Refining Strategies in Strategy Logic

Abstract

Nondeterministic strategies are strategies (or protocols, or plans) that, given a history in a game, assign a set of possible actions, all of which are winning. An important problem is that of refining such strategies. For instance, given a nondeterministic strategy that allows only safe executions, refine it to, additionally, eventually reach a desired state of affairs. We show that such problems can be solved elegantly in the framework of Strategy Logic (SL), a very expressive logic to reason about strategic abilities. Specifically, we introduce a variant of SL with nondeterministic strategies and a strategy refinement operator. We show that model checking this logic can be done at no additional computational cost with respect to standard SL, and can be used to solve problems such as module checking, synthesis, synthesis under constraints, and more.

1 Introduction

2 Nondeterministic strategies

[put nondeterministic strategies and refinement here - **Bastien**]

[maximal permissive strategies: look at litterature - **Bastien**]

3 Strategy Logic with refinement

In this section we introduce $\mathsf{SL}^{\prec}$, which extends SL with nondeterministic strategies, an *outcome quantifier* that quantifies over possible outcomes of a strategy profile, and more importantly, a refining operator that expresses that a strategy refines another. We first fix some basic notations.

3.1 Notations

Let Σ be an alphabet. A *finite* (resp. *infinite*) *word* over Σ is an element of Σ^* (resp. Σ^ω). The *length* of a finite word $w = w_0 w_1 \dots w_n$ is $|w| := n+1$, and $\mathrm{last}(w) := w_n$ is its last letter. Given a finite (resp. infinite) word w and $0 \leq i < |w|$ (resp. $i \in \mathbb{N}$), we let w_i be the letter at position i in w, $w_{\leq i}$ is the prefix of w that ends at position i and $w_{\geq i}$ is the suffix that starts at position i. We write $w \preccurlyeq w'$ if w is a prefix of w', and $\mathit{pref}(w)$ is the set of finite prefixes of word w. Finally, the domain of a mapping f is written $\mathit{dom}(f)$.

3.2 Syntax

For convenience we fix for the rest of the paper AP, a finite non-empty set of *atomic propositions*, Ag, a finite non-empty set of *agents* or *players*, and Var, a finite non-empty set of *variables*.

Definition 1. The syntax of $\mathsf{SL}^{\prec}$ is defined by the following grammar:

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \exists^{\mathrm{d}} x\, \varphi \mid x \preceq y \mid (a,x)\varphi \mid \mathbf{E}\psi$$
$$\psi := \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi \mathbf{U} \psi$$

where $p \in \mathrm{AP}$, $x, y \in \mathrm{Var}$ and $a \in \mathrm{Ag}$.

Formulas of type φ are called *state formulas*, those of type ψ are called *path formulas*, and $\mathsf{SL}^{\prec}$ consists of all state formulas.

Temporal operators, $\mathbf{X}$ (next) and $\mathbf{U}$ (until), have the usual meaning. The *refinement operator* expresses that the strategy denoted by a variable x is more restrictive than another one, or that it allows less behaviours: $x \preceq y$ reads as "strategy x refines strategy y". The two *strategy quantifiers* $\exists x$ and $\exists^{\mathrm{d}} x$ quantify on nondeterministic and deterministic strategies, respectively: $\exists x\, \varphi$ (resp. $\exists^{\mathrm{d}} x\, \varphi$) reads as "there exists a nondeterministic (resp. deterministic) strategy x such that φ holds", where x is a strategy variable. As usual, the *binding operator* (a,x) assigns a strategy to an agent, and $(a,x)\varphi$ reads as "when agent a plays strategy x, φ holds". Finally, the *outcome quantifier* $\mathbf{E}$ quantifies on outcomes of strategies currently in use: $\mathbf{E}\psi$ reads as "ψ holds in some outcome of the strategies currently used by the players".

We use usual abbreviations $\top := p \vee \neg p$, $\bot := \neg\top$, $\varphi \to \varphi' := \neg\varphi \vee \varphi'$, $\varphi \leftrightarrow \varphi' := \varphi \to \varphi' \wedge \varphi' \to \varphi$, $\mathbf{F}\varphi := \top\mathbf{U}\varphi$, and $\mathbf{G}\varphi := \neg\mathbf{F}\neg\varphi$. In addition we define the universal variants of the two strategy quantifiers: let $\forall x\, \varphi := \neg\exists x\, \neg\varphi$ and $\forall^{\mathrm{d}} x\, \varphi := \neg\exists^{\mathrm{d}} x\, \neg\varphi$.

For every formula $\varphi \in \mathsf{SL}^{\prec}$, we let $\mathit{free}(\varphi)$ be the set of variables that appear free in φ, i.e., that appear out of the scope of a strategy quantifier. A formula φ is a *sentence* if *free*(φ) is empty. Finally, we let the *size* $|\varphi|$ of a formula φ be the number of symbols in φ.

