Optimal Regression for Reasoning about Knowledge and Actions

Hans van Ditmarsch[1], Andreas Herzig[2] and Tiago de Lima[3]

[1] Department of Computer Science, University of Otago, New Zealand
hans@cs.otago.ac.nz
[2] Institut de Recherche en Informatique de Toulouse, France
herzig@irit.fr
[3] Institut de Recherche en Informatique de Toulouse, France
santos@irit.fr

Abstract. We show how in the propositional case Scherl & Levesque's solution to the frame problem with knowledge can be modelled in dynamic epistemic logic with announcements and assignments (DEL), and provide an optimal reasoning method for the latter. Our method is as follows: we encode Scherl & Levesque's framework into DEL; then, by extending Lutz' recent reduction method for public announcement logic to DEL, we establish optimal decision procedures: the satisfiability problem in DEL is NP-complete for one agent, PSPACE-complete for multiple agents and EXPTIME-complete when common knowledge is involved.

Keywords. reasoning about actions and change, reasoning about knowledge, dynamic epistemic logic, situation calculus, frame problem.

1 Introduction

Thielscher [1] distinguishes two versions of the frame problem. The *representational version* is the problem of designing a logical language and a semantics such that domains can be described without making explicit the interaction between every action and fluent: basically, when there are n actions and m fluents, the domain description should be much smaller than $2 \times n \times m$. The *inferential version* of the frame problem is more demanding: given a solution of the representational version, the problem is to design an "efficient" decision procedure, where "efficient" roughly means that its computational complexity should not be too high.

Reiter [2] solved the representational frame problem by means of successor state axioms (SSAs). In the propositional case fluents only have situations as arguments, and SSAs take the form

$$\forall x.\forall s.(p(do(x,s)) \leftrightarrow (Poss(x,s) \rightarrow (x = a_1 \wedge \gamma^+(a_1,p,s)) \vee \cdots \vee (x = a_n \wedge \gamma^+(a_n,p,s)) \vee (p(s) \wedge \neg(x = a'_1 \wedge \gamma^-(a'_1,p,s)) \wedge \cdots \wedge \neg(x = a'_m \wedge \gamma^-(a'_m,p,s)))))$$

Dagstuhl Seminar Proceedings 07351
Formal Models of Belief Change in Rational Agents
http://drops.dagstuhl.de/opus/volltexte/2007/1207

where $a_1, \ldots, a_n$ are the actions potentially making p true, and $a'_1, \ldots, a'_m$ are the actions potentially making p false. For a given action a_i, let us note $\mathit{Eff}^+(a_i)$ the set of those fluents which a_i may make true, and $\mathit{Eff}^-(a_i)$ the set of those fluents which a_i may make false (in [2] these sets are left implicit). Then for every fluent $p \in \mathit{Eff}^+(a_i)$, the formula $\gamma^+(a_i, p, s)$ characterizes the conditions under which a_i makes p true, and $\gamma^-(a_i, p, s)$ characterizes the conditions under which a_i makes p false. $\gamma^+(a_i, p, s)$ and $\gamma^-(a_i, p, s)$ must be *uniform in* s, which in particular means that they do not contain the function *do*.

Reiter's central idea is that due to inertia the sets $\mathit{Eff}^+(a_i)$ and $\mathit{Eff}^-(a_i)$ are 'small' subsets of the set of all fluents. For that reason the size of the set of all SSAs can be expected to be of the order of the number of actions, and thus much smaller than the product of the number of actions with the number of fluents. Hence SSAs count as a solution to the representational frame problem. Reiter's solution was extended in [3] to a framework containing knowledge-producing actions.

When SSAs are available for every fluent p, one can reduce (*regress*) any formula φ to an equivalent formula $\mathrm{reg}(\varphi)$ not mentioning actions. This leads to a straightforward decision procedure in the propositional case, that has been implemented in the GOLOG language [4]. However, the reduced formula can be exponentially larger than the original formula, and therefore the inferential frame problem has to be considered unsolved in Reiter's and in Scherl & Levesque's approaches.

In this paper we solve the inferential frame problem for the propositional case. For the extension to knowledge, among the epistemic actions we only consider *observations*: all agents observe *that* some proposition holds in the world, and update their epistemic state accordingly.[1] We give a satisfiability-preserving polynomial transformation eliminating action operators from formulas. This provides an optimal regression procedure for reasoning about actions: both in Reiter's case (without knowledge operators) and in the single-agent case the decision procedure works in nondeterministic polynomial time; in the multi-agent case it works in PSPACE, and in the case of common knowledge in EXPTIME. All these results are optimal because they match the complexity of the underlying epistemic logic.

Technically, our approach builds on recent progress in the field of *dynamic epistemic logics*. In this family of logics situation terms are left implicit, and there is no quantification over actions. Thus the central device in Reiter's solution is not available. We show that nevertheless one can do without it, and recast Reiter's and Scherl & Levesque's solution in the dynamic epistemic logic DEL proposed by [5] (see also [6]).[2] DEL being an extension of Plaza's public announcement logic, we extend Lutz' optimal decision procedure for the latter [8] to DEL, and show that we keep optimality: checking satisfiability of DEL-

[1] Note that observations are different from the sensing actions as studied in [3]. By performing the latter, the agents observe *whether* some proposition holds in the world or not.

[2] A similar idea is outlined independently in [7].

formulas is proved to have the same complexity as checking satisfiability in the underlying epistemic logic.

The remainder of the paper is organized as follows: Section 2 presents a useful fragment of the situation calculus introduced by [9] and its formulation of the solution to the representational frame problem. and Reiter's and Scherl & Levesque's solution to the frame problem with knowledge. The subsequent section introduces dynamic epistemic logics. Section 4 contains the translation of the latter approach into dynamic epistemic logic, and Section 5 contains decision procedures for satisfiability checking in DEL for the single-agent and the multi-agent case, as well as for the case of common knowledge.

2 A variant of situation calculus

The dialect of second-order logic called *situation calculus* was one of the first formalisms used in reasoning about actions [10]. After two decades, [2] proposed a partial solution to the frame problem in situation calculus, as well as a procedure for reasoning called *regression*, which leads to decidability of the satisfiability problem for a sub-class of formulas. Since then, situation calculus has been used to specify dynamic systems. Some programming languages, such as GOLOG [4], were built, and the formalism also received a number of extentions such as for concurrent actions [11], knowledge [12], and probabilistic (noisy) actions [13].

