Towards an Updatable Strategy Logic

Christophe Chareton Julien Brunel David Chemouil

Onera – The French Aerospace Lab
F-31055 Toulouse, France

`firstname.lastname@onera.fr`

This article is about temporal multi-agent logics. Several of these formalisms have been already presented (ATL-ATL*, ATL_{sc}, SL). They enable to express the capabilities of agents in a system to ensure the satisfaction of temporal properties. Particularly, SL and ATL_{sc} enable several agents to interact in a context mixing the different strategies they play in a semantical game. We generalize this possibility by proposing a new formalism, Updating Strategy Logic (USL). In USL, an agent can also refine its own strategy. The gain in expressive power rises the notion of *sustainable capabilities* for agents.

USL is built from SL. It mainly brings to SL the two following modifications: semantically, the successor of a given state is not uniquely determined by the data of one choice from each agent. Syntactically, we introduce in the language an operator, called an *unbinder*, which explicitly deletes the binding of a strategy to an agent. We show that USL is strictly more expressive than SL.

1 Introduction

Multi-agent logics are receiving growing interest in contemporary research. Since the seminal work of Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman [2], one major and recent direction (ATL with Strategy Context [3, 6, 7], Strategy Logic (presented first in [5] and then extended in [8, 10]) aims at contextualizing the statements of capabilities of agents.

Basically, multi-agent logics enable assertions about the capability of agents to ensure temporal properties. Thus, ATL-ATL* [2] appears as a generalization of CTL-CTL*, in which the path quantifiers **E** and **A** are replaced by *strategy quantifiers*. Strategy quantifiers (the existential $\langle\langle A\rangle\rangle$ and the universal $[\![A]\!]$) have a (coalition of) agent(s) as parameter. $\langle\langle A\rangle\rangle\varphi$ means that agents in A can act so as to ensure the satisfaction of temporal formula φ. It is interpreted in *Concurrent Game Structures* (CGS), where agents can make choices influencing the execution in the system. Formula $\langle\langle A\rangle\rangle\varphi$ is true if agents in A have a strategy so that if playing it they force the execution to satisfy φ, whatever the other agents do.

A natural question is: how to interpret the imbrication of several quantifiers? Precisely, in the interpretation of such formula as

$$\psi_1 := \langle\langle a_1\rangle\rangle\Box(\varphi_1 \wedge \langle\langle a_2\rangle\rangle\Box\varphi_2)$$

(where $\Box\varphi$ is the temporal operator meaning φ is always true, and a_1 and a_2 are agents), is the evaluation of φ_2 made in a context that takes into account both the strategy quantified in $\langle\langle a_1\rangle\rangle$ and the strategy quantified in $\langle\langle a_2\rangle\rangle$?

In ATL-ATL*, only a_2 is bound: subformula $\langle\langle a_2\rangle\rangle\Box\varphi_2$ is true iff a_2 may ensure $\Box\varphi_2$, whatever the other agents do. Then $\langle\langle a_2\rangle\rangle$ stands for three successive operations: First, each agent is unbound from its current strategy, then an existential quantification is made for strategy σ. At last, a_2 is bound to strategy σ.

ATL_{sc} [3, 6, 7], while keeping the ATL syntax, adapts the semantics in order to interpret formulas in a context which stores strategies introduced by earlier quantifiers.

Strategy Logic (SL [8, 10]) is another interesting proposition, which distinguishes between the quantifications over strategies and their bindings to agents. The operator $\langle\langle a\rangle\rangle$ is split into two different operators: a quantifier over strategies ($\langle\langle x\rangle\rangle$, where x is a strategy variable) and a binder ((a,x), where a is an agent) that stores into a context the information that a plays along the strategy instantiating variable x (let us write it σ_x in the remaining of this paper).The ATL formula ψ_1 syntactically matches the SL:

$$\psi_2 := \langle\langle x_1\rangle\rangle(a_1,x_1)\Box(\varphi_1 \wedge \langle\langle x_2\rangle\rangle(a_2,x_2)\Box\varphi_2)$$

In ψ_2, when evaluating $\Box\varphi_2$, a_1 remains bound to strategy σ_{x_1} except if a_1 and a_2 are the same agent. If they are the same, the binder (a_2,x_2) unbinds a from its current strategies before binding her to σ_{x_2}.

