_{\nu,h}(\tau) = 0$.

[3] Recall from the definitions that $\mu_{\nu\circ\beta,h}(\cdot)$ is the probability measure on the infinite paths starting with h and consistent with the joint-strategy $\nu \circ \beta$.

2.4 Discussion of the semantics

Restricted classes of strategies. A strategy σ is *deterministic* (or *pure*) if $\sigma(h)$ is a point distribution for every history h. A strategy σ is *finite-memory* if there is a deterministic finite-state machine (St, Q, q_0, δ) (over alphabet St) and output function $out : Q \times St \to Dist(Act)$ such that $\sigma(h) = out(\delta(q_0, h), last(h))$ for all $h \in Hist$. Intuitively, the strategy σ can be implemented with $\log |Q|$ bits of memory. In case $|Q| = 1$, we call the strategy *memoryless*. Equivalently, a strategy σ is memoryless if $last(h) = last(h')$ implies $\sigma(h) = \sigma(h')$. We can safely write a memoryless strategy as a function $\sigma : St \to Dist(Ac)$ whose domain is St instead of $Hist$, and a strategy profile of memoryless strategies as a function $\rho : St \to Dist(Ac^{Ag})$. In the definition of PSL semantics, if we let Σ be the set of memoryless strategies, we can replace histories h by states s and write, e.g., $G, \nu, s \models \varphi$. Also, in the semantics of $\mathbb{P}_\beta(\psi)$, the induced Markov chain $G_{\rho,s}$ where $\rho = \nu \circ \beta$, can be taken to be *finite-state*, i.e., the state set is St and the transition probability from v to w is $\sum_{a \in Ac^{Ag}} \rho(v)(a) \times Tr(v, a)(w)$.

PSL and non-probabilistic strategic logics. Restricting the semantics of PSL to deterministic transition systems and deterministic strategies results in (a notational variant of) SL. Indeed, in this case $\mathbb{P}_\beta(\psi) \in \{0, 1\}$, and $\mathbb{P}_\beta(\psi) = 1$ iff ψ holds. This way, PSL subsumes SL, and thus also, e.g., ATL*.

Expressing Nash equilibria. Since PSL includes comparisons between probabilities, it can express *the standard meaning of Nash Equilibrium (NE)*, i.e., that no deviation can increase an agent's expected payoff, see Example 3. In comparison, ordinary Strategy Logic (SL) [Mogavero *et al.*, 2014] can express NE for deterministic arenas and pure strategies. Stochastic Game Logic (SGL) [Baier *et al.*, 2012] is a similar logic to PSL, except it can only compare probabilities to constants (see Sec. 4 for a detailed comparison with SGL).

Vanilla PSL. We define a syntactic fragment of PSL that we call *vanilla* PSL. Intuitively, the restriction is similar to that put on CTL* to obtain CTL; a similar restriction is defined for a fragment of SL closely related to ATL*, see [Malvone *et al.*, 2018]. The syntax of vanilla PSL is the same as that of PSL except that the path formulas are given by the grammar: $\psi ::= \varphi \mid \mathsf{X}\varphi \mid \varphi \,\mathsf{U}\, \varphi$. The relevance of this fragment can be seen by noting that all the PSL formulas in the Examples in Section 2.2 are also vanilla PSL formulas assuming that the goal of agent i (for every $i \in Ag$) is a vanilla PSL path formula ψ_i. For instance, safety and reachability goals can be expressed in vanilla PSL. We will see that our algorithm for model-checking vanilla PSL (assuming all strategies are memoryless) has a substantially lower complexity than PSL.

3 Model checking PSL

PSL with unrestricted strategies is too expressive to be decidable. In fact, this is already true when restricted to a single agent, deterministic finite-memory strategies, and formulas of the form $\exists x.\varphi$ where φ is a formula of the grammar $p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbb{P}_\beta(\mathsf{F}\,\varphi) \bowtie c$ for $\bowtie \in \{>, =\}$ and $c \in \mathbb{Q} \cap [0, 1]$ (with $\beta(x) = 1$); see [Brázdil *et al.*, 2006].

