Journal of Artificial Intelligence Research 66 (2019) 197-223 Submitted 10/2018; published 09/2019

Timed ATL: Forget Memory, Just Count

Étienne André ETIENNE.ANDRE@LIPN.UNIV-PARIS13.FR
Laure Petrucci LAURE.PETRUCCI@LIPN.UNIV-PARIS13.FR
LIPN, CNRS UMR 7030, Université Paris 13,
Sorbonne Paris Cité, F-93430, Villetaneuse, France

Wojciech Jamroga W.JAMROGA@IPIPAN.WAW.PL
Michał Knapik MKNAPIK@IPIPAN.WAW.PL
Wojciech Penczek PENCZEK@IPIPAN.WAW.PL
Institute of Computer Science, Polish Academy of Sciences,
Warsaw, Poland

Abstract

In this paper we investigate the Timed Alternating-Time Temporal Logic (TATL), a discrete-time extension of ATL. In particular, we propose, systematize, and further study semantic variants of TATL, based on different notions of a strategy. The notions are derived from different assumptions about the agents' memory and observational capabilities, and range from timed perfect recall to untimed memoryless plans. We also introduce a new semantics based on counting the number of visits to locations during the play. We show that all the semantics, except for the untimed memoryless one, are equivalent when punctuality constraints are not allowed in the formulae. In fact, abilities in all those notions of a strategy collapse to the "counting" semantics with only two actions allowed per location. On the other hand, this simple pattern does not extend to the full TATL.

As a consequence, we establish a hierarchy of TATL semantics, based on the expressivity of the underlying strategies, and we show when some of the semantics coincide. In particular, we prove that more compact representations are possible for a reasonable subset of TATL specifications, which should improve the efficiency of model checking and strategy synthesis.

1. Introduction

The field of Multi-Agent Systems (MAS) concerns design, implementation, modeling, and verification of interacting agents, often assumed to be intelligent and autonomous. Alternating-time temporal logic ATL* and its fragment ATL (Alur, Henzinger, & Kupferman, 1997, 2002) provide a logical framework that allows for modeling, reasoning about, and verification of strategic interactions in MAS (Alur, Henzinger, Mang, Qadeer, Rajamani, & Tasiran, 1998; Alur, de Alfaro, Grossu, Henzinger, Kang, Kirsch, Majumdar, Mang, & Wang, 2001; Kacprzak & Penczek, 2004, 2005; Lomuscio & Raimondi, 2006; Bursztein & Goubault-Larrecq, 2007; Ezekiel & Lomuscio, 2009; Boureanu, Jones, & Lomuscio, 2012; Jamroga & Penczek, 2012; Huang & van der Meyden, 2014; Busard, Pecheur, Qu, & Raimondi, 2014; Lomuscio, Qu, & Raimondi, 2015; Jamroga, Konikowska, & Penczek, 2016; Boureanu, Kouvaros, & Lomuscio, 2016). However, ATL does not support timing constraints which are of high importance as they allow for expressing, e.g., that each request should be served in a reasonable amount of time, or that no agent should block the service for more than t time units.

In this paper, we investigate a discrete-time extension of ATL, called Timed Alternating-time Temporal Logic (TATL). The basic version of TATL was proposed in (Laroussinie, Markey, & Oreiby, 2006b), and enables to capture strategic properties that depend on both the visited locations (states of the environment) and the time measured along the paths. We propose and study a number of semantic variants, based on different constraints on the expressivity of strategies available to agents and coalitions. The variants range from timed perfect recall (the agents can memorize the whole history of the play, including all the visited locations and their time stamps) to untimed memoryless (only the current location can be used to determine the next action).

In this respect, we show that the timed perfect recall and timed memoryless semantics coincide for TATL. This corresponds to the well-known equivalence of the perfect recall and memoryless semantics for ATL, but the timed version of the result is much more intricate and difficult to prove. Moreover, we show that, for TATL without punctuality constraints, all the more expressive semantics collapse to the "counting" semantics where choices are based on how many times the current location has been visited during the play. In fact, it even suffices to consider the "1-threshold" strategies that specify only up to two actions per location. On the other hand, we prove that the semantics do *not* collapse for the full TATL (i.e. with $\leq$, $\geq$ and $=$ operators), leading to a whole hierarchy of different semantic variants corresponding to different observational and memory-related capabilities of agents.

The results can be interpreted in two ways. On one hand, they indicate that verification and strategy synthesis for $\mathrm{TATL}_{\leq,\geq}$ (i.e., Timed ATL without punctuality constraints) are relatively easy because the underlying notion of ability reduces to simple, almost Markovian strategies. This allows for a significant optimization of data structures and algorithms involved in the verification. As the same is not valid for the full TATL, the results suggest that $\mathrm{TATL}_{\leq,\geq}$ specifications should be used whenever possible.

At the same time, our results show that the language of $\mathrm{TATL}_{\leq,\geq}$ is not expressive enough to capture many subtleties of strategic play of agents who can observe and measure the flow of time. Thus, when expressivity is essential, one should rather sacrifice the complexity, and use the full language of TATL for specification of properties.

1.1 Motivation

The focus of the paper is theoretical: we study how the inclusion of timing information in a strategy influences the actual abilities of agents in the context of reachability and safety objectives. We note, however, that the issue has important practical implications. In particular, timing is highly relevant in strategic reasoning for broadly conceived security, e.g. potential terrorist attacks, hacking attacks, phishing etc. An excellent introduction based on empirical evidence can be found in Farhang and Grossklags (2017). For the purpose of motivating our research, we quote two real-life examples after that paper:

Hijacking incident of Ethiopian Airlines, 2014. On Monday, 17th of February 2014, an Ethiopian Airlines flight from Addis Ababa to Rome was hijacked, with its pilots planning to fly the plane over Switzerland to Geneva. The plane eventually landed in Geneva shortly after 6am. Interestingly, no escort could be provided by the Swiss Air Force because they did not operate before 8am on weekdays. They also did not work during lunch breaks and on weekends. Clearly, those timing aspects had been relevant for the hijackers' strategy (Saner, 2014; Farhang & Grossklags, 2017).

Data security. Three timing factors are recognised as highly important for the strategies of both the attacker and the system: protection time, detection time, and reaction time (Schwartau, 1999). The optimal strategy of the attacker depends also on the time needed to compromise the system. The strategic relevance of those factors has been demonstrated in multiple empirical studies. For instance, Nadella (2015) showed that the analysed organisations on average failed to detect attacks for over 225 days (!). Similarly, it was reported that for a majority of data breach incidents at Verizon (2016) the time needed to compromise was a matter of minutes, while the reaction time was typically a matter of days.

Our results show that, for a broad subclass of reachability and safety properties that may represent the objective of the attacker (hijacker, hacker, phishing agent), the relevant timing information to construct a winning strategy is actually quite limited. The same applies to defence strategies on the part of the system.

1.2 Structure of the Paper

The rest of the paper is organised as follows. In Section 2 we recall the basic definitions behind TATL from Laroussinie et al. (2006b), and illustrate them on a scenario concerning security of online services. In Section 3, we introduce several semantic variants of TATL. Namely, we consider timed vs. timeless, and memoryful vs. memoryless variants of strategies, followed by counting strategies and their n–threshold subclass. In Section 4 and Section 5, we explore the correspondence between the semantics. The positive results, showing equivalence or refinement between different semantic definitions are grouped in Section 4. The negative ones (i.e., non-equivalence proofs) are presented in Section 5. The paper ends in Section 6 with a short summary and an outline of future work.

1.3 Related Work

Strategic behaviour that uses timing information and strives to obtain timed properties have been referred to in many branches of computer science. For instance, timely fulfilment of contractual obligations is an important subject in deontic analysis of deadlines (Dignum & Kuiper, 1998; Broersen, Dignum, Dignum, & Meyer, 2004). As another example, the existence of *timing attacks*, where the attacker can compromise the system by analysing the time taken to execute different operations, create the need for timing analysis in verification of security (Kocher, 1996). The relevance of timing in construction of an attack strategy was demonstrated and discussed e.g. in Farhang and Grossklags (2017), Schwartau (1999), Saner (2014), Nadella (2015), Verizon (2016).

The work presented in this paper fits within the broad context of research on timed games (Maler, Pnueli, & Sifakis, 1995; Alur, Bernadsky, & Madhusudan, 2004; Cassez, David, Fleury, Larsen, & Lime, 2005; Brázdil, Forejt, Krcál, Kretínský, & Kucera, 2013; David, Fang, Larsen, & Zhang, 2014). Dense-timed games have been also specifically explored in Henzinger, Horowitz, and Majumdar (1999), Faella, La Torre, and Murano (2002), Bouyer, D'Souza, Madhusudan, and Petit (2003), Henzinger and Prabhu (2006), Jurdzinski and Trivedi (2007), Brihaye, Laroussinie, Markey, and Oreiby (2007), Faella, La Torre, and Murano (2014). Consequently, two kinds of semantic approaches are possible to logical reasoning about strategic ability: for strategies working in discrete vs. dense time. The former has been studied in Laroussinie et al. (2006b), with a discrete-time extension of the best known strategic logic ATL (Alur et al., 1997, 2002). The latter was investigated in Henzinger and Prabhu (2006), with an extension of a two-player fragment of ATL with physically meaningful strategies, real time, and freeze quantifiers.

As time is a resource, also the work that pertains to game logics describing behaviours of agents that can produce or consume resources should be mentioned. In particular, Alechina, Logan, Nguyen, and Rakib (2010), Alechina, Bulling, Demri, and Logan (2018) introduce and investigate Resource-Bounded ATL that allows for expressing strategic properties under limited reserves.

We build upon the theory introduced in Laroussinie et al. (2006b) that can be seen as the *simplest timed strategic logic*. Note that our aim is *not* to propose a new logical framework for reasoning about timed strategies. Instead, we identify natural restrictions on strategies in timed games, and study their relative expressiveness.

1.4 Previous Versions of the Material

The main ideas of this article appeared in the short conference paper (André, Knapik, Jamroga, & Petrucci, 2017). Here, we thoroughly extend and revise the material. The definitions are now properly formalized, we provide proofs of all the results, and add running examples. We also correct some erroneous statements regarding the relationships between non-equivalent semantic variants of the logic. To this end we provide an extended and corrected picture of these relationships and show that certain results hold only for the existential parts of TATL and $\mathrm{TATL}_{\leq,\geq}$.

2. Background

In this section we recall the basic definitions concerning Timed Alternating-time Temporal Logic (TATL), proposed in Laroussinie et al. (2006b). The framework of TATL is a natural platform for investigating the consequences of strategic decisions that take the flow of time explicitly into account. The strategies of agents and coalitions can depend on the visited locations in the model as well as the time measured along the paths. Note that, since we consider a discrete version of time and since transitions take at least one time unit, all the runs are non-Zeno. A remark in Laroussinie et al. (2006b) suggests that the results presented in the cited paper carry to the general case of non-progressive time, but the proof is missing. Therefore, we leave this avenue for future research and keep the assumption of strictly progressive time.

We begin by introducing the syntax of TATL. Then, we recall the semantics based on Tight Durational Concurrent Game Structures (TDCGS) and their unfoldings to Duration Transition Systems (DTS).

2.1 Syntax of TATL

Timed Alternating-time Temporal Logic (TATL) (Laroussinie et al., 2006b) extends Alternating-time Temporal Logic (ATL) (Alur et al., 1997, 2002) with timing constraints.

Definition 1 (TATL **Syntax)** *Let $\mathcal{AP}$ be a set of atomic propositions and $\mathbb{A}\mathrm{gt}$ be a finite set of all agents. The language of* TATL *is defined by the following grammar:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A \rangle\!\rangle X\, \varphi \mid \langle\!\langle A \rangle\!\rangle \varphi U_{\sim\eta} \varphi \mid \langle\!\langle A \rangle\!\rangle \varphi R_{\sim\eta} \varphi$$

where $\mathsf{p} \in \mathcal{AP}$*,* $A \subseteq \mathbb{A}\mathrm{gt}$*,* $\sim \in \{\leq, =, \geq\}$*, and* $\eta \in \mathbb{N}$*.*

We interpret $\langle\langle A\rangle\rangle\varphi$ as *"the coalition A can enforce φ"*, X stands for *"in the next state"*, U for *"until"*, and R for *"release"*. The Boolean constants $\top$ and $\bot$ are defined as usual. The $\sim\eta$ subscript of the U and R operators allow for expressing a timing constraint.

