

Computing the subjective possible locations of an object

François Schwarzentruher

ENS Rennes, France

Abstract. We present a prototype tool where we represent the state of knowledge of an agent about the position of another agent in the plane at any time.

1 Introduction

Dynamic epistemic logic ([2][5]) aims at expressing properties about the knowledge of agents and at modeling dynamics of knowledge in multi-agent settings. Recently, spatially grounded versions of Dynamic epistemic logic ([1], [3]) have been developed. In such versions, a world gives positions to all agents and epistemic relations are inferred from these positions. Figure 1 shows two indistinguishable worlds for agent a : in both worlds agent a sees the same thing. More precisely, agent a knows the positions of agents that she sees and only knows that the positions of agents that she does not see are outside her cone of vision.

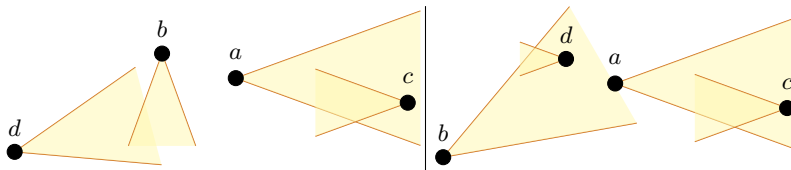


Fig. 1. Two indistinguishable worlds for agent a

Up to now, these spatially grounded epistemic logic variants have only the public announcement operator as a dynamic operators. With this demonstration, we show how we could add movements to agents. For the sake of simplicity, we only consider the following scenario: we represent the knowledge of one agent a about the position of agent b at any time. Note that our model is continuous both for time and space.

Potential applications could be video games [4] (we represent the knowledge of army a about army b in a strategic game), robotics (we represent the knowledge of a robot a about the position of robot b), building security (we simulate possible attacks and represent the knowledge of the attacker), etc.

In section 2, we present the demonstration. In section 3, we present the model and its current limitations. In section 4, we discuss the future work and propose some solutions to face the limitations.

2 Demonstration

The software is available here:

<http://www.irisa.fr/prive/fschwarz/knowledgeposition/>

During the demonstration, the user will be able to move agent a , to make a public announcement of the position of agent b at a given time and to travel back in time. We suppose that agent a knows the speed limit of agent b . The interface shows the region of the possible locations of agent b for agent a at a given time. It corresponds to the current knowledge of agent a at time t about the position of b at time $t' \leq t$. Figure 2 shows the graphical user interface of the system.

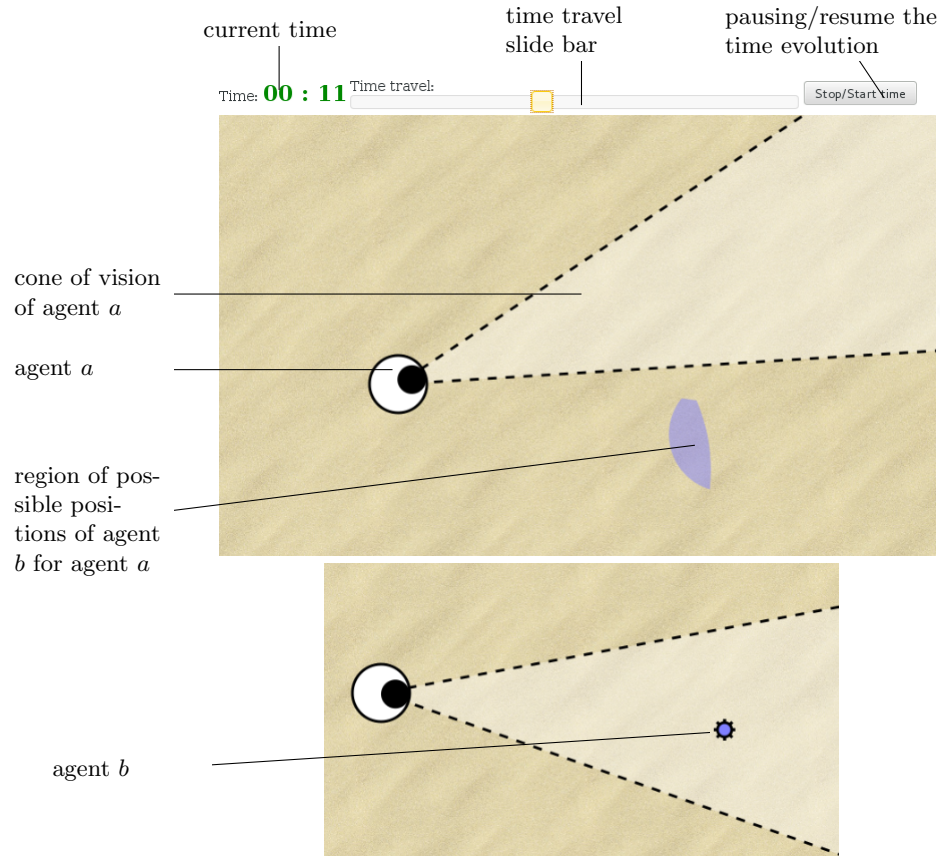


Fig. 2. Graphical interface

The user can perform a public announcement by pressing the space bar key. It will announce to agent a the current position of agent b .