In this paper we present USL, a logic obtained from SL by making explicit the unbinding of strategies and allowing new bindings without previous unbinding. For that, we introduce an explicit unbinder $(a \not\triangleright x)$ in the syntax (and the binder in USL is written $(a \triangleright x)$) and we interpret USL in models where the choices of agents are represented by the set of potential successors they enable from the current state. When there is no occurrence of an unbinder, each agent remains bound to her current strategies. Then different strategies can combine together even for a single agent, provided that they are *coherent*, which means they define choices in non-empty intersection (the notion is formally defined in Sect. 2).

The main interest in such introduction is to distinguish between cases where an agent composes strategies together and situations where she revokes a current strategy for playing an other one. If a_1 and a_2 are the same agents, then ψ_2 is written in SL:

$$\psi_3 := \langle\langle x_1\rangle\rangle(a,x_1)\Box(\varphi_1 \wedge \langle\langle x_2\rangle\rangle(a,x_2)\Box\varphi_2),$$

which syntactically matches the USL:

$$\psi_4 := \langle\langle x_1\rangle\rangle(a \triangleright x_1)\Box(\varphi_1 \wedge \langle\langle x_2\rangle\rangle(a \triangleright x_2)\Box\varphi_2)$$

In ψ_3, subformula $\langle\langle x_2\rangle\rangle(a,x_2)\Box\varphi_2$ states that a can adopt a new strategy that ensures $\Box\varphi_2$, no matter if it is coherent with the strategy σ_{x_1} previously adopted. In ψ_4, both strategies must combine coherently together. In natural language ψ_4 states that a can ensure φ_1 and leave open the possibility to ensure φ_2 in addition. The equivalent of ψ_3 in USL is actually not ψ_4 but

$$\psi_5 := \langle\langle x_1\rangle\rangle(a \triangleright x_1)\Box(\varphi_1 \wedge \langle\langle x_2\rangle\rangle(a \not\triangleright x_1)(a \triangleright x_2)\Box\varphi_2)$$

There indeed, in subformula $(a \not\triangleright x_1)(a \triangleright x_2)\Box\varphi_2$, a is first unbound from σ_{x_1} and then bound to σ_{x_2}.

A consequence of considering these compositions of strategies is the expressiveness of *sustainable capabilities* of agents. Let us now consider the USL formula:

$$\psi_6 := \langle\langle x_1\rangle\rangle(a \triangleright x_1)\Box(\langle\langle x_2\rangle\rangle(a \not\triangleright x_1)(a \triangleright x_2)\mathbf{X}\, p)$$

There the binder $(a \triangleright x_2)$ is used with the unbinder $(a \not\triangleright x_1)$, so that ψ_6 is equivalent to the SL:

$$\psi_7 := \langle\langle x_1\rangle\rangle(a,x_1)\Box(\langle\langle x_2\rangle\rangle(a,x_2)\mathbf{X}\, p)$$

It states that a can remain capable to perform the condition expressed by $\mathbf{X}\, p$ when she wants. But in case she actually performs it, the formula satisfaction does not require that she is still capable to perform it. The statement holds in state s_0 in structure $\mathcal{M}_1$ with single agent a. See Fig.1, where choices are defined by the set of transitions they enable. Since $\mathcal{M}_1$ interprets SL formulas with only agent a, the choices for a are deterministic: let s,s' be two states and c a choice, then the transition from s to s' is

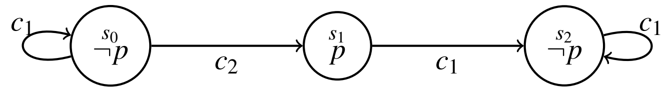

Figure 1: Structure $\mathscr{M}_1$

labelled with c iff $\{s'\}$ is a choice for a at s. Indeed, by always playing choice c_1, a remains in state s_0, where she can change her mind to ensure p. But if she chooses to reach p, she can do it only by moving to state s_1 and then to state s_2. Doing so, she loses her capability to ensure $\mathbf{X}\ p$ at any time. The only way for her to maintain her capability to reach p is to always avoid it, her capability is not sustainable.

