Timed ATL: Forget Memory, Just Count

(Extended Abstract)

Étienne André
and Laure Petrucci
Université Paris 13
{Etienne.Andre,Laure.Petrucci}
@lipn.univ-paris13.fr

Wojciech Jamroga
and Michał Knapik
Institute of Computer Science,
Polish Academy of Sciences
{mknapik, wojciech.jamroga}
@ipipan.waw.pl

Wojciech Penczek
Institute of Computer Science,
Polish Academy of Sciences
and University of Natural Sciences
and Humanities, Siedlce, Poland
penczek@ipipan.waw.pl

Keywords

Alternating-time temporal logic, Timed ATL, Strategies

1. INTRODUCTION

Multi-Agent Systems (MAS) describe interactions of agents that are often assumed to be intelligent and autonomous. Alternating-time temporal logic ATL* [4] and its fragment ATL are logics that allow for reasoning about strategic interactions in MAS. ATL is typically interpreted over Concurrent Game Structures (CGS), which are used to model MAS [6, 26, 7, 14, 19] and their strategic properties to be verified [3, 2, 28, 29, 32, 17, 24, 15, 31, 25]. However, the *"vanilla"* ATL does not support timing constraints.

In this paper, we investigate Timed Alternating-Time Temporal Logic (TATL) [30] which allows for expressing strategic properties that depend on both the visited locations and the time measured along the paths. In what follows, we provide a hierarchy of timed and untimed strategies and show that, unless a strict punctuality is needed, tracking the passage of time can be replaced with counting the number of visits.

The work presented in this paper fits within the broad context of research on timed games [1, 9, 16, 18, 33]. We build upon the theory introduced in [30] which in turn can be seen as the simplest discrete-time extension of ATL [4]. Real-time extensions of ATL and, more generally, dense-timed games are explored in e.g. [8, 10, 20, 21, 22, 23, 27].

2. TATL AND ITS SEMANTIC VARIANTS

TATL [30] extends ATL [4] with timing constraints.

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

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

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

As usual, we read $\langle\langle A\rangle\rangle\psi$ as *"the coalition A can enforce ψ along each path"*, X stands for *"in the next state"*, U for *"until"*, and R for *"release"*. We introduce the additional modality $\langle\langle A\rangle\rangle F_{\sim\eta}\phi := \langle\langle A\rangle\rangle \top U_{\sim\eta}\phi$, where F is interpreted as *"eventually"*. The additional subscript denotes the timing constraints imposed on modalities. $\mathrm{TATL}_{\leq,\geq}$ is the subset of TATL with $\sim\, \in \{\leq, \geq\}$, *i.e.* disallowing equality.

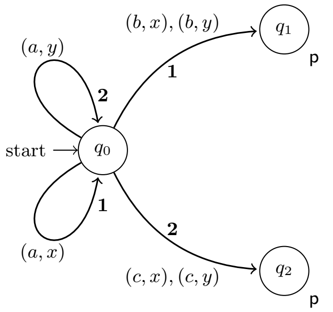

Figure 1: Punctuality needs clocks

We interpret our logic over Timed Durational CGS (TDCGS), *i.e.* graphs with vertices (set $\mathcal{Q}$) labeled with propositions and edges labeled with tuples of actions selected from Act, one per agent, and time durations. It is assumed that traversing an edge is an atomic action, taking as long as indicated by its time label. A path in the TDCGS is a sequence of states $\mathcal{S} = \mathcal{Q} \times \mathbb{N}$, *i.e.* pairs of current locations and time snapshots. We refer to [30] for details and focus on Fig. 1 to provide some further intuitions: This model contains three locations and its transitions are controlled by two agents. For example, in location q_0 agents 1 and 2 can first select actions a and x, respectively, to traverse the loop on q_0 in 1 time unit. Next, the agents can choose actions c and x, respectively, and move to location q_2 in 2 time units. This way, the system follows a finite path $\pi = (q_0, 0)(q_0, 1)(q_2, 3)$.