Following the choice made in Laroussinie et al. (2006b), and in contrast to the usual definitions of ATL, we also include both the "release" and the "until" modality in our syntax: indeed, it was proved in Laroussinie, Markey, and Oreiby (2008) that the release cannot be expressed using only $\langle\langle A\rangle\rangle U$ and $\langle\langle A\rangle\rangle G$.

Additional modalities $\langle\langle A\rangle\rangle F_{\sim\eta}\varphi \coloneqq \langle\langle A\rangle\rangle \top U_{\sim\eta}\varphi$ and $\langle\langle A\rangle\rangle G_{\sim\eta}\varphi \coloneqq \langle\langle A\rangle\rangle \bot R_{\sim\eta}\varphi$, are introduced, where F is interpreted as *"eventually,"* and G as *"always from now on"* are also introduced.

Note that following (Laroussinie et al., 2006b) and unlike in vanilla ATL we introduce $\langle\langle A\rangle\rangle G$ as a derived modality, as it is known (Laroussinie, Markey, & Oreiby, 2006a) that $\langle\langle A\rangle\rangle R$ cannot be expressed using only $\langle\langle A\rangle\rangle U$ and $\langle\langle A\rangle\rangle G$.

By TATL$_{\leq,\geq}$ we denote the subset of the TATL formulae with $\sim\in\{\leq,\geq\}$, i.e. where equality is disallowed. *Existential* TATL, denoted by $\exists$TATL, is the subset of TATL such that negation can be applied only to the atomic propositions.

Example 1 (Online services) *Consider a system that offers online services to its users, e.g. provides information about garbage collection in a municipality, handles submission of tax reports, conference registrations, etc. Let srv_a be an atomic proposition that holds whenever agent a has just accessed the service. Moreover, we assume that users are willing to spend up to t time units waiting for the service to be provided. The following formulae specify interesting properties that may (or shall not) hold for the system:*

Serviceability. $\langle\langle a\rangle\rangle F_{\leq t}\mathsf{srv}_a$: *agent a can use the service within t time units;*

Denial of service (DoS). $\langle\langle a\rangle\rangle G_{\leq t}\big(\bigwedge_{j\in\mathbb{A}\mathrm{gt}\setminus\{a\}}\neg\mathsf{srv}_j\big)$: *agent a can prevent all the others from using the service within t time units;*

Selective DoS. $\bigwedge_{j\in\mathbb{A}\mathrm{gt}\setminus\{a\}}\langle\langle a\rangle\rangle G_{\leq t}\neg\mathsf{srv}_j$: *agent a can prevent any other agent from using the service (though not necessarily all of them together) within t time units;*

Weak DoS. $\bigvee_{j\in\mathbb{A}\mathrm{gt}\setminus\{a\}}\langle\langle a\rangle\rangle G_{\leq t}\neg\mathsf{srv}_j$: *agent a can prevent another agent from using the service within t time units;*

Distributed DoS. $\langle\langle A\rangle\rangle G_{\leq t}\big(\bigwedge_{j\in\mathbb{A}\mathrm{gt}\setminus A}\neg\mathsf{srv}_j\big)$: *agents in A can collude to prevent all the other agents from using the service within t time units. Coalitional variants of weak and selective DoS can be obtained analogously;*

Service crash. $\langle\langle a\rangle\rangle G_{\geq t'}\big(\bigwedge_{j\in\mathbb{A}\mathrm{gt}}\neg\mathsf{srv}_j\big)$: *agent a can effectively disable the service from time t' on.*

In order to define the semantics of TATL we need to recall two semantic structures: Tight Durational Concurrent Game Structures and their Duration Transition Systems.

2.2 Tight Durational CGS

Tight Durational Concurrent Game Structures (Laroussinie et al., 2006b) are Concurrent Game Structures (Alur et al., 2002) with transitions labelled by positive integers interpreted as their durations.

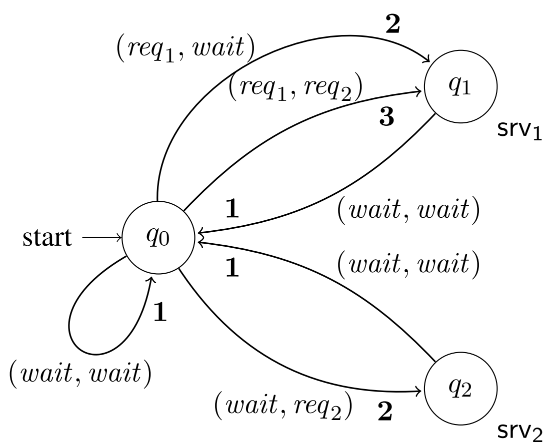

Figure 1: A simple model of online services

Definition 2 (TDCGS) *A* Tight Durational Concurrent Game Structure *is a tuple* $\mathcal{A} = (\mathbb{A}\mathrm{gt}, Act, \mathcal{Q}, \mathcal{AP}, \mathcal{L}, pr, t)$*, where:*

- $\mathbb{A}\mathrm{gt}$ *is a finite set of* agents*,*
- Act *is a finite set of* actions*,*
- $\mathcal{Q}$ *is a finite set of* locations*,*
- $\mathcal{AP}$ *is a set of* atomic propositions*,*
- $\mathcal{L}\colon \mathcal{Q} \to \mathcal{P}(\mathcal{AP})$ *is a* location labeling function*,*
- $pr\colon \mathbb{A}\mathrm{gt} \times \mathcal{Q} \to \mathcal{P}(Act) \setminus \{\emptyset\}$ *is a* protocol function*,*
- $t\colon \mathcal{Q} \times Act^{|\mathbb{A}\mathrm{gt}|} \to \mathcal{Q} \times \mathbb{N}_+$ *is a partial* transition function *such that* $t(q, act)$ *is defined iff* $act_{|a} \in pr(a, q)$ *for each* $a \in \mathbb{A}\mathrm{gt}$.

Transitions in TDCGS model the outcomes of behaviours of coalitions of agents. Each location is assigned a set of atomic propositions by the labelling function $\mathcal{L}$. In any location q, an agent a can take one among several possible actions, as described by its protocol function $pr(a, q)$. Note that this set of actions cannot be empty. Finally, the transition function t specifies the effect of the move of *all* agents such that each of them performs one action of its protocol. From a location and a tuple of actions (one per agent), the transition function t leads to another location with a duration specified by a positive integer.

Example 2 (Online services continued) *A very simple model of access to an online service can be defined by a* TDCGS $\mathcal{A}^k_{srv}$ *consisting of k agents (labelled* $1, \ldots, k$*). In the initial location, each agent can either do nothing (action* $wait$*) or request access to the service (action* req_i*). If all the agents do nothing, the system stays in the same location for 1 unit of time. If one agent sends a request, s/he is granted access and uses the service, which takes 2 time units. If two agents request access simultaneously, access is granted to the agent with the higher priority, i.e. the one with the lower id, and the overall step takes slightly longer (3 time units). Once the service is used, the agents can only execute* $wait$*, returning to the initial state, which takes 1 time unit. Finally, if more*

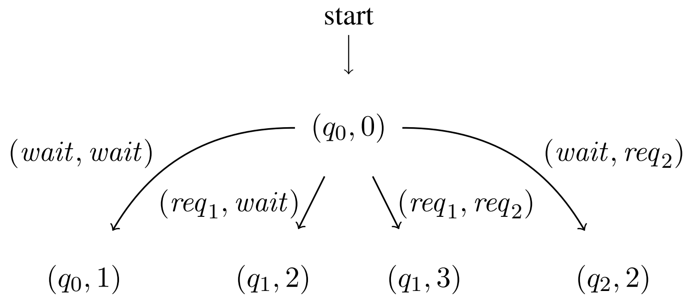

Figure 2: Partial unfolding of the TDCGS for services

than two agents send their requests at the same time, the system refuses to serve any of them, and the decision to refuse takes 2 time units. The variant $\mathcal{A}^2_{srv}$ for $k = 2$ agents is shown in Fig. 1.

In what follows, instead of $pr(a, q)$ we write $pr_a(q)$, for each $a \in \mathbb{A}\mathrm{gt}$ and $q \in \mathcal{Q}$. Moreover, we extend protocols to coalitions by taking $pr_A(q) = \prod_{a \in A} pr_a(q)$. Intuitively, $pr_A(q)$ contains all the tuples of actions viewed as possible synchronous moves of agents in A, enabled in the location q. If $A = \mathbb{A}\mathrm{gt}$, we omit the subscript, simply writing $pr(q)$.

We will use $\overline{A}$ as a shorthand for $\mathbb{A}\mathrm{gt} \setminus A$, Let $act \in pr_A(q)$ be a tuple of actions of all the agents in A and $act' \in pr_{\overline{A}}(q)$ a tuple of actions of all the agents in $\overline{A}$. Then, $(act, act') \in pr(q)$ denotes the unique action such that the projection of (act, act') on A (respectively, $\overline{A}$) yields act (respectively, act').

We define the execution semantics for TDCGS by means of an unfolding to Duration Transition Systems (DTS).

Definition 3 (DTS) *Let $\mathcal{A} = (\mathbb{A}\mathrm{gt}, Act, \mathcal{Q}, \mathcal{AP}, \mathcal{L}, pr, t)$ be a* TDCGS. *The* Duration Transition System *of $\mathcal{A}$ is a tuple $\mathcal{M}(\mathcal{A}) = (\mathcal{S}, \mathcal{E})$, where*

- $\mathcal{S} = \mathcal{Q} \times \mathbb{N}$ *is a set of* states, and
- $\mathcal{E} \colon \mathcal{S} \times Act^{|\mathbb{A}\mathrm{gt}|} \to \mathcal{S}$ is a (partial) transition function such that $\mathcal{E}((q, n), act) = (q', n + l)$ iff $t(q, act) = (q', l)$, for $q, q' \in \mathcal{Q}$, $act \in Act^{|\mathbb{A}\mathrm{gt}|}$, and $n \in \mathbb{N}_+$.

A state of a DTS is a pair composed of a location q and a timestamp n. An edge from a state (q, n) uses the actions act as in the transition $t(q, act)$ leading to a new state composed of the location q' and the time incremented with the transition duration.

Example 3 (Online services continued) *A part of a* DTS*, showing the first step of unfolding for the timed model of online services from Example 2, is shown in Fig. 2.*

Let $\pi \in \mathcal{S}^+ \cup \mathcal{S}^\omega$ be a finite or infinite sequence of states, and $i \in \mathbb{N}$. By $\pi(i)$ we denote the i–th state of π (starting from 0), by π_i its prefix of length i, and by π^i its suffix starting from $\pi(i)$. For each state $s = (q, n) \in \mathcal{S}$, its location q is denoted by $loc(s)$ and its time n by $\tau(s)$. If $\pi \in \mathcal{S}^+$ is a finite sequence of states, then its final state is denoted by π_F, and its *time length* is defined as $\tau(\pi) \coloneqq \tau(\pi_F)$. Moreover, by $locs(\pi)$ we denote the set of all locations present on π, i.e. $locs(\pi) = \{loc(\pi(k)) \mid k \in \mathbb{N}\}$. If $q \in \mathcal{Q}$ is a location, then by $\#_q(\pi)$ we denote the number of states of π whose location is q.

2.3 Semantics of TATL

Let $\mathcal{A} = (\mathbb{A}\mathrm{gt}, Act, \mathcal{Q}, \mathcal{AP}, \mathcal{L}, pr, t)$ be a TDCGS and $\mathcal{M}(\mathcal{A}) = (\mathcal{S}, \mathcal{E})$ its DTS. Moreover, let $a \in \mathbb{A}\mathrm{gt}$. A *strategy for* a is a function $\sigma_a \colon \mathcal{S}^+ \to Act$ such that, for each sequence $\pi \in \mathcal{S}^+$, we have $\sigma_a(\pi) \in pr_a(loc(\pi_F))$. That is, a strategy assigns choices saying how to continue to all possible histories of the game, where each history is a finite sequence of the visited *locations* together with their *time stamps*.

A *joint strategy* σ_A for a coalition $A \subseteq \mathbb{A}\mathrm{gt}$ is a tuple of strategies, one for each agent $a \in A$. Let $A = \{a_1, \dots, a_k\}$ for some $k \in \mathbb{N}$ and $\sigma_A = (\sigma_{a_1}, \dots, \sigma_{a_k})$ be a joint strategy for A. For each $i \in \mathbb{N}$ and $\pi \in \mathcal{S}^\omega$ we denote $\sigma_A(\pi_i) \coloneqq (\sigma_{a_1}(\pi_i), \dots, \sigma_{a_k}(\pi_i))$.