A more game theoretical view is to consider strategies as commitments. In such view, by adopting a strategy, a adopts a behavior that holds in the following execution, as far as it is not explicitly deleted. Formula

$$\psi_8 := \langle\langle x_1\rangle\rangle(a \triangleright x_1)\Box(\langle\langle x_2\rangle\rangle(a \triangleright x_2)\mathbf{X}\ p)$$

is the counterpart of formula ψ_7 with such interpretation of composing strategies for a single agent. If a plays σ_{x_2}, it must be coherently with σ_{x_1}. Thus, ψ_8 is false in structure $\mathscr{M}_1$, since a cannot achieve p more than once.

Formula ψ_8 distinguishes between structures $\mathscr{M}_1$ and $\mathscr{M}_2$ from Fig.2 (Note that in this second structure the choices are not deterministic: from a given state a choice may be compatible with several potential successors). In $\mathscr{M}_2$, ψ_8 is true at s_0 since the strategy *always play* c_1 ensure the execution to remain in state s_0 or s_1 and is always coherent with strategy *play* c_2 *first and then always play* c_1, which ensures $\mathbf{X}\ p$ from states s_0 and s_1. What is at stake with it is the difference between *sustainable capabilities* and *one shot capabilities*. Formulas ψ_7 and ψ_8 both formalize the natural language proposition *a can always achieve p*. One shot capability (ψ_7) means she can achieve it once for all and choose when. Sustainable capability (ψ_8) means she can achieve it and choose when without affecting nor losing this capability for the future.

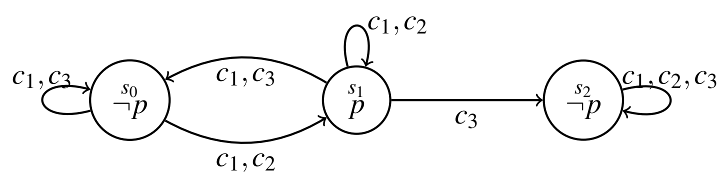

Figure 2: Structure $\mathscr{M}_2$

In Sect.3, we compare the expressive power of SL and USL by use of formula ψ_9, obtained from ψ_7 by adding to a the sustainable capability to ensure $\mathbf{X}\ \neg p$:

$$\psi_9 := \langle\langle x\rangle\rangle(a \triangleright x)\Box(\langle\langle x_0\rangle\rangle(a \triangleright x_0)\mathbf{X}\ p \wedge \langle\langle x_0\rangle\rangle(a \triangleright x_0)\mathbf{X}\ \neg p)$$

ψ_9 states that a has sustainable capability to decide whether p or $\neg p$ holds at next state. We say that a has *sustainable* control on property p: she is sustainably capable to decide the truth value of p.

The main purposes of USL are to give a formalism for the composition of strategies and to unify it with the classical branching-time mechanisms of strategy revocation. So, both treatments can be combined in a single formalism. In the remaining of this paper we define USL syntax and semantics, and we introduce the comparison of its expressive power with that of SL.

2 Syntax and semantics

In this section we present the syntax and semantics of USL, together with the related definitions they require. The USL formulas distinguish between *path* and *state* formulas.

Definition 1. *Let Ag be a set of agents, At a set of propositions and X a set of variables, USL (Ag, At, X) is given by the following grammar:*

- *State formulas:* $\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle x \rangle\rangle \varphi \mid (A \rhd x)\psi \mid (A \not\rhd x)\psi$
- *Path formulas:* $\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \psi\, \mathbf{U}\, \psi \mid \mathbf{X}\, \psi$

where $p \in At, A \subseteq Ag, x \in X$.