For memoryless strategies, the news is better:

Theorem 1. *1. Model-checking* PSL *restricted to memoryless strategies is decidable in space exponential in the system and triple-exponential in the formula.*

2. Model-checking vanilla PSL *restricted to memoryless strategies is decidable in space exponential in the system and the formula and double-exponential in the nesting-depth of probabilistic terms of the formula.*

In particular, for vanilla PSL with a fixed nesting depth of probabilistic terms, this gives an EXPSPACE algorithm. Note that in all the examples of Section 2.2 this depth is 1. To prove Theorem 1, we reduce to model-checking real arithmetic.

Real arithmetic. refers to first-order logic of the structure $\mathfrak{R} := (\mathbb{R}, +, \times, \leq, 0, 1)$. This logic is very expressive, e.g., it is not hard to see that solutions of systems of polynomial inequalities with rational coefficients are definable in this logic, and that one can define the relation $=$, and the operations $r - r'$ and r^{-1}, e.g., r^{-1} can be represented by a new variable r' and the constraint $r' \times r = 1$ (note that if $r = 0$ this constraint evaluates to false). A formula Φ of real arithmetic, with free variables in the set X, is interpreted given an evaluation $\chi : X \to \mathbb{R}$, and satisfaction is denoted $\mathfrak{R}, \chi \models \Phi$.

Theorem 2. *[Ben-Or* et al.*, 1986] Model-checking the structure $\mathfrak{R}$ of real-arithmetic against first-order logic sentences is decidable in space logarithmic in the size of the formula and exponential in the number of quantifiers in the formula.*

Translation of PSL to $\mathfrak{R}$. Fix a transition system G. Introduce real-valued first-order variables $r_{x,s,a}$ for every strategy variable $x \in Var$, state $s \in St$, and action $a \in Ac$. Intuitively, $r_{x,s,a}$ represents the probability that strategy x does action a in state s. An evaluation χ that maps $r_{x,s,a}$ to $\nu(x)(s)(a)$ (for every x, s, a) is said to be *compatible* with ν. The algorithm inductively translates a PSL history formula φ with free variables $\overline{x}$, and a state t, into a real arithmetic formula $\Phi_{\varphi,t}$ of $\mathfrak{R}$ with free variables $r_{x,s,a}$ for $x \in \overline{x}, s \in St$ and $a \in Ac$, such that for every valuation ν and every evaluation χ compatible with it, we have that

$$G, \nu, t \models \varphi \text{ iff } \mathfrak{R}, \chi \models \Phi_{\varphi,t} \tag{1}$$

If φ is a sentence then so is $\Phi_{\varphi,t}$. Thus, to model check $G, t \models \varphi$, simply model check $\mathfrak{R} \models \Phi_{\varphi,t}$.

Translating the Boolean cases The boolean cases are as follows: $\Phi_{p,t} :=$ true if $p \in Lab(t)$, and false otherwise; $\Phi_{\varphi_1 \vee \varphi_2, t} := \Phi_{\varphi_1,t} \vee \Phi_{\varphi_2,t}$, and $\Phi_{\neg\varphi,t} := \neg\Phi_{\varphi,t}$.

Translating the existential case The translation of $\exists x.\varphi$ is done by quantifying the corresponding real variables $r_{x,s,a}$, and expressing that they code a distribution. Formally:

$$\Phi_{\exists x.\varphi,t} := (\exists r_{x,s,a})_{s \in St, a \in Ac} \, [Dist_x \wedge \Phi_{\varphi,t}]$$

where $Dist_x$ is the conjunction of $\bigwedge_{s \in St, a \in Ac} r_{x,s,a} \geq 0$ and $\bigwedge_{s \in St} \sum_{a \in Ac} r_{x,s,a} = 1$.

Translating inequalities For every term τ introduce a variable r_τ, and let $ST(\tau)$ be all the arithmetic subterms of τ (including τ itself), i.e., the subterms treating $\mathbb{P}_\beta(\psi)$ as atomic.

The translation $\Phi_{\tau_1 \leq \tau_2, t}$ of $\tau_1 \leq \tau_2$ is

$$(\exists r_\tau)_{\tau \in ST(\tau_1) \cup ST(\tau_2)} \mathrm{EQN}_t(\tau_1) \wedge \mathrm{EQN}_t(\tau_2) \wedge r_{\tau_1} \leq r_{\tau_2}$$

where the construction of $\mathrm{EQN}_t(\tau_i)$ and the meaning of variables r_τ are given below. In particular, the formula $\mathrm{EQN}_t(\tau_i)$ has free variables $r_{x,s,a}$ (for all x, s, a) and r_τ (for $\tau \in ST(\tau_i)$), and satisfies the following property for every valuation ν and every evaluation χ compatible with it:

$$\mathfrak{R}, \chi \models \mathrm{EQN}_t(\tau_i) \text{ iff } \forall \tau \in ST(\tau_i), \chi(r_\tau) = val_{\nu,t}(\tau) \quad (2)$$