3.3 Semantics

The models of $\mathsf{SL}^{\prec}$ are the usual models of Strategy Logic, i.e.concurrent game structures.

Definition 2. A *concurrent game structure* (or CGS) is a tuple $\mathcal{G} = (\mathrm{Ac}, V, E, \ell, v_\iota)$ where

- Ac is a finite non-empty set of *actions*,
- V is a finite non-empty set of *positions*,
- $E : V \times \mathrm{Ac}^{\mathrm{Ag}} \to V$ is a *transition function*,
- $\ell : V \to 2^{\mathrm{AP}}$ is a *labelling function*, and
- $v_\iota \in V$ is an *initial position.*

In a position $v \in V$, where atomic propositions $\ell(v)$ hold, each player a chooses an action $c_a \in \mathrm{Ac}$, and the game proceeds to position $E(v, \boldsymbol{c})$, where $\boldsymbol{c} \in \mathrm{Ac}^{\mathrm{Ag}}$ stands for the *joint action* $(c_a)_{a\in\mathrm{Ag}}$. Given a joint action $\boldsymbol{c} = (c_a)_{a\in\mathrm{Ag}}$ and $a \in \mathrm{Ag}$, we let $\boldsymbol{c}_a$ denote c_a. A *finite* (resp. *infinite*) *play* is a finite (resp. infinite) word $\rho = v_0 \dots v_n$ (resp. $\pi = v_0v_1 \dots$) such that $v_0 = v_\iota$ and for every i such that $0 \leq i < |\rho| - 1$ (resp. $i \geq 0$), there exists a joint action $\boldsymbol{c}$ such that $E(v_i, \boldsymbol{c}) = v_{i+1}$. Given two finite plays ρ and ρ', we say that ρ' is a *continuation* of ρ if $\rho' \in \rho \cdot V^*$, and we write $\mathrm{Cont}(\rho)$ for the set of continuations of ρ.

Strategies. A (nondeterministic) *strategy* is a function $\sigma : \mathrm{Cont}(v_\iota) \to 2^{\mathrm{Ac}} \setminus \emptyset$ that maps each finite play in $\mathcal{G}$ to a nonempty finite set of actions that the player may play. A strategy σ is *deterministic* if for every finite play ρ, $\sigma(\rho)$ is a singleton. We let *Str* denote the set of all (nondeterministic) strategies, and $\mathit{Str}^{\mathrm{d}} \subset \mathit{Str}$ the set of deterministic ones (note that these sets depend on the CGS under consideration).

Formulas of $\mathsf{SL}^{\prec}$ will be evaluated at the end of a finite play ρ (which can be simply the initial position of the game). And since our logic contains only *future-time* temporal operators, the only relevant part of a strategy σ when evaluating a formula after finite play ρ is its definition on continuations of ρ. We thus define the *restriction* of σ to ρ as the restriction of σ to $\rho \cdot V^+$, that we write $\sigma_{|\rho} : \mathrm{Cont}(\rho) \to 2^{\mathrm{Ac}} \setminus \emptyset$. We will then say that a strategy σ refines another strategy σ' after a finite play ρ if the first one is more restrictive than the second one on continuations of ρ. More formally:

Definition 3. Strategy σ *refines* strategy σ' after finite play ρ if, for every $\rho' \in \mathrm{Cont}(\rho)$, $\sigma_{|\rho}(\rho') \subseteq \sigma'_{|\rho}(\rho')$.

An *assignment* $\chi : \mathrm{Ag} \cup \mathrm{Var} \rightharpoonup \mathit{Str}$ is a partial function that assigns a strategy to each player and strategy variable in its domain. For an assignment χ, player a and strategy σ, $\chi[a \mapsto \sigma]$ is the assignment of domain $\mathit{dom}(\chi) \cup \{a\}$ that maps a to σ and is equal to χ on the rest of its domain, and $\chi[x \mapsto \sigma]$ is defined similarly, where x is a variable. An assignment is *variable-complete* for a formula $\varphi \in \mathsf{SL}^{\prec}$ if its domain contains all free variables of φ.

For an assignment χ and a finite play ρ, we let $\mathrm{Out}(\chi, \rho)$ be the set of infinite plays that start with ρ and are then extended by letting players follow the strategies assigned by χ. Formally, $\mathrm{Out}(\chi, \rho)$ is the set of plays of the form $\rho \cdot v_1v_2 \dots$ such that for all $i \geq 0$, there exists $\boldsymbol{c}$ such that for all $a \in \mathit{dom}(\chi) \cap \mathrm{Ag}$, $\boldsymbol{c}_a \in \chi(a)(\rho \cdot v_1 \dots v_i)$ and $v_{i+1} = E(v_i, \boldsymbol{c})$, with $v_0 = \mathrm{last}(\rho)$.