These formulas hold a notion of *free* variable that is similar to that in [8, 10]: an atom has an empty set of free variables, a binder adds a free variable to the set of free variables of its direct subformula and a quantifier deletes it. Upon formulas on this grammar, those that can be evaluated with no context are the *sentences*. They are formulas with empty set of free variables, which means each of their bound variables is previously quantified. We now come to the definitions for USL semantics.

Definition 2. *A Non-deterministic Alternating Transition System (NATS) is a tuple* $\mathscr{M} = \langle Ag, M, At, v, Ch \rangle$ *where:*

- *M is a set of states, called the domain of the NATS, At is the set of atomic propositions and v is a valuation function, from M to $\mathscr{P}(At)$.*
- *Ch: $Ag \times M \to \mathscr{P}(\mathscr{P}(M))$ is a choice function mapping a pair $(agent, state)$ to a non-empty family of choices of possible next states. It is such that for every state $s \in M$ and for every agents a_1 and a_2 in Ag, for every $c_1 \in Ch(a_1, s)$ and $c_2 \in Ch(a_2, s), c_1 \cap c_2 \neq \emptyset$.*

We call a finite sequence of states in M a *track* τ. The last element of a track τ is denoted by $last(\tau)$. The set of tracks that are possible in $\mathscr{M}$ is denoted by $track_{\mathscr{M}}$: $\tau = s_0 s_1 \ldots s_k \in track_{\mathscr{M}}$ iff for every $i < k$, for every $a \in Ag$, there is $c_a \in \mathscr{P}(M)$ s.t. $c_a \in Ch(a, s_i)$ and $s_{i+1} \in c_a$. Similarly, an infinite sequence of states such that all its prefixes are in $track_{\mathscr{M}}$ is called a *path* (in $\mathscr{M}$).

Definition 3 (Strategies and coherence)**.** *A* strategy *is a function σ from $Ag \times track_{\mathscr{M}}$ to $\mathscr{P}(M)$ such that for all $(a, \tau) \in Ag \times track_{\mathscr{M}}, \sigma(a, \tau) \in Ch(a, last(\tau))$. By extension, we write $\sigma(A, \tau)$ for $\bigcap_{a \in A} \sigma(a, \tau)$ for every $A \subseteq Ag$. Two strategies σ_1 and σ_2 are* coherent *iff for all (a, τ) in $Ag \times track_{\mathscr{M}}, \sigma_1(a, \tau) \cap \sigma_2(a, \tau) \neq \emptyset$. In this case, we also say that $\sigma_1(a, \tau)$ and $\sigma_2(a, \tau)$ are* coherent choices.

A *commitment* κ is a finite sequence upon $(\mathscr{P}(Ag) \times X)$, representing the active bindings. An *assignment* α is a partial function from X to *Strat*. A *context* χ is a pair of an assignment and a commitment. Note that an agent can appear several times in a commitment. Furthermore commitments store the order in which pairs (A, x) are introduced. Therefore our notion of contexts differs from the notion of *assignments* that is used in SL [8, 10].

A context defines a function from $track_{\mathscr{M}}$ to $\mathscr{P}(M)$. We use the same notation for the context itself and its induced function. Let $\kappa_\emptyset$ be the empty sequence upon $(\mathscr{P}(Ag) \times X)$, then:

- $(\alpha, \kappa_\emptyset)(\tau) = M$
- $(\alpha, (A, x))(\tau) =$
  - $\bigcap_{a \in A} \alpha(x)(a, \tau)$ if $A \neq \emptyset$
  - else M
- $(\alpha, \kappa \cdot (A, x))(\tau) =$

- $(\alpha,\kappa)(\tau)\cap(\alpha,(A,x))(\tau)$ if this intersection is not empty.
- otherwise (which means the context induces contradictory choices), $(\alpha,\kappa)(\tau)$.