Observe that (2) implies that (1) holds in case $\varphi = \tau_1 \leq \tau_2$. Indeed, let ν be a valuation and χ a compatible evaluation. Suppose $G, \nu, t \models \tau_1 \leq \tau_2$. Then, by (2), $\mathrm{EQN}_t(\tau_1) \wedge \mathrm{EQN}_t(\tau_2) \wedge r_{\tau_1} \leq r_{\tau_2}$ holds under χ extended with the assignments $r_\tau \mapsto val_{\nu,t}(\tau)$, and thus $\mathfrak{R}, \chi \models \Phi_{\tau_1 \leq \tau_2, t}$. Conversely, suppose $\mathfrak{R}, \chi \models \Phi_{\tau_1 \leq \tau_2, t}$ and pick the r_τ's so that $\mathrm{EQN}_t(\tau_1) \wedge \mathrm{EQN}_t(\tau_2) \wedge r_{\tau_1} \leq r_{\tau_2}$ holds under χ. Thus, in particular, by (2), $r_{\tau_i} = val_{\nu,t}(\tau_i)$, and thus $r_{\tau_1} \leq r_{\tau_2}$ implies $G, \nu, t \models \tau_1 \leq \tau_2$.

Translating terms. Each subterm τ of φ is associated with a fresh variable r_τ, and the formula $\mathrm{EQN}_t(\tau)$ is defined inductively as follows:

- $\mathrm{EQN}_t(c) := r_c = c$,
- $\mathrm{EQN}_t(\tau^{-1}) := \mathrm{EQN}_t(\tau) \wedge r_{\tau^{-1}} = (r_\tau)^{-1}$, and
- $\mathrm{EQN}_t(\tau \oplus \tau') := \mathrm{EQN}_t(\tau) \wedge \mathrm{EQN}_t(\tau') \wedge r_{\tau \oplus \tau'} = r_\tau \oplus r_{\tau'}$ where $\oplus \in \{-, +, \times\}$.

It is not hard to see that (2) holds for these cases.

It remains to show how to construct $\mathrm{EQN}_t(\mathbb{P}_\beta(\psi))$. To do this, we use an automata-theoretic approach as follows.

As in the automata contruction for CTL* [Kupferman *et al.*, 2000] we view ψ as an LTL formula over atoms $\max(\psi)$, the set of maximal history-subformulas of ψ, and we construct a deterministic Rabin word automaton A_ψ (say with state set Q_ψ) that accepts exactly the set of words $w \in (2^{\max(\psi)})^\omega$ such that $w \models \psi$. Note, in particular, that this step translates the temporal operators.

We recall how to compute the probability that an LTL formula ψ holds in the Markov chain $G_{\nu \circ \beta, t}$ [Vardi, 1985; Courcoubetis and Yannakakis, 1995]. This amounts to solving a system of equations over the product Markov chain $G_{\nu \circ \beta, t} \times A_\psi$ with variables $r_{s,q}$ representing the probability that a path generated from s in the Markov chain is accepted by the automaton A_ψ starting in state q. Note, however, that in our case, the transition probabilities of the chain $G_{\nu \circ \beta, t}$ are not given explicitly (as numbers), but implicitly as variables $r_{x,s,a}$. It is not hard to see that these equations can be encoded in real-arithmetic (cf. [Baier *et al.*, 2012]), i.e., given a state t of G, β and A_ψ, one can define a formula ϕ^t_{β, A_ψ} over variables $r_{x,s,a}$ and the variable r_{t,q_ι} (here q_ι is the initial state of the automaton A_ψ), such that we have: for every valuation ν and every evaluation χ compatible with it,

$$\mathfrak{R}, \chi \models \phi^t_{\beta, A_\psi} \text{ iff } \chi(r_{t,q_\iota}) = val_{\nu,t}(\mathbb{P}_\beta(\psi)) \quad (3)$$