However, because situation calculus is defined axiomatically, properties about action theories that are not direct entailments are very hard to prove. An example of this difficulty is the very long proof given by [14] for the fact that if $\mathrm{K}\varphi$ entails $\mathrm{K}\psi \vee \mathrm{K}\chi$ in a theory Θ, then $\mathrm{K}\varphi$ entails $\mathrm{K}\psi$ in Θ, or $\mathrm{K}\varphi$ entails $\mathrm{K}\chi$ in Θ. Aiming at a "more workable" semantics for situation calculus, [15] (see also [9]) proposed a variant called ES. This logic is not as expressive as the entire situation calculus, but it handles the action theories defined by Reiter and, thereby, also his solution to the frame problem.

In the sequel we present syntax and semantics of ES and its solution to the frame problem.

2.1 Syntax and semantics of ES

The full language of ES is a many sorted second-order predicate language. In this work however, we focus our attention to the (quasi) propositional fragment. We drop functional symbols, restrict the number and type of predicates, and only consider fluents of arity zero.

Definition 1 (The language $\mathcal{L}_{\mathsf{ES}}$). *Let U be a set of countably many first-order variables, let P be a set of countably many fluents of arity 0, let A be a set of countably many actions, and let $Poss$ and SF be the only two predicate symbols of the language. Both predicates have arity 1 and are of sort A. The language $\mathcal{L}_{\mathsf{ES}}$ is the set of formulas φ and terms t defined by the following BNF:*

$$\varphi ::= t = t \mid p \mid Poss(t) \mid SF(t) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{K}\varphi \mid [t]\varphi \mid \Box\varphi \mid \forall x.\varphi$$
$$t ::= x \mid a$$

where x ranges over U, a ranges over A, and p ranges over P.

Formulas that do not mention $Poss$, SF, $\Box$, $[t]$ or K are called fluent formulas. Formulas without free variables are called sentences. *And* primitive sentences *are the formulas in the set $P \cup \{Poss(a) \mid a \in A\} \cup \{SF(a) \mid a \in A\}$.*

The predicate $Poss$ is used to model executability preconditions of actions. If $Poss(t)$ holds, then the action t is executable. The predicate SF is used to model the result of sensing actions. The formula $SF(t)$ is the formula whose truth value is known by the agent after the execution of the action t. The operator K is used to model knowledge (or belief) of the agent. The formula $\mathrm{K}\varphi$ is read 'the agent knows (or believes) that φ'. In other words, φ holds in all worlds that the agent considers possible.[3] The operator $[\cdot]$ is used to model the transitions associated to actions. A formula of the form $[a]\varphi$ is read 'φ holds after the execution of the action a'. And the formula $\Box\varphi$ is read 'φ holds after the execution of any sequence of actions'.

In addition, we use the common abbreviations for the formulas $\top$, $\bot$ and the operators $\neq$, $\vee$, $\rightarrow$, $\leftrightarrow$ and $\langle\cdot\rangle$.

Formulas in $\mathcal{L}_{\mathsf{ES}}$ are evaluated in tuples of the form $\langle e, w, \alpha\rangle$ such that:

- $w \in W$ is a function from primitive sentences and A^* to $\{0, 1\}$
- $e \subseteq W$ is the epistemic state of the agent; and
- $\alpha \in A^*$ is a (possibly empty) sequence of actions.

Let ϵ denote the empty sequence in A^*. To interpret what is known by the agent after a sequence of actions, we define that two worlds *agree* with respect to the sequence of actions α inductively by the following:

- $w \sim_\epsilon w'$ iff w and w' agree on the value of every primitive term and sentence; and
- $w \sim_{\alpha\cdot t} w'$ iff $w \sim_\alpha w'$ and $w(SF(t), \alpha) = w(SF(t), \alpha)$.

The satisfaction relation '$\Vdash$' is inductively defined by:

$$\begin{array}{lll}
\langle e, w\rangle \Vdash \varphi & \text{iff} & \langle e, w, \epsilon\rangle \Vdash \varphi \\
\langle e, w, \alpha\rangle \Vdash t_1 = t_2 & \text{iff} & w(t_1, \alpha) \text{ is identical to } w(t_2, \alpha) \\
\langle e, w, \alpha\rangle \Vdash p & \text{iff} & w(p, \alpha) = 1 \\
\langle e, w, \alpha\rangle \Vdash Poss(t) & \text{iff} & w(Poss(w(t, \alpha)), \alpha) = 1 \\
\langle e, w, \alpha\rangle \Vdash SF(t) & \text{iff} & w(SF(w(t, \alpha)), \alpha) = 1 \\
\langle e, w, \alpha\rangle \Vdash \neg\varphi & \text{iff} & \text{not } \langle e, w, \alpha\rangle \Vdash \varphi \\
\langle e, w, \alpha\rangle \Vdash \varphi \wedge \psi & \text{iff} & \langle e, w, \alpha\rangle \Vdash \varphi \text{ and } \langle e, w, \alpha\rangle \Vdash \psi \\
\langle e, w, \alpha\rangle \Vdash \forall x.\varphi & \text{iff} & \text{for all } a \in A, \langle e, w, \alpha\rangle \Vdash \varphi^a_x \\
\langle e, w, \alpha\rangle \Vdash \mathrm{K}\varphi & \text{iff} & \text{for all } w' \in e, w \sim_\alpha w' \text{ implies } \langle e, w', \alpha\rangle \Vdash \varphi \\
\langle e, w, \alpha\rangle \Vdash [t]\varphi & \text{iff} & \langle e, w, \alpha \cdot w(t, \alpha)\rangle \Vdash \varphi \\
\langle e, w, \alpha\rangle \Vdash \Box\varphi & \text{iff} & \text{for all } \alpha' \in A^*, \langle e, w, \alpha \cdot \alpha'\rangle \Vdash \varphi
\end{array}$$

[3] The original language of ES also contains the 'only knows' operator OK. It allows, for instance, to infer more about the ignorance of the agent. We do not consider it here.

Let $\Psi \subseteq \mathcal{L}_{\mathsf{ES}}$. A formula $\varphi \in \mathcal{L}_{\mathsf{ES}}$ is:

- a *valid* ES-*consequence* of Ψ (notation: $\Psi \models_{\mathsf{ES}} \varphi$) if and only if for all e, for all w, if for all $\psi \in \Psi$, $\langle e, w, \epsilon \rangle \Vdash \psi$, then $\langle e, w, \epsilon \rangle \Vdash \varphi$;
- ES-*valid* (notation: $\models_{\mathsf{ES}} \varphi$) if and only if $\emptyset \models_{\mathsf{ES}} \varphi$; and
- ES-*satisfiable* if and only if $\not\models_{\mathsf{ES}} \neg\varphi$.