Strategic abilities of coalitions of agents, *i.e.* the paths that can be enforced, depend on the allowed strategies. Following [9, 13, 34], we consider here several variants. Let $\pi \in \mathcal{S}^+$ be a finite sequence of states. By π_F we denote the final state of π, $lc(\pi_F)$ is the location of π_F, and $\#_F(\pi)$ denotes how the number of times $lc(\pi_F)$ appears along π.

Definition 2 (Classes of strategies).

- *A* timed perfect recall strategy *for agent a is a function $\sigma_a : \mathcal{S}^+ \to Act$. Σ_{T} denotes the set of such strategies.*
- *A* timed memoryless strategy *is a strategy $\sigma_a \in \Sigma_{\mathrm{T}}$ that assigns to $\pi \in \mathcal{S}^+$ an action based only on the final state π_F. These are denoted by Σ_{t}.*

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

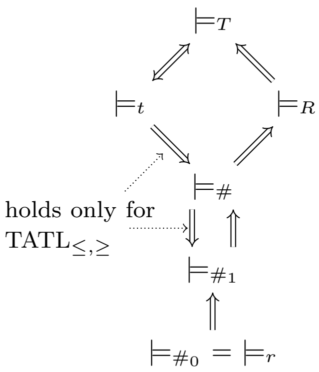

Figure 2: Correspondence between TATL semantics

- *A* timeless perfect recall strategy *is a strategy $\sigma_a \in \Sigma_{\mathrm{T}}$ that assigns to $\pi \in \mathcal{S}^+$ an action based on the sequence of locations visited along π. The temporal component of π is ignored. These are denoted by Σ_{R}.*
- Timeless memoryless strategies*, denoted by Σ_{r}, depend only on the final location $lc(\pi_F)$.*
- *A* counting strategy *is a strategy $\sigma_a \in \Sigma_{\mathrm{T}}$ such that, for each $\pi, \pi' \in \mathcal{S}^+$, if $lc(\pi_F) = lc(\pi'_F)$ and $\#_F(\pi) = \#_F(\pi')$, then $\sigma_a(\pi) = \sigma_a(\pi')$. The set of such strategies is denoted by $\Sigma_\#$.*

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 $q = lc(\pi_F)$ and $n = \#_{lc(\pi_F)}(\pi)$, for some π.

In addition to general counting strategies, we use those that are bounded by a threshold.

Definition 3 (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 $1 \leq j \leq n+1$: $\sigma_a^\#(q, k) = act_j$ if $k \in I_j$.*

The set of n–threshold strategies is denoted 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.

A strategy for a coalition $A \subseteq \mathbb{A}\mathrm{gt}$ is a set of strategies, one per agent. In what follows, for each type of strategy $\sigma \in \{\Sigma_{\mathrm{T}}, \Sigma_{\mathrm{t}}, \Sigma_{\mathrm{R}}, \Sigma_{\mathrm{r}}, \Sigma_\#\} \cup \bigcup_{n \in \mathbb{N}_+} \Sigma_{\#_n}$, we mean the corresponding satisfaction relation by using the appropriate subscript. For example, $q \models_\# \langle\langle 1 \rangle\rangle F_{\leq 5} \mathsf{srv}_1 \wedge \neg \langle\langle 2 \rangle\rangle F_{\leq 5} \mathsf{srv}_2$ may denote that the system is serviceable in five or less time units for agent 1 but not for agent 2, where the agents can use only counting strategies.

3. ANALYSIS OF SEMANTIC VARIANTS

Let us again refer to the model in Fig. 1. Observe that $q_0 \models \langle\langle 1 \rangle\rangle F_{=5} \mathsf{p}$. Indeed, agent 1 can follow a simple strategy of enforcing the loops on q_0 until the time reaches either 3 or 4, depending on the response of the second agent. Then, agent 1 selects action c or b, respectively, to reach one of the states labelled with p precisely at time 5. On the other hand, 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. We have, however, $q_0 \models \langle\langle 1 \rangle\rangle F_{\sim 5} \mathsf{p}$ and $q_0 \models_\# \langle\langle 1 \rangle\rangle F_{\sim 5} \mathsf{p}$ for $\sim \in \{\leq, \geq\}$.