Definition 4. The semantics of a state formula is defined on a CGS $\mathcal{G}$, an assignment χ that is variable-complete for φ, and a finite play ρ. For a path formula ψ, the finite play is replaced with an infinite play π and an index $i \in \mathbb{N}$. The definition by mutual induction is as follows:

$\mathcal{G}, \chi, \rho \models p$	if	$p \in \ell(\mathrm{last}(\rho))$
$\mathcal{G}, \chi, \rho \models \neg\varphi$	if	$\mathcal{G}, \chi, \rho \not\models \varphi$
$\mathcal{G}, \chi, \rho \models \varphi \vee \varphi'$	if	$\mathcal{G}, \chi, \rho \models \varphi$ or $\mathcal{G}, \chi, \rho \models \varphi'$
$\mathcal{G}, \chi, \rho \models \exists x\, \varphi$	if	$\exists\, \sigma \in \mathit{Str}$ s.t. $\mathcal{G}, \chi[x \mapsto \sigma], \rho \models \varphi$
$\mathcal{G}, \chi, \rho \models \exists^{\mathrm{d}} x\, \varphi$	if	$\exists\, \sigma \in \mathit{Str}^{\mathrm{d}}$ s.t. $\mathcal{G}, \chi[x \mapsto \sigma], \rho \models \varphi$
$\mathcal{G}, \chi, \rho \models x \preceq y$	if	$\chi(x)$ refines $\chi(y)$ after ρ
$\mathcal{G}, \chi, \rho \models (a, x)\varphi$	if	$\mathcal{G}, \chi[a \mapsto \chi(x)], \rho \models \varphi$
$\mathcal{G}, \chi, \rho \models \mathbf{E}\psi$	if	$\exists \pi \in \mathrm{Out}(\chi, \rho)$ s.t. $\mathcal{G}, \chi, \pi, \lvert\rho\rvert - 1 \models \psi$
$\mathcal{G}, \chi, \pi, i \models \varphi$	if	$\mathcal{G}, \chi, \pi_{\leq i} \models \varphi$
$\mathcal{G}, \chi, \pi, i \models \neg\psi$	if	$\mathcal{G}, \chi, \pi, i \not\models \psi$
$\mathcal{G}, \chi, \pi, i \models \psi \vee \psi'$	if	$\mathcal{G}, \chi, \pi, i \models \psi$ or $\mathcal{G}, \chi, \pi, i \models \psi'$
$\mathcal{G}, \chi, \pi, i \models \mathbf{X}\psi$	if	$\mathcal{G}, \chi, \pi, i + 1 \models \psi$
$\mathcal{G}, \chi, \pi, i \models \psi\mathbf{U}\psi'$	if	$\exists\, j \geq i$ s.t. $\mathcal{G}, \chi, \pi, j \models \psi'$ and, $\forall\, k$ s.t. $i \leq k < j$, $\mathcal{G}, \chi, \pi, k \models \psi$

TODO: example

To model check $\mathsf{SL}^{\prec}$ we extend the classic approach, which is to reduce to QCTL^*, the extension of CTL^* with (second-order monadic) quantification on atomic propositions. This logic is equivalent to MSO on infinite trees [Laroussinie and Markey, 2014], and it is easy to express that a strategy (or the atomic propositions that code for it) refines another one.

4 Model checking $\mathsf{SL}^{\prec}$

We first recall briefly the syntax and semantics of QCTL^*, to which we will reduce $\mathsf{SL}^{\prec}$.

Definition 5. The syntax of QCTL^* is defined by the following grammar:

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{E}\psi \mid \exists p\, \varphi$$
$$\psi := \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi$$

where $p \in \mathrm{AP}$.

Again, formulas of type φ are called *state formulas*, those of type ψ are called *path formulas*, and QCTL^* consists of all the state formulas defined by the grammar, and we use standard abbreviation $\mathbf{A}\psi := \neg\mathbf{E}\neg\psi$.

The models of QCTL^* are classic Kripke structures:

Definition 6. A *Kripke structure*, or KS, over AP is a tuple $\mathcal{S} = (S, R, \ell, s_\iota)$ where

- S is a set of *states*,
- $R \subseteq S \times S$ is a left-total[1] *transition relation*,
- $\ell : S \to 2^{\mathrm{AP}}$ is a *labelling function* and
- $s_\iota \in S$ is an *initial state*.

A *path* in $\mathcal{S}$ is an infinite sequence of states $\lambda = s_0 s_1 \dots$ such that for all $i \in \mathbb{N}$, $(s_i, s_{i+1}) \in R$. A *finite path* is a finite non-empty prefix of a path. Similar to continuations of finite plays, given a finite path λ we write $\mathrm{Cont}(\lambda)$ for the set of finite paths that start with λ. We may write $s \in \mathcal{S}$ for $s \in S$, and we define the *size* $|\mathcal{S}|$ of a KS $\mathcal{S} = (S, R, s_\iota, \ell)$ as its number of states: $|\mathcal{S}| := |S|$.

