Strategic Behaviour in Multi-Agent Systems Able to Perform Temporal Reasoning

Matei Popovici and Lorina Negreanu

Abstract. Temporal reasoning and strategic behaviour are important abilities of Multi-Agent Systems. We introduce a method suitable for modelling agents which can store and reason about the evolution of an environment, and which can reason strategically, that is, make a rational and self-interested choice, in an environment where all other agents will behave in the same way. We introduce a game-theoretic formal framework, and provide with a computational characterisation of our solution concepts, which suggests that our method can easily be put into practice.

1 Introduction

Multiagent Systems (MAS) have long been a successful means for modelling the interaction of software agents (or simply, programs) with themselves, other humans, and a given environment. In this context, there is an ever increasing need for MAS: (i) able to perform temporal inferences and (ii) capable of strategic reasoning.

In this paper, we equip MAS with memory and temporal inference capabilities, by deploying a method for temporal reasoning previously described in [2, 3, 6, 5], and we study the strategic behaviour of such systems. For the latter, we introduce the game-theoretic concept of Nash Equilibrium, and show how it can be used for enforcing MAS stability. We assume each agent has a particular goal, which is dependent on the current and past state(s) of the system. Goals are expressed using the temporal language $L_{\mathcal{H}}$, introduced and described in [5]. Each agent has some available actions which are able to change the system state Also, each action has associated a particular cost. We further assume that agents are *rational* and *self-interested*, meaning they will choose to execute those actions which: (i) make the goal satisfied, (ii) minimises the agent's costs. We are interested in those situations

Matei Popovici · Lorina Negreanu
Computer Science Department, University Politehnica of Bucharest,
Bucharest, Romania
e-mail: `matei.popovici@cs.pub.ro, lorina@moon.ro`

F. Zavoral et al. (eds.), *Intelligent Distributed Computing VII,*
Studies in Computational Intelligence 511,
DOI: 10.1007/978-3-319-01571-2_25,

where agents cannot individually satisfy their goals i.e. they may have (partially) conflicting and/or (partially) overlapping goals. In such situations, agents may deviate, i.e. change their set of actions to another, if the latter is a possible means for achieving a better outcome. The solution concept we use, the Nash Equilibrium, is aimed at identifying the outcomes where no agent has an incentive to deviate. We further study the computational complexity of related to the Nash Equilibrium, using the $L_{\mathcal{H}}$-model checking procedure described in [5] and provide with an upper complexity bound. We conjecture that such a bound is tight.

Related Work: We consider Boolean Games [4] to be one of the first frameworks which use logic for describing goal-based strategic interactions between agents. However, instead of propositional logic, we use the more expressive language $L_{\mathcal{H}}$, which also allows expressing domain-dependent temporal properties.

The choice of $L_{\mathcal{H}}$ over well-known temporal logics such as LTL [7] or CTL [1], is motivated, on one hand, on the increased computational complexity of model checking, in the case of LTL [7] and on the reduced expressive power of CTL, with respect to $L_{\mathcal{H}}$ [5].

The rest of the paper is structured as follows. In Section 2 we make a brief account on the temporal knowledge representation and reasoning method which we use. In Section 3 we introduce our formal framework, define the Nash Equilibrium solution concept and describe our complexity results, and in Section 4 we conclude and identify possible future work.

2 Temporal Representation and Reasoning

In what follows, we succinctly describe the approach for temporal representation and reasoning which we adopt. For a more detailed presentation, see [5].

Temporal Graph. The history of a given MAS is stored using *labelled* temporal graphs. A labelled temporal graph (or simply temporal graph) consists of: (i) *hypernodes* which model discrete moments of time, (ii) *labelled action nodes* which model instantaneous events that change the state of the current environment and (iii) *labelled quality edges* which model time-dependent properties.

An example of a temporal graph is shown in Figure 1, where h_i, $i = \overline{1,3}$ are hypernodes designating three distinct moments of time, a,b,c,d,e,f and a_{end} are action nodes, each associated to a certain hypernode, (a,b), (c,d) and (e,f) are quality edges denoting the time-dependent properties $On(ac)$, $On(h)$ and $On(v)$, respectively.