Definition 4 (Outcome) *The* outcome *of a joint strategy σ_A in a state $s \in \mathcal{S}$ is the set $out(s, \sigma_A) \subseteq \mathcal{S}^\omega$ such that $\pi \in out(s, \sigma_A)$ iff $\pi(0) = s$ and for each $i \in \mathbb{N}$ there is $act' \in pr_{\overline{A}}(loc(\pi(i)))$ such that $\mathcal{E}(\pi(i), act) = \pi(i+1)$, where $act|_A = \sigma_A(\pi_i)$ and $act|_{\overline{A}} = act'$.*

Intuitively, the outcome of a joint strategy σ_A in a state s is the set of all the possible paths that can occur when the agents of coalition A execute the strategy σ_A and each other agent, in $\overline{A}$, follows its own protocol. Since the action durations are strictly positive, the sequences of the outcome cannot form loops, i.e. for each $\pi \in out(s, \sigma_A)$ and $i, j \in \mathbb{N}$ if $\pi(i) = \pi(j)$, then $i = j$. In what follows, for technical convenience, instead of $out((q, 0), \sigma_A)$ we write $out(q, \sigma_A)$.

Definition 5 (Semantics of TATL**)** *Let $\mathcal{A} = (\mathbb{A}\mathrm{gt}, Act, \mathcal{Q}, \mathcal{AP}, \mathcal{L}, pr, t)$ be a* TDCGS*, $q \in \mathcal{Q}$ a location, $\varphi, \psi \in$* TATL*, and $A \subseteq \mathbb{A}\mathrm{gt}$ a set of agents. The semantics of* TATL *is given by the following clauses:*

- $\mathcal{A}, q \models \mathsf{p}$ *iff* $\mathsf{p} \in \mathcal{L}(q)$,
- $\mathcal{A}, q \models \neg\varphi$ *iff* $\mathcal{A}, q \not\models \varphi$,
- $\mathcal{A}, q \models \varphi \wedge \psi$ *iff* $\mathcal{A}, q \models \varphi$ *and* $\mathcal{A}, q \models \psi$,
- $\mathcal{A}, q \models \varphi \vee \psi$ *iff* $\mathcal{A}, q \models \varphi$ *or* $\mathcal{A}, q \models \psi$,
- $\mathcal{A}, q \models \langle\langle A \rangle\rangle X\varphi$ *iff there exists a joint strategy σ_A for A s.t. for each $\pi \in out(q, \sigma_A)$ we have $\mathcal{A}, loc(\pi(1)) \models \varphi$,*
- $\mathcal{A}, q \models \langle\langle A \rangle\rangle \varphi U_{\sim\eta} \psi$ *iff there exists a joint strategy σ_A for A s.t. for each $\pi \in out(q, \sigma_A)$ there exists $i \in \mathbb{N}$ such that $\tau(\pi(i)) \sim \eta$ and $\mathcal{A}, loc(\pi(i)) \models \psi$ and for all $0 \leq j < i$: $\mathcal{A}, loc(\pi(j)) \models \varphi$,*
- $\mathcal{A}, q \models \langle\langle A \rangle\rangle \varphi R_{\sim\eta} \psi$ *iff there exists a joint strategy σ_A for A s.t. for all $\pi \in out(q, \sigma_A)$ and for each $i \in \mathbb{N}$ such that $\tau(\pi(i)) \sim \eta$ we have $\mathcal{A}, loc(\pi(i)) \models \psi$ or there exists $0 \leq j < i$: $\mathcal{A}, loc(\pi(j)) \models \varphi$.*

The model can be omitted if it is clear from the context. As shown in Laroussinie et al. (2006b), the model checking problem for TATL is EXPTIME-complete, but it can be solved in polynomial time for its fragment $\mathrm{TATL}_{\leq,\geq}$. The EXPTIME-hardness (Laroussinie et al., 2006b, Theorem 13) follows from the reduction of the problem of the existence of a winning strategy in countdown games to the problem of model checking $\langle\langle A \rangle\rangle F_{=\eta}\varphi$ over TDCGS.

Example 4 (Online services continued) *Consider the* TDCGS $\mathcal{A}^2_{srv}$ *of Example 2 and Fig. 1. It is easy to see that $\mathcal{A}^2_{srv}, q_0 \models \langle\!\langle 1\rangle\!\rangle F_{\le 5}\mathsf{srv}_1 \wedge \neg\langle\!\langle 2\rangle\!\rangle F_{\le 5}\mathsf{srv}_2$ (the system is serviceable for agent 1 but not for agent 2). Moreover, agent 1 can perform the DoS attack: $\mathcal{A}^2_{srv}, q_0 \models \langle\!\langle 1\rangle\!\rangle G_{\le 5}\neg\mathsf{srv}_2$, the strategy being to request the access to the service at least twice in a row. On the other hand, agent 2 cannot do DoS unless the other agent is less patient: $\mathcal{A}^2_{srv}, q_0 \models \neg\langle\!\langle 2\rangle\!\rangle G_{\le 5}\neg\mathsf{srv}_1 \wedge \langle\!\langle 2\rangle\!\rangle G_{\le 2}\neg\mathsf{srv}_1$. Interestingly, when more agents are involved, any two of them can starve the other ones, e.g. $\mathcal{A}^k_{srv}, q_0 \models \langle\!\langle 1,2\rangle\!\rangle G_{\le t}\big(\bigwedge_{j=3,\dots,k}\neg\mathsf{srv}_j\big)$ for any $k \ge 3, t \in \mathbb{N}$ (the strategy being to always request access). Still, they cannot crash the service completely, which requires at least 3 agents to collaborate:*

$$\mathcal{A}^k_{srv}, q_0 \;\models\; \neg\langle\!\langle 1,2\rangle\!\rangle G_{\ge t}\big(\bigwedge_{j\in\mathbb{A}\mathrm{gt}}\neg\mathsf{srv}_j\big) \;\wedge\; \langle\!\langle 1,2,3\rangle\!\rangle G_{\ge t}\big(\bigwedge_{j\in\mathbb{A}\mathrm{gt}}\neg\mathsf{srv}_j\big).$$

Note that all the positive properties above can be obtained by strategies that make no use of time stamps. Thus, an interesting question arises: does the timing information increase the abilities of agents at all? We look closer into this issue in the next sections.

3. Hierarchy of Strategies and Semantic Variants of TATL

It was implicitly assumed in Laroussinie et al. (2006b) that the agents have perfect memory and can memorise not only the exact sequence of the past locations, but also their timing. On the other hand, Example 4 shows that such sophisticated capabilities are often redundant; agents can as well achieve their (timed) goals by using simpler strategies. In this section, we propose several semantic variants of TATL by restricting the allowed strategies. To this end, we introduce a hierarchy of strategy types.

The main types of strategies that we consider are timed vs. timeless on one hand, and perfect recall vs. memoryless on the other. We will also formally define *counting strategies*, including a very simple form of counting that we call *threshold counting*. As we prove in Section 4, whenever strict punctuality is not needed, most of the considered semantics are equivalent.

3.1 Strategy Types

Let $\mathcal{A} = (\mathbb{A}\mathrm{gt}, Act, \mathcal{Q}, \mathcal{AP}, \mathcal{L}, pr, t)$ be a TDCGS and $\mathcal{S} = \mathcal{Q} \times \mathbb{N}$ the underlying set of states. Following Brázdil et al. (2013), Bulling and Jamroga (2014), Schobbens (2004), we consider several variants of strategies for agents. To obtain a uniform view, we take the strategies from Section 2 as the starting point. The other classes of strategies are defined as subsets by imposing suitable constraints.

Definition 6 (Classes of strategies) *For each $a \in \mathbb{A}\mathrm{gt}$ we define five classes of strategies:*

(T) *A* timed perfect recall strategy for *a is a function $\sigma_a\colon \mathcal{S}^+ \to Act$ such that for each sequence $\pi \in \mathcal{S}^+$ we have $\sigma_a(\pi) \in pr_a(loc(\pi_F))$. I.e., these are exactly the strategies from Laroussinie et al. (2006b), that we already presented in Section 2. We denote the set of such strategies by Σ_{T}.*

(t) *A* timed memoryless strategy *is a strategy $\sigma_a \in \Sigma_{\mathrm{T}}$ such that, for each $\pi, \pi' \in \mathcal{S}^+$, if $\pi_F = \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$. We denote the set of such strategies by Σ_{t}.*

(R) *A* timeless perfect recall strategy *is a strategy $\sigma_a \in \Sigma_{\mathrm{T}}$ such that, for each $n \in \mathbb{N}$ and $\pi, \pi' \in \mathcal{S}^n$, if $loc(\pi(i)) = loc(\pi'(i))$ for all $0 \le i \le n$, then $\sigma_a(\pi) = \sigma_a(\pi')$. We denote the set of such strategies by Σ_{R}.*

(r) Timeless memoryless strategies *are defined analogously to timed memoryless strategies, and denoted by Σ_{r}.*

(#) *A* counting strategy *is a strategy $\sigma_a \in \Sigma_{\mathrm{T}}$ such that, for each $\pi, \pi' \in \mathcal{S}^+$, if $loc(\pi_F) = loc(\pi'_F)$ and $\#_{loc(\pi_F)}(\pi) = \#_{loc(\pi_F)}(\pi')$, then $\sigma_a(\pi) = \sigma_a(\pi')$. We denote the set of such strategies by $\Sigma_{\#}$.*

Intuitively, a perfect recall strategy selects an action based on the sequence of previous situations, whereas a memoryless strategy is a plan that looks only at the current situation. Similarly, a timed strategy can vary its choices depending on the time stamps, whereas timeless strategies cannot do so. Finally, counting strategies select actions by looking at the location and the number of its occurrences in the history so far. Notice that a timed memoryless strategy σ_a can be also defined by a function $\sigma'_a \colon \mathcal{S} \to Act$ such that $\sigma'_a(s) := \sigma_a(\pi)$ if $s = \pi_F$, for some π. Timeless perfect recall and timeless memoryless strategies can be defined analogously. Moreover, each counting strategy σ_a can be defined by a function $\sigma^{\#}_a \colon \mathcal{Q} \times \mathbb{N}_+ \to Act$ such that $\sigma^{\#}_a(q, n) := \sigma_a(\pi)$ if $q = loc(\pi_F)$ and $n = \#_{loc(\pi_F)}(\pi)$, for some π. Apart from general counting strategies, we will use some that are bounded in a special way, see below for the definition.

Definition 7 (Threshold strategies) *Let $n \in \mathbb{N}_+$. A counting strategy σ_a for a is called n–*threshold *iff for each $q \in \mathcal{Q}$ there exist actions $act_1, \ldots, act_{n+1} \in Act$, and integer intervals $I_1 = [1, i_1), I_2 = [i_1, i_2), \ldots, I_{n+1} = [i_n, \infty)$ such that for all $1 \le j \le n+1$: $\sigma^{\#}_a(q, k) = act_j$ if $k \in I_j$.*

We denote the set of n–threshold strategies by $\Sigma_{\#_n}$.

Intuitively, a counting strategy is n–threshold if for each location there exists a sequence of n thresholds, such that when the next threshold is exceeded, another action is used. Observe that once the final threshold is reached, the same action is executed whenever the location is visited.