Since we will interpret QCTL^* on unfoldings of KS, we now define infinite trees.

Trees. Let X be a finite set of *directions* (typically a set of states). An X-*tree* τ is a nonempty set of words $\tau \subseteq X^+$ such that **(1)** there exists $r \in X$, called the *root* of τ, such that each $u \in \tau$ starts with r ($r \preccurlyeq u$); **(2)** if $u \cdot x \in \tau$ and $u \cdot x \neq r$, then $u \in \tau$; **(3)** if $u \in \tau$ then there exists $x \in X$ such that $u \cdot x \in \tau$.

The elements of a tree τ are called *nodes*. If $u \cdot x \in \tau$, we say that $u \cdot x$ is a *child* of u. An X-tree τ is *complete* if for every $u \in \tau$ and $x \in X$, $u \cdot x \in \tau$. A *path* in τ is an infinite sequence of nodes $\lambda = u_0 u_1 \dots$ such that for all $i \in \mathbb{N}$, u_{i+1} is a child of u_i, and $Paths(u)$ is the set of paths that start in node u.

An AP-*labelled* X-*tree*, or (AP, X)-*tree* for short, is a pair $t = (\tau, \ell)$, where τ is an X-tree called the *domain* of t and $\ell : \tau \to 2^{\mathrm{AP}}$ is a *labelling*, which maps each node to the set of propositions that hold there. For $p \in \mathrm{AP}$, a p-*labelling* for a tree is a mapping $\ell_p : \tau \to \{0, 1\}$ that indicates in which nodes p holds, and for a labelled tree $t = (\tau, \ell)$, the p-labelling of t is the p-labelling $u \mapsto 1$ if $p \in \ell(u)$, 0 otherwise. The composition of a labelled tree $t = (\tau, \ell)$ with a p-labelling ℓ_p for τ is defined as $t \otimes \ell_p := (\tau, \ell')$, where $\ell'(u) = \ell(u) \cup \{p\}$ if $\ell_p(u) = 1$, and $\ell(u) \setminus \{p\}$ otherwise. A p-labelling for a labelled tree $t = (\tau, \ell)$ is a p-labelling for its domain τ. A *pointed labelled tree* is a pair (t, u) where u is a node of t.

Let $\mathcal{S} = (S, R, \ell, s_\iota)$ be a Kripke structure over AP. The *tree-unfolding of* $\mathcal{S}$ is the (AP, S)-tree $t_{\mathcal{S}} := (\tau, \ell')$, where τ is the set of all finite paths that start in s_ι, and for every $u \in \tau$, $\ell'(u) := \ell(\mathrm{last}(u))$.

Definition 7. We define by induction the satisfaction relation $\models$ of QCTL^*. Let $t = (\tau, \ell)$ be an AP-labelled tree, u a node and λ a path in τ:

$$
\begin{array}{lll}
t, u \models p & \text{if} & p \in \ell(u) \\
t, u \models \neg\varphi & \text{if} & t, u \not\models \varphi \\
t, u \models \varphi \vee \varphi' & \text{if} & t, u \models \varphi \text{ or } t, u \models \varphi' \\
t, u \models \mathbf{E}\psi & \text{if} & \exists\, \lambda \in Paths(u) \text{ s.t. } t, \lambda \models \psi \\
t, u \models \exists p\, \varphi & \text{if} & \exists\, \ell_p \text{ a } p\text{-labelling for } t \text{ s.t.} \\
 & & \qquad t \otimes \ell_p, u \models \varphi \\
t, \lambda \models \varphi & \text{if} & t, \lambda_0 \models \varphi \\
t, \lambda \models \neg\psi & \text{if} & t, \lambda \not\models \psi \\
t, \lambda \models \psi \vee \psi' & \text{if} & t, \lambda \models \psi \text{ or } t, \lambda \models \psi' \\
t, \lambda \models \mathbf{X}\psi & \text{if} & t, \lambda_{\geq 1} \models \psi \\
t, \lambda \models \psi \mathbf{U} \psi' & \text{if} & \exists\, i \geq 0 \text{ s.t. } t, \lambda_{\geq i} \models \psi' \text{ and} \\
 & & \qquad \forall j \text{ s.t. } 0 \leq j < i,\ t, \lambda_{\geq j} \models \psi
\end{array}
$$

We write $t \models \varphi$ for $t, r \models \varphi$, where r is the root of t. Given a KS $\mathcal{S}$ and a QCTL^* formula φ, we also write $\mathcal{S} \models \varphi$ if $t_{\mathcal{S}} \models \varphi$.