Now we can define the outcomes of a context χ, *out* (χ): let $\pi=\pi_0,\pi_1,\dots$ be an infinite sequence over M, then $\pi\in out(s,\chi)$ iff π is a path in $\mathscr{M}$, $s=\pi_0$ and for every $n\in\mathbb{N}$, $\pi_{n+1}\in\chi(\pi_0\dots\pi_n)$.

Definition 4 (Strategy and assignment translation)**.** *Let σ be a strategy and τ be a track. Then σ^τ is the strategy s.t. for every $\tau'\in track_{\mathscr{M}}$, $\sigma^\tau(\tau')=\sigma(\tau\tau')$. The notion is extended to an assignment: for every α,α^τ is the assignment with domain equal to that of α and s.t. for every $x\in dom(\alpha),\alpha^\tau(x)=(\alpha(x))^\tau$*

We also define the following transformations of commitments and assignments. Given a commitment κ, coalitions A and B, a strategy variable x, an assignment α and a strategy σ:

- $\kappa[A\to x]=\kappa\cdot(A\rhd x)$
- $((B,x)\cdot\kappa)[A\nrightarrow x]=(B\backslash A,x)\cdot(\kappa[A\nrightarrow x])$ and $\kappa_\emptyset[A\nrightarrow x]=\kappa_\emptyset$
- $\alpha[x\to\sigma]$ is the assignment with domain $dom(\alpha)\cup\{x\}$ s.t. $\forall y\in dom(\alpha)\backslash\{x\},\alpha[x\to\sigma](y)=\alpha(y)$ and $\alpha[x\to\sigma](x)=\sigma$

Definition 5 (Satisfaction relation)**.** *Let $\mathscr{M}$ be a NATS, then for every assignment α, commitment κ, state s and path π:*

- *State formulas:*
  - $\mathscr{M},\alpha,\kappa,s\models p$ *iff* $p\in v(s)$, *with* $p\in At$
  - $\mathscr{M},\alpha,\kappa,s\models\neg\varphi$ *iff it is not true that* $\mathscr{M},\alpha,\kappa,s\models\varphi$
  - $\mathscr{M},\alpha,\kappa,s\models\varphi_1\wedge\varphi_2$ *iff* $\mathscr{M},\alpha,\kappa,s\models\varphi_1$ *and* $\mathscr{M},\alpha,\kappa,s\models\varphi_2$
  - $\mathscr{M},\alpha,\kappa,s\models\langle\langle x\rangle\rangle\varphi$ *iff there is a strategy* $\sigma\in Strat$ *s.t.* $\mathscr{M},\alpha[x\to\sigma],\kappa,s\models\varphi$
  - $\mathscr{M},\alpha,\kappa,s\models(A\rhd x)\varphi$ *iff for every* π *in* $out(\alpha,\kappa[A\to x]),\mathscr{M},\alpha,\kappa[A\to x],\pi\models\varphi$
  - $\mathscr{M},\alpha,\kappa,s\models(A\not\rhd x)\varphi$ *iff for all* π *in* $out(\alpha,\kappa[A\nrightarrow x]),\mathscr{M},\alpha,\kappa[A\nrightarrow x],\pi\models\varphi$
- *Path formulas :*
  - $\mathscr{M},\alpha,\kappa,\pi\models\varphi$ *iff* $\mathscr{M},\alpha,\kappa,\pi_0\models\varphi$, *for every state formula* φ
  - $\mathscr{M},\alpha,\kappa,\pi\models\neg\psi$ *iff it is not true that* $\mathscr{M},\alpha,\kappa,\pi\models\psi$
  - $\mathscr{M},\alpha,\kappa,\pi\models\psi_1\wedge\psi_2$ *iff* $\mathscr{M},\alpha,\kappa,\pi\models\psi_1$ *and* $\mathscr{M},\alpha,\kappa,\pi\models\psi_2$
  - $\mathscr{M},\alpha,\kappa,\pi\models\mathbf{X}\ \psi$ *iff* $\mathscr{M},\alpha^{\pi_0},\kappa,\pi^1\models\psi$.
  - $\mathscr{M},\alpha,\kappa,\pi\models\psi_1\ \mathbf{U}\ \psi_2$ *iff there is* $i\in\mathbb{N}$ *s.t.* $\mathscr{M},\alpha^{\pi_0\dots\pi_{i-1}},\kappa,\pi^i\models\psi_2$ *and for every* $0\le j<i,\mathscr{M},\alpha^{\pi_0\dots\pi_{j-1}},\kappa,\pi^j\models\psi_1$

