

Small Undecidable Problems in Epistemic Planning

Sébastien Lê Cong¹, Sophie Pinchinat¹, François Schwarzentruber¹

¹ Univ Rennes, CNRS, IRISA

sebastien.le-cong@irisa.fr, sophie.pinchinat@univ-rennes1.fr, francois.schwarzentruber@ens-rennes.fr

Abstract

Epistemic planning extends classical planning with knowledge and is based on dynamic epistemic logic (DEL). The epistemic planning problem is undecidable in general. We exhibit a small undecidable subclass of epistemic planning over 2-agent S5 models with a fixed repertoire of one action, 6 propositions and a fixed goal. We furthermore consider a variant of the epistemic planning problem where the initial knowledge state is an automatic structure, hence possibly infinite. In that case, we show the epistemic planning problem with 1 public action and 2 propositions to be undecidable, while it is known to be decidable with public actions over finite models. Our results are obtained by reducing the reachability problem over small universal cellular automata. While our reductions yield a goal formula that displays the common knowledge operator, we show, for each of our considered epistemic problems, a reduction into an epistemic planning problem for a common-knowledge-operator-free goal formula by using 2 additional actions.

1 Introduction

Developing autonomous agents is central in artificial intelligence. Agents should be able to plan their actions for achieving a goal, and to reason about their knowledge and other agents' knowledge. *Epistemic planning* [Baral *et al.*, 2017] focuses on generating plans in a multi-agent context. Goals can be epistemic (for instance, the goal could be that “agent a knows that agent b does not know p ”). Some actions can change the physical world as well as the knowledge of agents. Typical actions are public actions where both knowledge and physical changes are commonly-known by the agents [Kominis and Geffner, 2015]. Traditionally, goals are expressed in epistemic logic (the goal above is expressed by formula $K_a \neg K_b p$). On top of the classical planning setting with pre/post-conditions for actions, the Dynamic Epistemic Logic [van Ditmarsch *et al.*, 2007; Bolander, 2017] framework offers a representation of knowledge change: *actions* are graphs [Baltag *et al.*, 1998] whose nodes are events (with pre/post-conditions) while edges denote epistemic relations.

In this setting, the epistemic planning problem consists in finding a sequence of actions leading to an epistemic state satisfying the goal. The problem is undecidable in general; there are several proofs in the literature, by reducing either the halting problem of Turing machines [Bolander and Andersen, 2011], or of two-counter machines [Aucher and Bolander, 2013; Charrier *et al.*, 2016]. Unfortunately, none of these proofs exhibit bounds to parameters (number of actions, number of propositions, etc.) involved in the problem.

In order to bound the parameters, we suggest to focus on one-dimensional three-neighbor cellular automata [von Neumann, 1951]. First, small universal cellular automata [Smith, 1968; Wolfram, 2002] have been exhibited. Second, DEL and cellular automata share theoretical features. On the one hand, all cells in a cellular automaton update their own state (called symbol later) *synchronously* according to a *local* rule. On the other hand, knowledge update in DEL consists in *synchronously* applying pre/post-conditions (that are *local* properties in epistemic logic) in several possible worlds.

Our first contribution is to take advantage of small universal cellular automata with a blank background (almost all cells' symbols except finitely many are \perp) to provide a small class of instances for the epistemic planning problem that is already undecidable (see Corollary 1). Interestingly, the repertoire of actions (made up of one action) and the goal are both fixed, only the initial knowledge state is part of the input.

Whereas public actions lead to decidability in epistemic planning [Kominis and Geffner, 2015; Belardinelli *et al.*, 2017], the second contribution is that public actions are enough for undecidability when the initial knowledge state is automatic [Blumensath and Grädel, 2000] (the gap between finite and automatic for structures is analogous to the gap between finite and regular for languages). This result is quite surprising since leaping from finite to automatic structures usually maintains decidability, for instance for first-order model checking. More surprisingly, undecidability is obtained for a very small class of instances (see Corollary 2) by taking advantage of Wolfram's Rule 110.

In a last contribution, we show how to remove the common knowledge operator from goal formulas (Corollaries 3 and 4).

The paper is organized as follows. We first recall the DEL setting and the epistemic planning problem in Section 2. Sections 3 and 4 are dedicated to the proofs of our two undecidability results and contain the material on cellular automata. Finally, Section 5 outlines the proof that common knowledge can be avoided in the goal formulas.

2 Background on Epistemic Planning

Knowledge states A *knowledge state* is a pointed Kripke model that describes an epistemic situation. In few words, it is a graph whose vertices are possible worlds with a distinguished world w_0 . In the literature, such models are often denoted by \mathcal{M}, w_0 , but here, we write them S for “states”; for ease of notation, we will improperly write the ordered pair (S, w) to denote state S whose actual world w_0 has been replaced with w . Let Ag be a finite set of agents and AP be a countable set of propositions.

Definition 1 A state $S = (W, (R_a)_{a \in Ag}, V, w_0)$ is defined by a non-empty set W of epistemic worlds, equivalence relations $R_a \subseteq W \times W$ called epistemic relations, a valuation function $V : W \rightarrow 2^{AP}$, and a world $w_0 \in W$ called the actual world.

In Definition 1, the valuation function V is a labeling for the worlds. The intuitive meaning of the epistemic relations is: $wR_a w'$ holds¹ if agent a considers world w' as possible in world w . The assumption of epistemic relations being equivalence relations is referred to as S5 [Chellas, 1980].