We have space for the briefest outline. The formula ϕ^t_{β, A_ψ} will have the form

$$(\exists r_{\varphi', s})^{s \in St}_{\varphi' \in \max(\psi)} (\mathrm{Max} \wedge (\exists r_{v,v'})_{v,v' \in St \times Q_\psi} (\mathrm{Prod} \wedge \mathrm{Rabin}))$$

where the fresh variables $r_{\varphi', s}$ are constrained by Max so that $r_{\varphi', s} > 0$ iff φ' holds in state s, and the fresh variables $r_{v,v'}$, for states v, v' of the product Markov chain $G_{\nu \circ \beta, t} \times A_\psi$, are constrained by Prod to be the transition probabilities in the product Markov chain $G_{\nu \circ \beta, t} \times A_\psi$, and variable r_{t,q_ι} is constrained by Rabin to be the probability that a random path in the product chain is accepted by the automaton A_ψ.

Finally, we can define $\mathrm{EQN}_t(\mathbb{P}_\beta(\psi))$ to be

$$\exists r_{t,q_\iota} \left(\phi^t_{\beta, A_\psi} \wedge r_{\mathbb{P}_\beta(\psi)} = r_{t,q_\iota} \right).$$

Observe that (3) implies that (2) holds in case $\tau_i = \mathbb{P}_\beta(\psi)$.

Computational complexity. Not surprisingly, the cost of translating $\mathbb{P}_\beta(\psi)$ dominates the other steps in the translation of a PSL formula. The size of $\mathbb{P}_\beta(\psi)$ is exponential in G, and double-exponential in ψ; and the number of quantifiers it uses is polynomial in G and double-exponential in ψ.

Thus, for a PSL formula φ, a worst-case blowup occurs if there is a linearly deep nesting of such probabilistic terms in φ. In this case, the size of $\Phi_{\varphi,s}$ is at most exponential in G and double-exponential in φ, and the number of quantifiers is polynomial in G and double-exponential in φ.

For vanilla PSL, each ψ has constant size, and so the size of $\Phi_{\varphi,s}$ is polynomial in φ, exponential in G and the nesting depth d of the probabilistic terms; and the number of quantifiers is polynomial in G and φ, and exponential in d.

Theorem 1 then follows from Theorem 2.

Extensions. PSL and the proof of the model-checking problem (for memoryless strategies) are highly extensible.

(1) Our algorithm immediately gives better complexity for other fragments of PSL, e.g., the fragment of PSL whose path formulas are $\psi ::= \varphi \mid \mathsf{X}\varphi \mid \varphi \,\mathsf{U}\, \varphi \mid \mathsf{G}\,\mathsf{F}\,\varphi$, is an extension of vanilla PSL that allows one to express some fairness properties. This has the same complexity as the vanilla fragment.

(2) We can add variables for deterministic strategies, and extend the translation of $\exists y.\varphi$ for such y by $(\exists r_{y,w,a})_{w \in St, a \in Ac} [Dist_y \wedge Point_y \wedge \Phi_{\varphi,\beta,s}]$ where $Point_y$ is the formula $\wedge_w \vee_a r_{y,w,a} = 1$. This allows one to express, e.g., the notion of pure-strategy ϵ-equilibrium.

(3) We can treat agents with partial observation, i.e., agent i has an indistinguishability relation $\equiv_i$ over states and the $\exists z.\varphi$ case is given by

$$(\exists r_{y,w,a})_{w \in St, a \in Ac} [Dist_y \wedge Unif_y \wedge \Phi_{\varphi,\beta,s}],$$

where $Unif_y$ is $\wedge_w \wedge_{v: v \equiv_i w} \wedge_a (r_{y,w,a} = r_{y,v,a})$. For comparison, strategy logic under imperfect information is studied in [Berthon *et al.*, 2017; Belardinelli *et al.*, 2017], and probabilistic extensions/variants of ATL* are studied in [Huang and Luo, 2013; Schnoor, 2013].

A lower bound. We have shown that one can model check, in EXPSPACE, the fragment of vanilla PSL in which the nesting depth d of probabilistic terms is bounded. As discussed in the introduction, we now show that there is a polynomial reduction from real-arithmetic to this fragment.

Theorem 3. *There is a polynomial reduction from model-checking $\mathfrak{R}$ to that of vanilla* PSL *restricted to a single agent, memoryless strategies, and a bounded nesting of probabilistic terms.*

Proof. We show how to translate a given real-arithmetic sentence φ in linear-time to a deterministic system G, a state s_0

of G, and a vanilla PSL formula $\widehat{\varphi}$ with no nesting of probabilistic terms, such that $\mathfrak{R} \models \varphi$ iff $G, s \models \widehat{\varphi}$.