[define alternation depth - **Bastien**]

Theorem 1. *The model-checking problem for* QCTL^* *is* $k+1$-EXPTIME-*complete for formulas of alternation depth* k.

4.1 Reduction to QCTL^*

We use a variant of the reductions presented in [Laroussinie and Markey, 2015; Fijalkow *et al.*, 2018; Berthon *et al.*, 2017; Maubert and Murano, 2018; Bouyer *et al.*, 2019], which transform instances of the model-checking problem for various strategic logics to (extensions of) QCTL^*.

Let $(\mathcal{G}, \Phi)$ be an instance of the SL model-checking problem, and assume without loss of generality that each strategy variable is quantified at most once in Φ. We define an equivalent instance of the model-checking problem for QCTL^*.

[say that the reduction preserves alternation depth - **Bastien**]

Define the KS $\mathcal{S}_{\mathcal{G}} := (S, R, s_\iota, \ell')$ where

- $S := \{s_v \mid v \in V\}$,
- $R := \{(s_v, s_{v'}) \mid \exists \boldsymbol{c} \in \mathrm{Ac}^{\mathrm{Ag}} \text{ s.t. } E(v, \boldsymbol{c}) = v'\} \subseteq S^2$,
- $s_\iota := s_{v_\iota}$, and
- $\ell'(s_v) := \ell(v) \cup \{p_v\} \subseteq \mathrm{AP} \cup \mathrm{AP}_v$.

For every finite play $\rho = v_0 \dots v_k$, define the node $u_\rho := s_{v_0} \dots s_{v_k}$ in $t_{\mathcal{S}_{\mathcal{G}}}$. Note that the mapping $\rho \mapsto u_\rho$ defines a bijection between the set of finite plays and the set of nodes in $t_{\mathcal{S}_{\mathcal{G}}}$.

Constructing the QCTL^* formulas $(\varphi)^f_s$. We now describe how to transform an $\mathsf{SL}^{\prec}$ formula φ and a partial function $f : \mathrm{Ag} \rightharpoonup \mathrm{Var}$ into a QCTL^* formula $(\varphi)^f_s$ (that will also depend on $\mathcal{G}$). Suppose that $\mathrm{Ac} = \{c_1, \dots, c_l\}$, and define $(\varphi)^f_s$ and $(\psi)^f_p$ by mutual induction on state and path formulas. The base cases are as follows: $(p)^f_s := p$ and $(\varphi)^f_p := (\varphi)^f_s$. Boolean and temporal operators are simply obtained by distributing the translation: $(\neg\varphi)^f_s := \neg(\varphi)^f_s$, $(\neg\psi)^f_p := \neg(\psi)^f_p$,

[1] i.e., for all $s \in S$, there exists s' such that $(s, s') \in R$.

$(\varphi_1 \vee \varphi_2)^f_s := (\varphi_1)^f_s \vee (\varphi_2)^f_s$, $(\psi_1 \vee \psi_2)^f_p := (\psi_1)^f_p \vee (\psi_2)^f_p$, $(\mathbf{X}\psi)^f_p := \mathbf{X}(\psi)^f_p$ and $(\psi_1 \mathbf{U} \psi_2)^f_p := (\psi_1)^f_p \mathbf{U} (\psi_2)^f_p$.

We continue with the strategy quantifiers:

$$\begin{array}{rll} & (\exists x\, \varphi)^f_s & := \exists p^x_{c_1} \ldots \exists^{\widetilde{o}} p^x_{c_l}.\varphi_{\mathrm{str}}(x) \wedge (\varphi)^f_s \\ \text{where} & \varphi_{\mathrm{str}}(x) & := \mathbf{AG} \bigvee_{c \in \mathrm{Ac}} p^x_c, \quad \text{and} \\ & (\exists^{\mathrm{d}} x\, \varphi)^f_s & := \exists p^x_{c_1} \ldots \exists^{\widetilde{o}} p^x_{c_l}.\varphi^{\mathrm{det}}_{\mathrm{str}}(x) \wedge (\varphi)^f_s \\ \text{where} & \varphi^{\mathrm{det}}_{\mathrm{str}}(x) & := \mathbf{AG} \bigvee_{c \in \mathrm{Ac}} (p^x_c \wedge \bigwedge_{c' \neq c} \neg p^x_{c'}). \end{array}$$

The intuition is that for each possible action $c \in \mathrm{Ac}$, an existential quantification on the atomic proposition p^x_c "chooses" for each node u_ρ of the tree $t_{\mathcal{S}_\mathcal{G}}$ whether strategy x allows action c in ρ or not. $\varphi_{\mathrm{str}}(x)$ checks that at least one action is allowed in each node, and thus that atomic propositions p^x_c indeed define a (nondeterministic) strategy. $\varphi^{\mathrm{det}}_{\mathrm{str}}(x)$ instead ensures that *exactly one* action is chosen for strategy x in each finite play, and thus that atomic propositions p^x_c characterise a deterministic strategy.