Lakemeyer & Levesque show that the same properties of knowledge as for situation calculus (and Hintika's epistemic logic EL) arise from this definition. For example, we have positive introspection, i.e., $\models_{\mathsf{ES}} \Box(\mathrm{K}\varphi \to \mathrm{K}\mathrm{K}\varphi)$, negative introspection, i.e., $\models_{\mathsf{ES}} \Box(\neg\mathrm{K}\varphi \to \mathrm{K}\neg\mathrm{K}\varphi)$, and also the following successor state axiom for knowledge:

$$\text{SSAK.} \qquad \models_{\mathsf{ES}} \Box([a]\mathrm{K}\varphi \leftrightarrow ((SF(a) \wedge \mathrm{K}(SF(a) \to [a]\varphi)) \quad \vee \\ (\neg SF(a) \wedge \mathrm{K}(\neg SF(a) \to [a]\varphi))))$$

2.2 Basic action theories

It was by means of situation calculus that [16] highlighted the representational frame problem in reasoning about actions. After that, several partial solutions were proposed, for example by [17], and by [18]. None of them however, is fully satisfactory. We present here yet another partial solution, proposed by [2]. We also incorporate the extension to incomplete knowledge proposed by [12] that was redesigned later by [3]. We call this solution the *RSL's partial solution to the frame problem*, or RSL for short.

RSL relies on a number of simplifying hypothesis. The most important are:

H1 All actions are deterministic.
H2 All the laws that define the behavior of the actions are known by the agent.
H3 All action occurrences are perceived by the agent.
H4 for each action constant a, it is possible to give a single formula $\psi(a)$ that characterizes the condition under which a is executable (*action precondition completeness*).
H5 for each fluent constant p, it is possible to give a finite set of action constants that may flip the truth value of p (*causal completeness*).
H6 for each fluent p, it is possible to give a single formula $\chi^+(a,p)$ (respectively $\chi^-(a,p)$ that characterizes all the conditions under which a flips the truth value of p to true (respectively to false) in the successor situation (*effect precondition completeness*).
H7 The length of the formulas $\chi^+(a,p)$ and $\chi^-(a,p)$ in H6 is roughly proportional to the number of actions that affect the value of the fluent.
H8 Relatively few actions affect a given fluent.
H9 The set of fluents affected by an action is much smaller than the entire set of fluents of the language (*inertia*).

Hypothesis H1 is about the nature of the world. This hypothesis is implemented by the fact that each $w \in W$ is a function. Hypotheses H2 and H3 say that the agent's knowledge about actions types and about actions instances are accurate. They are implemented by the indistinguishability relations '$\sim_\alpha$' between worlds. The other six hypotheses are implemented by requiring that the action preconditions and effects be described using a collection of formulas in some specific form. This collection of formulas is called a *basic action theory*.

Definition 2. *A basic action theory is a set $\Theta = \Theta_{una} \cup \Theta_{pre} \cup \Theta_{sense} \cup \Theta_{post}$, such that:*

- Θ_{una} *contains one formula of the form $a_1 \neq a_2$ for each pair of different action names (a_1, a_2);*
- Θ_{pre} *contains one formula of the form $\forall x.\Box(Poss(x) \leftrightarrow \varphi(x))$, where $\varphi(x)$ is a fluent formula that contains only x as free variable;*
- Θ_{sense} *contains one formula of the form $\forall x.\Box(SF(x) \leftrightarrow \psi(x))$, where $\psi(x)$ is a fluent formula that contains only x as free variable; and*
- Θ_{post} *contains one formula of the form $\forall x.\Box([x]p \leftrightarrow \chi(x,p))$ for each relevant p (i.e., each $p \in Eff^+(a) \cup Eff^-(a)$), where each $\chi(x,p)$ is a fluent formula that contains only x as free variable.*

Because $\varphi(x)$, $\psi(x)$ and $\chi(x,p)$ are fluent formulas, action preconditions and actions effects are completely determined by the current possible world. Moreover, under hypotheses H9 actions only change a small part of the world, leaving the rest unchanged. It follows that $|\Theta| = \mathcal{O}(|P| + |A|)$ [19]. In addition, note that H8 implies that there is no action changing the truth value of an infinity of fluents. Therefore, RSL is a solution to the representational frame problem.

Example 1 (The lady or the tiger). We illustrate RSL's solution by showing a formalization of a running example inspired by a puzzle of [20]. The environment consists of an agent that inhabits a room with two doors. These doors may be opened by the agent and, if so, behind each one the agent will either find the lady, or the tiger. If the agent opens a door and finds the lady, then she will marry him, and if he finds the tiger, then it will kill him. The available actions are:

- $listen_1$ and $listen_2$: by executing one of these actions, the agent listens to what happens behind the respective door, which results in hearing the tiger roaring if there is one behind the door; and
- $open_1$ and $open_2$: by executing one of these actions, the agent opens the respective door, which results in either marrying the lady or being killed by the tiger, depending on what is behind the door.

We the fluent $lady_1$ to mean that the lady is behind door 1. Thus the formula $\neg lady_1$ means that the lady is behind door 2 and the tiger is behind door 1. A basic action theory for this example can be formed by the following unique-name axioms for actions:

$$\Theta_{\text{una}} = \{listen_1 \neq listen_2,\ listen_1 \neq open_1,\ listen_1 \neq open_2,\\ listen_2 \neq open_1,\ listen_2 \neq open_2,\ open_1 \neq open_2\ \}$$

the following action precondition axiom:

$$\Theta_{\text{pre}} = \{\forall x.\Box(Poss(x) \leftrightarrow alive)\}$$

the following sensed fluent axiom:

$$\begin{aligned}\Theta_{\text{sense}} = \{\, \forall x.\Box(SF(x) \leftrightarrow (((x = open_1) \wedge \top) \vee \\ ((x = open_2) \wedge \top) \vee \\ ((x = listen_1) \wedge lady_1) \vee \\ ((x = listen_2) \wedge \neg lady_1)))\,\}\end{aligned}$$

and the following successor-state axioms:

$$\begin{aligned}\Theta_{\text{post}} = \{\, & \forall x.\Box([x]alive \leftrightarrow (((x = open_1) \wedge lady_1 \wedge alive) \vee \\ & \qquad ((x = open_2) \wedge \neg lady_1 \wedge alive))), \\ & \forall x.\Box([x]married \leftrightarrow (((x = open_1) \wedge (lady_1 \vee married)) \vee \\ & \qquad ((x = open_2) \wedge (\neg lady_1 \vee married)))), \\ & \forall x.\Box([x]lady_1 \leftrightarrow lady_1),\ \}\end{aligned}$$

Then, for instance we have that:

$$\Theta \models_{\mathsf{ES}} \langle listen_1\rangle(\mathrm{K}\,lady_1 \vee \mathrm{K}\neg lady_1)$$
$$\Theta \models_{\mathsf{ES}} (lady_1 \wedge alive) \rightarrow \langle listen_1\rangle\langle open_1\rangle\mathrm{K}\,married$$

2.3 Regression

In this section we present an effective procedure for reasoning using RSL's solution. It is based in goal regression, originally proposed by [21]. In the ES setting it corresponds to starting with a complex formula that represents the goal and applying successive "simplifications" in order to obtain an equivalent formula which does not mention actions. Once this formula is obtained, the problem has been reduced to theorem proving in S5 logic plus equality.