3 Over-approximation of the region

Up to now, we do not compute the exact knowledge of agent a but an over-approximation, that is, we guarantee that agent a knows that b is in the regions we compute.

W.l.o.g. we suppose that the speed of agent b is bounded by 1. Let $\alpha > 0$ be an angle. Let $x \in \mathbb{R}^2$ be a point in the plane and $d \in \mathbb{R}^2$ a vector. We denote by $C(x, d) \subseteq \mathbb{R}^2$ the closed cone centered at point x of direction d and of opening angle α .

Let $pos_a(t) \in \mathbb{R}^2$ and $pos_b(t) \in \mathbb{R}^2$ be respectively the position of agent a and b at time t . Let $dir_a(t) \in \mathbb{R}^2$ be the direction of agent a at time t (we assume the Euclidean norm of $dir_a(t)$ to be 1). In the demonstration, $pos_a(t)$ and $dir_a(t)$ are modified by the user and $pos_b(t)$ is modified by the program.

Let $\mathcal{A}$ be a finite set of elements (t, R) where $t \in \mathbb{R}$ is a given time and $R \subseteq \mathbb{R}^2$ is a region of the plane such that $pos_b(t) \in R$. An element (t, R) represents an announcement of the form ‘at time t , agent b is in the region R ’. We denote by $ext(R, t)$ the set $\{x \in \mathbb{R}^2 \mid d(x, R) \leq t\}$ where $d(x, R)$ is the distance from point x to the non-empty region R . Informally, $ext(R, t)$ is the set of positions at distance at most t from R or the reachable possible points of agent b for agent a at time t if agent a knows that agent b was in R at time 0.

We define $R_b(t)$ the set of possible positions for agent b at time t as follows.

- if $pos_b(t) \in C(pos_a(t), dir_a(t))$, then $R_b(t) := \{pos_b(t)\}$;
- if $pos_b(t) \notin C(pos_a(t), dir_a(t))$ then

$$R_b(t) := (\mathbb{R}^2 \setminus C(pos_a(t), dir_a(t))) \cap \bigcap_{(t_0, R) \in \mathcal{A}} ext(R, |t - t_0|).$$

By definition of $R_b(t)$ we have $pos_b(t) \in R_b(t)$.

When are announcements made? When the user presses the space bar button at time t then we add $(t, \{pos_b(t)\})$ to $\mathcal{A}$. But we also add elements (t, R) to $\mathcal{A}$ in the following situations:

- When the position of agent b becomes visible or invisible for agent a , we add $(t, \{pos_b(t)\})$ to $\mathcal{A}$;
- When the user moves agent a , then we add $(t, \mathbb{R}^2 \setminus C(pos_a(t), dir_a(t)))$ to $\mathcal{A}$.

How is $ext(t, R)$ computed? In this implementation, we have only two cases.

- If R is a singleton $\{x\}$, then $ext(t, \{x\})$ is the closed disc centered at x of radius t .
- If R is the complement of a closed cone $\mathbb{R}^2 \setminus C(x, d)$, then $ext(t, \mathbb{R}^2 \setminus C(x, d)) = \mathbb{R}^2 \setminus C(x + td, d)$.

How is $R_b(t)$ displayed? We use the clipping features provided by the html5 canvas to implement intersection and complement of discs and cones. The source code is available on the web page of the software for details.

4 Research questions

The first issue raised in this work is the computation of the *exact* region of positions known by agent a . In particular, in the current state, the software gives aberrant regions especially when $|t - t_0|$ is big in $ext(t, \{x\})$ as shown in Figure 3 whereas the region should be connected. The solution may consist in adapting a shortest path algorithm in a continuous setting with moving obstacles (the cone of agent a is considered as a moving obstacle) and use suitable data structures. Actually we also may consider obstacles. A simpler version would be to consider a discrete case.

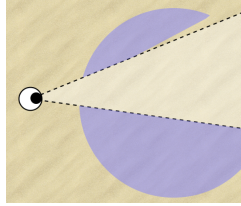


Fig. 3. Aberrant region of possible locations

A second issue is the movement of a : adding $(t, \mathbb{R}^2 \setminus C(pos_a(t), dir_a(t)))$ to $\mathcal{A}$ each time a is moving is costly in resources. It also suppose that the movement of a is discrete (a is moved each time we pressed an arrow button of the keyboard). In the future, we should consider the movement of a continuous even if the control of a by the user is discrete.

A third issue would be to generalize the model to the high-order knowledge. A challenge question would be how to represent graphically such high-order knowledge.

References

1. Philippe Balbiani, Olivier Gasquet, and François Schwarzenruber. Agents that look at one another. *Logic Journal of the IGPL*, 21(3):438–467, 2013.
2. Alexandru Baltag, Lawrence S Moss, and Slawomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, pages 43–56. Morgan Kaufmann Publishers Inc., 1998.
3. Olivier Gasquet, Valentin Goranko, and François Schwarzenruber. Big brother logic: logical modeling and reasoning about agents equipped with surveillance cameras in the plane. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, pages 325–332, 2014.
4. Ethan Kennerly, Andreas Witzel, Jonathan Zvesper, et al. Thief belief. In *Games Innovations Conference, 2009. ICE-GIC 2009. International IEEE Consumer Electronics Society's*, pages 169–172. IEEE, 2009.
5. Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Pieter Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.