Discounting in Strategy Logic

Munyque Mittelmann[1] , **Aniello Murano**[1] and **Laurent Perrussel**[2]

[1]University of Naples "Federico II"

[2]University of Toulouse - IRIT

{munyque.mittelmann, aniello.murano}@unina.it, laurent.perrussel@irit.fr

Abstract

Discounting is an important dimension in multi-agent systems as long as we want to reason about strategies and time. It is a key aspect in economics as it captures the intuition that the far-away future is not as important as the near future. Traditional verification techniques allow to check whether there is a winning strategy for a group of agents but they do not take into account the fact that satisfying a goal sooner is different from satisfying it after a long wait. In this paper, we augment Strategy Logic with *future discounting* over a set of discounted functions $\mathcal{D}$, denoted $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$. We consider "until" operators with discounting functions: the satisfaction value of a specification in $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ is a value in $[0, 1]$, where the longer it takes to fulfill requirements, the smaller the satisfaction value is. We motivate our approach with classical examples from Game Theory and study the complexity of model-checking $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$-formulas.

1 Introduction

The goal of this paper is to advance the research on strategic reasoning and formal verification by considering a discounting effect: the utility of agents decreases over time. Boolean state-transition models have been widely used to define the semantics of temporal and strategic logics, including Linear Temporal Logic (LTL) [Pnueli, 1977], Alternating-time Temporal Logic (ATL) [Alur *et al.*, 2002], Strategy Logic (SL) [Mogavero *et al.*, 2014; Chatterjee *et al.*, 2010]. In conjunction with model checking techniques [Clarke *et al.*, 2018], these formal frameworks are useful for the representation and verification of hardware and software systems. Given a strategic logic specification, the correctness of a system is a yes/no matter: either the system satisfies the specification or it does not. Complex systems that interact with a physical environment or that are composed of multiple autonomous agents may have quantitative aspects described by real numbers (e.g. utilities, time and costs). Evaluating the *quality* of such systems through the Boolean satisfaction of the specifications is often inadequate. Different levels of quality may exist, and this should be reflected in the output of the verification procedure [Almagor *et al.*, 2014].

In this work, we are interested in verifying Multi-Agent Systems (MAS) whose quality assessment needs to take into account that satisfying the goal sooner is different from satisfying it after a long wait. To illustrate this setting, consider an agent whose task is to organize a trip and who is facing the problem of booking a flight. An early booking is more susceptible to becoming unfeasible in the case of unforeseen changes in the travel plans. On the other hand, waiting to book may result in more important costs for the agent. Moreover, the trip-organizing agent may be a part of a system composed of other, self-interested, agents. In this case, the agents' interactions can also influence their ability to find reasonable flight options and price tags. On one side, there is a competitive aspect when agents dispute the last available tickets. Cooperation could also take place as some companies offer discounts for group booking. To address this problem for (single-agent) systems, researchers have suggested to augment Linear Temporal Logic with future discounting [De Alfaro *et al.*, 2005; Almagor *et al.*, 2014]. In the discounted setting, the satisfaction value of specifications is a numerical value, and it depends, according to some discounting function, on the time waited for eventualities to get satisfied.

Discounting is a key dimension in Economics and has been studied in Markov decision processes [Filar and Vrieze, 1996] as well as game theory [Shapley, 1953] and system theory [De Alfaro *et al.*, 2003] to capture the intuition that the far-away future is not as important as the near future. The multi-agent setting has also been widely investigated, including repeated games [Abreu, 1988; Fudenberg and Maskin, 2009; Pęski, 2014], the prisoner's dilemma game [Harris and Madden, 2002; Locey *et al.*, 2013], and negotiation protocols [Weg *et al.*, 1990; Fatima *et al.*, 2006], to name a few. Previous work [Jamroga, 2008b; Chen *et al.*, 2013] have initiated to study logics inspired on ATL and Markov chains for reasoning about discounting in stochastic MAS. Likewise ATL, these logics are unable to capture complex solution concepts in MAS (such as Nash equilibria), which are important when evaluating the possible outcomes of such systems.

Contribution In this work, we augment Strategy Logic with future discounting, denoted $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$, and study its complexity for model-checking. The main advantage of this logic is that it allows us to express and verify (i) the strategic abilities of agents to achieve certain goals while considering temporal discounts, and (ii) complex strategy concepts such

as Nash equilibrium of discounted games. Different from previous work, we focus on deterministic games and consider temporal discounting alongside a logic that quantifies over strategies. This enables an unbounded number of alternations from strategic operators which is necessary to capture complex solution concepts. In relation to technical results, we also studied the complexity of the model-checking problem under memoryless and perfect recall strategies, which was not established in [Jamroga, 2008b].

$\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ represents a family of logics, each one parameterized by a set of discounting functions. Considering a set of functions allows us to model games in which each agent, or a coalition of them, is affected differently by how long in the future events occur (e.g., patient vs hurried agents). We also provide complexity results for model-checking and motivate the approach with classical examples from Game Theory.

This is the first work to consider a Strategy Logic with discounting for strategic reasoning in MAS. We aim at paving the way for a new line of research that applies the formal techniques developed for verification and reasoning in MAS to game-theoretic problems involving future discounts.

Outline The paper is organized as follows: we start by discussing related work in Section 2. Then, we define Strategy Logic with future discounts, denoted $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ (Section 3). We proceed by introducing problems and concepts on using discounting in multi-agent games and illustrate the use of $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ (Section 4). Next, we study the complexity results for model checking (Section 5). Finally, we conclude the paper and point directions for future work (Section 6).

2 Related work

Weighted games have been studied in the literature in relation to various kinds of objectives, including parity [Emerson and Jutla, 1991], mean-payoff [Ehrenfeucht and Mycielski, 1979; Zwick and Paterson, 1996], energy [Chakrabarti et al., 2003; Bouyer et al., 2008], and combining qualitative and quantitative objectives in equilibrium [Gutierrez et al., 2021]. $\mathsf{SL}[\mathcal{F}]$ [Bouyer et al., 2019; Bouyer et al., 2023] was recently introduced as a quantitative extension of SL defined over weighted concurrent game structures. It extends $\mathsf{LTL}[\mathcal{F}]$ [Almagor et al., 2016], a multi-valued logic that augments LTL with quality operators. $\mathsf{SL}[\mathcal{F}]$ subsumes both SL and $\mathsf{LTL}[\mathcal{F}]$ and is expressive enough to express complex solution concepts such as Nash equilibrium and properties about quantities. An extension of $\mathsf{SL}[\mathcal{F}]$ with imperfect information and epistemic operators was recently proposed [Maubert et al., 2021]. Other quantitative extensions of LTL have been explored in the context of averaging [Bouyer et al., 2014], discounting [Almagor et al., 2014; Mandrali, 2012], and mean-payoff objectives [Bohy et al., 2013]. Quantitative extensions of ATL have also been investigated, such as timed ATL [Henzinger and Prabhu, 2006; Brihaye et al., 2007], multi-valued ATL [Jamroga et al., 2020], ATL with resource bounds [Alechina et al., 2017; Alechina et al., 2018], and weighted versions of ATL [Laroussinie et al., 2006; Bulling and Goranko, 2022; Vester, 2015]. Another related problem is prompt requirements (see, for instance, [Aminof et al., 2016; Fijalkow et al., 2020]), which consider a bound on the number of steps to satisfy the specification.

To encode the notion that the importance of events should be discounted according to how late they occur, De Alfaro et al. (2005) proposed an extension of the Computational Tree Logic with quantitative semantics. In this logic, path operators are discounted by a parameter that can be chosen to give more weight to states that are closer to the beginning of the path. Later, Almagor et al. 2014 proposed LTL augmented with an arbitrary set of discounting functions, denoted $\mathsf{LTL}^{\mathrm{disc}}[\mathcal{D}]$; and further explored with unary propositional quality operators and average-operator.