Using axiom SSAK above, the definition of the regression procedure for ES is very similar to that for situation calculus. Note that ES *regressable formulas* are sentences that do not mention the operator $\Box$.

Definition 3 (ES regression). *Let ϵ be an empty sequence of actions and let a basic action theory Θ be given. The regression operator* reg_Θ *is inductively defined on the sentence φ by:*

1. $\mathrm{reg}_\Theta(\varphi) = \mathrm{reg}_\Theta(\epsilon, \varphi)$
2. $\mathrm{reg}_\Theta(\alpha, t_1 = t_2) = (t_1 = t_2)$
3. $\mathrm{reg}_\Theta(\epsilon, p) = p$
4. $\mathrm{reg}_\Theta(\alpha \cdot t, p) = \mathrm{reg}_\Theta(\alpha, \chi(x, p)^t_x)$
5. $\mathrm{reg}_\Theta(\alpha, Poss(t)) = \mathrm{reg}_\Theta(\alpha, \varphi(x)^t_x)$
6. $\mathrm{reg}_\Theta(\alpha, SF(t)) = \mathrm{reg}_\Theta(\alpha, \psi(x)^t_x)$

7. $\mathrm{reg}_\Theta(\alpha, \neg\varphi) = \neg\, \mathrm{reg}_\Theta(\alpha, \varphi)$
8. $\mathrm{reg}_\Theta(\alpha, \varphi_1 \wedge \varphi_2) = \mathrm{reg}_\Theta(\alpha, \varphi_1) \wedge \mathrm{reg}_\Theta(\alpha, \varphi_2)$
9. $\mathrm{reg}_\Theta(\epsilon, \mathrm{K}\varphi) = \mathrm{K}\, \mathrm{reg}_\Theta(\epsilon, \varphi)$
10. $\mathrm{reg}_\Theta(\alpha \cdot t, \mathrm{K}\varphi) =$
    $\mathrm{reg}_\Theta(\alpha, (SF(t) \wedge \mathrm{K}(SF(t) \rightarrow [t]\varphi)) \vee (\neg SF(t) \wedge \mathrm{K}(\neg SF(t) \rightarrow [t]\varphi)))$
11. $\mathrm{reg}_\Theta(\alpha, [t]\varphi) = \mathrm{reg}_\Theta(\alpha \cdot t, \varphi)$
12. $\mathrm{reg}_\Theta(\alpha, \forall x.\varphi) = \forall x.\, \mathrm{reg}_\Theta(\alpha, \varphi)$

Theorem 1 ([9]). *$\Theta \models_{\mathsf{ES}} \varphi$ if and only if $\Theta_{una} \models_{\mathsf{ES}} \mathrm{reg}_\Theta(\varphi)$.*

Example 2. We use the theory Θ defined in Example 1 to regress the formula $[listen_1][open_1]\mathrm{K}\, married$. We have that:

$$\begin{aligned}
&\mathrm{reg}_\Theta([listen_1][open_1]\mathrm{K}\, married) = \\
&= \mathrm{reg}_\Theta(listen_1 \cdot open_1, \mathrm{K}\, married) \\
&= \mathrm{reg}_\Theta(listen_1, \mathrm{K}[open_1] married) \\
&= \mathrm{reg}_\Theta(listen_1, \mathrm{K}(lady_1 \vee married)) \\
&= \ \mathrm{reg}_\Theta(\epsilon, (lady_1 \wedge \mathrm{K}(lady_1 \rightarrow [listen_1](lady_1 \vee married))) \vee \\
&\qquad\qquad (\neg lady_1 \wedge \mathrm{K}(\neg lady_1 \rightarrow [listen_1](lady_1 \vee married))) \\
&= \ (lady_1 \wedge \mathrm{K}(lady_1 \rightarrow (lady_1 \vee married))) \vee \\
&\quad\ (\neg lady_1 \wedge \mathrm{K}(\neg lady_1 \rightarrow (lady_1 \vee married)))
\end{aligned}$$

which is equivalent to $lady_1 \vee (\neg lady_1 \wedge \mathrm{K}\, married)$.

3 Dynamic epistemic logic

A different tradition in modelling knowledge and change has been followed in, for example, [22], [23], [5], [24] and [6]. Logics in this tradition are in fact dynamic extentions of *epistemic logics*. The latter is a family of modal logics that use possible worlds semantics to represent agents' knowledge or beliefs. This idea, originally proposed by [25], has known great development in more recent works such as [26], [27] and [28].

In this section, we present epistemic logics and then present an extention wherein one can recast the RSL solution for the frame problem.

3.1 Epistemic logic

Definition 4 (The Languages $\mathcal{L}_{\mathsf{ELC}}$, $\mathcal{L}_{\mathsf{EL}}$ and $\mathcal{L}_{\mathsf{PL}}$). *Let P be a countable set of propositional letters, and let N be a finite set of agents. The language of epistemic logic with common knowledge $\mathcal{L}_{\mathsf{ELC}}$ is defined by the following BNF:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{K}_i\varphi \mid \mathrm{C}_G\varphi$$

where p ranges over P, i ranges over N, and G ranges over $\wp(N)$. We also define the language of epistemic logic without common knowledge $\mathcal{L}_{\mathsf{EL}}$ as the *language obtained from $\mathcal{L}_{\mathsf{ELC}}$ by dropping the operator 'C'; and the language of propositional logic $\mathcal{L}_{\mathsf{PL}}$ as the language obtained from $\mathcal{L}_{\mathsf{EL}}$ by dropping the operator 'K'.*

The formula $\mathrm{K}_i\varphi$ is read 'agent i knows (or believe) that φ', and the formula $\mathrm{C}_G\varphi$ is read 'all agents in group G commonly know (or believe) that φ'. We use the common abbreviations for the formulas $\top$, $\bot$, and for the operators $\vee$, $\rightarrow$, $\leftrightarrow$, and 'E'. The latter is defined by:

$$\mathrm{E}_G\varphi \stackrel{\text{def}}{=} \bigwedge_{i\in G} \mathrm{K}_i\varphi$$

where G ranges over $\wp(N)$. In addition, let M be one of the operators 'K_i', 'E_G' or 'C_G'. We sometimes write $\mathrm{M}^\ell\varphi$ to abbreviate the ℓ-fold nesting $\mathrm{M}\ldots\mathrm{M}\varphi$, if $\ell \geq 0$, where $\mathrm{M}^0\varphi$ is simply φ.