In Fig. 2 we present the main contribution of this paper: a roadmap of correspondences between semantic variants of TATL. A single-direction arrow between two semantic relation symbols indicates that the satisfaction of a given TATL (or TATL$_{\leq,\geq}$, in two cases) formula in the source semantics implies the satisfaction in the target semantics. A double-direction arrow indicates that the semantics are equivalent.

As it turns out, despite the removal of the timed component from the semantics, the counting strategies can implement properties expressed in TATL$_{\leq,\geq}$, *i.e.* without strict punctuality. Moreover a detailed analysis of counting strategies that result from the presented reduction revealed the simplicity of their structure. In fact, it is sufficient to consider 1-threshold strategies that utilise only two actions per location to implement any TATL$_{\leq,\geq}$ property. If equality is permitted, then counting and timed semantics do not coincide, the latter being more expressive. In general, there is no threshold that would allow for the counting strategies to be as powerful as the timed strategies.

Theorem 1 (Comparing semantics of TATL). *The following equivalences hold:*

- $\models_T \iff \models_t$*, for* TATL*,*
- $\models_{\#_1} \iff \models_\# \iff \models_R \iff \models_T$*, for* TATL$_{\leq,\geq}$*.*

4. CONCLUSIONS AND FUTURE WORK

In this paper we investigated TATL, a basic, natural extension of ATL with discrete time. We introduced a new type of semantics, where agents' decisions are based on the number of visits at locations encountered along the current execution path. We investigated in detail the correspondence between the semantic variants of the logic.

This work opens several research avenues that we plan to explore in future. Firstly, the strict binding of coalition selectors and temporal modalities in TATL can be loosened to obtain TATL*, similarly to ATL *vs.* ATL*. We feel that the correspondence between timed and counting semantics of TATL* is worth investigating. It is not difficult to see that in TATL* equality can be expressed using inequalities. Secondly, in this work we deal with agents equipped with perfect knowledge about their environment. Following [34] we plan to analyse the consequences of introducing indistinguishability relations to TDCGS. We expect that this modification will significantly influence the decidability of the model checking problem. Another natural extension of TATL consists in extending the logic [11], the models [5], or both [12] with parameters. Our preliminary analysis suggests that the decidability of associated *emptiness problem*, *i.e.* the existence of parameter valuations under which a given formula holds, depends both on the formula syntax and on the choice of place for parameter injection.

Acknowledgments

This work is partially sponsored by the National Centre for Research and Development (NCBR), Poland, under the PolLux project VoteVerif (POLLUX-IV/1/2016) and supported by the CNRS/PAS project "BehaPPi-BMC". We also thank Łukasz Mikulski for valuable suggestions and help in turning one of our conjectures into a theorem.

REFERENCES

[1] R. Alur, M. Bernadsky, and P. Madhusudan. Optimal reachability for weighted timed games. In *ICALP*, volume 3142 of *LNCS*, pages 122–133. Springer, 2004.

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

[3] R. Alur, T. Henzinger, F. Mang, S. Qadeer, S. Rajamani, and S. Tasiran. MOCHA: Modularity in model checking. In *CAV*, volume 1427 of *LNCS*, pages 521–525. Springer, 1998.

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

[5] R. Alur, T. A. Henzinger, and M. Y. Vardi. Parametric real-time reasoning. In *STOC*, pages 592–601. ACM, 1993.

[6] I. Boureanu, A. V. Jones, and A. Lomuscio. Automatic verification of epistemic specifications under convergent equational theories. In *AAMAS*, pages 1141–1148. ACM, 2012.

[7] I. Boureanu, P. Kouvaros, and A. Lomuscio. Verifying security properties in unbounded multiagent systems. In *AAMAS*, pages 1209–1217. ACM, 2016.