In the context of stochastic systems, Jamroga (2008a) proposed the Markov Temporal Logic, which extends the Branching-Time Temporal Logic and captures discounted goals. Later, this approach was extended to the multi-agent setting [Jamroga, 2008b]. Finally, Chen et al. (2013) considered a probabilistic extension of ATL, alongside discounted rewards.

3 Strategy Logic with Discounting

Strategy Logic with Discount ($\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$) generalizes SL by adding discounting temporal operators. The logic is actually a family of logics, each parameterized by a set $\mathcal{D}$ of discounting functions. A function $d : \mathbb{N} \to [0, 1]$ is a *discounting function* if $\lim_{i \to \infty} d(i) = 0$, and d is non-increasing. Examples of discounting functions include $d(i) = \lambda^i$, for some $\lambda \in (0, 1)$, and $d(i) = \frac{1}{i+1}$.

For the remainder of the paper, we fix a set of discounting functions $\mathcal{D}$, a set of atomic propositions AP, a set of agents Ag, and a set of strategy variables Var, except when stated otherwise. We let n be the number of agents in Ag.

The syntax of $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ adds to SL the operator $\varphi \mathbf{U}_d \psi$ (discounting-Until), for every function $d \in \mathcal{D}$. The logic is defined $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ as follows:

Definition 1. The syntax of $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ is defined by the grammar

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists s. \varphi \mid (a, s)\varphi \mid \mathbf{X}\varphi \mid \varphi \mathbf{U}\varphi \mid \varphi \mathbf{U}_d \varphi$$

where $p \in \mathrm{AP}$, $s \in \mathrm{Var}$, $a \in \mathrm{Ag}$, and $d \in \mathcal{D}$.

The intuitive reading of the operators is as follows: $\exists s. \varphi$ means that there exists a strategy such that φ holds; $(a, s)\varphi$ means that when strategy s is assigned (or "bound") to agent a, φ holds; $\mathbf{X}$ and $\mathbf{U}$ are the usual temporal operators "next" and "until". The intuition of the operator $\mathbf{U}_d$ is that events that happen in the future have a lower influence, and the rate by which this influence decreases depends on the function d.

A variable is *free* in a formula φ if it is bound to an agent without being quantified upon, and an agent a is free in φ if φ contains a temporal operator ($\mathbf{X}$, $\mathbf{U}$, $\mathbf{U}_d$) not in the scope of any binding for a. The set of free variables and agents in φ is written free(φ), and a formula φ is a *sentence* if free(φ) = $\emptyset$.

A state-transition model is a labeled directed graph, in which the vertices represent the system states, the edges the state changes (e.g., according to environment or agents' actions), and the labels the Boolean characteristics of the state

(i.e., the truth values of state atomic propositions). In this paper, we consider state-transition models in which there are multiple agents that act simultaneously and independently. These models are called concurrent game structures (CGS).

Definition 2. A *concurrent game structure* (CGS) is a tuple $\mathcal{G} = (\text{Ac}, V, v_\iota, \delta, \ell)$ where (i) Ac is a finite set of *actions*; (ii) V is a finite set of *positions*; (iii) $v_\iota \in V$ is an *initial position*; (iv) $\delta : V \times \text{Ac}^{\text{Ag}} \to V$ is a *transition function*; (v) $\ell : V \to 2^{\text{AP}}$ is a *labeling function*.

In a position $v \in V$, each player a chooses an action $c_a \in \text{Ac}$, and the game proceeds to position $\delta(v, \boldsymbol{c})$ where $\boldsymbol{c} \in \text{Ac}^{\text{Ag}}$ is an *action profile* $(c_a)_{a \in \text{Ag}}$.

We write $\boldsymbol{o}$ for a tuple of objects $(o_a)_{a \in \text{Ag}}$, one for each agent, and such tuples are called *profiles*. Given a profile $\boldsymbol{o}$ and $a \in \text{Ag}$, we let o_a be agent a's component, and $\boldsymbol{o}_{-a}$ is $(o_b)_{b \neq a}$. Similarly, we let $\text{Ag}_{-a} = \text{Ag} \setminus \{a\}$. For a group of n agents $A = \{a_1, ..., a_n\}$ and strategy profile $\sigma = \sigma_1, ..., \sigma_n$ we write (A, σ) as a shortcut for $(a_1, \sigma_1)...(a_n, \sigma_n)$.

A *play* $\pi = v_0 v_1 ...$ in $\mathcal{G}$ is an infinite sequence of positions such that $v_0 = v_\iota$ and for every $i \geq 0$ there exists an action profile $\boldsymbol{c}$ such that $\delta(v_i, \boldsymbol{c}) = v_{i+1}$. We write $\pi_i = v_i$ for the position at index i in play π. A *history* h is a finite prefix of a play, $\text{last}(h)$ is the last position of history h, $|h|$ is the length of h and Hist is the set of histories.

A (perfect recall) *strategy* is a function $\sigma : \text{Hist} \to \text{Ac}$ that maps each history to an action. A (memoryless) *strategy* is a function $\sigma : V \to \text{Ac}$ that maps each position to an action. We let Str^{R} (similarly Str^{r}) be the set of perfect recall strategies (resp. memoryless strategies). For the remainder of the paper, we use r and R to denote memoryless and perfect recall, respectively, and we let $\rho = \{\text{r}, \text{R}\}$.

An *assignment* $\chi : \text{Ag} \cup \text{Var} \to Str$ is a function from players and variables to strategies. For an assignment χ, an agent a and a strategy σ for a, $\chi[a \mapsto \sigma]$ is the assignment that maps a to σ and is otherwise equal to χ, and $\chi[s \mapsto \sigma]$ is defined similarly, where s is a variable.

For an assignment χ and a state v, we let $\text{Out}(\chi, v)$ be the unique play that continues v following the strategies assigned by χ. Formally, $\text{Out}(\chi, v)$ is the play $v v_0 v_1 ...$ such that for all $i \geq 0$, $v_i = \delta(v_{i-1}, \boldsymbol{c})$ where for all $a \in \text{Ag}$, $\boldsymbol{c}_a = \chi(a)(v v_1 ... v_{i-1})$.

Definition 3. Let $\mathcal{G} = (\text{Ac}, V, v_\iota, \delta, \ell)$ be a CGS, χ be an assignment, and $\rho \in \{\text{R}, \text{r}\}$. The satisfaction value $[\![\varphi]\!]_\chi^{\mathcal{G}, \rho}(v) \in [0, 1]$ of an $\text{SL}^{\text{disc}}[\mathcal{D}]$ formula φ in a state v is defined as follows, where π denotes $\text{Out}(\chi, v)$:

$$[\![p]\!]_\chi^{\mathcal{G}, \rho}(v) = \begin{cases} 1 & \text{if } p \in \ell(v) \\ 0 & \text{otherwise} \end{cases}$$

$$[\![\exists s. \varphi]\!]_\chi^{\mathcal{G}, \rho}(v) = \max_{\sigma \in Str} [\![\varphi]\!]_{\chi[s \mapsto \sigma]}^{\mathcal{G}, \rho}(v)$$

$$[\![(a, s)\varphi]\!]_\chi^{\mathcal{G}, \rho}(v) = [\![\varphi]\!]_{\chi[a \mapsto \chi(s)]}^{\mathcal{G}, \rho}(v)$$

$$[\![\varphi_1 \vee \varphi_2]\!]_\chi^{\mathcal{G}, \rho}(v) = \max([\![\varphi_1]\!]_\chi^{\mathcal{G}, \rho}(v), [\![\varphi_2]\!]_\chi^{\mathcal{G}, \rho}(v))$$

