

# A Logic for Strategy Updates

Can Başkent

Department of Computer Science, Graduate Center, City University of New York  
cbaskent@gc.cuny.edu

## 1 Introduction

While people play games, they observe, learn, recollect and update their strategies *during* the game as well as adopting deontological strategies and goals before the game. In this paper, we focus on what we call *move updates* where some moves become unavailable during the game, and present a formal framework for extending *strategy logic* which was introduced by Ramanujam and Simon [2].

## 2 Restricted Strategy Logic

Restricted strategy logic (RSL) is based on the strategy logic (SL) [1,2]. The focus is on games played between two players. The most basic constructions in SL are strategy specifications. The specification  $[\varphi \rightarrow a]^i$  at player  $i$ 's position stands for “play  $a$  whenever  $\varphi$  holds”. The specification  $\sigma_1 + \sigma_2$  means that the strategy of the player conforms to the specification  $\sigma_1$  or  $\sigma_2$  and  $\sigma_1 \cdot \sigma_2$  means that the strategy of the player conforms to the specifications  $\sigma_1$  and  $\sigma_2$ . In RSL, we add an additional specification  $[\sigma!a]^i$  that stands for the updated specification where the player is not allowed to make an  $a$  move. Then,  $[\sigma!a]$  conforms with the model where the outgoing move is not  $a$ . Now, based on the strategy specifications, the syntax of the strategy logic SL is given as follows.

$$p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle a \rangle\varphi \mid (\sigma)_i : a \mid \sigma \rightsquigarrow_i \psi$$

We read  $(\sigma)_i : a$  as “at the current state the strategy specification  $\sigma$  for player  $i$  suggests that the move  $a$  can be played”, and  $\sigma \rightsquigarrow_i \psi$  as “following strategy  $\sigma$  player  $i$  can ensure  $\psi$ ”. The syntax of RSL, then, is the same as SL as we have incorporated the update modality  $[\sigma!a]$  at the specification level.

The truth definition for the strategy formulas are as follows:

$$\begin{aligned} M, s \models \langle a \rangle\varphi & \quad \text{iff } \exists s' \text{ such that } s \xrightarrow{a} s' \text{ and } M, s' \models \varphi \\ M, s \models (\sigma)_i : a & \quad \text{iff } a \in \sigma(s) \\ M, s \models \sigma \rightsquigarrow_i \psi & \quad \text{iff } \forall s' \text{ such that } s \Rightarrow_{\sigma}^* s' \text{ in } T_s|_{\sigma}, \text{ we have} \\ & \quad M, s' \models \psi \wedge (\mathbf{turn}_i \rightarrow \mathbf{enabled}_{\sigma}) \end{aligned}$$

where  $\sigma(s)$  is the set of enabled moves at  $s$  for strategy  $\sigma$ , and  $\Rightarrow_{\sigma}^*$  denotes the reflexive transitive closure of  $\Rightarrow_{\sigma}$ . Furthermore,  $T_s$  is the tree that consists of the unique path from the root ( $s_0$ ) to  $s$  and the subtree rooted at  $s$ , and  $T_s|_{\sigma}$  is the least subtree of  $T_s$  that contains a unique path from  $s_0$  to  $s$  and from  $s$

onwards, for each player  $i$  node, all the moves enabled by  $\sigma$ , and for each node of the opponent player, all possible moves. The proposition  $\mathbf{turn}_i$  denotes that it is  $i$ 's turn to play. Finally, define  $\mathbf{enabled}_\sigma = \bigvee_{a \in \Sigma} (\langle a \rangle \top \wedge (\sigma)_i : a)$ .

The axioms of RSL is given as follows based on SL [2].