Formally, given a *vocabulary* $\sigma_A \cup \sigma_Q$ and $\sigma_A \cup \sigma_Q$-structure $\mathfrak{I}$, over universe I (a set of individuals), a labelled temporal graph (or **t**-graph) is a structure $\mathcal{H} = \langle A, H, \mathcal{T}, E, \mathcal{L}_A, \mathcal{L}_Q \rangle$, where A is a set of *action nodes*, H is a set of hypernodes, $\mathcal{T} : A \rightarrow H$ is a onto (or surjective) function which assigns for each action node a, the hypernode $\mathcal{T}(a)$ when a occurs, $E \subseteq A^2$ is a *quality edge* relation, the function $\mathcal{L}_Q : E \rightarrow \mathcal{P}(\cup_{R \in \sigma_Q} R^I)$ **labels** each quality edge with a set of relation instances from $\mathfrak{I}$ and the function $\mathcal{L}_A : A \rightarrow \mathcal{P}(\cup_{R \in \sigma_A} R^I)$ **labels** each action node with a set of action labels from $\Im$. Given the quality edge $(a,b) \in E$, we say that action nodes a or b are the *constructor* or *destructor* nodes of (a,b), respectively.

The language $L_{\mathcal{H}}$. We use the language $L_{\mathcal{H}}$ in order to express complex temporal relations between time-dependent properties, such as *has the air conditioner been opened in the same time a window has been opened* or *find all properties On(ac) which are destroyed precisely when a property Opened(x,y) is created.*

Due to limited space, we provide with a simplified syntax and semantics of $L_{\mathcal{H}}$. The original one(s) can be found in [5].

Let $\mathbb{V}\text{ars}$ be a set of variables, $\Im$ be a $\sigma_Q \cup \sigma_A$-structure and $\mathcal{H}_\Im$ be a labelled **t**-graph. The ***syntax*** of a (Q-)formula is recursively defined with respect to $\Im$, as follows. If $R \in \sigma_Q$ with $arity(R) = n$ and $\bar{t} \in (\mathbb{V}\text{ars} \cup I)^n$, then $R(\bar{t})$ is an **atomic** Q-formula (or an atom). If ϕ is a (Q-)formula then (ϕ) is also a (Q-)formula. If ϕ, ψ are (Q-)formulae then ϕ **b** ψ, ϕ ¬**b** ψ, ϕ **m** ψ, ϕ **a** ψ, ϕ ¬**a** ψ are also (Q-)formulae. **b** stands for *before* and ¬**b** for *not before*. Similarly, **m** stand for *meets* and **a** for *after*. The following are valid $L_{\mathcal{H}}$ formulae: (i) $On(x)$ (ii) $On(ac)$ **b** $Opened(x,y)$ (iii) $On(x)$ **m** $Opened(John,x)$. Formula (i) refers to all properties On associated to some individual x, which can occur at any time in the evolution of the domain. Formula (ii) refers to those properties *On(ac)* which occur before any relation *Opened*, which may enrol arbitrary individuals x,y. Similarly, formula (iii) refers to those properties *On(ac)* which end at the very moment when some relation *Opened(John,x)* is created.

$L_{\mathcal{H}}$ formulae are evaluated over *paths* from labelled temporal graphs. The evaluation is defined by the mapping $||\cdot||^Q_{\mathcal{H}} : L_{\mathcal{H}} \to 2^E$ which assigns for each formula $\phi \in L_{\mathcal{H}}$, a set of quality edges $||\phi||^Q_{\mathcal{H}}$ which satisfy it. In what follows, we omit the formal definition of the $L_{\mathcal{H}}$ semantics, and defer the interested reader to [5]. Finally, we note that, in the unrestricted case considered in [5], $L_{\mathcal{H}}$-formulae can also express more general constraints, for instance between actions and properties (and not just properties, as considered here). Returning to the temporal graph, we have that: $||On(x)||^Q_{\mathcal{H}} = \{(a_1,a_2)\}$, since (a_1,a_2) is labelled with *On(ac)*, which unifies with *On(x)*. $||On(ac)\ \mathbf{b}\ Opened(John,x)||^Q_{\mathcal{H}} = \{(a_1,a_2)\}$ since (a_1,a_2) also occurs before quality edge (a_3,a_4), which is labelled with *Opened(John,win)*. Finally, $||On(x)\ \mathbf{m}\ Opened(John,x)||^Q_{\mathcal{H}} = \emptyset$ since there is no individual $i \in I$, for which properties *On(i)* and *Opened(John,i)* exist in our labelled temporal graph.