$$[\![\neg\varphi]\!]_\chi^{\mathcal{G}, \rho}(v) = 1 - [\![\varphi]\!]_\chi^{\mathcal{G}, \rho}(v)$$

$$[\![\mathbf{X}\varphi]\!]_\chi^{\mathcal{G}, \rho}(v) = [\![\varphi]\!]_\chi^{\mathcal{G}, \rho}(\pi_1)$$

$$[\![\varphi_1 \mathbf{U} \varphi_2]\!]_\chi^{\mathcal{G}, \rho}(v) = \sup_{i \geq 0} \min\big([\![\varphi_2]\!]_\chi^{\mathcal{G}, \rho}(\pi_i),$$
$$\min_{0 \leq j < i} [\![\varphi_1]\!]_\chi^{\mathcal{G}, \rho}(\pi_j)\big)$$

$$[\![\varphi_1 \mathbf{U}_d \varphi_2]\!]_\chi^{\mathcal{G}, \rho}(v) = \sup_{i \geq 0} \min\big(d(i)[\![\varphi_2]\!]_\chi^{\mathcal{G}, \rho}(\pi_i),$$
$$\min_{0 \leq j < i} d(j)[\![\varphi_1]\!]_\chi^{\mathcal{G}, \rho}(\pi_j)\big)$$

If φ is a sentence, its satisfaction value does not depend on the assignment, and we write $[\![\varphi]\!]^{\mathcal{G}, \rho}(v)$ for $[\![\varphi]\!]_\chi^{\mathcal{G}, \rho}(v)$ where χ is any assignment. We also let $[\![\varphi]\!]^{\mathcal{G}, \rho} = [\![\varphi]\!]^{\mathcal{G}, \rho}(v_\iota)$.

Classical abbreviations are defined as follows: $\bot := \neg\top$, $\varphi \wedge \varphi' := \neg(\neg\varphi \vee \neg\varphi')$, $\varphi \to \varphi' := \neg\varphi \vee \varphi'$, $\mathbf{F}\psi := \top \mathbf{U} \psi$, $\mathbf{G}\psi := \neg\mathbf{F}\neg\psi$ and $\forall s. \varphi := \neg\exists s. \neg\varphi$. The quantitative counterparts of $\mathbf{G}$ and $\mathbf{F}$, denoted $\mathbf{G}_d$ and $\mathbf{F}_d$, are defined analogously.

Relation with $\text{LTL}^{\text{disc}}[\mathcal{D}]$, SL and $\text{SL}[\mathcal{F}]$: $\text{LTL}^{\text{disc}}[\mathcal{D}]$ [Almagor *et al.*, 2014] is the fragment of $\text{SL}^{\text{disc}}[\mathcal{D}]$ without strategy quantification and bindings. Considering that the satisfactions values 1 and 0 represent *true* and *false* (resp.), SL [Mogavero *et al.*, 2014] is a syntactical restriction of $\text{SL}^{\text{disc}}[\mathcal{D}]$ (without the discounted-Until). SL cannot express that the value of a formula decays over time. To notice the difference, assume a CGS $\mathcal{G}$, an assignment χ and states v, v' such that and $\text{Out}(\chi, v') = \pi_0 \pi$ and $\pi = \text{Out}(\chi, v)$, that is, the outcome from v' is the outcome from v with the first state repeated. Assuming that $p \in \ell(\pi_i)$ and $d(i) \neq d(i-1)$, for some $i \geq 1$ and $p \in \text{AP}$, we have that $[\![\mathbf{F}_d p]\!]_\chi^{\mathcal{G}, \rho}(v) \neq [\![\mathbf{F}_d p]\!]_\chi^{\mathcal{G}, \rho}(\pi')$. However, using the classical until, we have that $[\![\mathbf{F} p]\!]_\chi^{\mathcal{G}, \rho}(v) = [\![\mathbf{F} p]\!]_\chi^{\mathcal{G}, \rho}(\pi')$. As for $\text{SL}[\mathcal{F}]$ [Bouyer *et al.*, 2019], notice it is interpreted over different classes of models from $\text{SL}^{\text{disc}}[\mathcal{D}]$, namely weighted CGS, which uses weight functions for atomic propositions in place of propositional labeling of states. $\text{SL}[\mathcal{F}]$ is defined over a set of functions $\mathcal{F}$ over $[0, 1]$, but its semantics does not enable to use these functions to capture the effect of future discounts. This is because functions are applied over the satisfaction value of formulas in a given state, independent from *how far* in the play they are being evaluated w.r.t. the initial state.

4 Discounting in multi-agent games

We now introduce problems and concepts from Game Theory that motivated reasoning about discounts in MAS.

4.1 Nash equilibrium for $\text{SL}^{\text{disc}}[\mathcal{D}]$ goals

Nash equilibrium (NE) is a central solution concept in game theory that captures the notion of a stable solution, that is a solution from which no single player can individually improve his or her welfare by deviating [Nisan *et al.*, 2007]. Deterministic concurrent multi-player Nash equilibrium can be expressed using SL (or its extensions) for Boolean valued goals [Mogavero *et al.*, 2014] and quantitative goals [Bouyer *et al.*, 2019]. With $\text{SL}^{\text{disc}}[\mathcal{D}]$, we can express that agent's goals are affected by how long in the future they are achieved.

Let the $\text{LTL}^{\text{disc}}[\mathcal{D}]$-formula ψ_a (i.e., an $\text{SL}^{\text{disc}}[\mathcal{D}]$ formula without bindings and strategy quantification) denote the goal

of agent a. We can express whether a strategy profile $\boldsymbol{\sigma} = (\sigma_a)_{a \in \text{Ag}}$ is a *Nash equilibrium* through the $\mathsf{SL}^{\text{disc}}[\mathcal{D}]$ formula

$$\varphi_{\text{NE}}(\boldsymbol{\sigma}) := (\text{Ag}, \boldsymbol{\sigma}) \bigwedge_{a \in \text{Ag}} \left(\forall t. (a, t)\psi_a\right) \rightarrow \psi_a$$

The existence of a Nash equilibrium is captured by the formula $\hat{\varphi}_{\text{NE}} := \exists \boldsymbol{\sigma}(\varphi_{\text{NE}}(\boldsymbol{\sigma}))$. This is a classical problem in game theory and, more precisely when studying games with future discounting [Fudenberg and Maskin, 1990].

As we shall see in the next sections, the goal ψ_a of an agent a may involve temporal discounts. In the booking agent example, for instance, the discounted goal

$$\psi_a := \text{priceunder}_\vartheta \mathbf{U}_d \text{booked}_a$$

specifies that the flight ticket is affordable (that is, below a threshold ϑ) until agent a booked her ticket. The value obtained from achieving the goal later is reduced according to the discounted function d.

4.2 Secretary problem

The classical secretary problem studies the problem of an agent selecting online an element (called a "secretary") the *maximum value* from a known number of candidates to be presented one by one in random order. As each item is presented she must either accept it, in which case the game ends, or reject it. In the second case, the next item in the sequence is presented and the agent faces the same choice as before [Freeman, 1983]. Applications of this problem include agents' facing the decision of buying a house or hiring employees. Several variants and extensions of the secretary problem are considered in the literature, including using time-dependent discount factors to reduce the benefit derived from selecting a secretary at a later time [Babaioff *et al.*, 2009]. The discounted setting captures the cost of rejecting elements. For instance, when seeking to purchase a house, an agent may prefer to chose a suboptimal house at the beginning of the game than wait longer to pick her most desirable house. Recently, Do *et al.* (2022) investigated the selection of k secretaries by a multi-agent selection committee. The hiring decision is made by a group of voting agents that specify whether they consider acceptable to hire the current candidate or not.