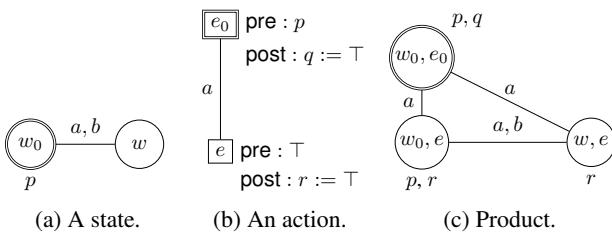


Figure 1: A state, an action and their product.

Consider the state $S = (W, (R_a)_{a \in Ag}, V, w_0)$ where $W = \{w_0, w\}$, $R_a = R_b = W \times W$, $V(w_0) = \{p\}$, and $V(w) = \emptyset$. This state has 2 agents and 1 proposition, and is depicted in Figure 1a, with the convention that the pointed world is in a double circle and equivalent worlds are linked with plain lines (in order to lighten the pictures, loops are omitted). In the epistemic situation described by S , agents a and b do not distinguish w_0 from w , and therefore do not know that p holds.

Epistemic language The language \mathcal{L}_{EL} extends the propositional language with epistemic modal operators and is defined by: $\varphi ::= \top \mid p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid K_a \varphi \mid C_G \varphi$, with $p \in AP$, $a \in Ag$, $G \subseteq Ag$. Formula $K_a \varphi$ is read “agent a knows that φ holds” and formula $C_G \varphi$ is read “ φ is commonly known among agents in G ”. We define the usual abbreviations $(\varphi_1 \wedge \varphi_2)$ for $\neg(\neg\varphi_1 \vee \neg\varphi_2)$, $\hat{K}_a \varphi$ for $\neg K_a \neg\varphi$

¹This is a standard notation for $(w, w') \in R_a$.

and $\hat{C}_G \varphi$ for $\neg C_G \neg\varphi$. The semantics of \mathcal{L}_{EL} is defined by induction over the formula (Boolean cases are omitted).

- $(W, (R_a)_{a \in Ag}, V, w) \models p$ if $p \in V(w)$;
- $(W, (R_a)_{a \in Ag}, V, w) \models K_a \varphi$ if for all w' s.t. $wR_a w'$, $(W, (R_a)_{a \in Ag}, V, w') \models \varphi$;
- $(W, (R_a)_{a \in Ag}, V, w) \models C_G \varphi$ if for all w' s.t. $w (\bigcup_{a \in G} R_a)^* w'$, $(W, (R_a)_{a \in Ag}, V, w') \models \varphi$

where $(\bigcup_{a \in G} R_a)^*$ is the reflexive and transitive closure of the union of the R_a for $a \in G$.

Actions Actions are modeled by so-called pointed event models² [van Ditmarsch *et al.*, 2007]. They are graphs whose nodes are events with a distinguished event e_0 that represents the real event taking place (in the current world) when action \mathbf{A} is executed. In more formal words:

Definition 2 An action $\mathbf{A} = (E, (\mathfrak{R}_a)_{a \in Ag}, \text{pre}, \text{post}, e_0)$ is given by a non-empty finite set of events E , equivalence relations $(\mathfrak{R}_a)_{a \in Ag} \subseteq E \times E$, a precondition function $\text{pre} : E \rightarrow \mathcal{L}_{EL}$, a postcondition function $\text{post} : E \times AP \rightarrow \mathcal{L}_{EL}$ and an event $e_0 \in E$ called the current event.

The precondition function defines under which condition an event can take place. The postcondition describes the effect of an event on the world: after an event e has taken place, the truth of p is assigned to the value $\text{post}(e, p)$. Actions made up of a single event ($\text{card}(E) = 1$) are *public actions*. Figure 1b shows the action \mathbf{A} where $E = \{e_0, e\}$, $\mathfrak{R}_a = E \times E$, $\mathfrak{R}_b = \{(e_0, e_0), (e, e)\}$, $\text{pre}(e_0) = p$, $\text{pre}(e) = \top$, $\text{post}(e_0, q) = \top$ and $\text{post}(e, r) = \top$. In figures and formal definitions, when a postcondition $\text{post}(e, p)$ is omitted, we mean that $\text{post}(e, p) = p$.

Product The update of a state S by an action \mathbf{A} is a new state written $S \otimes \mathbf{A}$ and defined as a product. The product is defined only if the current event in \mathbf{A} can take place, namely its precondition holds in the current world of S .

Definition 3 Let $S = (W, (R_a)_{a \in Ag}, V, w_0)$ be a state and $\mathbf{A} = (E, (\mathfrak{R}_a)_{a \in Ag}, \text{pre}, \text{post}, e_0)$ be an action. If $S \models \text{pre}(e_0)$, we define the product of S and \mathbf{A} as the state $S \otimes \mathbf{A} = (W', (R_a')', V', (w, e_0))$ where:

- $W' = \{(w, e) \in W \times E \mid (S, w) \models \text{pre}(e)\}$;
- $(w, e) R_a' (w', e')$ iff $wR_a w'$ and $e \mathfrak{R}_a e'$;
- $V'((w, e)) = \{p \in AP \mid (S, w) \models \text{post}(e, p)\}$.

Figure 1c shows the product $S \otimes \mathbf{A}$ of state S of Figure 1a and action \mathbf{A} of Figure 1b. For instance, formula $K_b(p \wedge q \wedge K_a(\neg q \rightarrow r))$ holds in state $S \otimes \mathbf{A}$. In the following, the expression $S \otimes \mathbf{A}_1, \dots, \mathbf{A}_m$ is a concise notation for $(\dots ((S \otimes \mathbf{A}_1) \otimes \mathbf{A}_2) \dots \otimes \mathbf{A}_m)$.

The epistemic planning problem introduced by [Bolander and Andersen, 2011] can now be defined.

²In the literature, pointed event models are often denoted by \mathcal{E}, e . We also decided to simplify the notations and to call them *actions*.