[8] P. Bouyer, D. D'Souza, P. Madhusudan, and A. Petit. Timed control with partial observability. In *CAV*, volume 2725 of *LNCS*, pages 180–192. Springer, 2003.

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

[10] T. Brihaye, F. Laroussinie, N. Markey, and G. Oreiby. Timed concurrent game structures. In *CONCUR*, volume 4703 of *LNCS*, pages 445–459. Springer, 2007.

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

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

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

[14] E. Bursztein and J. Goubault-Larrecq. A logical framework for evaluating network resilience against faults and attacks. In *ASIAN*, pages 212–227, 2007.

[15] S. Busard, C. Pecheur, H. Qu, and F. Raimondi. Improving the model checking of strategies under partial observability and fairness constraints. In *Formal Methods and Software Engineering*, volume 8829 of *LNCS*, pages 27–42. Springer, 2014.

[16] F. Cassez, A. David, E. Fleury, K. G. Larsen, and D. Lime. Efficient on-the-fly algorithms for the analysis of timed games. In *CONCUR*, volume 3653 of *LNCS*, pages 66–80, 2005.

[17] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis. PRISM-games: A model checker for stochastic multi-player games. In *TACAS*, volume 7795 of *LNCS*, pages 185–191. Springer, 2013.

[18] A. David, H. Fang, K. G. Larsen, and Z. Zhang. Verification and performance evaluation of timed game strategies. In *FORMATS*, volume 8711 of *LNCS*, pages 100–114. Springer, 2014.

[19] J. Ezekiel and A. Lomuscio. Combining fault injection and model checking to verify fault tolerance in multi-agent systems. In *AAMAS*, pages 113–120. IFAAMAS, 2009.

[20] M. Faella, S. La Torre, and A. Murano. Dense real-time games. In *LICS*, pages 167–176. IEEE Computer Society, 2002.

[21] M. Faella, S. La Torre, and A. Murano. Automata-theoretic decision of timed games. *Theoretical Computer Science*, 515:46–63, 2014.

[22] T. A. Henzinger, B. Horowitz, and R. Majumdar. Rectangular hybrid games. In *CONCUR*, volume 1664 of *LNCS*, pages 320–335. Springer, 1999.

[23] T. A. Henzinger and V. S. Prabhu. Timed alternating-time temporal logic. In *FORMATS*, volume 4202 of *LNCS*, pages 1–17. Springer, 2006.

[24] X. Huang and R. van der Meyden. Symbolic model checking epistemic strategy logic. In *AAAI*, pages 1426–1432, 2014.

[25] W. Jamroga, B. Konikowska, and W. Penczek. Multi-valued verification of strategic ability. In *AAMAS*, pages 1180–1189, 2016.

[26] W. Jamroga and W. Penczek. Specification and verification of multi-agent systems. In *Lectures on Logic and Computation*, volume 7388 of *LNCS*, pages 210–263. Springer, 2012.

[27] M. Jurdzinski and A. Trivedi. Reachability-time games on timed automata. In *ICALP*, volume 4596 of *LNCS*, pages 838–849. Springer, 2007.

[28] M. Kacprzak and W. Penczek. Unbounded model checking for alternating-time temporal logic. In *AAMAS*, pages 646–653. IEEE Computer Society, 2004.

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

[30] F. Laroussinie, N. Markey, and G. Oreiby. Model checking timed ATL for durational concurrent game structures. In *FORMATS*, volume 4202 of *LNCS*, pages 245–259. Springer, 2006.

[31] A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: An open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer*, 2015. Availabe online.

[32] A. Lomuscio and F. Raimondi. MCMAS: A model checker for multi-agent systems. In *TACAS*, volume 4314 of *LNCS*, pages 450–454. Springer, 2006.

[33] O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of discrete controllers for timed systems. In *STACS*, LNCS, pages 229–242. Springer, 1995.

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