- All the substitutional instances of the tautologies of propositional calculus
- $[a](\varphi \rightarrow \psi) \rightarrow ([a]\varphi \rightarrow [a]\psi)$
- $\langle a \rangle \varphi \rightarrow [a]\varphi$
- $\langle a \rangle \top \rightarrow ([\psi \rightarrow a]^i : a)_i$  for all  $a \in \Sigma$
- $[\mathbf{turn}_i \wedge \psi \wedge ([\psi \rightarrow a]^i : a)] \rightarrow \langle a \rangle \top$
- $\mathbf{turn}_i \wedge ([\psi \rightarrow a]^i : c) \leftrightarrow \neg \psi$  for all  $a \neq c$
- $(\sigma + \sigma')_i : a \leftrightarrow (\sigma : a)_i \vee (\sigma' : a)_i$
- $(\sigma \cdot \sigma')_i : a \leftrightarrow (\sigma : a)_i \wedge (\sigma' : a)_i$
- $\sigma \rightsquigarrow_i \psi \rightarrow [\psi \wedge \mathbf{inv}_i^\sigma(a, \psi) \wedge \mathbf{inv}_{-i}^\sigma(\psi) \wedge \mathbf{enabled}_\sigma]$
- $(\sigma!a)_i : c \leftrightarrow \mathbf{turn}_i \wedge \neg((\sigma)_i : a) \wedge (\sigma)_i : c$

Here,  $\mathbf{inv}_i^\sigma(a, \psi) = (\mathbf{turn}_i \wedge (\sigma)_i : a) \rightarrow [a](\sigma \rightsquigarrow_i \psi)$  which expresses the fact that after an  $a$  move by  $i$  which conforms to  $\sigma$ , the statement  $\sigma \rightsquigarrow_i \psi$  continues to hold, and  $\mathbf{inv}_{-i}^\sigma(\psi) = \mathbf{turn}_i \rightarrow \odot(\sigma \rightsquigarrow_i \psi)$  states that after any move of  $-i$ ,  $\sigma \rightsquigarrow_i \psi$  continues to hold. Moreover,  $\bigcirc \varphi \equiv \bigvee_{a \in \Sigma} \langle a \rangle \varphi$  and  $\odot \varphi \equiv \neg \bigcirc \neg \varphi$ . The inference rules that SL employs is as follows: modus ponens and generalization for  $[a]$  for each  $a \in \Sigma$ . The induction rule is a bit more complex: From the formulas  $\varphi \wedge (\mathbf{turn}_i \wedge (\sigma)_i : a) \rightarrow [a]\varphi$ ,  $\varphi \wedge \mathbf{turn}_{-i} \rightarrow \odot \varphi$ , and  $\varphi \rightarrow \psi \wedge \mathbf{enabled}_\sigma$  derive  $\varphi \rightarrow \sigma \rightsquigarrow_i \psi$ . Note that the additional axiom we need for RSL is the last one, and the previous ones are the axioms for SL. The final axiom explains how updates behave in RSL. Note that SL is sound and complete with respect to the given semantics [2].

**Theorem 1.** *RSL is complete with respect to the given semantics.*

**Theorem 2.** *The model checking problem for SL and RSL are in PSPACE.*

The proof of the completeness of RSL is by reduction. We can show that any RSL formula can be reduced to an SL formula.

For the complexity of model checking, we can use either the standard arguments for the model checking for basic modal logic, or we can give a polynomial translation between CTL\* and RSL.

*Acknowledgements.* We especially acknowledge the help and encouragement of Sujata Ghosh and Rineke Verbrugge. This paper is the product of author's visit to the Department of Artificial Intelligence of the University of Groningen. We also thank Sergei Artemov and Rohit Parikh for their feedback.

## References

1. Ghosh, S., Ramanujam, R., Simon, S.: On strategy composition and game composition (2010) (unpublished manuscript)
2. Ramanujam, R., Simon, S.: A logical structure for strategies. In: Logic and the Foundations of Game and Decision Theory (LOFT 7). Texts in Logic and Games, vol. 3, pp. 183–208. Amsterdam University Press, Amsterdam (2008)