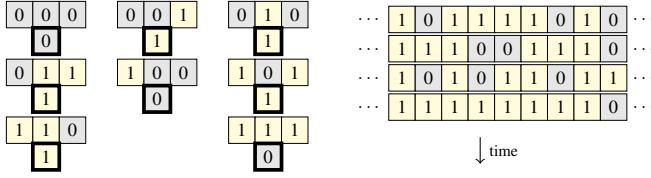


Figure 2: Rule 110 transition function ($f_{110}(1, 0, 1) = 1$, etc.) and some successive configurations.

Definition 4 (Epistemic planning problem)

- *input:* a state S , a finite set of actions \mathbb{A} and a formula $\varphi \in \mathcal{L}_{EL}$;
- *output:* yes if there exists a sequence of actions (a plan) $\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathbb{A}$ such that $S \otimes \mathbf{A}_1, \dots, \mathbf{A}_m \models \varphi$; no otherwise.

3 Small Epistemic Planning Problems

In this section, we show the undecidability of two restrictions of the epistemic planning problem, where \mathbb{A} is fixed and no longer part of the input. Undecidability is proven by reduction from undecidable reachability problems on cellular automata. From now on throughout the paper, $Ag = \{a, b\}$; recall that all models (i.e. states and actions) are S5 (epistemic relations are equivalence relations).

3.1 Universal 1D Cellular Automata

In this article, we only consider one-dimensional three-neighbor cellular automata. An infinite sequence of cells are settled on a line; each cell is in a state represented by a symbol³ of a finite alphabet Σ . Given a cell, a transition function f maps a three-neighbor (left-cell symbol, current symbol, right-cell symbol) to the new symbol of the cell.

Definition 5 A cellular automaton is a pair $\mathcal{A} = (\Sigma, f)$ where Σ is a finite alphabet and $f : \Sigma^3 \rightarrow \Sigma$ is a transition function.

Example 1 (Rule 110 [Wolfram, 2002]) The Rule 110 cellular automaton is the two-symbol cellular automaton $\mathcal{A}_{R110} = (\{0, 1\}, f_{110})$ where f_{110} is defined by the Boolean formula $f_{110}(x, y, z) := (x \wedge y \wedge \neg z) \vee (x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge z) \vee (\neg x \wedge y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z)$.

A configuration, that is the symbols of cells on an infinite line, is modeled by an infinite word $c \in \Sigma^{\mathbb{Z}}$, that is, a map that assigns a symbol $c[i]$ to any integer $i \in \mathbb{Z}$. A computation step is performed by the following rule.

Definition 6 Given a cellular automaton \mathcal{A} , given an infinite word $c \in \Sigma^{\mathbb{Z}}$, we define the successor of c by \mathcal{A} to be the infinite word c' defined by $c'[i] := f(c[i-1], c[i], c[i+1])$. We write $c \rightarrow_{\mathcal{A}} c'$.

Figure 2 shows the transition function f_{110} graphically and some successive configurations.

³We use ‘symbol’ instead of ‘cell state’, to avoid confusion with a knowledge state.

A cellular automaton is deemed *universal* if it can simulate any Turing machine; the quest for finding such *small* universal cellular automata started in the 1960s. A common hypothesis is to assume a *blank background*: we consider that alphabets always contain a special symbol \perp and that transition functions map $\perp\perp\perp$ to \perp . Furthermore, we assume that configurations are *finite*, in the sense that almost all cell symbols are \perp except a finite number – configurations are of the form $\perp^{\omega} \alpha \perp^{\omega}$ where α is a finite word, called the *support* of the configurations. Starting from a finite configuration only leads to finite configurations.

Smith proved that any m -symbol n -state Turing machine can be simulated by a one-dimensional $(m + 2n)$ -symbol⁴ 3-neighbor cellular automaton with a blank background (see Theorem 4 in [Smith, 1968]). As Minsky constructed a 4-symbol 7-state universal Turing machine M_{Minsky} [Minsky, 1967], there exists a $4 + 2 \times 7 = 18$ -symbol universal cellular automaton $\mathcal{A}_{\text{Smith}} = (\Sigma_{\text{Smith}}, f_{\text{Smith}})$, that simulates M_{Minsky} . As a consequence, we obtain the following undecidability result for the *reachability problem* for $\mathcal{A}_{\text{Smith}}$ with blank background.

Theorem 1 There exists a finite word⁵ h_{Smith} such that it is undecidable to determine, given a finite word α , whether $\perp^{\omega} \alpha \perp^{\omega} \rightarrow_{\mathcal{A}_{\text{Smith}}}^* c$ where the configuration c contains the pattern h_{Smith} .

3.2 Finite Linear States

The main result of this section is Corollary 1 which provides bounds to the parameters of the epistemic planning problem. This is achieved by simulating executions of cellular automata with blank background. First of all, we introduce sufficiently many propositions to encode symbols of the alphabet. Typically, for alphabet Σ_{Smith} with 18 symbols that we respectively write $\ell_0, \ell_1, \dots, \ell_{17}$, only 5 propositions p_1, \dots, p_5 suffice. Given a symbol ℓ , we note $\text{enc}(\ell)$ the encoding of ℓ : $\text{enc}(\ell_0) = \neg p_1 \wedge \dots \wedge \neg p_5$, $\text{enc}(\ell_1) = p_1 \wedge \neg p_2 \wedge \dots \wedge \neg p_5$, etc. W.l.o.g., we suppose that symbol \perp is ℓ_0 and is encoded by the valuation making all p_i ’s false. In the rest, \vec{p} denotes the sequence of propositions p_i ’s.