For strategy refinement, the translation is as follows:

$$(x \preceq y)^f_s := \mathbf{AG} \bigwedge_{c \in \mathrm{Ac}} p^x_c \to p^y_c.$$

Here are the remaining cases:

$$\begin{array}{rll} & ((a,x)\varphi)^f_s & := (\varphi)^{f[a \mapsto x]}_s \qquad \text{for } x \in \mathrm{Var} \cup \{?\} \\ \text{and} & (\mathbf{E}\psi)^f_s & := \mathbf{E}\,(\psi^f_{\mathrm{out}} \wedge (\psi)^f_p), \text{ where} \end{array}$$

$$\psi^f_{\mathrm{out}} := \mathbf{G} \bigvee_{v \in V} \Big(p_v \wedge \bigvee_{\boldsymbol{c} \in \mathrm{Ac}^{\mathrm{Ag}}} (\bigwedge_{a \in dom(f)} p^{f(a)}_{\boldsymbol{c}_a} \wedge \mathbf{X}\, p_{E(v,\boldsymbol{c})}) \Big).$$

ψ^f_{out} checks that each player a in the domain of f follows the strategy coded by the $p^{f(a)}_c$.

To prove the correctness of the translation we need some additional definitions. First, given a strategy σ and a strategy variable x we let $\ell^x_\sigma := \{\ell_{p^x_c} \mid c \in \mathrm{Ac}\}$ be the family of p^x_c-labellings for tree $t_{\mathcal{S}_\mathcal{G}}$ defined as follows: for each finite play ρ and $c \in \mathrm{Ac}$, we let $\ell_{p^x_c}(u_\rho) := 1$ if $c \in \sigma(\rho)$, 0 otherwise. For a labelled tree t with same domain as $t_{\mathcal{S}_\mathcal{G}}$ we write $t \otimes \ell^x_\sigma$ for $t \otimes \ell_{p^x_{c_1}} \otimes \ldots \otimes \ell_{p^x_{c_l}}$.

Second, given an infinite play π and a point $i \in \mathbb{N}$, we let $\lambda_{\pi,i}$ be the infinite path in $t_{\mathcal{S}_\mathcal{G}}$ that starts in node $u_{\pi_{\leq i}}$ and is defined as $\lambda_{\pi,i} := u_{\pi_{\leq i}} u_{\pi_{\leq i+1}} u_{\pi_{\leq i+2}} \ldots$

Finally, we say that a partial function $f : \mathrm{Ag} \rightharpoonup \mathrm{Var}$ is *compatible* with an assignment χ if $dom(\chi) \cap \mathrm{Ag} = dom(f)$ and for all $a \in dom(f)$, $\chi(a) = \chi(f(a))$.

Proposition 2. *For every state subformula φ and path subformula ψ of Φ, finite play ρ, infinite play π, point $i \in \mathbb{N}$, for every assignment χ variable-complete for φ (resp. ψ) and partial function $f : \mathrm{Ag} \rightharpoonup \mathrm{Var}$ compatible with χ, assuming also that no x_i in $dom(\chi) \cap \mathrm{Var} = \{x_1, \ldots, x_k\}$ is quantified in φ or ψ, we have*

$$\mathcal{G}, \chi, \rho \models \varphi \quad \textit{iff} \quad t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi(x_k)}, u_\rho \models (\varphi)^f_s$$

$$\mathcal{G}, \chi, \pi, i \models \psi \quad \textit{iff} \quad t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi(x_k)}, \lambda_{\pi,i} \models (\psi)^f_p$$

In addition, $\mathcal{S}_\mathcal{G}$ is of size linear in $|\mathcal{G}|$, and $(\varphi)^f_s$ and $(\psi)^f_p$ are of size linear in $|\mathcal{G}|^2 + |\varphi|$.

Proof. The proof is by induction on φ. We detail the case for binding, strategy quantification, strategy refinement and outcome quantification, the others follow simply by definition of $\mathcal{S}_\mathcal{G}$ for atomic propositions and induction hypothesis for remaining cases.