With CGS, we can represent deterministic perfect information instances of the secretary problem. Let us consider the selection of k secretaries by multiple voting agents. For each candidate j from a finite set of candidates C, we let the atomic propositions present_j denote whether she was presented and hired_j denote whether she was hired. Proposition k-hired specifies whether k secretaries were already selected[1].

The $\mathsf{SL}^{\text{disc}}[\mathcal{D}]$ formula $\mathbf{F}_d\, k\text{-hired}$ represents the goal of having enough candidates hired in the future. The satisfaction value of this goal decreases according to d, denoting that it is preferable to hire k candidates as soon as possible. The discounted goal

$$\exists s \forall \boldsymbol{t}(a, s)(\text{Ag}_{-a}, \boldsymbol{t})(\bigvee_{j \in C} \neg\text{present}_j)\mathbf{U}_d\, k\text{-hired}$$

---

[1] The formalization of the game as a CGS is left to the reader. Examples on how to model similar problems using CGS can be found in [Mittelmann and Perrussel, 2020; Maubert *et al.*, 2021].
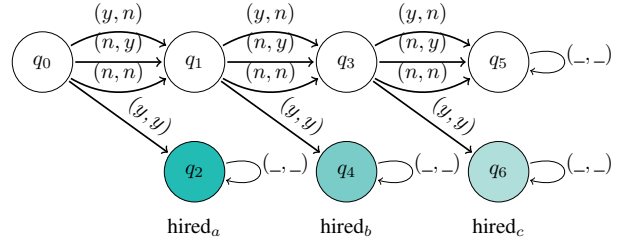

Figure 1: $\mathcal{G}_{sec}$ representing the secretary problem with three candidates (a, b and c) and two voters (Ann and Bob). In state q_0 (similarly, q_1 and q_3), Ann and Bob vote on whether to hire candidate a (resp. b and c). States q_2, q_4, and q_6 represent the situation in which candidate a, b and c were hired, respectively.

represents that the voter a has a strategy to ensure that, no matter the strategies of the other agents, there are candidates still not presented until enough secretaries were hired.

In Figure 1, we exemplify the CGS $\mathcal{G}_{sec}$ representing an instance of the secretary problem with two voting agents, Ann and Bob, and three candidates, a, b, and c. In the initial state (q_0), the agents vote on whether they want to hire candidate a by performing the action y or n. Candidate a is hired only if both agents play y, in which case the game moves to state q_2. Otherwise, the game proceeds to state q_1 in which they can vote for candidate b (and similarly, for candidate c in state q_3. The game ends when one secretary is hired (states q_2, q_4, and q_6) or all candidates have been presented (state q_5).

We let the following $\mathsf{SL}^{\text{disc}}[\mathcal{D}]$ formulas denote agent a's and agent b's goals, resp.:

$$\psi_{\text{Ann}} := \mathbf{F}\,\text{hired}_b \vee \mathbf{F}_{d_{\text{Ann}}}\,1\text{-hired}$$

$$\psi_{\text{Bob}} := \mathbf{F}_{d_{\text{Bob}}}\,1\text{-hired}$$

and we assume the discount functions $d_{\text{Ann}}(i) = \frac{1}{i+1}$ and $d_{\text{Bob}}(i) = (\frac{1}{2})^i$. In other words, Ann's goal is to hire candidate b in the future or to hire any candidate (with a discount according to d_{Ann}), while Bob's goal is to hire a candidate in the future (with a discount given by d_{Bob}). Notice that without the discount functions, hiring a secretary earlier would be similar to hiring later. The two discount functions stress that Bob is more eager to hire a secretary than Ann. Table 1 shows the value of the functions in each time i.

i	0	1	2	3
d_{Ann}	1	0.5	0.333	0.25
d_{Bob}	1	0.5	0.25	0.125

Table 1: Values for $d_{\text{Ann}}(i)$ and $d_{\text{Ann}}(i)$

The satisfaction value of the agents' goals is only different from 0 in the states in which a candidate were hired. Let σ_{abc} denote the strategy of playing y for each candidate (that is, $\sigma_{abc}(q_0) = \sigma_{abc}(q_1) = \sigma_{abc}(q_3) = y$), σ_{bc} denote the strategy of playing y only for candidates b and c, and σ_c denote the strategy of playing y only for c. Table 2 shows the satisfaction value of agents goals' from the initial state q_0 for different assignments of strategies.

| | | $\chi(\text{Bob})$ | |
	σ_{abc}	σ_{bc}	σ_c
σ_{abc}	$(0.5, 0.5)$	$(1, 0.25)$	$(0.25, 0.125)$
$\chi(\text{Ann})$ σ_{bc}	$(1, 0.25)$	$(1, 0.25)$	$(0.25, 0.125)$
σ_c	$(0.25, 0.125)$	$(0.25, 0.125)$	$(0.25, 0.125)$

Table 2: Value of $(\llbracket\psi_{\text{Ann}}\rrbracket^{\mathcal{G}_{sec},\mathbf{r}}_\chi(q_0), \llbracket\psi_{\text{Bob}}\rrbracket^{\mathcal{G}_{sec},\mathbf{r}}_\chi(q_0))$ for different strategy assignments χ.

As illustrated on Table 2, the strategy profile $(\sigma_{bc}, \sigma_{abc})$ is a Nash equilibrium and thus $\llbracket\hat{\varphi}_{\text{NE}}\rrbracket^{\mathcal{G}_s,\mathbf{r}} \neq 0$ (note memoryless strategies are enough for this problem).

4.3 Negotiation with time constraints

Let us consider a second context where discounting is a key issue. Negotiation is type of interaction in MAS in which disputing agents decide how to divide a resource. Time constraints, which may be in the form of both deadlines and discount factors, are an essential element from negotiation, because the interaction cannot go on indefinitely and must end within a reasonable time limit [Livne, 1979]. Here we consider the problem of negotiation with time constraints studied in [Rubinstein, 1982; Fatima *et al.*, 2006], and generalize to the multiple agent case. In this problem, agents want to determine how to divide (single or multiple) issues, called "pies", of size 1 among themselves. The negotiation must end in at most $n \in \mathbb{N}^+$ rounds. This deadline can be represented with an arbitrary discounting function d_n such that $d_n(n) = 0$. In this case, a goal in the form $\mathbf{F}_{d_n}\psi$ motivates agents to achieve ψ before the n-th stage of the negotiation.

The negotiation process is made by alternating offers from the agents. Initially, a starts by making an offer on how to divide a pie to the other agents Ag_{-a}. Agents in Ag_{-a} can either accept or reject this offer. If agents in Ag_{-a} accept, the negotiation ends in an agreement with the proposed share. Otherwise, an agent $b \neq a$ makes a counteroffer in the next round. The negotiation proceeds until there is an agreement on accepting an offer. The key feature of this problem is that the pie is assumed to shrink (i.e., to lose value) with time [Rubinstein, 1982]. This represents the situation in which the pie perishes with time or is affected by inflation. The pie shrinkage is represented with by a discount function d_{pie}. At time $i = 1$, the size of the pie is 1, but in all subsequent time periods $i > 1$, the pie shrinks to $d_{pie}(i)$.

Figure 2 shows the CGS $\mathcal{G}_{ngt}$, which illustrates an instance of the negotiation problem with a single-issue and two agents, Alice and Beth. The game starts in state q_0, where Alice can make an offer to split the pie either so as to take half or two thirds of it for herself (while the remaining of the pie is left for Beth). In the next state (either q_1 or q_2, according to Alice's action), Beth can perform the action acc to accept the offer or she can make a counteroffer and pass the turn to Alice. As soon as an agent accepts an offer, the negotiation ends and the pie is divided (e.g., states q_3, q_6, q_9, ans q_{12}).