Assume, wlog, that variable names in φ are not reused, and let X be the variables occuring in φ. Let $Var := X$, $AP := \{x_1, x_2 : x \in X\}$. Define the deterministic system $G = (Ac, St, Tr, Lab)$, in which $|Ag| = 1$, where

- $Ac := \{\bot\} \cup AP$ and $St := \{s_0\} \cup AP$;
- Tr is given by $s_0 \xrightarrow{x_i} x_i$, $s_0 \xrightarrow{\bot} s_0$, and $x_i \xrightarrow{a} x_i$, for every a, x, i;
- $Lab(s_0) = \emptyset, Lab(x_i) = \{x_i\}$.

Let β_x be the assignment of the strategy x to the agent. Construct $\widehat{\phi}$ from ϕ by: replacing every occurence of a term x by the term $enc(x) := (\mathbb{P}_{\beta_x}(\mathsf{X}\, x_1))^{-1} - (\mathbb{P}_{\beta_x}(\mathsf{X}\, x_2))^{-1}$; then replacing every inequality $\varphi := \tau_1 \leq \tau_2$ with the formula $\varphi \wedge \bigwedge_{x \in free(\varphi)} \bigwedge_{i=1,2} \mathbb{P}_{\beta_x}(\mathsf{X}\, x_i) \neq 0$.

For every subformula φ of ϕ, the corresponding translated subformula $\widehat{\varphi}$ has the following property: for every evaluation χ there exists a valuation ν such that $\mathfrak{R}, \chi \models \varphi$ iff $G, \nu, start \models \widehat{\varphi}$. The proof is a straightforward induction that uses the fact that every real number r can be written as $(r_1)^{-1} - (r_2)^{-1}$ for $r_1, r_2 \in (0, \frac{1}{2})$. Then, a strategy that, for example, assigns the probability r_i to the action x_i ($i = 1, 2$) and the probability $1 - (r_1)^{-1} - (r_2)^{-1}$ to the action $\bot$, has the property that $enc(x) = r$. □

4 Related Work

Probabilistic strategic logics. Early probabilistic extensions of ATL are PATL/PATL* [Chen and Lu, 2007]. They use operators of the form $\langle\langle A \rangle\rangle Pr(\psi) \bowtie c$ read "agents in A have strategies such that for all strategies of agents not in A, the probability that the path formula ψ holds is $\bowtie c$", where $\bowtie \in \{<, =, >\}$ and $c \in [0, 1]$ is rational. Just as SL subsumes ATL/ATL*, so PSL subsumes PATL/PATL*. To see this, note that $\langle\langle A \rangle\rangle Pr(\psi) \bowtie c$ is equivalent to the PSL formula (allowing shorthands) $(\exists x_i)_{i \in A} (\forall x_j)_{j \notin A} Pr_\beta(\psi) \bowtie c$. Another variant of probabilistic ATL, ATL with probabilistic success, is studied in [Bulling and Jamroga, 2009].

The logics rPATL/rPATL* [Chen *et al.*, 2013a; Kwiatkowska *et al.*, 2018] extend PATL/PATL* with operators that can enforce an expected *reward* $\bowtie c$. The main differences with PSL are: (i) rPATL* can express cumulative rewards given by the transition system, while PSL can only express rewards for path formulas holding; (ii) unlike PSL, rPATL* is not designed to express solution concepts. Model-checking rPATL* is 2EXPTIME-complete.

Closely related works. rPATL+NE [Kwiatkowska *et al.*, 2019] extends rPATL with an equilibrium operator expressing that there exists a subgame perfect Nash equilibrium between the coalitions A and $Ag \setminus A$ under which the sum of the (probability or reward) objectives for the coalitions is $\bowtie c$. The main differences with PSL are: (i) rPATL+NE (like rPATL before it) can express more complex rewards than PSL, (ii) PSL can express other complex concepts from game-theory.