Proposition 1 ([5]). *Let $\mathcal{H}$ be a temporal graph where temporal nodes H are equipped with a* temporal order*: $\langle H,<\rangle$. We designate by $L^<_{\mathcal{H}}$ the language $L_{\mathcal{H}}$ defined over such temporal graphs. Given a formula $\phi \in L^<_{\mathcal{H}}$ and a property Q, the decision problem $Q \in ||\phi||^Q_{\mathcal{H}}$ is NP-complete.*

3 Formal Setting

In what follows, we formally introduce the concept of Nash Equilibrium adjusted to our setting, as well as other supporting concepts.

Definition 1 (t-frame). Let $\mathfrak{I}$ be a structure over vocabulary $\sigma_Q \cup \sigma_A$ and with universe I, $\mathcal{A}set = \{R_i(\bar{i}) : R_i \in \sigma_A, \bar{i} \in R_i^I\}$, and $Qset = \{R_i(\bar{i}) : R_i \in \sigma_Q, \bar{i} \in R_i^I\}$.

A **t**-frame is given by $\mathcal{F} = (N, Own, Ont, c, (\phi_n)_{n\in N}, (v_n)_{n\in N}, \langle H, <\rangle)$, where N is a finite set of agents, $Own : N \to 2^{\mathcal{A}set}$ is a function that assigns to each agent a subset of action *types* he can control. $Own(n)$ models the types of changes agent n can perform to the environment. The sets $Own(1) \ldots Own(|N|)$ are a **partition** of $\mathcal{A}set$, $Ont \subseteq \mathcal{A}set \times Qset \times \mathcal{A}set$ indicates what are the labels of action nodes that can create and destroy quality edges with a certain label. In this sense, Ont can be interpreted as a (simple) *ontology*. Ont must take into account that each quality edge (and hence quality label) must be created/destroyed by a unique action node (hence action label), $c : \mathcal{A}set \to \mathbb{R}^{\geq 0}$ assigns a cost to each action label, $(\phi_n)_{n\in N}$ is a set of goals, one for each agent ($\phi_n \in L_{\mathcal{H}}$ for $n \in N$), $(v_n)_{n\in N}$ a goal value for each agent, measuring the amount of resources each agent is willing to spend for achieving his goal, a finite and ordered set $\langle H, <\rangle$ of hypernodes.

*Example 1 (**t**-frame).* Let $\mathcal{F}$ be a three-agent frame where h(home cinema), ac(air conditioner) and v(ventilation) are individuals and $\mathcal{A}set = \{turnOn(x) : x \in I\} \cup \{\text{turnOff}(x) : x \in I\}$ and $Qset = \{On(x) : x \in I\}$. Each agent 1,2,3 can turn on or off device ac, h, v, respectively. The costs of each action are fixed to 1. The goals of each agent are as follows: $\phi_1 = \text{On}(ac)\ \mathbf{b}\ \text{On}(h)$ (*the air conditioner must be started before the home cinema was started*), $\phi_2 = \text{On}(ac)$ (*at some point in time, the air conditioner should be started.*), $\phi_3 = \text{On}(v)\ \mathbf{m}\ \text{On}(h)$ (*the ventilation should be stopped precisely when the home cinema starts*). Finally, $v_n = 5$ for $n \in \{1,2,3\}$, and $\langle H, <\rangle$ is given by: $h_1 < h_2 < h_3$.

In what follows, we assume that $\mathcal{F}$ is a **t**-frame and $h_{max} \in H$ is the *most recent* hypernode, i.e., $\nexists h' \in H$ such that $h_{max} < h'$: An ***action*** of a agent n is given by: $\langle A_n, \mathcal{L}_{A_n}, \mathcal{T}_n\rangle$ where A_n is a set of action nodes, $\mathcal{L}_{A_n} : A_n \to \mathcal{A}set$ is a labelling function and $\mathcal{T}_n : A_n \to H$ is a mapping of A_n on the set of hypernodes. We note that, in this paper, we have restricted $\mathcal{L}_{A_n}$, such that each action node receives a unique label. This restriction is for convenience only.