The formula $\mathrm{E}_G\varphi$ is read 'every agent in group G knows (or believes) that φ'. We indeed need the operator 'C' of common knowledge (or common belief) to represent, for example, that something is a convention in a group of agents. In this sense, saying that something is known by everybody in the group of agents G is different from saying that something is commonly known in the group of agents G. Discussions about the differences between these two concepts can be found, for example, in [29] and in [30].

Definition 5 (Epistemic model). *An epistemic model is a tuple $\langle W, R, V\rangle$, such that:*

- *W is a nonempty set of possible worlds;*
- *$R : N \rightarrow \wp(W \times W)$ associates an accessibility relation R_i to each $i \in N$; and*
- *$V : P \rightarrow \wp(W)$ associates an interpretation $V_p \subseteq W$ to each $p \in P$.*

Let $M = \langle W, R, V\rangle$ be an epistemic model and let $w \in W$, we call the pair (M, w) a pointed epistemic model. For convenience, we also define $R_i(w)$ as the set $\{w' \mid (w, w') \in R_i\}$.

Definition 6 (The satisfaction relation). *Let (M, w) be a pointed epistemic model. The satisfaction relation '$\Vdash$' is inductively defined as follows:*

$M, w \Vdash p$	*iff*	$w \in V_p$
$M, w \Vdash \neg\varphi$	*iff*	*not* $M, w \Vdash \varphi$
$M, w \Vdash \varphi \wedge \psi$	*iff*	$M, w \Vdash \varphi$ *and* $M, w \Vdash \psi$
$M, w \Vdash \mathrm{K}_i\varphi$	*iff*	$R_i(w) \subseteq [\![\varphi]\!]_M$
$M, w \Vdash \mathrm{C}_G\varphi$	*iff*	$(\bigcup_{i\in G} R_i)^+(w) \subseteq [\![\varphi]\!]_M$

where $[\![\varphi]\!]_M = \{w \in W \mid M, w \Vdash \mathsf{ELC}\varphi\}$ is the extension of φ in the model M, and the operator '+' is transitive closure.

In the pointed epistemic model (M, w), the distinguished world w is interpreted as the *actual* world. And the set $R_i(w)$ is the set of worlds that agent i considers possible at w. Then, a formula φ is known by agent i if and only if φ holds in all possible worlds that are accessible for agent i, i.e., iff φ holds in all worlds of $R_i(w)$. For the operator 'C' we use the transitive closure of relations R_i. Then a formula φ is commonly known by all agents, if and only if φ holds in all possible worlds that are "reachable" from w.

A *class of epistemic models*, is the set of all epistemic models that respect a subset of the properties above. Below, we list the classes that we address in this paper and their respective names.

- K: no restrictions;
- KT: each R_i is reflexive;
- $\mathsf{S4}$ (or $\mathsf{KT4}$): each R_i is reflexive and transitive;
- $\mathsf{KD45}$: each R_i is serial, transitive and euclidian; and
- $\mathsf{S5}$ (or $\mathsf{KT5}$): each R_i is reflexive and euclidian.

We note two things. First, K is also the class of all epistemic models. Second, we have the following relations between the above classes: $\mathsf{S5} \subset \mathsf{S4} \subset \mathsf{KT} \subset \mathsf{K}$, and also $\mathsf{S5} \subset \mathsf{S4} \subset \mathsf{KD45} \subset \mathsf{K}$. From now on whenever an epistemic model is in K, we call it a K-model. Similarly, we also use the terms KT-, $\mathsf{KD45}$-, $\mathsf{S4}$-, and $\mathsf{S5}$-model.

Definition 7 (Validity, satisfiability and valid consequence).
Let $\mathsf{C} \in \{\mathsf{K}, \mathsf{KT}, \mathsf{S4}, \mathsf{KD45}, \mathsf{S5}\}$, and let $\Psi \subseteq \mathcal{L}_{\mathsf{ELC}}$. A formula $\varphi \in \mathcal{L}_{\mathsf{ELC}}$ is:

- *C-valid (notation: $\models_{\mathsf{C}} \varphi$) if and only if for all pointed C-models (M, w), $(M, w) \Vdash \mathsf{C}\varphi$;*
- *C-satisfiable if and only if $\not\models_{\mathsf{C}} \neg\varphi$; and*
- *a valid C-consequence of Ψ (notation: $\Psi \models_{\mathsf{C}} \varphi$) if and only if for all pointed C-models (M, w), if for all $\psi \in \Psi$, $(M, w) \Vdash \mathsf{C}\psi$, then $(M, w) \Vdash \mathsf{C}\varphi$.*

Because of the relations between classes of models mentioned above, we obviously have that ELC-validities are also K-, KT-, $\mathsf{S4}$-, $\mathsf{KD45}$- and $\mathsf{S5}$-validities. We therefore abuse notation and write $\models_{\mathsf{ELC}} \varphi$ (instead of $\models_{\mathsf{K}} \varphi$) to mean that φ is valid in all classes of epistemic models.

Before closing this section, we list several known complexity results for epistemic logics. All of them were shown by [31]. Without the operator 'C' (i.e., for $\mathcal{L}_{\mathsf{EL}}$), satisfiability checking is:

- NP-complete for $|N| = 1$ in $\mathsf{KD45}$ and $\mathsf{S5}$;
- PSPACE-complete for $|N| \geq 2$ in $\mathsf{KD45}$ and $\mathsf{S5}$; and
- PSPACE-complete for any number of agents in K, KT and $\mathsf{S4}$.

With the operator 'C' (i.e., for $\mathcal{L}_{\mathsf{ELC}}$), satisfiability checking is:

- PSPACE-complete for $|N| = 1$ in $\mathsf{S4}$, $\mathsf{KD45}$ and $\mathsf{S5}$;
- EXPTIME-complete for $|N| \geq 2$ in $\mathsf{S4}$, $\mathsf{KD45}$ and $\mathsf{S5}$; and
- EXPTIME-complete for any number of agents in K and KT.