Example 5 (Online services continued) *Consider strategies of agent 1 in the model of online services (Example 2). An example* timed perfect recall strategy *prescribes action req_1 whenever the system has stayed in q_0 for more than 15 time units in a row, and action $wait$ otherwise, i.e. $s_1(h) = req_1$ if $h = (q^0, t^0), \ldots, (q^n, t^n), \ldots, (q^{n+k}, t^{n+k})$, $q^n = \cdots = q^{n+k} = q_0$, $t^{n+k} - t^n > 15$, and otherwise $s_1(h) = wait$. An example* timed memoryless strategy *prescribes req_1 whenever the system is in q_0 and the time stamp is a prime number, and $wait$ otherwise. A* timeless perfect recall strategy *can specify to do req_1 if and only if the latest 15 locations have been q_0. A* timeless memoryless strategy *must fix a single action (e.g. req_1) for all states where the location is q_0. A* counting strategy *may prescribe req_1 iff the current location is q_0 and it has been visited a prime number of times. An example* 1–threshold strategy *specifies req_1 for the first 15 occurrences of q_0, and $wait$ after that.*

Remark 1 *Clearly, more expressive strategy types subsume the more restricted ones. The inclusions are graphically depicted in Figure 3.*

Again, a joint strategy for $A \subseteq \mathbb{A}\mathrm{gt}$ is a tuple of individual strategies, one per agent in A. The notion of the outcome of a strategy is the same as in Section 2.3.

$$\Sigma_{\mathrm{r}} = \Sigma_{\#_0} \subsetneq \Sigma_{\mathrm{t}} \subsetneq \Sigma_{\mathrm{T}}$$

$$\Sigma_{\#_0} \subsetneq \Sigma_{\#_1} \subsetneq \cdots \subsetneq \Sigma_{\#_{n-1}} \subsetneq \Sigma_{\#_n} \subsetneq \cdots \subsetneq \Sigma_{\#} \subsetneq \Sigma_{\mathrm{R}} \subsetneq \Sigma_{\mathrm{T}}$$

Figure 3: Relationships between strategy types

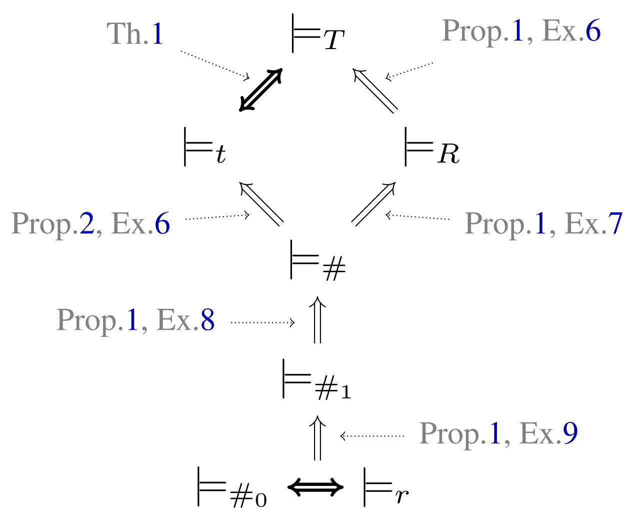

Figure 4: Correspondence between semantic variants for full TATL. The thick arrows indicate that the relationship holds for the whole language of TATL. The thin arrows indicate that the relationship holds only for the existential fragment of the language.

3.2 Semantic Variants of TATL

The classes of strategies give rise to different semantic variants in a natural way.

Definition 8 (Semantic variants for TATL**)** *We define the semantic relations $\models_Y$, parametrised by strategy types $Y \in \{\mathrm{T}, \mathrm{t}, \mathrm{R}, \mathrm{r}, \#\} \cup \{\#_n \mid n \in \mathbb{N}_+\}$, by replacing "$\models$" with "$\models_Y$" and "joint strategy" with "joint Y-strategy" in the clauses of Definition 5.*

Proposition 1 *For every $\mathcal{A}$, q, and formula $\varphi \in \exists$TATL, if $\Sigma_Y \subseteq \Sigma_Z$ then $\mathcal{A}, q \models_Y \varphi$ implies $\mathcal{A}, q \models_Z \varphi$.*

Proof. Straightforward induction on the structure of φ. □

We further investigate the relationships between the semantic variants in the subsequent sections.

4. Getting Your Timing Right Without the Clock

In the next two sections, we study the relationship between semantic variants of TATL defined by different restrictions on strategies, proposed in Section 3. This section presents the positive results; Section 5 shows which implications do not hold. A summary of all the results is presented in Fig. 4

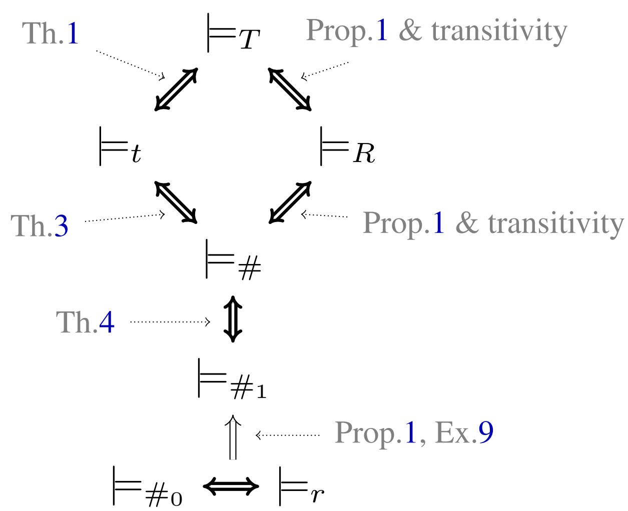

Figure 5: Correspondence between semantic variants for $\text{TATL}_{\leq,\geq}$. The thick arrows indicate that the relationship holds for the whole language of TATL. The thin arrows indicate that the relationship holds only for the existential fragment of the language.

and Fig. 5. A double-direction arrow indicates that the two semantics are equivalent. That is, given $\mathcal{A}$, q, and φ, we have that $\mathcal{A}, q \models_1 \varphi$ iff $\mathcal{A}, q \models_2 \varphi$. A single-direction arrow from $\models_1$ to $\models_2$ indicates that satisfaction in $\models_1$ implies satisfaction in $\models_2$. That is, given $\mathcal{A}$, q, and φ, we have that $\mathcal{A}, q \models_1 \varphi$ always implies $\mathcal{A}, q \models_2 \varphi$. We also point to the formal results that establish the relationships in the rest of paper.

The most striking conclusion is that, for a significant subset of TATL, keeping track of the timestamps does not increase the agents' abilities. For goals without punctuality constraints, strategies with counting up to one threshold per location are as good as strategies based on the whole sequence of locations and clock readings. In the rest of the section, we prove that the relationships are indeed as depicted in the graphs.

4.1 Two Technical Lemmas

We begin with two technical lemmas that will be useful in proving the subsequent results. The first lemma states that it suffices to consider only existential formulae in order to prove the equivalence of any two semantic variants of TATL. The second lemma considers the relationship between abilities based on timed memoryless strategies and counting strategies. Its main importance lies in the technique used to prove the lemma; we will later reuse it to prove other results, too. The readers not interested in technical details of the proofs are advised to skip this material, and proceed directly to Section 4.2. We have also provided figures that illustrate the proofs of Lemma 2 and Theorem 1, so that the reader can consult them while reading the proofs.

Lemma 1 *If* $\forall_{q\in\mathcal{Q}}\forall_{\varphi\in\exists\text{TATL}}\ (q \models_Y \varphi \iff q \models_Z \varphi)$*, then also* $\forall_{q\in\mathcal{Q}}\forall_{\varphi\in\text{TATL}}\ (q \models_Y \varphi \iff q \models_Z \varphi)$*, for any* $Y, Z \in \{\mathrm{T}, \mathrm{t}, \mathrm{R}, \mathrm{r}, \#\} \cup \{\#_n \mid n \in \mathbb{N}_+\}$.

The same applies to proving equivalence for semantics of $\text{TATL}_{\leq,\geq}$.

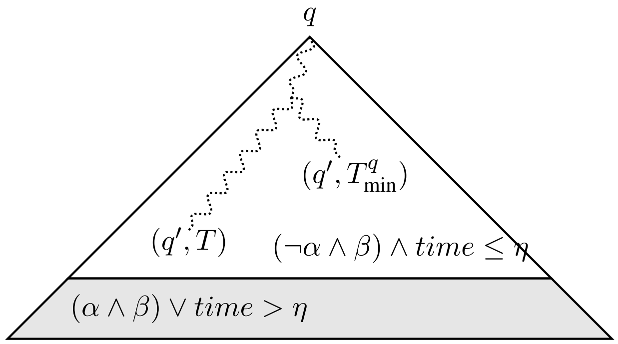

Figure 6: The outcome of σ_A enforcing $\alpha R_{\leq\eta}\beta$, cut after time exceeds η or β is released by α. For each $q' \in \mathcal{Q}$, $T^q_{\min}(\sigma_A, q')$ is the smallest time value among all T such that (q', T) is reached in the pictured segment.

Proof. The proof is by contradiction. Let us assume that $\forall_{q\in\mathcal{Q}}\forall_{\varphi\in\exists\text{TATL}}\ (q \models_Y \varphi \iff q \models_Z \varphi)$. Suppose that ψ is a shortest TATL formula such that either $(\mathcal{A}, q \models_Y \psi$ and $\mathcal{A}, q \not\models_Z \psi)$ or $(\mathcal{A}, q \not\models_Y \psi$ and $\mathcal{A}, q \models_Z \psi)$, i.e. ψ holds in q of $\mathcal{A}$ under one of the semantics, but not under the other. It is easy to see that if $\psi \in \mathcal{AP}$ or ψ is of the form $\neg\alpha, \alpha \wedge \beta$, or $\alpha \vee \beta$, then we immediately obtain a contradiction with the minimality of ψ. Therefore, let us suppose that $\psi \in \{\langle\!\langle A\rangle\!\rangle X\alpha, \langle\!\langle A\rangle\!\rangle \alpha U_\eta \beta, \langle\!\langle A\rangle\!\rangle \alpha R_\eta \beta\}$. Next, replace the subformulae α and β in ψ with new, fresh atomic propositions p_α and p_β, respectively, to obtain the formula $\psi_{\exists\text{TATL}}$ which is in $\exists$TATL. Define a new model, $\mathcal{A}_{\exists\text{TATL}}$, which is obtained from $\mathcal{A}$ in such a way that p_ξ is assigned to the states where ξ holds, for $\xi \in \{\alpha, \beta\}$. Now we have $\mathcal{A}, q \models_Y \psi$ iff $\mathcal{A}_{\exists\text{TATL}}, q \models_Y \psi_{\exists\text{TATL}}$ and $\mathcal{A}, q \models_Z \psi$ iff $\mathcal{A}_{\exists\text{TATL}}, q \models_Z \psi_{\exists\text{TATL}}$. To conclude, observe that by our assumption we have $\mathcal{A}_{\exists\text{TATL}}, q \models_Y \psi_{\exists\text{TATL}}$ iff $\mathcal{A}_{\exists\text{TATL}}, q \models_Z \psi_{\exists\text{TATL}}$. So, this contradicts our assumption that ψ holds in q of $\mathcal{A}$ under one of the semantics, but not under the other.

The proof for $\text{TATL}_{\leq,\geq}$ is analogous. □

As shown in Lemma 2 below, satisfaction of $\text{TATL}_{\leq,\geq}$ formulae under timed memoryless semantics implies satisfaction in the counting semantics. The techniques employed in the proof of the lemma will be reused in further parts of the paper.

Lemma 2 $\forall_{q\in\mathcal{Q}}\forall_{\varphi\in\exists\text{TATL}_{\leq,\geq}}(q \models_t \varphi \implies q \models_\# \varphi)$.

Proof. The proof follows by induction on the structure of φ, with $\varphi \in \exists\text{TATL}_{\leq,\geq}$. Let $\alpha, \beta \in \exists\text{TATL}_{\leq,\geq}$ and $\eta \in \mathbb{N}$. Let us assume $q \in \mathcal{Q}$ such that $q \models_t \varphi$. The proof is straightforward if $\varphi \in \mathcal{AP}$ or φ is of the form $\neg\alpha, \alpha \wedge \beta$, $\alpha \vee \beta$, or $\langle\!\langle A\rangle\!\rangle X\alpha$. Therefore, we consider only the remaining cases. For simplicity, we treat memoryless strategies as functions from the set of states to the set of actions.

The case of $\varphi = \langle\!\langle A\rangle\!\rangle \alpha U_{\leq\eta}\beta$ follows from Laroussinie et al. (2006b, Lemma 6), which shows that a perfect recall strategy for which each outcome satisfies $\alpha U_{\leq\eta}\beta$ can be modified in such a way that for the resulting strategy along each of its outcomes any location is present in at most one state until a state in which β holds is reached. This is sufficient for building a counting strategy that implements φ.