An ***action profile*** consists of a vector $(\langle A_n, \mathcal{L}_{A_n}, \mathcal{T}_n\rangle)_{n\in N}$ of actions, one for each agent in N. An action profile **induces** a labelled temporal graph $\langle A, H, \mathcal{T}, E, \mathcal{L}_A, \mathcal{L}_Q\rangle$ where: $A = (\bigcup_{n\in N} A_n) \cup a_{end}$, for each pair $a, b \in A$, if there exists a label $X \in Qset$ such that $(\mathcal{L}_A(a), X, \mathcal{L}_A(b)) \in Ont$, then create $(a,b) \in E$, and for each such X, $X \in \mathcal{L}_Q(a,b)$, and if there exists an action node $a \in A$ and **no** action node $b \in A$ such that $(\mathcal{L}_A(a), X, \mathcal{L}_A(b))$, then create $(a, a_{end}) \in E$, and for each such X, $X \in \mathcal{L}_Q(a,b)$. Finally, $\mathcal{T}(x) = \mathcal{T}_i(x)$ if $x \in A_i$ and $\mathcal{T}(x) = h_{max}$ if $x = a_{end}$ and $\mathcal{L}_A(x) = \{\mathcal{L}_{A_i}(x)\}$ if $x \in A_i$, for $i = \overline{1 \ldots |N|}$.

Proposition 2. *Given a labelled temporal graph $\mathcal{H}$, there is a **unique** action profile $a^* \in \times_{n\in N} \mathcal{A}ctions_n$ such that a^* induces $\mathcal{H}$.*

Given a labelled **t**-graph $\mathcal{H}$ and a formula ϕ_k of agent k, we say ϕ_k is **satisfied** in $\mathcal{H}$ iff $||\phi_k||^X_{\mathcal{H}} \neq \emptyset$. Thus, a goal ϕ_k is satisfied in $\mathcal{H}$ if there exists at least one quality edge in $\mathcal{H}$ that belongs to $||\phi_k||^Q_{\mathcal{H}}$. A labelled temporal graph $\mathcal{H}$ induces a ***cost***

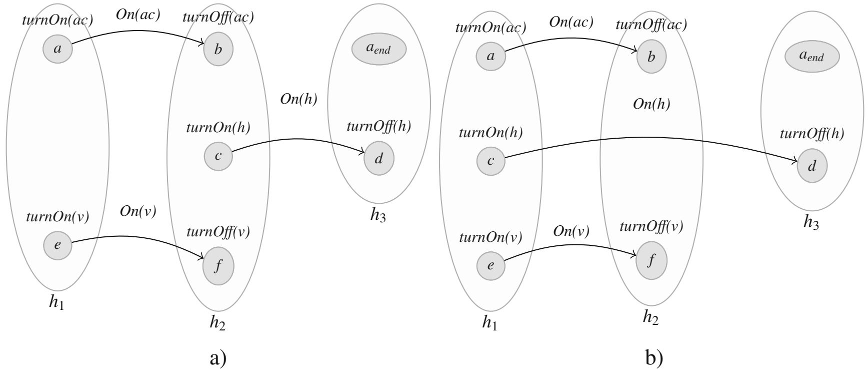

Fig. 1 a) A temporal graph induced from an action profile (and a Nash Equilibrium) and b) A temporal graph which is not a Nash Equilibrium

on a agent n, denoted $Cost_n(\mathcal{H})$, which is: $Cost_n(\mathcal{H}) = \sum_{a\in A}\left(\sum_{X\in Own(n)\cap\mathcal{L}_A(a)} c(X)\right)$ $Cost(n)$ is the sum of costs of actions performed by agent n. The utility $u_n(\mathcal{H})$ of agent n in a labelled temporal graph $\mathcal{H}$ is defined as: $u_n(\mathcal{H}) = v_n - Cost_n(\mathcal{H})$ if ϕ_n is satisfied in $\mathcal{H}$, and as $u_n(\mathcal{H}) = -Cost_n(\mathcal{H})$, otherwise. We say a agent n prefers a labelled temporal graph $\mathcal{H}$ over $\mathcal{H}'$ and write $\mathcal{H} \succ_n \mathcal{H}'$, if $u_n(\mathcal{H}) > u_n(\mathcal{H}')$. We also extend the preference relation over action profiles. Given two action profiles a, a' that induce temporal graphs $\mathcal{H}$ and $\mathcal{H}'$, respectively, we write $a \succ_n a'$ iff $\mathcal{H} \succ_n \mathcal{H}'$. Let $\mathcal{H}$ be the labelled temporal graph from Figure 1 a) and $\mathcal{H}'$ be the temporal graph from Figure 1 b). We have that $u_1(\mathcal{H}) = 5-(1+1) = 3$ and $u_1(\mathcal{H}') = 0-(1+1) = -2$, since agent 1's goal is satisfied in $\mathcal{H}$ but not in $\mathcal{H}'$. Therefore $\mathcal{H} \succ_1 \mathcal{H}'$.