3.2 Adding dynamic operators

Definition 8 (The languages $\mathcal{L}_{\mathsf{DELC}}$ and $\mathcal{L}_{\mathsf{DEL}}$). *The language of dynamic epistemic logic with common knowledge $\mathcal{L}_{\mathsf{DELC}}$ is defined by the following BNF:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{K}_i\varphi \mid \mathrm{C}_G\varphi \mid [!\varphi]\varphi \mid [!!\varphi]\varphi \mid [\sigma]\varphi$$
$$\sigma ::= \epsilon \mid p := \varphi, \sigma$$

where p ranges over P, i ranges over N, G ranges over $\wp(N)$, and ϵ is an empty assignment. Similarly to ELC*, we define the language of dynamic epistemic logic without common knowledge $\mathcal{L}_{\mathsf{DEL}}$ as the language obtained from $\mathcal{L}_{\mathsf{DELC}}$ by dropping the operator '*C*'.*

Again, the formula $[\alpha]\varphi$ is read 'φ holds after all possible executions of α'. The action $!\varphi$ is the public announcement of φ. The action $!!\varphi$ is the public test of φ.[4] The action $p := \varphi$ is the public assignment of the truth value of φ to the atom p. For example, $p := \bot$ is a public assignment, and $\mathrm{K}_i[p := \bot]\neg p$ is a formula. When assignments are made in parallel, the same propositional letter can appear only once on the left hand side of the operator '$:=$'. For convenience, we identify a complex announcement of the form $(p_1 := \varphi_1, \ldots, p_n := \varphi_n)$ with the set $\{p_1 := \varphi_1, \ldots, p_n := \varphi_n\}$. Thus ϵ is identified with $\emptyset$.

The fragment of DELC without assignments and without tests is Plaza's public announcement logic with common knowledge (PALC) [22], whose fragment without common knowledge we note PAL.

Announcements and tests model epistemic actions, while assignments model ontic actions. For example, if we do not consider execution preconditions of actions, then the epistemic action listen_1 of Example 1 can be modelled as $!!\mathit{lady}_1$. And the ontic action open_1 can be modelled as the complex assignment

$$\sigma_{\mathit{open}_1} = \{\mathit{alive} := (\mathit{lady}_1 \wedge \mathit{alive}), \mathit{married} := (\mathit{lady}_1 \vee \mathit{married})\}$$

Definition 9 (The satisfaction relation). *Formulas in $\mathcal{L}_{\mathsf{DELC}}$ are interpreted in epistemic models. The satisfaction relation '$\Vdash$' is extended with the following three clauses:*

$M, w \Vdash [!\varphi]\psi$	*iff*	$M, w \Vdash \varphi$ *implies* $M^{!\varphi}, w \Vdash \psi$
$M, w \Vdash [!!\varphi]\psi$	*iff*	$M, w \Vdash [!\varphi]\psi$ *and* $M, w \Vdash [!\neg\varphi]\psi$
$M, w \Vdash [\sigma]\varphi$	*iff*	$M^{\sigma}, w \Vdash \varphi$

where $M^{!\varphi}$ and M^{σ} are updates of the epistemic model M that are defined as:

$$M^{!\varphi} = \langle W^{!\varphi}, R^{!\varphi}, V^{!\varphi}\rangle$$
$$W^{!\varphi} = W \cap [\![\varphi]\!]_M$$
$$R_i^{!\varphi} = R_i \cap ([\![\varphi]\!]_M \times [\![\varphi]\!]_M)$$
$$V^{!\varphi}(p) = V(p) \cap [\![\varphi]\!]_M$$

[4] Note that both announcement and test operators in DELC are different from the test operator in propositional dynamic logic (noted '?'): the latter does not have epistemic effects, while the other ones do.

and

$$M^\sigma = \langle W, R, V^\sigma \rangle$$
$$V^\sigma(p) = [\![\sigma(p)]\!]_M$$

and where $\sigma(p)$ is the formula assigned to p in σ. If there is no such a formula, i.e., if there is no $p := \varphi$ in σ, then $\sigma(p) = p$. (In particular $\epsilon(p) = p$ for all p.)

Clearly, the public test operator can be defined in terms of the public announcement operator and does not increase the expressivity of DELC. Nevertheless, its definition as a primitive operator is important in order to provide a polynomial translation of sensing actions into DELC. If tests were defined as abbreviations, the resulting translated formula might increase exponentially in size.

Definition 10 (Validity and satisfiability). *A formula $\varphi \in \mathcal{L}_{\mathsf{DELC}}$ is:*

- DELC-*valid (notation: $\models_{\mathsf{DELC}} \varphi$) if and only if for all pointed epistemic models (M, w), $(M, w) \Vdash \varphi$; and*
- DELC-*satisfiable if and only if $\not\models_{\mathsf{DELC}} \neg\varphi$.*

It has been proved that every formula in DEL can be *reduced* to an equivalent formula in EL by the following method.

Theorem 2 (DEL reduction [6]). *Let $\varphi \in \mathcal{L}_{\mathsf{DEL}}$. Then $\models_{\mathsf{DEL}} \varphi$ if and only if $\models_{\mathsf{EL}} \mathrm{red}(\varphi)$, where $\mathrm{red}(\varphi)$ is inductively defined by:*