Let us use the atomic propositions twothird$_a$, half$_a$, and onethird$_a$ to denote whether agent $a \in \{$Alice, Beth$\}$ has received two thirds, half, or one-third of the pie. Agents may have different preferences for how much of the pie they re-
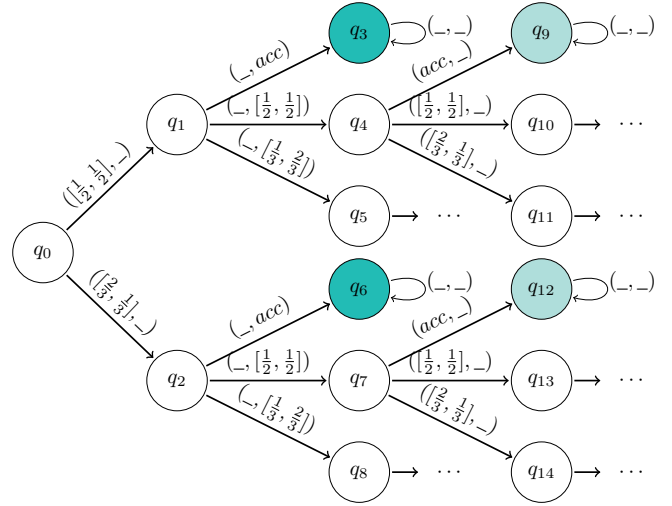

Figure 2: $\mathcal{G}_{ngt}$ representing the single-issue negotiation problem with two agents, who alternate into proposing a division of the resource. The negotiation ends when one of the agents agree with the proposed division (e.g., at the colored states q_3, q_6, q_9, q_{12}).

ceive. Discounting functions can be used to capture the share they are more eager to receive. For instance, let

$$\psi_a := \mathbf{F}_{d_{2/3}}\text{twothird}_a \vee \mathbf{F}_{d_{1/2}}\text{half}_a \vee \mathbf{F}_{d_{1/3}}\text{onethird}_a$$

be the goal of agents $a \in \{Alice, Beth\}$, with the discounting functions defined as $d_{n/m} := \frac{n}{m}d_{pie}(i)$ for $n, m \in \{1, 2, 3\}$. This goal stresses that agent a prefers to get two-thirds of the pie over half or one-third, and half of the pie over one-third. Note that for the sake of simplicity of this example, deadlines are not considered in ψ_a.

To continue the example, consider that the discounting function d_{pie} is defined as follows

$$d_{pie}(i) = \begin{cases} 1 & \text{if } i \leq 2 \\ \left(\frac{1}{2}\right)^i & \text{otherwise} \end{cases}$$

This represents that the pie starts shrinking only after the 2nd game stage (states q_9, q_{10}, q_{11} and so on). After that, the pie shrinks by half in each successive state. In this case, the rate in which the pie shrinks motivates agents to accept the first proposed division.

Given the discount function $d_{pie}(i)$ and the goals ψ_{Alice} and ψ_{Beth}, a Nash equilibrium from the game is the strategy profile $(\sigma_{\text{Alice}}, \sigma_{\text{Beth}})$, where σ_{Alice} and σ_{Beth} are strategies such that $\sigma_{\text{Alice}}(q_0) = [\frac{2}{3}, \frac{1}{3}]$ and $\sigma_{\text{Beth}}(q) = acc$ for any state q. Thus, we have that $\llbracket\hat{\varphi}_{\text{NE}}\rrbracket^{\mathcal{G}_n,\mathbf{r}} \neq 0$.

5 Model checking SL with Discounting

In this section, we study the quantitative model checking problem for SL$^{\text{disc}}[\mathcal{D}]$. Let us first define it formally.

Definition 4. The threshold *model-checking problem* for SL$^{\text{disc}}[\mathcal{D}]$ consists in deciding, given a formula φ, a CGS $\mathcal{G}$, $\rho \in \{\text{R}, \text{r}\}$, and a threshold $\vartheta \in [0, 1]$, whether $\llbracket\varphi\rrbracket^{\mathcal{G},\rho} \geq \vartheta$.

5.1 Memoryless strategies

Model-checking $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ with memoryless agents is no harder than model-checking LTL or classical SL with memoryless agents.

Theorem 1. *Assuming that functions in $\mathcal{D}$ can be computed in polynomial space, model checking $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ with memoryless agents is* PSPACE-*complete.*

Proof. The lower bound is inherited from SL [Cermák *et al.*, 2018][2], which is captured by $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$. For the upper bound, we first show that each recursive call only needs at most polynomial space. Most cases are treated analogously to the proof of Theorem 2 in [Maubert *et al.*, 2021]. We focus on the case $[\![\varphi_1 \mathbf{U}_d \varphi_2]\!]_\chi^{\mathcal{G},\mathrm{r}}(v)$. Let $\pi = \mathrm{Out}(v,\chi)$. When evaluating a discounted operator on π, one can restrict attention to two cases: either the satisfaction value of the formula goes below ϑ, in which case this happens after a bounded prefix (with index $m \geq 0$), or the satisfaction value always remains above ϑ, in which case we can replace the discounted operator with a Boolean one. This allows us to look only at a finite number of stages. In the first case, let $m \geq 0$ denote the first index in which the satisfaction value of the formula goes below ϑ. Let $\varphi = \varphi_1 \mathbf{U}_d \varphi_2$, it follows that

$$[\![\varphi]\!]_\chi^{\mathcal{G},\mathrm{r}}(v) = \sup_{i \geq 0} \min\left(d(i)[\![\varphi_2]\!]_\chi^{\mathcal{G},\mathrm{r}}(\pi_i), \min_{0 \leq j < i}[\![\varphi_1]\!]_\chi^{\mathcal{G},\mathrm{r}}(\pi_j)\right)$$

$$= \max_{0 \leq i \leq m} \min\left(d(j)[\![\varphi_2]\!]_\chi^{\mathcal{G},\mathrm{r}}(\pi_i), \min_{0 \leq j < i}[\![\varphi_1]\!]_\chi^{\mathcal{G},\mathrm{r}}(\pi_j)\right)$$

This can be computed by a while loop that increments i, computes $[\![\varphi_2]\!]_\chi^{\mathcal{G},\mathrm{r}}(\pi_i)$, $\min_{0 \leq j < i}[\![\varphi_1]\!]_\chi^{\mathcal{G},\mathrm{r}}(\pi_j)$ and their minimum, records the result if it is bigger than the previous maximum, and stops upon reaching a position that has already been visited. This requires storing the current value of $\min_{0 \leq j < i}[\![\varphi_1]\!]_\chi^{\mathcal{G},\mathrm{r}}(\pi_j)$, the current maximum, and the list of positions already visited, which are at most $|V|$. The second case is treated as for Boolean until.

Next, the number of nested recursive calls is at most $|\varphi|$, so the total space needed is bounded by $|\varphi|$ times a polynomial in the size of the input, and is thus polynomial. □

5.2 Perfect recall

Our solution to the problem of $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ model checking for perfect recall applies the automata-theoretic approach [Thomas, 1990; Vardi and Wolper, 1986]. The solution opportunely combines the techniques used for model-checking in [Almagor *et al.*, 2014; Mogavero *et al.*, 2014]. Let us recall relevant definitions from automata theory (see [Kupferman *et al.*, 2000] for details).

Alternating tree automata An *alternating tree automaton* (ATA) is a tuple $\mathcal{A} = \langle \Sigma, \Delta, Q, \delta, q_0, \aleph \rangle$, where Σ, Δ, and Q are, respectively, non-empty finite sets of *input symbols*, *directions*, and *states*, $q_0 \in Q$ is an *initial state*, $\aleph$ is an *acceptance condition*, and $\delta : Q \times \Sigma \to \mathcal{B}^+(\Delta \times Q)$ is an

[2]The proof provided in [Cermák *et al.*, 2018] (Thm. 1) considers SL with epistemic operators, but by carefully reading the proof one can notice removing the operators and restricting to perfect information do not affect the complexity results.

alternating transition function that maps each pair of states and input symbols to a positive Boolean combination on the set of propositions of the form $(d,q) \in \Delta \times Q$, called *moves*.