Now, we encode the finite supports of configurations by means of *finite linear states* – such states appear in real epistemic puzzles such as the *consecutive number puzzle* [van Ditmarsch and Kooi, 2015]. They are states of the form $\mathcal{I}_n = (\{-n, \dots, -1, 0, 1, \dots, n\}, (R_a)_{a \in Ag}, V, 0)$ with odd n and:

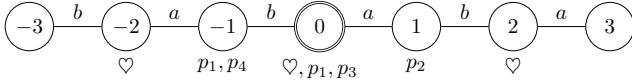
1. $R_a = \{(k, k) \mid k \in \llbracket 1; n \rrbracket\} \cup \{(2k, 2k+1), (2k+1, 2k) \mid \frac{-n+1}{2} \leq k \leq \frac{n-1}{2}\};$
2. $R_b = \{(k, k) \mid k \in \llbracket 1; n \rrbracket\} \cup \{(2k, 2k-1), (2k-1, 2k) \mid \frac{-n+1}{2} \leq k \leq \frac{n-1}{2}\};$
3. $\heartsuit \in V(k)$ iff k is even;
4. for all i , $p_i \notin V(-n), V(-n+1), V(n-1), V(n)$.

A finite linear state \mathcal{I}_n encodes a finite configuration c whose support is of length smaller than $2n - 3$ when $V(k)$

⁴Referred to as $(m + 2n)$ -state in [Smith, 1968].

⁵According to the notation of Table 14.8-1 p. 279 in [Minsky, 1967], word h_{Smith} is $q_3 0$.

makes $\text{enc}(\ell_i)$ true iff $c[k] = \ell_i$; we require that the two terminal ($-n$ and n) and the two pre-terminal ($n - 1, -n + 1$) worlds encode \sqcup (Condition 4). For instance, the $\mathcal{A}_{\text{Smith}}$ configuration $\sqcup \sqcup \sqcup \ell_9 \ell_5 \ell_2 \sqcup \sqcup \sqcup$ where $c[-1] = 9$ can be represented by:



Given a finite word α , let us set once and for all what finite linear state S_α will encode α : substituting $\alpha \sqcup^m$ for α with $m \in \{0, 1, 2, 3\}$ so that $|\alpha|$ be congruent to 3 modulo 4, we set $S_\alpha = (\mathcal{I}_{\lfloor \alpha \rfloor + 3}, (R_a)_{a \in Ag}, V, 0)$ where $V(k - \frac{|\alpha| - 1}{2})$ makes $\text{enc}(\ell_i)$ true iff $\alpha[k] = \ell_i$ for $0 \leq k \leq |\alpha| - 1$, and makes $\text{enc}(\sqcup)$ true outside of this range. These are only technicalities ensuring a sufficiently large odd index for the interval state to respect constraints with a pseudo-centered word.

3.3 Simulating Cellular Automata in DEL

We define action \mathbf{F} mimicking one computation step of the cellular automaton: if S is a finite linear state encoding a configuration c , then $S \otimes \mathbf{F}$ is (isomorphic to) a finite linear state encoding the successor of c .

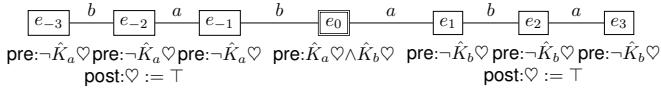


Figure 3: Skeleton of action \mathbf{F} .

Action \mathbf{F} is partially given by Figure 3. Intuitively, the actual event e_0 copies every non-terminal world; event e_{-1} keeps the left-tip world, while e_{-2} and e_{-3} clone it to append two new worlds to the left. Events e_1, e_2, e_3 play a similar part. In the end, action \mathbf{F} adds two new worlds on each side, while preserving the canonical knowledge state structure that we aim for, including the tips' asymmetry relatively to the agents.

We finish the definition of \mathbf{F} by adding postconditions for p_j 's, corresponding to the application of a transition function f . Suppose w.l.o.g. that \heartsuit holds in a given world $k \in \{-n + 1, \dots, n - 1\}$. Bits of $c[k - 1]$ are obtained by taking the b -transition from the $\neg\heartsuit$ -world of world k . They are: $\langle \hat{K}_b(\neg\heartsuit \wedge p_i) \rangle_i$. In the same way, the values of \vec{p} in world $k + 1$ are given by the vector $\langle \hat{K}_a(\neg\heartsuit \wedge p_i) \rangle_i$. The case where \heartsuit does not hold in the current world is symmetric. We model f by Boolean formulas $f_j(\vec{p}^-, \vec{p}, \vec{p}^+)$ over three sequences of atomic propositions \vec{p}^- (left cell symbol), \vec{p} (middle cell symbol), \vec{p}^+ (right cell symbol) that return the value of the j^{th} bit of the new symbol at the middle cell. Bits of the new symbol are: $\langle f_j(\vec{p}^-, \vec{p}, \vec{p}^+) \rangle_j$.

The postconditions for p_j 's in \mathbf{F} is thus defined as follows. First, $\text{post}(e_k)(p_j) = \perp$ for all $k \neq 0$. Only e_0 effectively applies f and $\text{post}(e_0)(p_j)$ is formula

$$\begin{aligned} (\heartsuit \rightarrow f_j(\langle \hat{K}_b(\neg\heartsuit \wedge p_i) \rangle_i, \vec{p}, \langle \hat{K}_a(\neg\heartsuit \wedge p_i) \rangle_i)) \wedge \\ (\neg\heartsuit \rightarrow f_j(\langle \hat{K}_a(\heartsuit \wedge p_i) \rangle_i, \vec{p}, \langle \hat{K}_b(\heartsuit \wedge p_i) \rangle_i)) . \end{aligned}$$

We will refer to $\mathbf{F}_{\text{Smith}}$ for the action model that corresponds to the transition function f_{Smith} of the cellular automaton $\mathcal{A}_{\text{Smith}}$.