1. $\mathrm{red}(p) = p$;
2. $\mathrm{red}(\neg\varphi) = \neg\,\mathrm{red}(\varphi)$;
3. $\mathrm{red}(\varphi_1 \wedge \varphi_2) = \mathrm{red}(\varphi_1) \wedge \mathrm{red}(\varphi_2)$;
4. $\mathrm{red}(\mathrm{K}_i\varphi) = \mathrm{K}_i\,\mathrm{red}(\varphi)$;
5. $\mathrm{red}([\sigma]p) = \mathrm{red}(\sigma(p))$;
6. $\mathrm{red}([\sigma]\neg\varphi) = \mathrm{red}(\neg[\sigma]\varphi)$;
7. $\mathrm{red}([\sigma](\varphi_1 \wedge \varphi_2)) = \mathrm{red}([\sigma]\varphi_1 \wedge [\sigma]\varphi_2)$;
8. $\mathrm{red}([\sigma]\mathrm{K}_i\varphi) = \mathrm{red}(\mathrm{K}_i[\sigma]\varphi)$;
9. $\mathrm{red}([!\psi]p) = \mathrm{red}(\psi \to p)$;
10. $\mathrm{red}([!\psi]\neg\varphi) = \mathrm{red}(\psi \to \neg[!\psi]\varphi)$;
11. $\mathrm{red}([!\psi](\varphi_1 \wedge \varphi_2) = \mathrm{red}([!\psi]\varphi_1 \wedge [!\psi]\varphi_2)$;
12. $\mathrm{red}([!\psi]\mathrm{K}_i\varphi) = \mathrm{red}(\psi \to \mathrm{K}_i[!\psi]\varphi)$;
13. $\mathrm{red}([!!\psi]p) = \mathrm{red}(p)$;
14. $\mathrm{red}([!!\psi]\neg\varphi) = \mathrm{red}(\neg[!!\psi]\varphi)$;
15. $\mathrm{red}([!!\psi](\varphi_1 \wedge \varphi_2) = \mathrm{red}([!!\psi]\varphi_1 \wedge [!!\psi]\varphi_2)$;
16. $\mathrm{red}([!!\psi]\mathrm{K}_i\varphi = \mathrm{red}((\psi \wedge \mathrm{K}_i(\psi \to [!!\psi]\varphi)) \vee (\neg\psi \wedge \mathrm{K}_i(\neg\psi \to [!!\psi]\varphi)))$.

However, DEL reduction has the same problem as ES regression: the size of the formula $\mathrm{red}(\varphi)$ can be exponentially larger than the size of φ. An example is the family of formulas defined in [8, Theorem 2]:

$$\varphi_0 = \top$$
$$\varphi_{n+1} = \langle\langle\varphi_n\rangle\neg\mathrm{K}_i\neg\top\rangle\neg\mathrm{K}_j\neg\top$$

Moreover, no such equivalences exist for the operator 'C' [23]. In Section 5 we provide a method that can do such a reduction polynomially.

4 From ES to DEL

The regression procedure in Definition 3 is very similar to the reduction in Theorem 2. This suggests that formulas in ES could in some way be encoded in DEL. In this section we show that:

- the executability preconditions ($Poss$) of actions can be modelled as public announcements;
- the sensing results (SF) of actions can be modelled as public tests; and
- the changes brought about by actions can be modelled as public assignments.

It must be noted however, that DEL is a propositional language. Then, it clearly does not have the same expressivity as the whole language of ES defined in [9]. What we provide in this section is a translation from the *variable-free fragment* of $\mathcal{L}_{\mathsf{ES}}$ to $\mathcal{L}_{\mathsf{DEL}}$. This fragment is defined below.

Definition 11 (The language $\mathcal{L}^0_{\mathsf{ES}}$). *Let P be a set of countably many fluents of arity 0, and let A be a set of countably many actions. The language $\mathcal{L}^0_{\mathsf{ES}}$ is defined by the following BNF:*

$$\varphi ::= a = a \mid p \mid Poss(a) \mid SF(a) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{K}\varphi \mid [a]\varphi$$

where p ranges over P and a ranges over A.

That is, $\mathcal{L}^0_{\mathsf{ES}}$ is obtained from $\mathcal{L}_{\mathsf{ES}}$ by dropping variables, quantifiers and the operator $\Box$. The reader can also notice that the fragment $\mathcal{L}^0_{\mathsf{ES}}$ differs from $\mathcal{L}_{\mathsf{DEL}}$ only on the $[\cdot]$ operator and equality. Given the observations made in the beginning of the section, the translation from this fragment to $\mathcal{L}_{\mathsf{DEL}}$ is almost straightforward.

Definition 12 (Translation from ES to DEL). *Let a basic action theory Θ as in Definition 2 be given. We define the translation tra_Θ as a function from $\mathcal{L}^0_{\mathsf{ES}}$ to single-agent $\mathcal{L}_{\mathsf{DEL}}$ inductively as follows:*

1. $\mathrm{tra}_\Theta(a_1 = a_2) = \begin{cases} \bot \text{ if } a_1 \neq a_2 \in \Theta_{una} \\ \top \text{ else} \end{cases}$
2. $\mathrm{tra}_\Theta(p) = p$
3. $\mathrm{tra}_\Theta(Poss(a)) = \mathrm{tra}_\Theta(\varphi(x)^a_x)$
4. $\mathrm{tra}_\Theta(SF(a)) = \mathrm{tra}_\Theta(\psi(x)^a_x)$
5. $\mathrm{tra}_\Theta(\neg\varphi) = \neg\,\mathrm{tra}_\Theta(\varphi)$
6. $\mathrm{tra}_\Theta(\varphi_1 \wedge \varphi_2) = \mathrm{tra}_\Theta(\varphi_1) \wedge \mathrm{tra}_\Theta(\varphi_2)$
7. $\mathrm{tra}_\Theta(\mathrm{K}\varphi) = \mathrm{K}_i\,\mathrm{tra}_\Theta(\varphi)$
8. $\mathrm{tra}_\Theta([a]\varphi) = [!!\,\mathrm{tra}_\Theta(\psi(x)^a_x)][\sigma_a]\,\mathrm{tra}_\Theta(\varphi)$

where σ_a is the complex assignment defined by:

$$\{p := \mathrm{tra}_\Theta(\chi(x,p)^a_x) \mid p \in \mathit{Eff}(a)\}$$

where $\mathit{Eff}(a) = \mathit{Eff}^+(a) \cup \mathit{Eff}^-(a)$ is the set of fluents such that the truth value can be flipped by the action a, defined by:

$$\{p \mid \forall x.\Box([x]p \leftrightarrow \chi(x,p)) \in \Theta_{post} \text{ and } a \text{ is mentioned in } \chi(x,p)\}$$

For example, $\mathrm{tra}_\Theta[listen_1]\mathrm{K}\,lady_1 = [!!\varphi(x)_x^{listen_1}][\emptyset]\mathrm{K}_i\,lady_1$, which is equivalent to $[!!lady_1]\mathrm{K}_i\,lady_1$. For an example with an ontic action, we can take

$$\mathrm{tra}_\Theta([open_1]\,alive) = \\ [!!\varphi(x)_x^{open_1}][alive := lady_1 \land alive, married := lady_1 \lor married]\,alive$$

which is equivalent to $[!!\top](lady_1 \land alive)$, and therefore also equivalent to $lady_1 \land alive$.