A Σ-labeled tree is a pair $\langle T, v \rangle$ where T is a tree and $V : T \to \Sigma$ maps each node of T to a letter in Σ.

Run A run of an ATA $\mathcal{A} = \langle \Sigma, \Delta, Q, \delta, q_0, \aleph \rangle$ on a Σ-labeled Δ-tree $\tau = \langle T, v \rangle$ is a $(\Delta \times Q)$-tree R such that for all nodes $x \in R$, where $x = \prod_{i=1}^n (d_i, q_i)$ and $y = \prod_{i=1}^n d_i$ with $n \in [0, \omega[$, it holds that (i) $y \in T$ and (ii) there is a set of moves $S \subseteq \Delta \times Q$ with $S \models \delta(q_n, v(y))$ such that $x \cdot (d,q) \in R$ for all $(d,q) \in S$.

Alternating parity tree automata (APT) are alternating tree automata along with a *parity acceptance condition* [Grädel *et al.*, 2002]. We consider ATAs along with the parity acceptance condition (APT) $\aleph = (F_1, ..., F_k) \in (2^Q)^+$ with $F_1 \subseteq ... \subseteq F_k = Q$. A nondeterministic parity tree automaton (NPT) is a special case of APT in which each conjunction in the transition function δ has exactly one move (d,q) associated with each direction d.

APT Acceptance An APT $\mathcal{A} = \langle \Sigma, \Delta, Q, \delta, q_0, \aleph \rangle$ accepts a Σ-labeled Δ-tree τ if and only if is there exists a run R of $\mathcal{A}$ on τ such that all its infinite branches satisfy the acceptance condition $\aleph$. By $\mathcal{L}(\mathcal{A})$ we denote the language accepted by the APT $\mathcal{A}$, that is, the set of trees τ accepted by $\mathcal{A}$. The emptiness problem for $\mathcal{A}$ is to decide whether $\mathcal{L}(\mathcal{A}) = \emptyset$.

From $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ to APT

We reuse the structure of the model-checking approach for SL [Mogavero *et al.*, 2014]. Precisely, given a CGS $\mathcal{G}$, a state v, and an SL-sentence φ, the procedure consists of building an NPT that is non-empty if φ is satisfied in $\mathcal{G}$ at state v (Thm 5.8 [Mogavero *et al.*, 2014]). As an intermediate step to obtain the NPT, the construction builds an APT $\mathcal{A}$ that accepts a tree encoding of $\mathcal{G}$ containing the information on an assignment χ iff the CGS satisfies the formula of interest for χ. The NPT $\mathcal{N}$ is obtained by using an APT direction projection with distinguished direction v to the APT $\mathcal{A}$ (Thm 5.4 [Mogavero *et al.*, 2014]). The size of the APT $\mathcal{A}$ is polynomial in the size of $\mathcal{G}$ and exponential in the number k of alternations of strategy quantifiers. Then, building the NPT $\mathcal{N}$ and checking its emptiness requires an additional exponent on top of the number of alternations k, which leads to a final complexity $(k+1)$-EXPTIME-complete (and PTIME in the size of the $\mathcal{G}$). For adapting this procedure to model checking of $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ with perfect recall, we need to unpack and extend the construction of the APT shown in Lemma 5.6 in [Mogavero *et al.*, 2014], which we do here in the rest of this section.

We define a translation for each $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ formula φ to an APT $\mathcal{A}$ that recognizes a tree encoding τ of a CGS $\mathcal{G}$, containing the information on the assignment χ iff $[\![\psi]\!]_\chi^{\mathcal{G},\mathrm{R}}(v_\iota) \geq \vartheta$.

Defining the appropriate transition function for the $\mathcal{A}$ follows the semantics of $\mathsf{SL}^{\mathrm{disc}}[\mathcal{D}]$ in the expected manner. The transitions involving the discounting operators need a careful treatment, as discounting formulas can take infinitely many satisfaction values. As for $\mathsf{LTL}^{\mathrm{disc}}[\mathcal{D}]$ [Almagor *et al.*, 2014], given a threshold ϑ and a computation π, when evaluating a discounted operator on π, one can restrict attention to two cases: either the satisfaction value of the formula goes below

ϑ, in which case this happens after a bounded prefix, or the satisfaction value always remains above ϑ, in which case we can replace the discounted operator with a Boolean one.

As for [Mogavero *et al.*, 2014], we use the concept of encoding for a CGS assignment. First, let $\text{Val}_\varphi := \text{free}(\varphi) \to \text{Ac}$.

Assignment-State Encoding Let $\mathcal{G}$ be a CGS, $v \in V$ be a state, and χ be an assignment. Then, the assignment-encoding for χ is the $(\text{Val}_\varphi \times V)$-labeled V-tree τ, $\langle T, u \rangle$, such that T is the set of histories h of $\mathcal{G}$ given χ starting in v and $u(h) := (f, q)$, where q is the last state in h and $f : \text{free}(\varphi) \to \text{Ac}$ is defined by $f(s) := \chi(s)(h)$ for each free variable $s \in \text{free}(\psi)$.

Lemma 1. *Let $\mathcal{G}$ be a CGS, φ an $\mathsf{SL}^{disc}[\mathcal{D}]$ formula, and $\vartheta \in [0, 1]$ be a threshold. Then, there exists an $\mathcal{A}_{\varphi, \vartheta} = \langle \text{Val}_\varphi \times V, V, Q, \delta, q_0, \aleph \rangle$ such that, for all states $q \in Q$, and assignments χ, it holds that $\llbracket \varphi \rrbracket^{\mathcal{G}, R}_\chi (v) > \vartheta$ iff $\tau \in \mathcal{L}(\mathcal{A}_{\varphi, \vartheta})$, where τ is the assignment-state encoding for χ.*

Proof sketch. The construction of the APT $\mathcal{A}_{\varphi, \vartheta}$ is done recursively on the structure of the formula φ. Let $xcl(\varphi)$ be the extended closure of φ defined analogously to [Almagor *et al.*, 2014]. The state space Q consists of two types of states. Type-1 states are assertions of the form $(\psi > t)$ or $(\psi < t)$, where $\psi \in xcl(\varphi)$ is not an SL formula and $t \in [0, 1]$. Type-2 states correspond to SL formulas. The precise definition of $xcl(\varphi)$, Type 1 and Type 2 states is analogously to [Almagor *et al.*, 2014]. Let S be the set of Type-1 and Type-2 states for all $\psi \in xcl(\varphi)$ and thresholds $t \in [0, 1]$. Then, Q is the subset of S constructed on-the-fly according to the transition function defined below.

The transition function $\delta : (\text{Val}_\varphi \times V) \to \mathcal{B}^+(V \times Q)$ is defined as follows. For Type-2 states, the transitions are as in the standard translation from SL to APT [Mogavero *et al.*, 2014]. For the other states, we define the transitions as follows. Let $(f, v) \in (\text{Val}_\varphi \times V)$ and $\oplus \in \{<, >\}$.