A ***temporal game*** is defined as $\mathcal{TG} = (\mathcal{F}, (\mathcal{A}ctions_n)_{n\in N}, (\succ_n)_{n\in N})$ where, for each $n \in N$, $\mathcal{A}ctions_n$ designates the set of actions available to agent n and $\succ_n$ is the preference relation of agent n over temporal graphs. A *Nash Equilibrium* (NE) of a ***temporal game*** is a labelled temporal graph induced by an action profile: $(a^*_1, \ldots, a^*_n, \ldots, a^*_{|N|}) \in \times_{n\in N}\mathcal{A}ctions_n$ such that, for each agent $n \in N$ and any action $a_n \in \mathcal{A}ctions_n$: $(a^*_1, \ldots, a^*_n, \ldots, a^*_{|N|}) \succ_n (a^*_1, \ldots, a_n, \ldots, a^*_{|N|})$. Nash Equilibria capture those situations in which each individual agent n cannot change his action (namely a^*_n) to one that ensures a higher utility. The **t**-graph in Figure 1 a) is a Nash Equilibrium. All three agents have their goals satisfied, and no individual agent can change his action to achieve a higher utility. The temporal graph from Figure 1 b) is **not** a Nash Equilibrium. The goal of agent 1 is not satisfied and there is no action that 1 can take in order to change this. Therefore, 1 would prefer not to execute any action node(i.e., remain passive), and therefore achieve 0 utility (instead of -2). The same is true with respect to agent 3.

Proposition 3 (Complexity Results). *Checking whether a temporal graph $\mathcal{H}$ is a Nash Equilibrium of a temporal game $\mathcal{TG}$ with goals formulated in the language $L^<_{\mathcal{H}}$, is a Π_2-problem. Finding if there exists a temporal graph $\mathcal{H}$ such that $\mathcal{H}$ is a Nash Equilibrium of a temporal game $\mathcal{TG}$ with goals formulated in the language $L^<_{\mathcal{H}}$, is a problem in Σ_3.*

4 Conclusions and Future Work

While the scope of our paper is rather formal, we believe our results can be easily put into practice. Our complexity results show that implementations are possible, even if they require an increased computational effort. We consider this effort to be tolerable. Also, by introducing *limited memory*, that is, by truncating temporal graphs to a fixed number of hypernodes, the computational effort may be further controlled. As suggested by all the examples above, our modelling method can be used for identifying stable behaviour of agents in intelligent environments (i.e., buildings equipped with programmable sensors and control devices). Such an approach is currently a work-in-progress. However, we consider our method to be general enough for application in a wide variety of scenarios that require modelling temporal reasoning together with strategic behaviour.

Acknowledgements. The authors wish to acknowledge the help and support from prof. Cristian Giumale, which was the first to think about temporal graphs, and whose guidance patroned the work presented in this paper.

The work has been funded by Project 264207, ERRIC-Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1.

References

1. Arnold, A., Crubille, P.: A linear algorithm to solve fixed-point equations on transition systems. Inf. Process. Lett. 29(2), 57–66 (1988)
2. Giumale, C., Negreanu, L.: Reasoning with fluid qualities. In: 17th International Conference on Control Systems and Computer Science, CSCS-17, vol. 2, pp. 197–203 (December 2009)
3. Giumale, C., Negreanu, L., Muraru, M., Popovici, M.: Modeling ontologies for time-dependent applications. In: International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, pp. 202–208 (2010)
4. Harrenstein, P., van der Hoek, W., Meyer, J.-J., Witteveen, C.: Boolean games. In: Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge, TARK 2001, pp. 287–298. Morgan Kaufmann Publishers Inc., San Francisco (2001)
5. Popovici, M.: Using evolution graphs for describing topology-aware prediction models in large clusters. In: Fisher, M., van der Torre, L., Dastani, M., Governatori, G. (eds.) CLIMA XIII 2012. LNCS, vol. 7486, pp. 94–109. Springer, Heidelberg (2012)
6. Popovici, M., Muraru, M., Agache, A., Giumale, C., Negreanu, L., Dobre, C.: A modeling method and declarative language for temporal reasoning based on fluid qualities. In: Andrews, S., Polovina, S., Hill, R., Akhgar, B. (eds.) ICCS-ConceptStruct 2011. LNCS, vol. 6828, pp. 215–228. Springer, Heidelberg (2011)
7. Sistla, A.P., Clarke, E.M.: The complexity of propositional linear temporal logics. J. ACM 32(3), 733–749 (1985)