For $\varphi = x \preceq y$, assume that $\mathcal{G}, \chi, \rho \models x \preceq y$. First, observe that since χ is variable-complete for φ, x and y are in $dom(\chi)$. Now we have that $\chi(x)_{|\rho}(\rho') \subseteq \chi(y)_{|\rho}(\rho')$ for every $\rho' \in \mathrm{Cont}(\rho)$. By definition of $\ell^x_{\chi(x)} = \{\ell_{p^x_c} \mid c \in \mathrm{Ac}\}$ and $\ell^y_{\chi(y)} = \{\ell_{p^y_c} \mid c \in \mathrm{Ac}\}$, it follows that for each $c \in \mathrm{Ac}$ and $\rho' \in \mathrm{Cont}(\rho)$, if $\ell_{p^x_c}(\rho') = 1$, then $\ell_{p^y_c}(\rho') = 1$, and thus

$$t_{\mathcal{S}_\mathcal{G}} \otimes \ell^x_{\chi(x)} \otimes \ell^y_{\chi(y)} \models \mathbf{AG} \bigwedge_{c \in \mathrm{Ac}} p^x_c \to p^y_c$$

Because the labellings $\ell^x_{\chi(x)}$ touch distinct sets of atomic propositions for each variable x in $dom(\chi) \cap \mathrm{Var}$, we can conclude this direction.

For the other direction let $t = t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi(x_k)}$ and assume that

$$t, u_\rho \models \mathbf{AG} \bigwedge_{c \in \mathrm{Ac}} p^x_c \to p^y_c.$$

This implies that for every $\rho' \in \mathrm{Cont}(\rho)$,

$$t, u_{\rho'} \models \bigwedge_{c \in \mathrm{Ac}} p^x_c \to p^y_c,$$

and thus $\chi(x)_{|\rho}$ refines $\chi(y)_{|\rho}$.

For $\varphi = (a,x)\varphi'$, we have $\mathcal{G}, \chi, \rho \models (a,x)\varphi'$ if and only if $\mathcal{G}, \chi[a \mapsto \chi(x)], \rho \models \varphi'$. The result follows by using the induction hypothesis with assignment $\chi[a \mapsto x]$ and function $f[a \mapsto x]$. This is possible because $f[a \mapsto x]$ is compatible with $\chi[a \mapsto x]$: indeed $dom(\chi[a \mapsto x]) \cap \mathrm{Ag}$ is equal to $dom(\chi) \cap \mathrm{Ag} \cup \{a\}$ which, by assumption, is equal to $dom(f) \cup \{a\} = dom(f[a \mapsto x])$. Also by assumption, for all $a' \in dom(f)$, $\chi(a') = \chi(f(a'))$, and by definition $\chi[a \mapsto \chi(x)](a) = \chi(x) = \chi(f[a \mapsto x](a))$.

For $\varphi = \exists x\, \varphi'$, assume first that $\mathcal{G}, \chi, \rho \models \exists x\, \varphi'$. There exists a nondeterministic strategy σ such that

$$\mathcal{G}, \chi[x \mapsto \sigma], \rho \models \varphi'.$$

Since f is compatible with χ, it is also compatible with assignment $\chi' = \chi[x \mapsto \sigma]$. By assumption, no variable in $\{x_1, \ldots, x_k\}$ is quantified in φ, so that $x \neq x_i$ for all i, and thus $\chi'(x_i) = \chi(x_i)$ for all i; and because no strategy variable is quantified twice in a same formula, x is not quantified in φ', so that no variable in $\{x_1, \ldots, x_k, x\}$ is quantified in φ'. By induction hypothesis

$$t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi'(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi'(x_k)} \otimes \ell^x_{\chi'(x)}, u_\rho \models (\varphi')^f_s.$$

It follows that

$$t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi'(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi'(x_k)}, u_\rho \models \exists^{\widetilde{o}} p^x_{c_1} \ldots \exists p^x_{c_l}.\varphi_{\mathrm{str}}(x) \wedge (\varphi')^f_s.$$

Finally, since $\chi'(x_i) = \chi(x_i)$ for all i, we conclude that

$$t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi(x_k)}, u_\rho \models (\exists x\, \varphi')^f_s.$$

For the other direction, assume that

$$t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi(x_k)}, u_\rho \models (\varphi)^f_s,$$

and recall that $(\varphi)^f_s = \exists^{\widetilde{o}} p^x_{c_1} \ldots \exists^{\widetilde{o}} p^x_{c_l} . \varphi_{\text{str}}(x) \wedge (\varphi')^f_s$. Write $t = t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi(x_k)}$. There exist $\ell_{p^x_c}$-labellings such that

$$t \otimes \ell_{p^x_{c_1}} \otimes \ldots \otimes \ell_{p^x_{c_l}} \models \varphi_{\text{str}}(x) \wedge (\varphi')^f_s.$$

By $\varphi_{\text{str}}(x)$, these labellings code for a strategy σ. Let $\chi' = \chi[x \mapsto \sigma]$. For all $1 \leq i \leq k$, by assumption $x \neq x_i$, and thus $\chi'(x_i) = \chi(x_i)$. The above can thus be rewritten

$$t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi'(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi'(x_k)} \otimes \ell^{x}_{\chi'(x)} \models \varphi_{\text{str}}(x) \wedge (\varphi')^f_s.$$

By induction hypothesis we have $\mathcal{G}, \chi[x \mapsto \sigma], \rho \models \varphi'$, hence $\mathcal{G}, \chi, \rho \models \exists x\, \varphi'$.