Let $\alpha_\emptyset$ be the unique assignment with empty domain. Let φ be a sentence in USL (Ag,At,X). Then $\mathscr{M},s\models\varphi$ iff $\mathscr{M},\alpha_\emptyset,\kappa_\emptyset\models\varphi$.

Let us give the following comment over these definitions: for every context $\chi=(\alpha,\kappa)$, the definition of *out* (χ) ensures that the different binders encoded in χ compose their choices together, *as far as possible*. In case two contradictory choices from an agent are encoded in the context, the priority is given to the first binding that was introduced in this context (the left most binding in the formula). This guarantees that a formula requiring the composition of two contradictory strategies is false. For example, suppose that $\langle\langle x_1\rangle\rangle(a\rhd x_1)\varphi_1$ and $\langle\langle x_2\rangle\rangle(a\rhd x_2)\varphi_2$ are both true in a state of a model, and suppose that strategies σ_{x_1} and σ_{x_2} necessarily rely on contradictory choices of a (this means that a cannot play in a way that ensures both φ_1 and φ_2). Then, $\langle\langle x_1\rangle\rangle(a\rhd x_1)(\varphi_1\wedge\langle\langle x_2\rangle\rangle(a\rhd x_2)\varphi_2)$ is false in the same state of the same model. If the priority was given to the most recent binding (right most binding in the formula), the strategy σ_{x_1} would be revoked and the formula would be satisfied.

3 Comparison with SL [8, 10]

SL syntax can be basically described from SL by deleting the use of the unbinder. Furthermore, the binders are limited to sole agents and are written (a,x) instead of $(a \rhd x)$. USL appears to be more expressive than SL [8, 10]. More precisely, SL can be embedded in USL, while ψ_9 is not expressible in SL, even by extending its semantics to non-deterministic models. Here we give the three related propositions. By lack of space, the proofs are only sketched in this article. Detailed proofs of these propositions can be found in [4]. Note that, since SL is strictly more expressive than $\mathrm{ATL_{sc}}$ [6], the following results also hold for comparing USL with $\mathrm{ATL_{sc}}$.

Proposition 1. *There is an embedding of SL into USL.*

Proof (Sketch). The embedding consists in a parallel transformation from SL models and formulas to that of USL. The transformation preserves the satisfaction relation. The differences between SL and USL lie both in the definition of strategies in SL semantics and the difference of interpretation for the binding operator. The first is treated by defining an internal transformation for SL. By this transformation, the constraints of agents playing the same choices, issued from SL actions framework, are expressed in the syntax. Then we define a new operator in USL that is equivalent to SL binding, and show the equivalence: the operator $[a \rhd x]$ is an abbreviation for a binder $(a \rhd x)$ preceded by the set of unbinders $(a \not\rhd x_i)$, one for every variable x_i in the language. □

Proposition 2. *A model is said* deterministic *if the successor of a state is uniquely determined by one choice for every agent. Then, sustainable control is not expressible over deterministic models, neither in SL nor in USL.*

Proof (Sketch). One checks that for every deterministic NATS $\mathscr{M}$, for any state s of $\mathscr{M}$, $\mathscr{M}, s \not\models \psi_9$. Proposition 1 then straightly brings proposition 2 □

Proposition 3. *Sustainable control is not expressible in SL interpreted over NATSs.*