- $\delta((p > t), (f, v)) = \begin{cases} true & \text{if } p \in \ell(v) \text{ and } t < 1, \\ false & \text{otherwise.} \end{cases}$

- $\delta((p < t), (f, v)) = \begin{cases} false & \text{if } p \in \ell(v) \text{ or } t = 0, \\ true & \text{otherwise.} \end{cases}$

- $\delta((\exists s \psi) \oplus t), (f, v)) = \bigvee_{c \in \text{Ac}} \delta'(\psi \oplus t, (f[s \mapsto c], v))$ where δ'_ψ is obtained by nondeterminizing the APT $\mathcal{A}_{\psi, t}$, by applying the classic transformation [Muller and Schupp, 1987] which gives the equivalent NPT $\mathcal{N}_{\psi, t} = \langle \text{Val}_\psi \times V, V, Q', \delta', q'_0, \aleph' \rangle$.

- $\delta(((s, a) \psi \oplus t), (f, v)) = \delta'((\psi \oplus t), (f', v))$ where $f' = f[t \mapsto f(s)]$ if $t \in \text{free}(\psi)$, and $f' = f$ otherwise.

The remaining cases are a simple adaptation of the proof in [Almagor *et al.*, 2014] (Thm 1) to the input symbols $\text{Val}_\varphi \times V$.

The initial state of $\mathcal{A}_{\varphi, \vartheta}$ is $(\varphi > \vartheta)$. The accepting states are these of the form $(\psi_1 \mathbf{U} \psi_2 < t)$ for Type-1 states, as well as accepting states that arise in the standard translation of Boolean SL to APT for Type-2 states. While the construction as described above is infinite only finitely many states are reachable from the initial state, and we can compute these states in advance. □

Using the threshold and the discounting behavior of the discounted-Until, we can restrict attention to a finite resolution of satisfaction values, enabling the construction of a finite automaton. Its size depends on the functions in $\mathcal{D}$. Intuitively, the faster the discounting tends to 0, the fewer states there will be. Thus, the exact complexity of model checking $\mathsf{SL}^{disc}[\mathcal{D}]$ (which relies on the size of the APT) depends on two aspects. First, the alternation of quantifiers in the formula and, second, the type of discounting functions considered. In the specific setting where $\mathcal{D}$ is composed of exponential-discounting functions, (i.e., $\mathcal{D} \subseteq \{d(j) = \lambda^j : j \in (0, 1) \cap \mathbb{Q}\}$), the overall complexity remains as it is for SL. Exponential discounting functions are perhaps the most common class of discounting functions, as they describe many natural processes (e.g., temperature change and effective interest rate [Shapley, 1953; De Alfaro *et al.*, 2003]).

Theorem 2. *Assuming that functions in $\mathcal{D}$ are exponential-discounting, model checking $\mathsf{SL}^{disc}[\mathcal{D}]$ with memoryfull agents is $(k+1)$-EXPTIME and k-EXPSPACE w.r.t the number k of quantifiers alternations in the specification.*

Proof sketch. The model checking procedure from [Mogavero *et al.*, 2014] is $(k + 1)$-EXPTIME-complete and k-EXPSPACE w.r.t the number k of quantifiers alternations in the specification. Let $\vartheta \in (0, 1)$ be a threshold. When discounting by an exponential-discounting function $d(j) = \lambda^j \in \mathcal{D}$, the number of states in the APT constructed as per Lemma 1 is proportional to the maximal number j such that $\lambda^j < \vartheta$, which is polynomial in the description length of ϑ and λ [Almagor *et al.*, 2014]. □

6 Conclusion and Discussion

In this paper, we proposed Strategy Logic with discounting ($\mathsf{SL}^{disc}[\mathcal{D}]$), which contains an operator that captures the idea that the longer it takes to fulfill a requirement, the smaller the satisfaction value is. This work expends the research on temporal and strategic reasoning in Game Theory. As advocated by Pauly and Wooldridge (2003), logics for strategic reasoning can have an important role in the specification and verification of game-theoretical problems and, in particular, related to Automated Mechanism Design (AMD). Indeed, recent works have proposed a new approach for AMD based on model-checking and synthesis from specifications in $\mathsf{SL}[\mathcal{F}]$ [Maubert *et al.*, 2021; Mittelmann *et al.*, 2022]. Remarkably, $\mathsf{SL}^{disc}[\mathcal{D}]$ provides less complicated machinery in relation to $\mathsf{SL}[\mathcal{F}]$, as it is defined over classical concurrent game structures. More importantly, it brings a new dimension for reasoning about mechanisms that take into consideration how events are affected by how long in the future they occur.

There are several interesting directions for future work, including considering synthesis from $\mathsf{SL}^{disc}[\mathcal{D}]$-specifications as well as the setting of imperfect information. With SL already, imperfect information yields undecidability, but known tractable fragments exist [Berthon *et al.*, 2021; Belardinelli *et al.*, 2020]. We will investigate them in the case of $\mathsf{SL}^{disc}[\mathcal{D}]$.

Acknowledgments

References

[Abreu, 1988] Dilip Abreu. On the theory of infinitely repeated games with discounting. *Econometrica: Journal of the Econometric Society*, pages 383–396, 1988.

[Alechina *et al.*, 2017] Natasha Alechina, Brian Logan, Hoang Nga Nguyen, and Franco Raimondi. Model-checking for resource-bounded ATL with production and consumption of resources. *J. Comput. Syst. Sci.*, 88:126–144, 2017.

[Alechina *et al.*, 2018] Natasha Alechina, Nils Bulling, Stéphane Demri, and Brian Logan. On the complexity of resource-bounded logics. *Theor. Comput. Sci.*, 750:69–100, 2018.

[Almagor *et al.*, 2014] Shaull Almagor, Udi Boker, and Orna Kupferman. Discounting in LTL. In *Proc. of TACAS*, 2014.

[Almagor *et al.*, 2016] Shaull Almagor, Udi Boker, and Orna Kupferman. Formally reasoning about quality. *Journal of the ACM*, 63(3):1–56, 2016.

[Alur *et al.*, 2002] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[Aminof *et al.*, 2016] Benjamin Aminof, Aniello Murano, Sasha Rubin, and Florian Zuleger. Prompt alternating-time epistemic logics. In *Proc. of KR*, 2016.

[Babaioff *et al.*, 2009] Moshe Babaioff, Michael Dinitz, Anupam Gupta, Nicole Immorlica, and Kunal Talwar. Secretary problems: weights and discounts. In *Proc. of SODA*, 2009.

[Belardinelli *et al.*, 2020] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. Verification of multi-agent systems with public actions against strategy logic. *Artificial Intelligence*, 285, 2020.

[Berthon *et al.*, 2021] Raphaël Berthon, Bastien Maubert, Aniello Murano, Sasha Rubin, and Moshe Y. Vardi. Strategy logic with imperfect information. *ACM Trans. Comput. Logic*, 22(1), 2021.

[Bohy *et al.*, 2013] Aaron Bohy, Véronique Bruyère, Emmanuel Filiot, and Jean-François Raskin. Synthesis from LTL specifications with mean-payoff objectives. In *Proc. of TACAS*, 2013.

[Bouyer *et al.*, 2008] Patricia Bouyer, Ulrich Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Jirí Srba. Infinite runs in weighted timed automata with energy constraints. In *Proc. of FORMATS*, 2008.

[Bouyer *et al.*, 2014] Patricia Bouyer, Nicolas Markey, and Raj Mohan Matteplackel. Averaging in LTL. In *Proc. of CONCUR*, 2014.

[Bouyer *et al.*, 2019] Patricia Bouyer, Orna Kupferman, Nicolas Markey, Bastien Maubert, Aniello Murano, and Giuseppe Perelli. Reasoning about quality and fuzziness of strategic behaviours. In *Proc. of IJCAI*, 2019.

[Bouyer *et al.*, 2023] Patricia Bouyer, Orna Kupferman, Nicolas Markey, Bastien Maubert, Aniello Murano, and Giuseppe Perelli. Reasoning about quality and fuzziness of strategic behaviors. *ACM Trans. Comput. Logic*, 24(3), 2023.