Let us now move to the case of $\varphi = \langle\!\langle A\rangle\!\rangle \alpha R_{\leq\eta}\beta$. Let σ_A be a timed memoryless strategy such that $\forall_{\pi\in out(q,\sigma_A)}\pi \models_t \alpha R_{\leq\eta}\beta$. Using σ_A we build a counting strategy $\sigma^\#_A$ such that

$\forall_{\pi \in out(q,\sigma_A^\#)} \pi \models_\# \alpha R_{\leq \eta} \beta$. Similarly to the previous case, the strategy $\sigma_A^\#$ associates the choice of an action with each location, thus the time component of a state is not relevant. For each $\pi \in out(q, \sigma_A)$ let $\pi^R_{\min}$ be the shortest prefix of π such that $\pi' \models_t \alpha R_{\leq \eta} \beta$ for each extension π' of $\pi^R_{\min}$ (its existence follows from the progress of time). Formally, let $\pi^R_{\min} = \pi_k$, where $k \in \mathbb{N}$ is the cutting point defined as the minimal index such that the following alternative is satisfied:

$$\tau(\pi(k)) \geq \eta \wedge \tau(\pi(k-1)) < \eta \tag{1}$$

$$\text{or } \big(\tau(\pi(k)) < \eta \wedge ((\pi(k) \models_t \alpha \wedge \beta) \wedge (k \geq 1 \implies (\pi(k-1) \models_t \neg\alpha \wedge \beta)))\big). \tag{2}$$

Recall that we assumed that $\alpha R_{\leq \eta} \beta$ holds along π. Eq. (1) covers the case where β is satisfied and not released by α along π until time η is reached, thus $\forall_{i<k} \pi(i) \models_t \beta$ and $\pi(k) \models_t \beta$ if $\tau(\pi(k)) = \eta$. Eq. (2) deals with the case where α releases β for the first time along π before time reaches η, therefore $\pi(k) \models_t \alpha \wedge \beta$ and $\forall_{i<k} \pi(i) \models_t \neg\alpha \wedge \beta$. To compute $\pi^R_{\min}$ we select the minimal index k for which any of the above cases is satisfied.
Now, for each $q' \in \mathcal{Q}$ we define:

$$T^q_{\min}(\sigma_A, q') := \min \left\{ t \mid \exists_{\pi \in out(q,\sigma_A)} \exists_{i \in \mathbb{N}}\ \pi^R_{\min}(i) = (q', t) \right\}$$

if the right side of the above is properly defined (i.e. min operates over a non-empty set) or $T^q_{\min}(\sigma_A, q') := 0$ otherwise. Intuitively, $T^q_{\min}(\sigma_A, q')$ is the smallest time value such that q' can be encountered by starting from q and following the strategy σ_A until the traversed path satisfies $\alpha R_{\leq \eta} \beta$. If such value does not exist for a given q', then a state with this location either cannot be reached from q or can be reached only along a path that fulfills the path condition $\alpha R_{\leq \eta} \beta$ over a prefix that does not visit q'. In this case we arbitrarily assign $T^q_{\min}(\sigma_A, q') = 0$. This construction is illustrated in Fig. 6. Now, we can define the counting strategy $\sigma_A^\#$ as follows:

$$\forall_{q' \in \mathcal{Q}} \forall_{n \in \mathbb{N}_+}\ \sigma_A^\#(q', n) := \sigma_A(q', T^q_{\min}(\sigma_A, q')).$$

Note that $\tau(s) \geq T^q_{\min}(\sigma_A, loc(s))$ for all states $s \in \mathcal{S}$ that appear along the paths of $out(q, \sigma_A^\#)$. We now prove that $\sigma_A^\#$ enforces $\alpha R_{\leq \eta} \beta$ along each outcome. Let us consider a path in $out(q, \sigma_A^\#)$. Along the path, $\sigma_A^\#$ follows the earliest decisions of σ_A that concern the location component of the current state, traversing states labelled with β that are not released by α. A state s encountered on the path escapes the part of the model that satisfies $\neg\alpha \wedge \beta$ only if σ_A was able to do so in $s' = (loc(s), T^q_{\min}(\sigma_A, loc(s)))$. This can happen only if the relevant successor of s' satisfies $\alpha \wedge \beta$ or its time component exceeds η. Both of these properties transfer to the successor of s on the considered path, which ends the proof of the case.

Next, we consider the case of $\varphi = \langle\langle A \rangle\rangle \alpha U_{\geq \eta} \beta$. As previously, let σ_A be a timed memoryless strategy such that $\forall_{\pi \in out(q,\sigma_A)} \pi \models_t \alpha U_{\geq \eta} \beta$ and for each $\pi \in out(q, \sigma_A)$ let $\pi^U_{\min}$ be the shortest prefix of π such that $\pi' \models_t \alpha U_{\geq \eta} \beta$ for each extension π' of $\pi^U_{\min}$. Formally, let $\pi^U_{\min} = \pi_k$, where $k \in \mathbb{N}$ is the cutting point defined as the minimal index such that:

$$\tau(\pi(k)) \geq \eta \wedge \pi(k) \models_t \beta \wedge (k \geq 1 \implies (\pi(k-1) \models_t \alpha)).$$

The cutting point is indeed obtained when β is reached for the first time after time η.

For each $q' \in \mathcal{Q}$ we define $T^{U,q}_{\min}(\sigma_A, q')$ as in the previous case, substituting $\alpha U_{\geq \eta} \beta$ for $\alpha R_{\leq \eta} \beta$. We also define:

$$T^{U,q}_{\max}(\sigma_A, q') := \max \left\{ t \mid \exists_{\pi \in out(q,\sigma_A)} \exists_{i \in \mathbb{N}}\ \pi^U_{\min}(i) = (q', t) \right\}$$

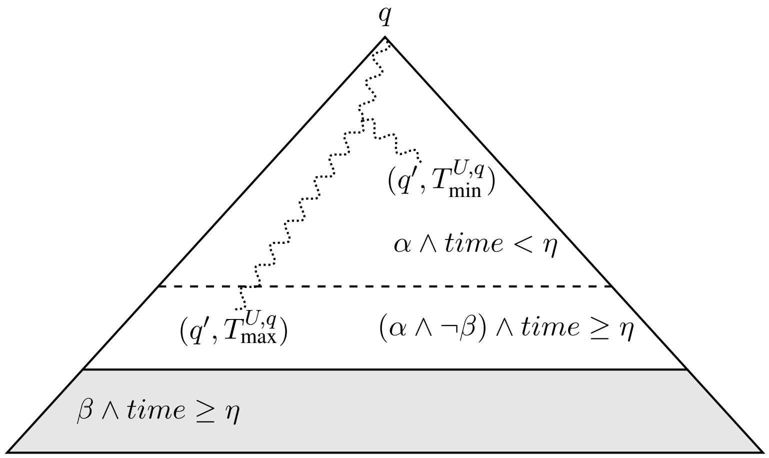

Figure 7: The outcome of σ_A enforcing $\alpha U_{\geq\eta}\beta$, cut after reaching a state where β holds and time is at least η. For each $q' \in \mathcal{Q}$, by $T^{U,q}_{\min}(\sigma_A, q')$ (resp. $T^{U,q}_{\max}(\sigma_A, q')$) we denote the smallest (resp. greatest) time value among all T such that (q', T) is reached in the pictured segment.

whenever the set over which $\max$ operates is non-empty and $T^{U,q}_{\max}(\sigma_A, q') := 0$ otherwise. This construction is illustrated in Fig. 7. Intuitively, $T^{U,q}_{\max}(\sigma_A, q')$ and $T^{U,q}_{\min}(\sigma_A, q')$ are the largest and smallest, respectively, time values such that q' can be encountered by starting from q and following the strategy σ_A until the traversed path satisfies $\alpha U_{\geq\eta}\beta$. We now define the counting strategy $\sigma^{\#}_A$ as follows:

$$\forall_{q' \in \mathcal{Q}} \forall_{1 \leq n < \eta}\ \sigma^{\#}_A(q', n) := \sigma_A(q', T^{U,q}_{\min}(\sigma_A, q')), \tag{3}$$

$$\forall_{q' \in \mathcal{Q}} \forall_{n \geq \eta}\ \sigma^{\#}_A(q', n) := \sigma_A(q', T^{U,q}_{\max}(\sigma_A, q')). \tag{4}$$

The strategy $\sigma^{\#}_A$ selects the actions of σ_A such α holds until we reach a time greater than or equal to η (the case of Eq. (3)) and then reaches a state in which β holds (the case of Eq. (4)). To observe that $\forall_{\pi \in out(q, \sigma^{\#}_A)} \pi \models_{\#} \alpha U_{\geq\eta}\beta$, notice that every path $\pi \in out(q, \sigma^{\#}_A)$ first proceeds by traversing along the states labelled with α and with time component smaller than η. In the initial fragment of π, strategy $\sigma^{\#}_A$ assigns to each state s the actions $\sigma_A(s')$, where s' is either time-earliest (via Eq. (3)) or time-latest (via Eq. (4)) state with the same location as s, selected from the bounded unfolding of $out(q, \sigma^{\#}_A)$, pictured in Fig. 7. Observe that the Eq. (3)-based part of strategy σ_A can only be executed a finite number of times and Eq. (4)-based part does not allow two states with the same location along a path. Thus, eventually the progression of π along states labelled with α must lead to reaching a state s'' such that $loc(s'') = loc(s''')$, for some $s''' \in \{\pi^{U}_{\min F} \mid \pi \in out(q, \sigma_A)\}$, i.e. the dark bottom of Fig. 7. Of course, β is true in s''', hence it is also true in s''. Now, we only need to check that the time component of s''' is at least η. This, however stems from the fact that, if s'' is reached while executing the Eq. (3)-based part of the strategy, then its time component is not smaller than the time component of s'''. On the other hand, if s'' is reached while executing the other part of the strategy, then the time must have exceeded η by definition (recall that time is discrete and each transition takes at least one time unit).

Remark 2 *The construction exhibited in our proof aims at correctness but not necessarily at efficiency. Consider the single-agent* TDCGS *in Fig. 8 and formula* $\varphi = \langle\langle A\rangle\rangle \alpha U_{\geq 10}\beta$. *Let* σ_A *be a*

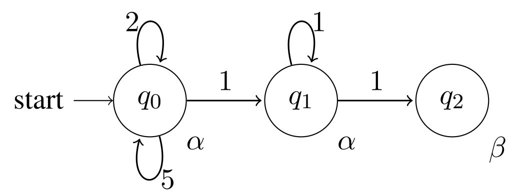

Figure 8: Exemplifying the complexity in the proof of $\varphi = \langle\langle A \rangle\rangle \alpha U_{\geq\eta}\beta$

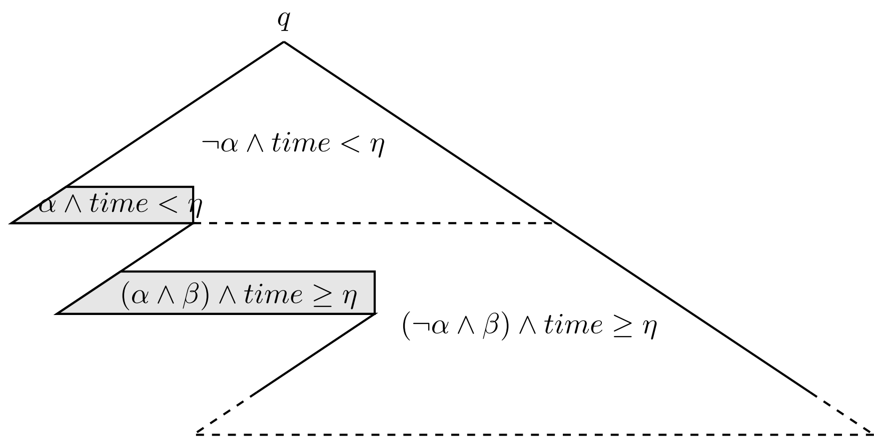

Figure 9: The outcome of σ_A enforcing $\alpha R_{\geq\eta}\beta$, cut after β is released by α.

timed memoryless strategy such that: (1) $\sigma_A(q_0, t)$ selects the loop labelled with 2 for $t < 2$, the loop labelled with 5 for $t = 2$, and the transition to q_1 if $t > 2$; (2) $\sigma_A(q_1, t)$ chooses to loop in q_1 if $t < 9$ and to move to q_2 if $t \geq 9$. As it can be readily seen, following σ_A enforces a single path and proves that $q_0 \models_t \varphi$. Moreover, a "natural" counting strategy built from σ_A is to select the 2-labelled transition when in q_0 for the first time, the 5-labelled transition for the second time, and the transition to q_1 for the third time. Then in q_1, the strategy would be to firstly follow the loop and then to move to q_2. However, using our construction, the strategy built from σ_A loops nine times on q_0 via transition labelled with 2 before moving to q_1 where it loops nine times, again, followed by transition to q_2.