On top of proposition \heartsuit , when considering the particular cellular automaton $\mathcal{A}_{\text{Smith}}$, we need no more than 5 extra propositions to encode all symbols of alphabet Σ_{Smith} of cardinal 18, so that an overall set of 6 propositions suffices.

Now, we define formulas encoding the occurrence of a pattern h in the configuration. We first define the formula $\text{wenc}(h) := \text{wenc}_\heartsuit(h) \vee \text{wenc}_{\neg\heartsuit}(h)$ where formulas $\text{wenc}_\heartsuit(h)$ and $\text{wenc}_{\neg\heartsuit}(h)$ are inductively defined by:

- $\text{wenc}_\heartsuit(\epsilon) = \heartsuit$ and $\text{wenc}_{\neg\heartsuit}(\epsilon) = \neg\heartsuit$;
- for all letters ℓ , $\text{wenc}_\heartsuit(\ell h) = \heartsuit \wedge \text{enc}(\ell) \wedge \hat{K}_a \text{wenc}_{\neg\heartsuit}(h)$;
- for all letters ℓ , $\text{wenc}_{\neg\heartsuit}(\ell h) = \neg\heartsuit \wedge \text{enc}(\ell) \wedge \hat{K}_b \text{wenc}_\heartsuit(h)$.

We can now state the following theorem that derives from our ability to simulate the behavior of the cellular automaton $\mathcal{A}_{\text{Smith}}$ and the use of the dual of the common knowledge operator \hat{C}_{Ag} to search over a finite linear state for the pattern $\text{wenc}(h_{\text{Smith}})$.

Theorem 2 *Given a finite linear state S over 6 propositions, it is undecidable to determine whether or not a state satisfying $\hat{C}_{Ag} \text{wenc}(h_{\text{Smith}})$ is reachable from S by executing a finite sequence of actions $\mathbf{F}_{\text{Smith}}$.*

PROOF.

By reduction from the undecidable reachability problem of Theorem 1. An instance α of the latter is translated into S_α . \blacksquare

Corollary 1 *The epistemic planning problem over 2-agent S5 finite models is undecidable, even if the repertoire is $\{\mathbf{F}_{\text{Smith}}\}$, the goal is $\hat{C}_{Ag} \text{wenc}(h_{\text{Smith}})$ and we only use at most 6 different propositions.*

The interested reader can run simulations in DEL of cellular automata using the online software *Hintikka's world* [Schwarzentruber, 2018] available at the following address: <http://hintikkasworld.irisa.fr>.

4 Epistemic Planning over Infinite States

We extend the epistemic planning problem to structures that can be infinite, but which can be fully described by a finite presentation, in the sense of *automatic structures* [Blumensath and Grädel, 2000].

Automatic structures should be understood as all logical structures whose domain and relations can be represented by an encoding in some regular finite-word language and (multi-tape) synchronous⁶ finite-state automata operating on said encoding, respectively.

In our setting, we let a state $S = (W, (R_a)_{a \in Ag}, V, w_0)$ be *automatic* if the following holds⁷.

⁶All heads of the tapes progress synchronously, provided the inputs have been “aligned” with a technical trick.

⁷We take the convention to use letter \mathcal{B} for (multi-tape) finite-state automata.

- W can be encoded as a regular language, hence accepted by some one-tape finite-state automaton \mathcal{B}_W over some alphabet;
- both relations R_a and R_b are characterized by some two-tape automata \mathcal{B}_a and \mathcal{B}_b respectively: for two finite words η and ζ accepted by \mathcal{B}_W (henceforth, they are encodings of two worlds w_η and w_ζ) of S , the pair (η, ζ) is accepted by the two-tape automaton \mathcal{B}_a if, and only if, $w_\eta R_a w_\zeta$ (similarly for \mathcal{B}_b);
- for each proposition $p \in AP$, there is a one-tape automaton \mathcal{B}_p accepting exactly words η s. th. $p \in V(w_\eta)$.

If the word η encodes world w_0 , the tuple $(\mathcal{B}_W, (\mathcal{B}_a)_{a \in Ag}, (\mathcal{B}_p)_p, \eta)$ is called an *automatic presentation* of $(W, (R_a)_{a \in Ag}, V, w_0)$. We can now introduce the following epistemic planning problem.

Definition 7 The epistemic planning problem over automatic structures is the following.

- *input: an automatic presentation of a state S , a finite set of actions \mathbb{A} and a formula φ ;*
- *output: yes if there exists a sequence of actions (a plan) $\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathbb{A}$ such that $S \otimes \mathbf{A}_1, \dots, \mathbf{A}_m \models \varphi$; no otherwise.*

In order to prove this problem to be undecidable, we first recall a well-known result on *universal cellular automata with periodic background*.

4.1 Cellular Automata with Periodic Background

In an attempt to reduce the number of possible cell symbols of universal cellular automata as introduced in Section 3, the notion of *periodic background* was introduced: a *periodic background* is of the form $\alpha^\omega \beta \gamma^\omega$, where $\alpha, \beta, \gamma \in \Sigma^*$ and $|\alpha|, |\gamma| > 1$.

Lindgren and Nordahl exhibited a 7-symbol universal cellular automaton [Lindgren and Nordahl, 1990]. Their results were subsumed by Wolfram's \mathcal{A}_{R110} cellular automaton with only 2 symbols [Wolfram, 2002]. Automaton \mathcal{A}_{R110} displays complex behaviors, and more importantly, is universal [Cook, 2008; 2004; Sutner, 2003]. As stated in [Larsson, 2013] (Th. 5.1), the following reachability problem is undecidable.