SGL (Stochastic Game Logic) [Baier *et al.*, 2012] is like ATL with strategy contexts, and uses a probabilistic operator $P_{\bowtie c}(D, \phi_1, \cdots, \phi_k)$, where D is a Rabin word automaton over the alphabet $[k]$, and each ϕ_i is an SGL formula, which expresses that the probability is $\bowtie c$ that a path π, when viewed as the sequence whose jth element consist of the indices $i \in [k]$ such that ϕ_i holds in the suffix $\pi_{\geq i}$, is accepted by D. Model-checking SGL is undecidable in general; and restricted to memoryless strategies it is decidable via a translation to real-arithmetic (an analysis of their algorithm yields a 2EXPSPACE upper bound). Our proof of Theorem 1 is based on this translation, and differs from it in two ways: we compile path formulas into automata over truth values of subformulas; and we translate inequalities between arithmetic terms. The main differences with PSL are: (i) there are no terms in SGL that *compare probabilities of path formulas*, and thus, one cannot, in general, express any of the examples of PSL formulas in Section 2.2, including Nash equilibrium (intuitively, the common fragment of SGL and PSL is the fragment of PSL in which the terms $\tau \leq \tau$ are replaced by $\mathbb{P}_\beta(\psi) \leq c$); (ii) SGL uses automata instead of LTL to represent path formulas; if we had used automata instead of LTL our model-checking algorithm would work in 2EXPSPACE (instead of 3EXPSPACE), but we chose not to do so because automata are not as readable as formulas.

Overall, PSL stands out for two reasons: (i) it is not subsumed by any of the logics mentioned (and it subsumes aspects of each of them), and (ii) its syntax is, we believe, simpler and more natural than those of the logics mentioned.

5 Conclusion

Strategy Logic (SL) can elegantly express important solution concepts in game-theory, e.g., Nash Equilibria, dominant strategies, subgame-perfect equilibria [Mogavero *et al.*, 2014]. Although extended in a number of useful ways, e.g., imperfect information [Berthon *et al.*, 2017; Belardinelli *et al.*, 2017] and counting strategies [Aminof *et al.*, 2018; Malvone *et al.*, 2018], until now it has only been investigated in the restricted setting of pure strategies and determinisitic systems. This is unfortunate, as game-theory and MAS often involve randomised strategies and stochastic systems. We fill this gap by introducing PSL, a probabilistic extension of SL.

The importance of PSL is further highlighted by a combination of three facts, as we have shown: it has a natural syntax which mimics first-order logic; it is expressive; and it is highly extensible. Thus, PSL takes an important step to bring us closer to the ideal of having a natural, expressive, and decidable logic for strategic reasoning in multi-agent systems.

There are a number of non-trivial open problems regarding the exact complexity of model-checking PSL and its fragments. Also, the decidability of model-checking the *qualitative* fragment of PSL (with no restriction on the strategies) is open, i.e., the qualitative fragment restricts terms τ to be of the form $0 \mid 1 \mid \mathbb{P}_\beta(\psi)$, and thus one can express, e.g., $\mathbb{P}_\beta(\psi) = 1$, but not $\mathbb{P}_\beta(\psi) = 0.5$.

Acknowledgements

Benjamin Aminof acknowledges the Austrian Science Fund (FWF) P 32021, and Marta Kwiatkowska acknowledges partial support from the EPSRC Mobile Autonomy Programme Grant EP/M019918/1.

References

[Alur *et al.*, 2002] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J.ACM*,49(5):672–713, 2002.

[Aminof *et al.*, 2018] B. Aminof, V. Malvone, A. Murano, and S. Rubin. Graded modalities in strategy logic. *Inf. Comput.*, 261:634–649, 2018.

[Baier *et al.*, 2012] C. Baier, T. Brázdil, M. Größer, and A. Kucera. Stochastic game logic. *Acta Inf.*, 49(4):203–224, 2012.

[Belardinelli *et al.*, 2017] F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin. Verification of broadcasting multi-agent systems against an epistemic strategy logic. In *IJCAI*, 2017.

[Ben-Or *et al.*, 1986] M. Ben-Or, D. Kozen, and J. H. Reif. The complexity of elementary algebra and geometry. *JCSS*, 32(2):251–264, 1986.

[Berman, 1980] L. Berman. The complexity of logical theories. *Theor. Comput. Sci.*, 11(1):71 – 77, 1980.

[Berthon *et al.*, 2017] R. Berthon, B. Maubert, A. Murano, S. Rubin, and M.Y. Vardi. Strategy logic with imperfect information. In *LICS*, 2017.

[Bouyer *et al.*, 2016] P. Bouyer, P. Gardy, and N. Markey. On the semantics of strategy logic. *IPL*, 116(2):75–79, 2016.