Remark 3 *The considered case is the only one where a 1–threshold strategy is constructed. In all the remaining cases 0–threshold strategies are built.*

We now move to the final case of the proof, i.e. $\varphi = \langle\langle A \rangle\rangle \alpha R_{\geq\eta}\beta$. Let σ_A be a memoryless strategy such that $\forall_{\pi \in out(q,\sigma_A)} \pi \models_t \alpha R_{\geq\eta}\beta$. For each $\pi \in out(q, \sigma_A)$ let $T^R_{\text{release}}(\pi)$ be the time of the earliest appearance of α along the path (or ∞, if α is not visited). Formally: $T^R_{\text{release}}(\pi) = \min\big\{\{\tau(\pi(k)) \mid k \in \mathbb{N} \wedge \pi(k) \models_t \alpha\}, \infty\big\}$. Let $q' \in \mathcal{Q}$. Depending on the structure of π we define $T^R_{\text{encounter}}(\pi, q')$ as: (1) the time of the latest encounter of q' along π, up to the moment of reaching α; (2) the time of the earliest encounter of q' along π, after exceeding η, when α never holds along π; (3) the arbitrary value of 0 if q' is visited along π only after reaching α or not at all. Formally we have:

$$T^R_{\text{encounter}}(\pi, q') = \begin{cases} \max\{\tau(\pi(k)) \mid loc(\pi(k)) = q' \wedge \tau(\pi(k)) \leq T^R_{\text{release}}(\pi)\} \\ \qquad\qquad \text{if } T^R_{\text{release}}(\pi) < \infty \wedge q' \in locs(\pi) \\ \min\{\tau(\pi(k)) \mid loc(\pi(k)) = q' \wedge \tau(\pi(k)) \geq \eta\} \\ \qquad\qquad \text{if } T^R_{\text{release}}(\pi) = \infty \wedge q' \in locs(\pi) \\ 0 \qquad\qquad \text{otherwise} \end{cases}$$

The final building block of our counting strategy is defined as:

$$T^{R,q}_{\text{sel}}(\sigma_A, q') = \max_{\pi \in out(q,\sigma_A)} \{T^R_{\text{encounter}}(\pi, q')\}$$

Let us define the counting strategy $\sigma^{\#}_A$ as follows:

$$\forall_{q' \in \mathcal{Q}} \forall_{n \in \mathbb{N}_+}\ \sigma^{\#}_A(q', n) := \sigma_A(q', T^{R,q}_{\text{sel}}(\sigma_A, q'))$$

We now show that for all $\forall_{\pi \in out(q,\sigma^{\#}_A)} \pi \models_{\#} \alpha R_{\geq\eta}\beta$. Let $\pi \in out(q, \sigma^{\#}_A)$ and π' be a finite prefix of π. Observe that if $\pi' \not\models_{\#} \alpha R_{\geq\eta}\beta$, then $\tau(\pi'_F) \geq \eta$ implies that $T^{R,q}_{\text{sel}}(\sigma_A, \pi'_F) \geq \eta$. This in turn means that $\pi'_F \models_t \beta$. Therefore, if β is not released by α over π before exceeding time η, then either β holds along π until infinity or eventually $\pi' \not\models_{\#} \alpha R_{\geq\eta}\beta$ for a prefix π' of π. Let us refer to Fig. 9 to provide more intuitions. Strategy $\sigma^{\#}_A$ selects actions by copying the relevant choices from the pictured segment of outcome of σ_A. If the time of a current state along a path enforced by $\sigma^{\#}_A$ reaches or exceeds η, then the strategy copies actions of σ_A associated with those states of the outcome whose time component exceeds η. All these states satisfy β and the only way to escape is to reach the darker fragment where $\alpha \wedge \beta$ holds. This concludes the proof of the case and the lemma. □

4.2 Time Makes Memory Obsolete

We now establish that the memoryless and perfect recall timed semantics of TATL are equivalent, similarly to the well-known result for standard (timeless) ATL (Alur et al., 2002).

Theorem 1 (Perfect recall = Memoryless) $\forall_{q \in \mathcal{Q}} \forall_{\varphi \in \text{TATL}} (q \models_T \varphi \iff q \models_t \varphi)$.

Proof. First, observe that by Lemma 1 it suffices to prove that $q \models_T \varphi \iff q \models_t \varphi$ for all $\varphi \in \exists\text{TATL}$. Moreover, we can focus only on $q \models_T \varphi \implies q \models_t \varphi$, as the other direction easily follows from the definition.

Let $\varphi = \langle\langle A \rangle\rangle \alpha U_{=\eta}\beta$ and $\sigma_A \in \Sigma_{\text{T}}$ be a strategy such that $\alpha U_{=\eta}\beta$ holds along each path from $out(q, \sigma_A)$. Let $\pi' \in \{\pi_k \mid \pi \in out(q, \sigma_A) \wedge k \in \mathbb{N} \wedge \tau(\pi(k)) \leq \eta\}$ be such that $\pi'_F = (q', n)$, for some $q' \in \mathcal{Q}$ and $n \leq \eta$. We are targeting (q', n) and plan to build from σ_A a new strategy that unifies all the selections that reach this state. Notice that $q' \models_T \langle\langle A \rangle\rangle \alpha U_{=(\eta-n)}\beta$. We now alter σ_A to obtain $\widehat{\sigma_A}^{\pi'}$ as follows:

$$\widehat{\sigma_A}^{\pi'}(\pi) = \begin{cases} \sigma_A(\pi'\pi^2) & \text{if } \pi = \pi^1\pi^2 \text{ and } \pi^1_F = (q', n) \\ \sigma_A(\pi) & \text{otherwise} \end{cases}$$

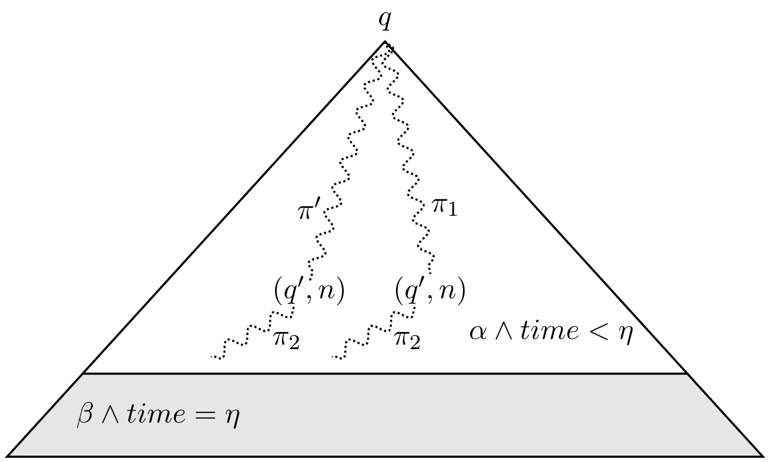

Figure 10: Timed memoryful strategy σ_A enforcing $\alpha U_{=\eta}\beta$ is unified over (q', n). A finite sequence π' reaching (q', n) is fixed. A choice of action made for $\pi_1\pi_2$ in the unified strategy is the same as after traversing $\pi'\pi_2$ under σ_A.

for all $\pi \in \mathcal{S}^+$. Intuitively (see Fig. 10), in $\widehat{\sigma_A}^{\pi'}$ we replace the current path in the memory by π' once (q', n) is encountered while travelling from q. Observe that $\alpha U_{=\eta}\beta$ holds along each path from $out(q, \widehat{\sigma_A}^{\pi'})$.

We can now iteratively update the strategy σ_A in accordance with the following: $\forall q' \in \mathcal{Q}, \forall n < \eta : \exists \pi' \in out(q, \sigma_A), \pi'_F = (q', n) \implies \sigma_A := \widehat{\sigma_A}^{\pi'}$. Note that this process cannot take more than $|\mathcal{Q}| \cdot \eta$ steps.

Finally, we alter the updated strategy σ_A by assigning to each finite sequence $\pi \in \mathcal{S}^+$ such that $\tau(\pi_F) \geq \eta$ any fixed selection of actions that only depends on $loc(\pi_F)$. Thus, we have built a timed memoryless strategy such that $\alpha U_{=\eta}\beta$ holds along each of its outcomes from q.

As the same technique can be used to deal with the case of $\varphi = \langle\!\langle A \rangle\!\rangle \alpha R_{=\eta}\beta$, we conclude the proof.

□

Additionally, we get the following as a corollary.

Proposition 2 $\forall_{q \in \mathcal{Q}} \forall_{\varphi \in \exists\text{TATL}} (q \models_\# \varphi \implies q \models_t \varphi)$.

Proof. Let φ be a formula of $\exists$TATL. By Remark 1 and Proposition 1, we have that $q \models_\# \varphi \implies q \models_R \varphi \implies q \models_T \varphi$. Moreover, by Theorem 1, we have $q \models_T \varphi \implies q \models_t \varphi$, which concludes the proof. □

4.3 Too Much Time Will Not Help You

As it turns out, time is also of limited importance, once it exceeds a certain value. To see this, we need the following definition.

Definition 9 (n–time–agnostic strategy) *Let $n \in \mathbb{N}$. A joint memoryless strategy σ_A for a coalition $A \subseteq \mathbb{A}\text{gt}$ is* n–time–agnostic *iff for each $s, s' \in \mathcal{S}$ such that $loc(s) = loc(s')$ if $\tau(s) \geq n$ and $\tau(s') \geq n$, then $\sigma_A(s) = \sigma_A(s')$. We denote the set of such strategies by $\Sigma_{t(n)}$.*

Intuitively, a strategy is n–time–agnostic if it does not track the passage of time once it exceeds n. For example, consider the model from Example 2 and strategy σ_1 for agent 1 such that $\sigma_1(q_0, k) = req_1$ for $k \in \{0, 1, 2\}$ and $\sigma_1(q_0, k) = wait$ for $k \geq 3$. This strategy is 3–time–agnostic and allows agent 1 to enforce three consecutive visits in q_1 after which the location is avoided indefinitely.

The memoryless n–time–agnostic semantics of TATL, denoted by $\models_{\mathrm{t}(n)}$, is obtained similarly to the previous semantics (cf. Definition 5), with strategies from $\Sigma_{\mathrm{t}(n)}$ used instead.

Theorem 2 *Let $\varphi \in$ TATL, and $c_\varphi \in \mathbb{N}$ be the maximal integer occurring in φ (with $c_\varphi = 0$ if φ contains no time constraints). Then, $\forall_{q \in \mathcal{Q}}\ (q \models_t \varphi \iff q \models_{\mathrm{t}(c_\varphi)} \varphi)$.*

Proof. The proof follows by contradiction.[1] Let φ be a formula of minimal size among the formulae whose truth value in some location q is different under timed memoryless semantics than under $\models_{\mathrm{t}(c_\varphi)}$. That is, we assume that $q \models_t \varphi$ and $q \not\models_{\mathrm{t}(c_\varphi)} \varphi$, or $q \not\models_t \varphi$ and $q \models_{\mathrm{t}(c_\varphi)} \varphi$. It is not difficult to see that the negation and the conjunction cannot be the main operators of φ, as this would conflict with the minimality of the formula. We can thus assume that $\varphi = \langle\langle A \rangle\rangle \gamma$ for some $A \subseteq \mathbb{A}\mathrm{gt}$. Moreover, a c_φ–time–agnostic strategy is a timed memoryless strategy, hence $q \models_{\mathrm{t}(c_\varphi)} \varphi$ implies $q \models_t \varphi$. We can thus assume that $q \models_t \varphi$ and $q \not\models_{\mathrm{t}(c_\varphi)} \varphi$.

Let $\gamma \in \{X\alpha, \alpha U_{\sim\eta}\beta, \alpha R_{\sim\eta}\beta\}$, where $\sim\, \in \{\leq, =\}$ and $\eta \in \mathbb{N}$. Let σ_A be a strategy under which φ holds in q. As there is no need to track the passage of time after it exceeds η, formula φ holds in q under any extension of $\sigma_{A|\{s \in \mathcal{S} \mid \tau(s) \leq \eta\}}$. In particular we can use c_φ–time–agnostic extension, because by definition we have $\eta \leq c_\varphi$. Therefore, we necessarily have $q \models_{\mathrm{t}(c_\varphi)} \varphi$, which contradicts the aforementioned assumption. Hence, we can assume that $\sim\, = \,\geq$. At this stage we inferred that $\gamma \in \{\alpha U_{\geq\eta}\beta, \alpha R_{\geq\eta}\beta\}$, for some $\eta \in \mathbb{N}$.