[Brihaye *et al.*, 2007] Thomas Brihaye, François Laroussinie, Nicolas Markey, and Ghassan Oreiby. Timed concurrent game structures. In *Proc. of CONCUR*, 2007.

[Bulling and Goranko, 2022] Nils Bulling and Valentin Goranko. Combining quantitative and qualitative reasoning in concurrent multi-player games. *Auton. Agents Multi Agent Syst.*, 36(1):2, 2022.

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

[Chakrabarti *et al.*, 2003] Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Mariëlle Stoelinga. Resource interfaces. In *Proc. of EMSOFT*, 2003.

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

[Chen *et al.*, 2013] Taolue Chen, Vojtech Forejt, Marta Z. Kwiatkowska, David Parker, and Aistis Simaitis. Automatic verification of competitive stochastic systems. *Formal Methods Syst. Des.*, 43(1):61–92, 2013.

[Clarke *et al.*, 2018] Edmund M Clarke, Thomas A Henzinger, Helmut Veith, Roderick Bloem, et al. *Handbook of model checking*, volume 10. Springer, 2018.

[De Alfaro *et al.*, 2003] Luca De Alfaro, Thomas A Henzinger, and Rupak Majumdar. Discounting the future in systems theory. In *Proc. of ICALP*, 2003.

[De Alfaro *et al.*, 2005] Luca De Alfaro, Marco Faella, Thomas A Henzinger, Rupak Majumdar, and Mariëlle Stoelinga. Model checking discounted temporal properties. *Theor. Comput. Sci.*, 345(1):139–170, 2005.

[Do *et al.*, 2022] Virginie Do, Matthieu Hervouin, Jérôme Lang, and Piotr Skowron. Online approval committee elections. In *Proc. of IJCAI*, 2022.

[Ehrenfeucht and Mycielski, 1979] Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979.

[Emerson and Jutla, 1991] E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy. In *Proc. of FOCS*, 1991.

[Fatima *et al.*, 2006] Shaheen Fatima, Michael Wooldridge, and Nicholas Jennings. Multi-issue negotiation with deadlines. *J. Artif. Intell. Res.*, 27:381–417, 2006.

[Fijalkow *et al.*, 2020] Nathanaël Fijalkow, Bastien Maubert, Aniello Murano, and Moshe Y. Vardi. Assume-guarantee synthesis for prompt linear temporal logic. In *In Proc. of IJCAI*, 2020.

[Filar and Vrieze, 1996] Jerzy Filar and Koos Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, Berlin, Heidelberg, 1996.

[Freeman, 1983] P.R. Freeman. The secretary problem and its extensions: A review. *Int. Statistical Review/Revue Int. e de Statistique*, pages 189–206, 1983.

[Fudenberg and Maskin, 1990] Drew Fudenberg and Eric Maskin. Nash and perfect equilibria of discounted repeated games. *J. of Econ, Theory*, 51(1):194–206, 1990.

[Fudenberg and Maskin, 2009] Drew Fudenberg and Eric Maskin. The folk theorem in repeated games with discounting or with incomplete information. In *A long-run collaboration on long-run games*. World Scientific, 2009.

[Grädel *et al.*, 2002] Erich Grädel, Wolfgang Thomas, and Thomas Wilke. *Automata, logics, and infinite games: a guide to current research*, volume 2500. Springer Science & Business Media, 2002.

[Gutierrez *et al.*, 2021] Julian Gutierrez, Aniello Murano, Giuseppe Perelli, Sasha Rubin, Thomas Steeples, and Michael J. Wooldridge. Equilibria for games with combined qualitative and quantitative objectives. *Acta Informatica*, 58(6):585–610, 2021.

[Harris and Madden, 2002] Andrew C Harris and Gregory J Madden. Delay discounting and performance on the prisoner's dilemma game. *The Psychological Record*, 52(4):429–440, 2002.

[Henzinger and Prabhu, 2006] Thomas A. Henzinger and Vinayak S. Prabhu. Timed alternating-time temporal logic. In *Proc. of FORMATS*, 2006.

[Jamroga *et al.*, 2020] Wojciech Jamroga, Beata Konikowska, Damian Kurpiewski, and Wojciech Penczek. Multi-valued verification of strategic ability. *Fundam. Informaticae*, 175(1-4):207–251, 2020.

[Jamroga, 2008a] Wojciech Jamroga. A temporal logic for markov chains. In *Proc. of AAMAS*, 2008.

[Jamroga, 2008b] Wojciech Jamroga. A temporal logic for stochastic multi-agent systems. In *Proc. of PRIMA*, 2008.

[Kupferman *et al.*, 2000] Orna Kupferman, Moshe Y Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, 2000.

[Laroussinie *et al.*, 2006] François Laroussinie, Nicolas Markey, and Ghassan Oreiby. Model-checking timed. In *Proc. of FORMATS*, 2006.

[Livne, 1979] Zvi A Livne. *The role of time in negotiations*. PhD thesis, Massachusetts Institute of Technology, 1979.

[Locey *et al.*, 2013] Matthew L Locey, Vasiliy Safin, and Howard Rachlin. Social discounting and the prisoner's dilemma game. *Journal of the experimental analysis of behavior*, 99(1):85–97, 2013.

[Mandrali, 2012] Eleni Mandrali. Weighted LTL with discounting. In *Proc. of CIAA*, 2012.

[Maubert *et al.*, 2021] Bastien Maubert, Munyque Mittelmann, Aniello Murano, and Laurent Perrussel. Strategic reasoning in automated mechanism design. In *Proc. of KR*, 2021.

[Mittelmann and Perrussel, 2020] Munyque Mittelmann and Laurent Perrussel. Auction description language (ADL): General framework for representing auction-based markets. In *Proc. of ECAI*, 2020.

[Mittelmann *et al.*, 2022] Munyque Mittelmann, Bastien Maubert, Aniello Murano, and Laurent Perrussel. Automated synthesis of mechanisms. In *Proc. of IJCAI*, 2022.

[Mogavero *et al.*, 2014] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. Reasoning about strategies: On the model-checking problem. *ACM Trans. Comput. Log.*, 15(4), 2014.

[Muller and Schupp, 1987] David E. Muller and Paul E. Schupp. Alternating automata on infinite trees. *Theor. Comput. Sci.*, 54(2):267–276, 1987.

[Nisan *et al.*, 2007] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[Pauly and Wooldridge, 2003] M. Pauly and M. Wooldridge. Logic for mechanism design–a manifesto. In *Workshop on Game Theory and Decision Theory in Agent Syst.*, 2003.

[Pęski, 2014] Marcin Pęski. Repeated games with incomplete information and discounting. *Theoretical Economics*, 9(3):651–694, 2014.

[Pnueli, 1977] Amir Pnueli. The temporal logic of programs. In *Prof. of FOCS*, 1977.

[Rubinstein, 1982] Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, pages 97–109, 1982.

[Shapley, 1953] Lloyd S Shapley. Stochastic games. *Proc. of national academy of sciences*, 39(10):1095–1100, 1953.

[Thomas, 1990] Wolfgang Thomas. Automata on infinite objects. In *Formal Models and Semantics*, pages 133–191. Elsevier, 1990.

[Vardi and Wolper, 1986] Moshe Y Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *Proc. of LICS*, 1986.

[Vester, 2015] Steen Vester. On the complexity of model-checking branching and alternating-time temporal logics in one-counter systemss. In *Proc. of ATVA*, 2015.

[Weg *et al.*, 1990] Eythan Weg, Amnon Rapoport, and Dan S Felsenthal. Two-person bargaining behavior in fixed discounting factors games with infinite horizon. *Games and Economic Behavior*, 2(1):76–95, 1990.

[Zwick and Paterson, 1996] Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996.