Proof (Sketch). The proof uses a generalization of SL semantics over *NATSs*. Its definition is in [4] and holds, for example, the following cases:

- $\mathscr{M}, \alpha, \kappa, s \models_{\mathrm{NATS}} \mathbf{X}\, \varphi$ iff for every $\pi \in out(s,(\alpha,\kappa)), \mathscr{M}, \alpha^{\pi_0}, \kappa, \pi_1 \models_{\mathrm{NATS}} \varphi$
- $\mathscr{M}, \alpha, \kappa, \pi \models_{\mathrm{NATS}} \varphi_1\ \mathbf{U}\ \varphi_2$ iff for every $\pi \in out(s,(\alpha,\kappa))$, there is $i \in \mathbb{N}$ s.t. $\mathscr{M}, \alpha^{\pi_0 \dots \pi_{i-1}}, \kappa, \pi^i \models_{\mathrm{NATS}} \varphi_2$ and for all $0 \le j \le i, \mathscr{M}, \alpha^{\pi_0 \dots \pi_{i-1}}, \kappa, \pi^j \models_{\mathrm{NATS}} \varphi_1$.
- $\mathscr{M}, \alpha, \kappa, s \models_{\mathrm{NATS}} \langle\langle x \rangle\rangle \varphi$ iff there is a strategy $\sigma \in Strat$ s.t. $\mathscr{M}, \alpha[x \to \sigma], \kappa, s \models_{\mathrm{NATS}} \varphi$.
- $\mathscr{M}, \alpha, \kappa, s \models_{\mathrm{NATS}} (a,x)\varphi$ iff $\mathscr{M}, \alpha, \kappa[x \backslash \kappa(a)], s \models_{\mathrm{NATS}} \varphi$.

where $\kappa[x \backslash \kappa(a)]$ designates the context obtained from κ by replacing every (a,y) in it by (a,x).

Formula ψ_9 states that a can always control whether p or not. Suppose there is a formula φ in SL equivalent to ψ_9 and let us call *existential* a formula in SL in which every occurrence of $\langle\langle x \rangle\rangle$ is under an even number of quantifiers. If φ is existential then under binary trees it is equivalent to a formula in Σ_1^1 (the fragment of second order logic with only existential set quantifiers).

We now consider a set of formulas $\{\Gamma_i\}_{i \in \mathbb{N}}$, each one stating that a can choose i times between p and $\neg p$. The set $\{\Gamma_i\}_{i \in \mathbb{N}}$ is defined by induction over i:

- $\Gamma_0 := \langle\langle x \rangle\rangle (a,x) \Box (\langle\langle x_0 \rangle\rangle (a,x_0) \mathbf{X}\, p \wedge \langle\langle x_0 \rangle\rangle (a,x_0) \mathbf{X}\, \neg p)$

- for all $i \in \mathbb{N}, \Gamma i+1 = \Gamma_i[p \wedge \Box(\langle\langle x_{i+1}\rangle\rangle(a,x_{i+1})\mathbf{X}\ p \wedge \langle\langle x_{i+1}\rangle\rangle(a,x_{i+1})\mathbf{X}\ \neg p \backslash p]$ $[\neg p \wedge \Box(\langle\langle x_{i+1}\rangle\rangle(a,x_{i+1})\mathbf{X}\ p \wedge \langle\langle x_{i+1}\rangle\rangle(a,x_{i+1})\mathbf{X}\ \neg p) \backslash \neg p]$.

where the notation $\theta_1[\theta_2 \backslash \theta_3]$ designates the formula obtained from θ_1 by replacing any occurrence of subformula θ_3 in it by θ_2. $\{\Gamma_i\}_{i\in\mathbb{N}}$ is equivalent to φ. A compactness argument shows that it is not equivalent to a formula in Σ_1^1 under binary trees, hence φ is not an existential formula. Then, we notice that φ is true in structures where, from any state, a can ensure any labelling of sequences over p. So, if φ has a subformula $(a,x)\psi$ where x is universally quantified, ψ must be equivalent to $\Box(p \vee \neg p)$. Then, by iteration, φ is equivalent to an existential formula in SL. Hence a contradiction. $\Box$

4 Conclusion

In this article we defined a strategy logic with updatable strategies. By updating a strategy, agents remain playing along it but add further precision to their choices. This mechanism enables to express such properties as sustainable capability and sustainable control. To the best of our knowledge, this is the first proposition for expressing such properties. Especially, the comparison introduced with SL in this article could be adapted to ATL with Strategy Context [3].