[Brázdil *et al.*, 2006] T. Brázdil, V. Brozek, V. Forejt, and A. Kucera. Stochastic games with branching-time winning objectives. In *LICS*, 2006.

[Bulling and Jamroga, 2009] N. Bulling and W. Jamroga. What agents can probably enforce. *Fundam. Inform.*, 93(1-3):81–96, 2009.

[Cermák *et al.*, 2018] P. Cermák, A. Lomuscio, F. Mogavero, and A. Murano. Practical verification of multi-agent systems against SLK specifications. *Inf. Comput.*, 261:588–614, 2018.

[Chatterjee and Henzinger, 2012] K. Chatterjee and T.A. Henzinger. A survey of stochastic ω-regular games. *JCSS*, 78(2):394–413, 2012.

[Chatterjee *et al.*, 2010] K. Chatterjee, T.A. Henzinger, and N. Piterman. Strategy Logic. *Inf. Comput.*, 208(6):677–693, 2010.

[Chen and Lu, 2007] T. Chen and J. Lu. Probabilistic alternating-time temporal logic and model checking algorithm. In *FSKD*, 2007.

[Chen *et al.*, 2013a] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis. Automatic verification of competitive stochastic systems. *FMSD*, 43(1):61–92, 2013.

[Chen *et al.*, 2013b] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis. PRISM-games: A model checker for stochastic multi-player games. In *TACAS*, 2013.

[Courcoubetis and Yannakakis, 1995] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.

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

[Fisman *et al.*, 2010] D. Fisman, O. Kupferman, and Y. Lustig. Rational synthesis. In *TACAS*, 2010.

[Halpern, 2008] J. Y. Halpern. Beyond nash equilibrium: Solution concepts for the 21st century. In *KR*, 2008.

[Huang and Luo, 2013] Xiaowei Huang and Cheng Luo. A logic of probabilistic knowledge and strategy. In *AAMAS*, pages 845–852, 2013.

[Kemeny and Snell, 1976] J. G. Kemeny and J. L. Snell. *Finite markov chains*. Springer, 1976.

[Kupferman *et al.*, 2000] O. Kupferman, M.Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *J. ACM*, 47(2):312–360, 2000.

[Kupferman *et al.*, 2016] O. Kupferman, G. Perelli, and M.Y. Vardi. Synthesis with rational environments. *Ann. Math. Artif. Intell.*, 78(1):3–20, 2016.

[Kwiatkowska *et al.*, 2012] M. Kwiatkowska, G. Norman, and D. Parker. The PRISM benchmark suite. In *QEST*, 2012.

[Kwiatkowska *et al.*, 2018] M. Kwiatkowska, G. Norman, D. Parker, and G. Santos. Automated verification of concurrent stochastic games. In *QEST*, 2018.

[Kwiatkowska *et al.*, 2019] M. Kwiatkowska, G. Norman, D. Parker, and G. Santos. Equilibria-based probabilistic model checking for concurrent stochastic games. *CoRR*, abs/1811.07145, 2019.

[Laroussinie and Markey, 2015] F. Laroussinie and N. Markey. Augmenting ATL with strategy contexts. *Inf. Comput.*, 245:98–123, 2015.

[Malvone *et al.*, 2018] V. Malvone, F. Mogavero, A. Murano, and L. Sorrentino. Reasoning about graded strategy quantifiers. *Inf. Comput.*, 259(3):390–411, 2018.

[Mogavero *et al.*, 2014] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. Reasoning about strategies: On the model-checking problem. *ACM Trans. Comput. Log.*, 15(4):34:1–34:47, 2014.

[Pnueli, 1977] A. Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57, 1977.

[Russell and Norvig, 1995] S.J. Russell and P. Norvig. *Artificial Intelligence. A Modern Approach*. Prentice, 1995.

[Schnoor, 2013] H. Schnoor. Epistemic and probabilistic ATL with quantification and explicit strategies. In *ICAART*, volume 449, pages 131–148, 2013.

[Svorenová and Kwiatkowska, 2016] M. Svorenová and M. Kwiatkowska. Quantitative verification and strategy synthesis for stochastic games. *EJC*, 30:15–30, 2016.

[Ummels and Wojtczak, 2011] M. Ummels and D. Wojtczak. The complexity of nash equilibria in stochastic multiplayer games. *LMCS*, 7(3), 2011.

[Vardi, 1985] M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS*, 1985.