Theorem 3 ([Cook, 2008]) Given three finite words $\alpha, \beta, \gamma \in \{0, 1\}^*$ with $|\alpha|, |\gamma| \geq 1$, deciding whether $\alpha^\omega \beta \gamma^\omega \xrightarrow{*_{\mathcal{A}_{R110}}} c$ where c contains the finite word $h_{R110} = 01101001101000$ is undecidable.

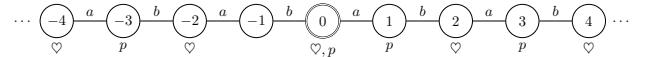
We now explain how the behavior of \mathcal{A}_{R110} can be simulated in DEL. We first show how to represent configurations by infinite linear states.

4.2 Infinite Linear States

To represent infinite configurations of cellular automata, we introduce *infinite linear states* that are states $\mathcal{Z} = (\mathbb{Z}, (R_a)_{a \in Ag}, V, 0)$ where:

- $R_a = \{(k, k), (2k, 2k+1), (2k+1, 2k) \mid k \in \mathbb{Z}\}$;
- $R_b = \{(k, k), (2k, 2k-1), (2k-1, 2k) \mid k \in \mathbb{Z}\}$;
- $\heartsuit \in V(k)$ iff k is even.

For a configuration (a two-way infinite word) $c \in \{0, 1\}^{\mathbb{Z}}$, we define S_c to be the infinite linear state such that $p \in V(k)$ iff $c[k] = 1$. For example, consider the infinite word $(100)^\omega 110(100)^\omega$, where word 110 in the middle is anchored in 0, namely, $\sigma[0] = 1, \sigma[1] = 1, \sigma[2] = 0$. Its corresponding infinite linear state⁸ $S_{(100)^\omega 110(100)^\omega}$ has the form (with the convention that we do not draw self-loops):



Proposition 1 Given three finite words $\alpha, \beta, \gamma \in \{0, 1\}^*$ with $|\alpha|, |\gamma| \geq 1$, $S_{\alpha^\omega \beta \gamma^\omega}$ is automatic and has an effectively computable automatic presentation.

We now turn to simulating the dynamics of \mathcal{A}_{R110} which will entail Theorem 4.

4.3 Epistemic Planning over Automatic Structures

The simulation of the cellular automaton \mathcal{A}_{R110} (see Definition 1) over the configuration $\alpha^\omega \beta \gamma^\omega$ follows the spirit of Section 3, but by simplifying action \mathbf{F} since linear states are infinite: the initial configuration of \mathcal{A}_{R110} is represented by the infinite linear state $S_{\alpha^\omega \beta \gamma^\omega}$. The application of f_{110} to each cell is simulated by executing the public action \mathbf{F}_{R110} consisting of a single event e_{R110} with $\text{pre}(e_{R110}) = \top$ and whose postcondition reflects f_{110} and leaves \heartsuit unchanged. The postcondition of e_{R110} is a 'single-bit' version of the postcondition for event e_0 for action \mathbf{F} of Section 3, as only one proposition is involved. $\text{post}(e_{R110})(p)$ is:

$$\begin{aligned} & (\heartsuit \rightarrow f_{110}(\hat{K}_b(\neg \heartsuit \wedge p), p, \hat{K}_a(\neg \heartsuit \wedge p))) \wedge \\ & (\neg \heartsuit \rightarrow f_{110}(\hat{K}_a(\heartsuit \wedge p), p, \hat{K}_b(\heartsuit \wedge p))). \end{aligned}$$

It is easy to see that the reachability problem for the cellular automaton \mathcal{A}_{R110} over a periodic background (given an initial periodic background $\alpha^\omega \beta \gamma^\omega$, does there exist a reachable configuration c that contains the finite word h_{R110} ?) can be rephrased as the epistemic planning problem over input $(S_{\alpha^\omega \beta \gamma^\omega}, \{\mathbf{F}_{R110}\}, \hat{C}_{Ag}(\text{wenc}(h_{R110})))$.

Theorem 4 Given an automatic presentation of an infinite linear state S (that requires only the 2 propositions \heartsuit and p), it is undecidable to know whether or not some state satisfying $\hat{C}_{Ag}(\text{wenc}(h_{R110}))$ can be reached by executing finitely many times the public action \mathbf{F}_{R110} .

PROOF.

We show a reduction from the undecidable reachability problem of Theorem 3. An instance (α, β, γ) of the latter is effectively translated into an automatic presentation of $S_{\alpha^\omega \beta \gamma^\omega}$ thanks to Proposition 1. ■

By Theorem 4, where only 1 public action (\mathbf{F}_{R110}) and 2 propositions (\heartsuit and p) were involved, we get the following corollary.

Corollary 2 The epistemic planning problem over automatic 2-agent S5 models is undecidable, even if the repertoire is fixed to 1 public action, the goal is fixed to $\hat{C}_{Ag}(\text{wenc}(h_{R110}))$ and if we use at most 2 different propositions.

⁸We deliberately forget the word's anchor in our notation.

5 Common Knowledge Elimination

In this section, we explain how to remove the dual common knowledge operator \hat{C}_{Ag} from the two goal formulas. Since the halting word (h_{Smith} or h_{R110}) can be far from the actual world, the basic idea is to apply shifts (whether rightward or leftward) so that the window of the halting word is at the actual world. It can then be checked by a \hat{C}_{Ag} -free goal formula. Any new plan (if any) will then be comprised of an old plan mingled with a shifting phase (i.e. a sequence of shifts).