The revocation of strategies is also questioned in [1]. The authors propose a formalism with definitive strategies, that completely determine the behaviour of agents. They also underline the difference between these strategies and revocable strategies in the classical sense. We believe that updatable strategies offer a synthesis between both views: updatable strategies can be modified without being revoked.

Strategies in USL can also be explicitly revoked. This idea is already present in [3] with the operator $\cdot\rangle A\langle\cdot$. But the operator $\langle\cdot A\cdot\rangle$ also implicitly unbinds current strategy for agents in A before binding them a new strategy. Thus it prevents agents from updating their strategy or composing several strategies.

Further study perspectives about USL mainly concern the model checking. Further work will provide it with a proof of non elementary decidability, adapted from the proof in [10]. We are also working on a semantics for USL under memory-less strategies and **PSPACE** algorithm for its model-checking. Satisfiability problem should also be addressed. Since SL SAT problem is not decidable, similar result is expectable for USL. Nevertheless, decidable fragments of USL may be studied in the future, in particular by following the directions given in [9].

References

[1] Thomas Ågotnes, Valentin Goranko & Wojciech Jamroga (2007): *Alternating-time temporal logics with irrevocable strategies*. In: Theoretical aspects of rationality and knowledge, pp. 15–24, doi:10.1145/1324249.1324256.

[2] Rajeev Alur, Thomas A. Henzinger & Orna Kupferman (2002): *Alternating-time temporal logic*. J. ACM 49(5), pp. 672–713, doi:10.1145/585265.585270.

[3] T. Brihaye, A. Da Costa, F. Laroussinie & N. Markey (2009): *ATL with strategy contexts and bounded memory*. Logical Foundations of Computer Science, pp. 92–106, doi:10.1007/978-3-540-92687-0_7.

[4] Christophe Chareton, Julien Brunel & David Chemouil (2013): *Updatable Strategy Logic*. hal-00785659. Available at `http://hal.archives-ouvertes.fr/hal-00785659`. Submitted.

[5] Krishnendu Chatterjee, Thomas A. Henzinger & Nir Piterman (2010): *Strategy logic*. Inf. & Comp. 208(6), pp. 677–693, doi:10.1016/j.ic.2009.07.004.

[6] Arnaud Da Costa Lopes (2011): *Propriétés de jeux multi-agents*. Phd thesis, École normale supérieure de Cachan.

[7] Arnaud Da Costa Lopes, François Laroussinie & Nicolas Markey (2010): *ATL with Strategy Contexts: Expressiveness and Model Checking*. In: *FSTTCS*, pp. 120–132, doi:10.4230/LIPIcs.FSTTCS.2010.120.

[8] Fabio Mogavero, Aniello Murano, Giuseppe Perelli & Moshe Y. Vardi (2011): *Reasoning About Strategies: On the Model-Checking Problem*. *CoRR* abs/1112.6275. Available at `http://arxiv.org/abs/1112.6275`.

[9] Fabio Mogavero, Aniello Murano, Giuseppe Perelli & Moshe Y. Vardi (2012): *What Makes Atl* Decidable? A Decidable Fragment of Strategy Logic*. In: *CONCUR*, pp. 193–208, doi:10.1007/978-3-642-32940-1_15.

[10] Fabio Mogavero, Aniello Murano & Moshe Y. Vardi (2010): *Reasoning about strategies*. In: *FSTTCS*, 8, pp. 133–144, doi:10.4230/LIPIcs.FSTTCS.2010.133.