Let us consider the case of $\varphi = \langle\langle A \rangle\rangle \alpha U_{\geq\eta}\beta$, where $\eta \in \mathbb{N}$, The proof utilizes elements of the proof of the same subcase of Lemma 2. Let σ_A be a strategy such that $\alpha U_{\geq\eta}\beta$ holds along each path of $out(q, \sigma_A)$. Recall that for each $q' \in \mathcal{Q}$ we define $T^{U,q}_{\max}(\sigma_A, q')$ as the greatest time value $\tau(s)$ among all the states s such that $loc(s) = q'$, encountered while following any outcome of strategy σ_A from $(q, 0)$ before reaching a state where β holds and its time component reaches or exceeds η (cf. Fig. 7). If there is no such state, then $T^{U,q}_{\max}(\sigma_A, q') = 0$. We define the c_φ–time–agnostic strategy σ'_A as follows:

$$\forall_{\{s \in \mathcal{S} \mid \tau(s) < \eta\}}\ \sigma'_A(s) \coloneqq \sigma_A(s), \tag{5}$$

$$\forall_{\{s \in \mathcal{S} \mid \tau(s) \geq \eta\}}\ \sigma'_A(s) \coloneqq \sigma_A(loc(s).T^{U,q}_{\max}(\sigma_A, loc(s))). \tag{6}$$

Strategy σ'_A enforces in $(q, 0)$ (via Eq. (5)) the same initial fragments of its outcome as σ_A (hence traversing along states labelled with α), until time reaches or exceeds η. After this moment it switches (via Eq. (6)) to consistently assigning to each state s the actions $\sigma_A(s')$ of the latest appearance along any $\pi^U_{\min}$ such that $\pi \in out(q, \sigma_A)$ of a state s' satisfying $loc(s) = loc(s')$. Similarly to the proof of the same subcase of Lemma 2 we can now prove that once the second part of definition of strategy σ'_A is activated, no location can appear twice before reaching a state labelled with β. Given the fact that the number of locations is finite and the path keeps traversing states that satisfy α, this concludes the proof of the case. To provide a more intuitive explanation of this proof, refer to Fig. 7 and observe that every outcome σ'_A starts by following the same paths as enforced by σ_A (the upper part of the triangle), until it reaches the dashed line that depicts time η. Afterwards, it

1. A direct proof is also possible, but it is less readable due to the larger number of subcases.

consistently selects the same actions as the latest choice of σ_A in the lower part of the triangle. From the fact that the latest choices are made, at this stage σ'_A cannot visit two different states sharing the same location, so it eventually needs to enter the bottom of the triangle.

We have one case left to consider, i.e. $\varphi = \langle\langle A\rangle\rangle \alpha R_{\geq\eta}\beta$. It is simple, however, since $q \models_t \varphi$ implies the existence of a counting strategy $\sigma^{\#}_A$ that enforces φ in q and the strategy built in the proof of Lemma 2 for this subcase assigns to each state s a selection of actions based on its location component $loc(s)$, ignoring the time. We have therefore reached a contradiction and the end of the proof. □

4.4 Time is Counting (for Unpunctual Agents)

The following theorem states that in TATL$_{\leq,\geq}$, i.e. when strict punctuality is not needed, the strategies based on observing the current time and based on tracking the number of visits in the current location are of the same expressive power.

Theorem 3 (Time is Counting) $\forall_{q\in\mathcal{Q}}\forall_{\varphi\in \text{TATL}_{\leq,\geq}}(q \models_t \varphi \iff q \models_\# \varphi)$.

Proof. Let $\varphi \in \exists\text{TATL}_{\leq,\geq}$ and assume that $q \models_\# \varphi$. By Remark 1 and Proposition 1, each counting strategy is a timed perfect recall strategy, hence we also have $q \models_T \varphi$. By Theorem 1 this implies $q \models_t \varphi$. Combining this with Lemma 2 we obtain $q \models_t \varphi \iff q \models_\# \varphi$. It now suffices to apply Lemma 1 to extend this result to TATL$_{\leq,\geq}$. □

The proof of Lemma 2 revealed the simplicity of the structure of strategies that implement formulae in TATL$_{\leq,\geq}$. Namely, we presented how to transform each counting strategy implementing a strategic formula $\varphi \in \text{TATL}_{\leq,\geq}$ such that it still enforces φ and assigns to each location at most two actions. We thus have the following as a corollary.

Theorem 4 (Simple Counting Is Enough) $\forall_{q\in\mathcal{Q}}\forall_{\varphi\in \text{TATL}_{\leq,\geq}}(q \models_\# \varphi \iff q \models_{\#_1} \varphi)$.

In the next section we show that, in the other cases, the semantics being considered are not equivalent.

5. Some Things that Make the Difference

We have just showed that TATL for agents with full perfect recall is in fact equivalent to the semantics based on memoryless timed strategies. Moreover, for TATL$_{\leq,\geq}$, all the semantic variants – except for the one based on untimed memoryless strategies – collapse to $\models_{\#_1}$. Even the abilities of agents with perfect recall of locations and timestamps can be verified by looking at 1-threshold strategies. In this section, we prove that the semantic hierarchy does not collapse further. To this end, we present a series of examples showing that the converse of the implications depicted in Fig. 4 and Fig. 5 do not hold.

5.1 Time Matters for Punctual Agents

We start by showing that Theorem 3 cannot be extended to full TATL. That is, there exist $\mathcal{A}$, q, and $\varphi \in$ TATL, such that $\mathcal{A}, q \models_t \varphi$ but $\mathcal{A}, q \not\models_\# \varphi$. The same example can be used to show that $\models_T$ does not imply $\models_R$.

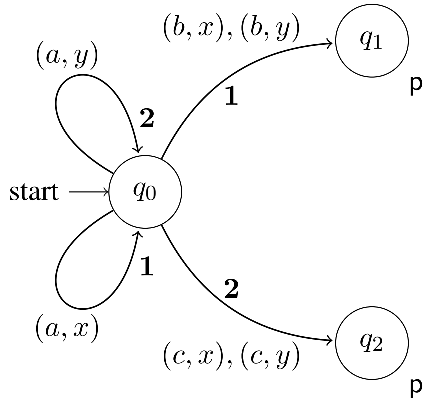

Figure 11: Punctuality needs clocks

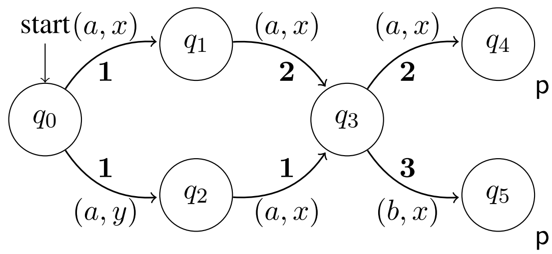

Figure 12: Timeless perfect recall is stronger than counting

Example 6 *Consider the* TDCGS *presented in Fig. 11. The model contains three locations:* q_0, q_1*, and* q_2*, of which* q_1 *and* q_2 *are labelled by* p*. In* q_0*, agent 1 can select one of its three actions, while agent 2 selects one of its two actions. The protocols are thus as follows:* $pr(1, q_0) = \{a, b, c\}$*, and* $pr(2, q_0) = \{x, y\}$*. To see that* $q_0 \models_t \langle\langle 1 \rangle\rangle F_{=5}\mathsf{p}$*, observe that agent 1 can follow a simple strategy of enforcing the loops in* q_0 *until the time reaches either 3 or 4, depending on the response of the second agent. Then, agent 1 selects the action* c *or* b*, respectively, to reach one of the states labelled with* p *precisely at time 5. It is easy to see that* $q_0 \not\models_\# \langle\langle 1 \rangle\rangle F_{=5}\mathsf{p}$*, as there is no counting strategy that allows to decide when to leave* q_0 *for a location labelled with* p *and which branch to take in order to reach the target in 5 time units. It is also not difficult to observe that* $q_0 \not\models_R \langle\langle 1 \rangle\rangle F_{=5}\mathsf{p}$.

5.2 Smart Agents Recall, and Not Only Count

Next, we show that $\models_R$ does not imply $\models_\#$ for the full language of TATL.

Example 7 *Consider the* TDCGS *presented in Fig. 12. Observe that agent 2 controls the branch selection in the location* q_0 *while agent 1 selects the branch to take in* q_3*. Under timeless perfect recall, agent 1 that started in* q_0 *knows in* q_3 *whether it has visited the location* q_1 *or* q_2*. The agent is thus able to select the next action in such a way that* p *is reached in precisely* 5 *time units, i.e.* $q_0 \models_R \langle\langle 1 \rangle\rangle F_{=5}\mathsf{p}$*. However,* $q_0 \not\models_\# \langle\langle 1 \rangle\rangle F_{=5}\mathsf{p}$.

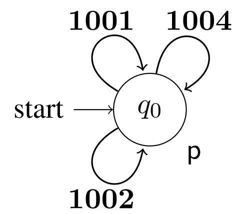

Figure 13: Three distinct actions needed in q_0 to reach p in exactly 3007 time units

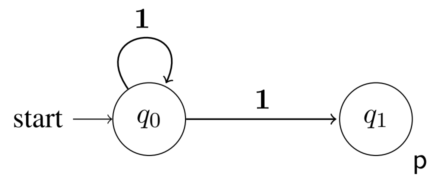

Figure 14: Two distinct actions are needed in q_0 to reach p in at least 2 time units

5.3 Punctual Agents Have to Count More

As we have shown, 1–threshold strategies are sufficient to implement any property expressed in $\text{TATL}_{\leq,\geq}$. The following example presents a case where at least 2–threshold strategies are required for full TATL.

Example 8 *Consider the one-agent* TDCGS *in Fig. 13. A simple 2–threshold strategy of firing each action one after another and repeating the final one ad infinitum shows that* $q_0 \models_{\#_2} \langle\langle 1\rangle\rangle F_{=3007}\mathsf{p}$. *However, it can be verified by hand calculations that* 3007 *cannot be represented as a natural canonical combination (i.e. a linear combination with non-negative natural coefficients) of* $1001, 1002$ *and* 1004, *other than* $1001 + 1002 + 1004$. *We therefore have* $q_0 \not\models_{\#_k} \langle\langle 1\rangle\rangle F_{=3007}\mathsf{p}$ *for any* $k \in \{0, 1\}$.

This result can be extended to an arbitrary value $n \in \mathbb{N}_+$. To this end, define a TDCGS $\mathcal{A}_n^{eq}$ that contains a single state q_0 labelled with p and n transition loops $act_1^n, \ldots, act_n^n$ such that $\tau(act_i^n) = 10^n + 2^i$ for all $0 \leq i < n$. It can be proved using elementary arguments that $\sum_{i=0}^{n-1} \tau(act_i^n)$ can be obtained only by this single canonical combination of time values. Therefore, for each $n \in \mathbb{N}_+$ we have $\mathcal{A}_n^{eq} \models_{\#_{n-1}} q_0\langle\langle 1\rangle\rangle F_{=\sum_{i=0}^{n-1} 10^n+2^i}\mathsf{p}$ and $\mathcal{A}_n^{eq} \not\models_{\#_k} q_0\langle\langle 1\rangle\rangle F_{=\sum_{i=0}^{n-1} 10^n+2^i}\mathsf{p}$ for all $0 \leq k < n-1$.

5.4 Untimed Memoryless Agents Are Dumb Beyond Count

Finally, the following example illustrates that a single action per location is not sufficient to implement properties expressed in $\text{TATL}_{\leq,\geq}$, i.e. $\models_{\#_1}$ does not imply $\models_{\#_0}$.

Example 9 *Consider the two-state* TDCGS *in Fig. 14. Observe that* $q_0 \models_{\#_1} \langle\langle 1\rangle\rangle F_{\geq 2}\mathsf{p}$ *iff the agent decides first to traverse the loop in* q_0 *and then to move to* q_1. *It is straightforward to see that this 1–threshold strategy cannot be reduced to a 0–threshold strategy.*

5.5 Summary

This concludes our comparative analysis of semantic variants of TATL, based on different notions of strategic play. The relationships were already summarized in Fig. 4 and Fig. 5. The most important conclusion is that timed perfect recall strategies provide no extra ability over timed memoryless strategies. Moreover, all the timed strategies collapse to simple counting strategies for goals without equality constraints. This is somewhat similar to the classical results for standard ATL (Alur et al., 2002; Schobbens, 2004), and suggests that $\mathrm{TATL}_{\leq,\geq}$ is only slightly more expressive that untimed ATL.