For $\varphi = \exists^{\text{d}} x\, \psi$, the proof is similar, using $\varphi^{\text{det}}_{\text{str}}(x)$ instead of $\varphi_{\text{str}}(x)$.

For $\varphi = \mathbf{E}\psi$, assume first that $\mathcal{G}, \chi, \rho \models \mathbf{E}\psi$. There exists a play $\pi \in \text{Out}(\chi, \rho)$ s.t. $\mathcal{G}, \chi, \pi, |\rho| - 1 \models \psi$. By induction hypothesis,

$$t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi(x_k)}, \lambda_{\pi,|\rho|-1} \models (\psi)^f_p.$$

Since π is an outcome of χ, each agent $a \in dom(\chi) \cap \text{Ag}$ follows strategy $\chi(a)$ in π. Because $dom(\chi) \cap \text{Ag} = dom(f)$ and for all $a \in dom(f)$, $\chi(a) = \chi(f(a))$, each agent $a \in dom(f)$ follows the strategy $\chi(f(a))$, which is coded by atoms $p^{f(a)}_c$ in the translation of Φ. Therefore $\lambda_{\pi,|\rho|-1}$ also satisfies ψ^{χ}_{out}, hence

$$t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi(x_k)}, \lambda_{\pi,|\rho|-1} \models \psi^{\chi}_{\text{out}} \wedge (\psi)^f_p,$$

and we are done.

For the other direction, assume that

$$t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi(x_k)}, u_\rho \models \mathbf{E}(\psi^f_{\text{out}} \wedge (\psi)^f_p).$$

There exists a path λ in $t_{\mathcal{S}_\mathcal{G}} \otimes \ell^{x_1}_{\chi(x_1)} \otimes \ldots \otimes \ell^{x_k}_{\chi(x_k)}$ starting in node u_ρ that satisfies both ψ^f_{out} and $(\psi)^f_p$. By construction of $\mathcal{S}_\mathcal{G}$ there exists an infinite play π such that $\pi_{\leq|\rho|-1} = \rho$ and $\lambda = \lambda_{\pi,|\rho|-1}$. By induction hypothesis, $\mathcal{G}, \chi, \pi, |\rho| - 1 \models \psi$. Because $\lambda_{\pi,|\rho|-1}$ satisfies ψ^f_{out}, $dom(\chi) \cap \text{Ag} = dom(f)$, and for all $a \in dom(f)$, $\chi(a) = \chi(f(a))$, it is also the case that $\pi \in \text{Out}(\chi, \rho)$, hence $\mathcal{G}, \chi, \rho \models \mathbf{E}\psi$.

□

Applying Proposition 2 to the sentence Φ, $\rho = v_\iota$, any assignment χ, and the empty function $\emptyset$, we get:

$$\mathcal{G} \models \Phi \quad \text{if and only if} \quad t_{\mathcal{S}_\mathcal{G}} \models (\Phi)^{\emptyset}_s.$$

5 Applications

[Module checking - **Bastien**] [synthesis - **Bastien**] [synthesis under constraints (ex: rational synthesis) - **Bastien**] [synthesis of maximal permissive strategies - **Bastien**]

References

[Berthon *et al.*, 2017] Raphael Berthon, Bastien Maubert, Aniello Murano, Sasha Rubin, and Moshe Y. Vardi. Strategy logic with imperfect information. In *LICS'17*, pages 1–12. IEEE, 2017.

[Bouyer *et al.*, 2019] Patricia Bouyer, Orna Kupferman, Nicolas Markey, Bastien Maubert, Aniello Murano, and Giuseppe Perelli. Reasoning about quality and fuzziness of strategic behaviours. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 1588–1594, 2019.

[Fijalkow *et al.*, 2018] Nathanaël Fijalkow, Bastien Maubert, Aniello Murano, and Sasha Rubin. Quantifying bounds in strategy logic. In *27th EACSL Annual Conference on Computer Science Logic, CSL 2018, September 4-7, 2018, Birmingham, UK*, pages 23:1–23:23, 2018.

[Laroussinie and Markey, 2014] François Laroussinie and Nicolas Markey. Quantified CTL: expressiveness and complexity. *LMCS*, 10(4), 2014.

[Laroussinie and Markey, 2015] François Laroussinie and Nicolas Markey. Augmenting ATL with strategy contexts. *Inf. Comput.*, 245:98–123, 2015.

[Maubert and Murano, 2018] Bastien Maubert and Aniello Murano. Reasoning about knowledge and strategies under hierarchical information. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018*, pages 530–540, 2018.