We sketch a proof for finite states in Section 5.1, and then proceed in a similar fashion with infinite ones in Section 5.2.

5.1 Finite States

We wish to clearly express the epistemic planning problem with common knowledge associated to Smith's cellular automaton (EPC_{Smith}). As we only consider \mathcal{A}_{Smith} with its DEL simulation as described in Section 3.3, both the checked goal formula ($\hat{C}_{Ag}wenc(h_{Smith})$) and the set of actions \mathbb{A}_{Smith} ($= \{F_{Smith}\}$) are fixed, and so is the set of 6 propositions; the only input is the initial state S_α , which depends on finite word α , as defined in the aforementioned section.

Now, we will define the epistemic planning problem *without* common knowledge associated to Smith's cellular automaton (EPK_{Smith}), where the goal formula is now merely $wenc(h_{Smith})$ and the new set of actions \mathbb{A}'_{Smith} will also be fixed, with the intent to reduce EPC_{Smith} to EPK_{Smith} . Incidentally, the initial input state will be the same.

We enrich the set of actions \mathbb{A}_{Smith} of the initial DEL model: we choose to set $\mathbb{A}'_{Smith} = \{F_{Smith}, R_{Smith}, L_{Smith}\}$; the two new actions are defined below:

- R_{Smith} (*right shift*) is defined by the same structure as F_{Smith} is, except for the following postconditions on event e_0 : for all i , p_i 's assignment is replaced with $p_i := (\heartsuit \wedge \hat{K}_b(\neg \heartsuit \wedge p_i)) \vee (\neg \heartsuit \wedge \hat{K}_a(\heartsuit \wedge p_i))$;
- L_{Smith} (*left shift*) is akin to R_{Smith} ; let us just give the postcondition for p_i on e_0 since nothing else differs: $p_i := (\heartsuit \wedge \hat{K}_a(\neg \heartsuit \wedge p_i)) \vee (\neg \heartsuit \wedge \hat{K}_b(\heartsuit \wedge p_i))$.

This construction amounts to allowing nondeterministic shifts at any time, while still enforcing growth of the state in order to avoid information overflow on each side. We can now state the following theorem.

Theorem 5 EPC_{Smith} reduces to EPK_{Smith} .

In a nutshell, the proof relies on the two following lemmas; here and later, we use the exponent and update product notations somewhat freely to indicate iterating some action application, as there is no ambiguity.

Lemma 1 *If S is a finite linear state⁹, then, for all $n \in \mathbb{N}$, $(S, k) \models wenc(h_{Smith})$ iff $(S \otimes R_{Smith}^n, k + n) \models wenc(h_{Smith})$ iff $(S \otimes L_{Smith}^n, k - n) \models wenc(h_{Smith})$.*

⁹Recall that terminal and pre-terminal worlds encode \perp ; this property — which is an invariant — guarantees that there is no information loss about non-blank symbols after any shift, since actions R_{Smith} and L_{Smith} are designed in such a way that only those worlds' encodings cannot be transferred.

Lemma 2 *If S is a finite linear state, then $S \otimes R_{Smith} \otimes F_{Smith}$ is isomorphic to $S \otimes F_{Smith} \otimes R_{Smith}$; the result still holds when replacing L_{Smith} for R_{Smith} .*

Now, any plan in EPC_{Smith} can be completed to a plan in EPK_{Smith} : indeed, if $(S_\alpha \otimes \pi, k) \models wenc(h_{Smith})$ where π is a plan in EPC_{Smith} , then, by Lemma 1, if $k \geq 0$, $\pi \otimes L_{Smith}^k$ is a plan in EPK_{Smith} , and if $k < 0$, $\pi \otimes R_{Smith}^{-k}$ is a plan in EPK_{Smith} .

Conversely, given a plan π in EPK_{Smith} , Lemma 2 allows us to change its order to get an equivalent plan made up of π' with no shifts catenated with a sequence of shifts; using Lemma 1, we assert that π' is a plan for EPC_{Smith} .

Since we enriched the initial DEL model with exactly 2 actions, we have the following corollary.

Corollary 3 *The epistemic planning problem over finite linear initial states with 6 atomic propositions, with the fixed set of actions $\{F_{Smith}, L_{Smith}, R_{Smith}\}$, and the fixed goal $wenc(h_{Smith})$ is undecidable.*

5.2 Infinite States

We now turn to the Rule 110 cellular automaton with the same mindset: the epistemic planning problem with common knowledge associated to the Rule 110 cellular automaton EPC_{R110} relies on the simulation of Section 4, where $\mathbb{A}_{R110} = \{F_{R110}\}$, $AP = \{\heartsuit, p\}$ and the goal formula is $\hat{C}_{Ag}wenc(h_{R110})$. Recall that the input (α, β, γ) defines initial state $S_{\alpha^\omega \beta \gamma^\omega}$.

Problem EPK_{R110} will have $wenc(h_{R110})$ as its goal formula and the remainder of its setting will be that of EPC_{R110} , while its fixed set of actions is $\mathbb{A}'_{R110} = \{F_{R110}, L_{R110}, R_{R110}\}$. As states are infinite linear states, these two shifting actions can be defined as public actions:

- R_{R110} (*right shift*) is a public action whose single event precondition is \top and whose postcondition for p is $p := (\heartsuit \wedge \hat{K}_b(\neg \heartsuit \wedge p)) \vee (\neg \heartsuit \wedge \hat{K}_a(\heartsuit \wedge p))$;
- L_{R110} (*left shift*) is akin to R_{R110} ; let us just give the postcondition for p since nothing else differs: $p := (\heartsuit \wedge \hat{K}_a(\neg \heartsuit \wedge p)) \vee (\neg \heartsuit \wedge \hat{K}_b(\heartsuit \wedge p))$.