While we do not provide the detailed complexity analysis, we can infer that model checking of $\mathrm{TATL}_{\leq,\geq}$ over the counting semantics is in PTIME. This follows immediately from Theorem 3 and the fact that model checking of $\mathrm{TATL}_{\leq,\geq}$ over the timed semantics is in PTIME (Laroussinie et al., 2006b). Moreover, we conjecture that model checking of the entire TATL over the counting semantics is in PSPACE w.r.t. the product of the size of statespace, the number of bits needed to represent the largest number present in the verified formula, and the length of the formula. This is in contrast to Laroussinie et al. (2006b) where it is shown that the problem is EXPTIME-complete for the timed semantics. Our intuitions base on Theorem 2 and standard techniques for implementing non-deterministic bit counters (Sipser, 1997).

6. Conclusions and Future Work

In this paper, we investigate TATL which is a natural extension of Alternating-time Temporal Logic with discrete time. We propose and study a hierarchy of strategy types, leading to a hierarchy of semantic variants that differ in the assumptions about the agents' mental capabilities. In particular, we introduce several variants of *counting strategies*, where the agents' decisions are based on the number of visits at locations encountered along the execution path. As we have shown, those counting strategies play a key role in defining the expressivity $\mathrm{TATL}_{\leq,\geq}$, i.e. the logic of timed strategic ability for goals without strict punctuality constraints. In fact, it is sufficient to consider 1-threshold strategies that use only two actions per location to reason about any $\mathrm{TATL}_{\leq,\geq}$ property, even for agents with perfect recall of the past.

If equality constraints are allowed in a formula, then the picture is much more intricate. Still, we prove that the full power of perfect recall strategies with timestamps is never needed, and one can use timed memoryless strategies instead.

The work opens several interesting paths for future research. First, the strict coupling of strategic and temporal modalities in TATL can be relaxed to obtain $\mathrm{TATL}^{\star}$, similarly to $\mathrm{ATL}^{\star}$. The correspondence between various timed and counting semantics of $\mathrm{TATL}^{\star}$ is definitely worth investigating, especially since in $\mathrm{TATL}^{\star}$ equality can be expressed using inequality constraints. Secondly, TATL deals only with agents that have perfect information about the current state of the environment. Following Schobbens (2004), Jamroga and van der Hoek (2004), we would like to study the consequences of introducing indistinguishability relations to TDCGS. We expect that this should significantly influence the complexity and decidability of the model checking problem. Another natural extension of TATL consists in a parametric variant of the logic, with parameters added to the formulae (Bruyère, Dall'olio, & Raskin, 2008), the models (Alur, Henzinger, & Vardi, 1993), or both (Bruyère & Raskin, 2007). Our preliminary analysis suggests that the decidability of the associated *emptiness problem*, i.e. the existence of parameter valuations under which a given formula holds, depends on both the formula and the choice of place for parameter injection.

Finally, we plan to investigate the applications of Timed ATL in the context of cyber-security, where the lack of a timely response is often the culprit for the system's failure.

Acknowledgements

We would like to thank Łukasz Mikulski for his help in extending Example 8 to the general case. W. Jamroga, W. Penczek, and M. Knapik acknowledge the support of the National Centre for Research and Development (NCBR), Poland, and the Luxembourg National Research Fund (FNR), under the PolLux/FNR-INTER project VoteVerif (POLLUX-IV/1/2016). W. Jamroga is also affiliated with the Interdisciplinary Centre for Security, Reliability, and Trust, SnT, University of Luxembourg. This work was also supported by the CNRS/PAS project PARTIES and the invited professorship programme of University Paris 13.

References

Alechina, N., Logan, B., Nguyen, H., & Rakib, A. (2010). Resource-bounded alternating-time temporal logic. In *Proceedings of AAMAS*, pp. 481–488.

Alechina, N., Bulling, N., Demri, S., & Logan, B. (2018). On the complexity of resource-bounded logics. *Theor. Comput. Sci.*, *750*, 69–100.

Alur, R., de Alfaro, L., Grossu, R., Henzinger, T., Kang, M., Kirsch, C., Majumdar, R., Mang, F., & Wang, B.-Y. (2001). jMocha: A model-checking tool that exploits design structure. In *ICSE*, pp. 835–836. IEEE Computer Society Press.

Alur, R., Henzinger, T., Mang, F., Qadeer, S., Rajamani, S., & Tasiran, S. (1998). MOCHA: Modularity in model checking. In *CAV*, Vol. 1427 of *Lecture Notes in Computer Science*, pp. 521–525. Springer.

Alur, R., Henzinger, T. A., & Kupferman, O. (1997). Alternating-time Temporal Logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 100–109. IEEE Computer Society Press.

Alur, R., Bernadsky, M., & Madhusudan, P. (2004). Optimal reachability for weighted timed games. In *ICALP*, Vol. 3142 of *Lecture Notes in Computer Science*, pp. 122–133. Springer.

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

Alur, R., Henzinger, T. A., & Vardi, M. Y. (1993). Parametric real-time reasoning. In *STOC*, pp. 592–601. ACM.

André, É., Knapik, M., Jamroga, Wojciech Penczek, W., & Petrucci, L. (2017). Timed ATL: Forget memory, just count. In Larson, K., Winikoff, M., Das, S., & Durfee, E. (Eds.), *AAMAS*, pp. 1460–1462. ACM.

Boureanu, I., Jones, A. V., & Lomuscio, A. (2012). Automatic verification of epistemic specifications under convergent equational theories. In *AAMAS*, pp. 1141–1148. ACM.

Boureanu, I., Kouvaros, P., & Lomuscio, A. (2016). Verifying security properties in unbounded multiagent systems. In *AAMAS*, pp. 1209–1217. ACM.

Bouyer, P., D'Souza, D., Madhusudan, P., & Petit, A. (2003). Timed control with partial observability. In *CAV*, Vol. 2725 of *Lecture Notes in Computer Science*, pp. 180–192. Springer.

Brázdil, T., Forejt, V., Krcál, J., Kretínský, J., & Kucera, A. (2013). Continuous-time stochastic games with time-bounded reachability. *Information and Computation*, *224*, 46–70.

Brihaye, T., Laroussinie, F., Markey, N., & Oreiby, G. (2007). Timed concurrent game structures. In *CONCUR*, Vol. 4703 of *Lecture Notes in Computer Science*, pp. 445–459. Springer.

Broersen, J. M., Dignum, F., Dignum, V., & Meyer, J. C. (2004). Designing a deontic logic of deadlines. In *Deontic Logic in Computer Science, Proceedings of DEON*, pp. 43–56.

Bruyère, V., Dall'olio, E., & Raskin, J.-F. (2008). Durations and parametric model-checking in timed automata. *ACM Transactions on Computational Logic*, *9*(2), 12:1–12:23.

Bruyère, V., & Raskin, J. (2007). Real-time model-checking: Parameters everywhere. *Logical Methods in Computer Science*, *3*(1:7), 1–30.

Bulling, N., & Jamroga, W. (2014). Comparing variants of strategic ability: How uncertainty and memory influence general properties of games. *Autonomous Agents and Multi-Agent Systems*, *28*(3), 474–518.

Bursztein, E., & Goubault-Larrecq, J. (2007). A logical framework for evaluating network resilience against faults and attacks. In *ASIAN*, pp. 212–227.

Busard, S., Pecheur, C., Qu, H., & Raimondi, F. (2014). Improving the model checking of strategies under partial observability and fairness constraints. In *Formal Methods and Software Engineering*, Vol. 8829 of *Lecture Notes in Computer Science*, pp. 27–42. Springer.

Cassez, F., David, A., Fleury, E., Larsen, K. G., & Lime, D. (2005). Efficient on-the-fly algorithms for the analysis of timed games. In *CONCUR*, Vol. 3653 of *Lecture Notes in Computer Science*, pp. 66–80.

David, A., Fang, H., Larsen, K. G., & Zhang, Z. (2014). Verification and performance evaluation of timed game strategies. In *FORMATS*, Vol. 8711 of *Lecture Notes in Computer Science*, pp. 100–114. Springer.

Dignum, F., & Kuiper, R. (1998). Specifying deadlines with continuous time using deontic and temporal logic. *International Journal of Electronic Commerce*, *3*(2), 67–85.

Ezekiel, J., & Lomuscio, A. (2009). Combining fault injection and model checking to verify fault tolerance in multi-agent systems. In *AAMAS*, pp. 113–120. IFAAMAS.

Faella, M., La Torre, S., & Murano, A. (2002). Dense real-time games. In *LICS*, pp. 167–176. IEEE Computer Society.

Faella, M., La Torre, S., & Murano, A. (2014). Automata-theoretic decision of timed games. *Theoretical Computer Science*, *515*, 46–63.

Farhang, S., & Grossklags, J. (2017). When to invest in security? empirical evidence and a game-theoretic approach for time-based security. *CoRR*, *abs/1706.00302*.

Henzinger, T. A., Horowitz, B., & Majumdar, R. (1999). Rectangular hybrid games. In *CONCUR*, Vol. 1664 of *Lecture Notes in Computer Science*, pp. 320–335. Springer.

Henzinger, T. A., & Prabhu, V. S. (2006). Timed alternating-time temporal logic. In *FORMATS*, Vol. 4202 of *Lecture Notes in Computer Science*, pp. 1–17. Springer.

Huang, X., & van der Meyden, R. (2014). Symbolic model checking epistemic strategy logic. In *AAAI*, pp. 1426–1432.

Jamroga, W., Konikowska, B., & Penczek, W. (2016). Multi-valued verification of strategic ability. In *AAMAS*, pp. 1180–1189.

Jamroga, W., & Penczek, W. (2012). Specification and verification of multi-agent systems. In *Lectures on Logic and Computation*, Vol. 7388 of *Lecture Notes in Computer Science*, pp. 210–263. Springer.

Jamroga, W., & van der Hoek, W. (2004). Agents that know how to play. *Fundamenta Informaticae*, *63*(2–3), 185–219.

Jurdzinski, M., & Trivedi, A. (2007). Reachability-time games on timed automata. In *ICALP*, Vol. 4596 of *Lecture Notes in Computer Science*, pp. 838–849. Springer.

Kacprzak, M., & Penczek, W. (2004). Unbounded model checking for alternating-time temporal logic. In *AAMAS*, pp. 646–653. IEEE Computer Society.

Kacprzak, M., & Penczek, W. (2005). Fully symbolic unbounded model checking for alternating-time temporal logic. *Autonomous Agents and Multi-Agent Systems*, *11*(1), 69–89.

Kocher, P. C. (1996). Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology, Proceedings of CRYPTO*, pp. 104–113.

Laroussinie, F., Markey, N., & Oreiby, G. (2006a). Expressiveness and complexity of ATL. Tech. rep. LSV-06-03, CNRS & ENS Cachan, France.

Laroussinie, F., Markey, N., & Oreiby, G. (2006b). Model checking timed ATL for durational concurrent game structures. In *FORMATS*, Vol. 4202 of *Lecture Notes in Computer Science*, pp. 245–259. Springer.

Laroussinie, F., Markey, N., & Oreiby, G. (2008). On the expressiveness and complexity of ATL. *Logical Methods in Computer Science*, *4*(2).

Lomuscio, A., Qu, H., & Raimondi, F. (2015). MCMAS: An open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer*, *24*, 84–90. Available online.

Lomuscio, A., & Raimondi, F. (2006). MCMAS: A model checker for multi-agent systems. In *TACAS*, Vol. 4314 of *Lecture Notes in Computer Science*, pp. 450–454. Springer.

Maler, O., Pnueli, A., & Sifakis, J. (1995). On the synthesis of discrete controllers for timed systems. In *STACS*, Lecture Notes in Computer Science, pp. 229–242. Springer.

Nadella, S. (2015). Enterprise security in a mobile-first, cloud-first world. Available at http://news.microsoft.com/security2015/.

Saner, E. (2014). The Swiss air force: armed and dangerous, but only in office hours. *The Guardian*, *19 Feb 2014*. Available at https://www.theguardian.com/world/shortcuts/2014/feb/19/swiss-air-force-ethiopian-airlines-hijacking-office-hours.

Schobbens, P. Y. (2004). Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, *85*(2), 82–93.

Schwartau, W. (1999). *Time based security: Practical and provable methods to protect enterprise and infrastructure*. Interpact Press.

Sipser, M. (1997). *Introduction to the theory of computation*. PWS Publishing Company.

Verizon (2016). Data breach investigations report (DBIR). Available at http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/.