Theorem 6 EPC_{R110} reduces to EPK_{R110} .

Informally, the proof of Theorem 6 can be viewed as a fairly straightforward deduction from the demonstration of Theorem 5, through a projection of finite linear states onto infinite ones, while reinterpreting F_{R110} , R_{R110} and L_{R110} as pruned versions of F_{Smith} , R_{Smith} and L_{Smith} respectively, and adjusting propositions to the Rule 110 setting.

Notice that in the previous construction, both R_{R110} and L_{R110} are public actions as well as F_{R110} , which gives us the following corollary.

Corollary 4 *The epistemic planning problem over automatic structures with 2 atomic propositions, with the fixed set of public actions $\{F_{R110}, L_{R110}, R_{R110}\}$, and the fixed goal $wenc(h_{R110})$ is undecidable.*

Notice that actions F_{R110} and R_{R110} can be merged (so that the final phase of any successful plan is some — possibly empty — sequence of left shifts), although we did not proceed with this optimization here for the sake of clarity.

6 Conclusion

This work makes a connection between epistemic planning in DEL and cellular automata. We claim that many other similar undecidability results could be transferred to epistemic planning. Decidability techniques (see [Codd, 1968]) could be of use for finding decidable cases and for sharpening the decidability/undecidability frontier of epistemic planning.

Acknowledgments

We thank Maurice Margenstern for email discussions about Lindgren and Nordahl's cellular automaton. We are also grateful for our reviewers' helpful remarks.

References

- [Aucher and Bolander, 2013] Guillaume Aucher and Thomas Bolander. Undecidability in epistemic planning. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013, Beijing, China, August 3-9, 2013*, pages 27–33, 2013.
- [Baltag *et al.*, 1998] Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements and common knowledge and private suspicions. In *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98), Evanston, IL, USA, July 22-24, 1998*, pages 43–56, 1998.
- [Baral *et al.*, 2017] Chitta Baral, Thomas Bolander, Hans van Ditmarsch, and Sheila McIlraith. Epistemic Planning (Dagstuhl Seminar 17231). *Dagstuhl Reports*, 7(6):1–47, 2017.
- [Belardinelli *et al.*, 2017] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. Verification of multi-agent systems with imperfect information and public actions. In Kate Larson, Michael Winikoff, Sanmay Das, and Edmund H. Durfee, editors, *Proceedings of the 16th Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 1268–1276. ACM, 2017.
- [Blumensath and Grädel, 2000] Achim Blumensath and Erich Grädel. Automatic structures. In *Logic in Computer Science, 2000. Proceedings. 15th Annual IEEE Symposium on*, pages 51–62. IEEE, 2000.
- [Bolander and Andersen, 2011] Thomas Bolander and Mikkel Birkegaard Andersen. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.
- [Bolander, 2017] Thomas Bolander. A gentle introduction to epistemic planning: The DEL approach. In *Proceedings of the Ninth Workshop on Methods for Modalities, M4M@IJCAI 2017, Indian Institute of Technology, Kanpur, India, 8th to 10th January 2017.*, pages 1–22, 2017.
- [Charrier *et al.*, 2016] Tristan Charrier, Bastien Maubert, and François Schwarzenbuber. On the impact of modal depth in epistemic planning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1030–1036, 2016.
- [Chellas, 1980] Brian F Chellas. *Modal logic: an introduction*. Cambridge university press, 1980.
- [Codd, 1968] Edgar F Codd. *Cellular automata*. Academic Press, 1968.
- [Cook, 2004] Matthew Cook. Universality in elementary cellular automata. *Complex Systems*, 15(1), 2004.
- [Cook, 2008] Matthew Cook. A concrete view of rule 110 computation. In *Proceedings International Workshop on The Complexity of Simple Programs, CSP 2008, Cork, Ireland, 6-7th December 2008.*, pages 31–55, 2008.
- [Kominis and Geffner, 2015] Filippos Kominis and Hector Geffner. Beliefs in multiagent planning: From one agent to many. In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015.*, pages 147–155, 2015.
- [Larsson, 2013] Urban Larsson. Impartial games emulating one-dimensional cellular automata and undecidability. *J. Comb. Theory, Ser. A*, 120(5):1116–1130, 2013.
- [Lindgren and Nordahl, 1990] Kristian Lindgren and Mats G. Nordahl. Universal computation in simple one-dimensional cellular automata. *Complex Systems*, 4(3), 1990.
- [Minsky, 1967] Marvin L Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., 1967.
- [Schwarzenbuber, 2018] François Schwarzenbuber. Hintikka's world: agents with higher-order knowledge (demo). In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI) and the 23rd European Conference on Artificial Intelligence (ECAI), Stockholm, 13-19 July 2018*, 2018.
- [Smith, 1968] Alvy Ray Smith. Simple computation-universal cellular spaces and self-reproduction. In *9th Annual Symposium on Switching and Automata Theory, Schenectady, New York, USA, October 15-18, 1968*, pages 269–277, 1968.
- [Sutner, 2003] Klaus Sutner. Almost periodic configurations on linear cellular automata. *Fundam. Inform.*, 58(2003):223–240, 2003.
- [van Ditmarsch and Kooi, 2015] Hans van Ditmarsch and Barteld Kooi. *One Hundred Prisoners and a Light Bulb*. Springer, 2015.
- [van Ditmarsch *et al.*, 2007] Hans van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.
- [von Neumann, 1951] John von Neumann. The general and logical theory of automata. *Cerebral mechanisms in behavior*, 1(41):1–2, 1951.
- [Wolfram, 2002] Stephen Wolfram. *A new kind of science*. Wolfram-Media, 2002.