In order to prove that this translation is polynomial, we define the function len that returns the *length* of a given expression. In the case of sets and tuples, we count the length of each element and also the commas and delimiters. That is, the length of a given set X is:

$$\mathrm{len}(X) = 1 + \sum_{x \in X} (1 + \mathrm{len}(x))$$

while for a given tuple $Y = \langle y_1, \ldots, y_n \rangle$ it is

$$\mathrm{len}(Y) = 1 + \sum_{k=1}^{n} (1 + \mathrm{len}(y_k))$$

For formulas in $\mathcal{L}_{\mathsf{ELC}}$, we use the inductive definition that follows (note that G is a set):

$$\begin{aligned} \mathrm{len}(p) &= 1 \\ \mathrm{len}(\neg\varphi) &= 1 + \mathrm{len}(\varphi) \\ \mathrm{len}(\varphi \land \psi) &= 1 + \mathrm{len}(\varphi) + \mathrm{len}(\psi) \\ \mathrm{len}(\mathrm{K}_i\varphi) &= 2 + \mathrm{len}(\varphi) \\ \mathrm{len}(\mathrm{C}_G\varphi) &= 1 + \mathrm{len}(G) + \mathrm{len}(\varphi) \end{aligned}$$

For formulas in $\mathcal{L}^0_{\mathsf{ES}}$ we also use:

$$\mathrm{len}(t_1 = t_2) = 3\,\mathrm{len}([a]\varphi) = 2 + \mathrm{len}(\varphi)$$

and for formulas in $\mathcal{L}_{\mathsf{DELC}}$ we also use (we consider σ as a set of assignments):

$$\begin{aligned} \mathrm{len}([!\varphi]\psi) &= 1 + \mathrm{len}(\varphi) + \mathrm{len}(\psi) \\ \mathrm{len}([!!\varphi]\psi) &= 1 + \mathrm{len}(\varphi) + \mathrm{len}(\psi) \\ \mathrm{len}([\sigma]\varphi) &= 1 + \mathrm{len}(\sigma) + \mathrm{len}(\varphi) \\ \mathrm{len}(p := \varphi) &= 2 + \mathrm{len}(\varphi) \end{aligned}$$

For example, $\mathrm{len}([\{p := q, q := p \land q\}]\mathrm{K}_i p = 1 + \mathrm{len}(\{p := q, q := p \land q\}) + \mathrm{len}(\mathrm{K}_i p) = 12 + 2 + 1 = 15$.

Theorem 3 (Polynomial translation). *Let Θ be a basic action theory and let $\varphi \in \mathcal{L}^0_{\mathsf{ES}}$. Then $\mathrm{len}(\mathrm{tra}_\Theta(\varphi)) \leq \mathcal{O}(\mathrm{len}(\Theta) \times \mathrm{len}(\varphi))$.*

The proof is by induction on the structure of φ. We omit details here.

Theorem 4. *Let Θ be a basic action theory and let $\varphi \in \mathcal{L}^0_{\mathsf{ES}}$. $\Theta \models_{\mathsf{ES}} \varphi$ if and only if $\models_{\mathsf{DEL}} \mathrm{tra}_\Theta(\varphi)$.*

Again, the proof is by induction on the structure of φ. It uses the regresssion procedures for ES and DEL.

Hence, when the formulas in Θ respect some restrictions, namely when φ, $\psi(x)$, $\chi(x)$ and $\psi(x,p)$ are in $\mathcal{L}^0_{\mathsf{ES}}$, the problem of deciding whether $\Theta \models_{\mathsf{ES}} \varphi$ can be polynomially reduced to a validity problem in DEL.

5 Optimal reduction

We now give a polynomial reduction from DELC to ELC. The idea is first eliminate assignments, and then apply Lutz' reduction [8] to eliminate announcements. In this section, we consider that $\mathcal{L}_{\mathsf{DELC}}$ does not have the public test operator '!!'. As mentioned earlier, it does not restrict the expressivity of DELC, since tests can be decomposed into two announcements. Remember that the formula $[!!\varphi]\psi$ is equivalent to $[!\varphi]\psi \wedge [!\neg\varphi]\psi$. The reason of doing this restriction is that the reduction procedure to be presented, when extended for handling public tests is not polynomial any more.

We also remark that the reduction does not impose any other restriction on formulas. In particular, this means that the entire method works for basic action theories where the formulas $\varphi(x)$, $\psi(x)$ and $\chi(x)$ mention knowledge and common knowledge operators. This is a generalization of Lakemeyer & Levesque's approach where these formulas are fluent formulas (in particular, without the operator 'K').

5.1 Eliminating assignments

We apply a technique that is fairly standard in automated theorem proving [32].

Theorem 5 (Assignment elimination). *Let $[p_1 := \varphi_1, \ldots, p_n := \varphi_n]\psi$ be a subformula of a formula χ in $\mathcal{L}_{\mathsf{DELC}}$. Let ψ' be obtained from ψ by substituting every occurrence of p_k by x_{p_k}, where x_{p_k} is a new propositional letter not occurring in χ. Let χ' be obtained from χ by replacing $[p_1 := \varphi_1, \ldots, p_n := \varphi_n]\psi$ by ψ'. Let B abbreviate the conjunction $\bigwedge_{1 \le k \le n}(x_{p_k} \leftrightarrow \varphi_k)$ (of bi-implications).*

1. *For $\chi \in \mathcal{L}_{\mathsf{DEL}}$ and $|N| = 1$: χ is DEL-satisfiable if and only if*

$$\chi' \wedge \bigwedge_{\ell \le \mathrm{md}(\varphi)} \mathrm{K}_i^\ell B$$

is DEL-satisfiable, where the modal depth $\mathrm{md}(\varphi)$ is the maximal number of nested modal operators of ψ.

2. *For* $\chi \in \mathcal{L}_{\mathsf{DEL}}$ *and* $|N| \geq 2$*:* χ *is* DEL*-satisfiable if and only if*

$$\chi' \wedge \bigwedge_{\ell \leq \mathrm{md}(\varphi)} \mathrm{E}_N^\ell B$$

is DEL*-satisfiable.*

3. *For* $\chi \in \mathcal{L}_{\mathsf{DELC}}$*:* χ *is* DELC*-satisfiable if and only if*

$$\chi' \wedge \mathrm{C}_N B$$

is DELC*-satisfiable.*

Proof. To simplify suppose a single assignment, i.e., that the subformula of χ is $[p := \varphi]\psi$. We do the last case here. The other ones are analogous, and left to the reader.

From the left to the right. Suppose that $M = \langle W, R, V \rangle$ is an epistemic model such that $M, w \Vdash \chi$. Then we construct an epistemic model $M_{x_p} = \langle W, R, V_{x_p} \rangle$, where

$$V_{x_p}(p) = V(p) \quad \text{for all } p \neq x_p \text{ and}$$
$$V_{x_p}(x_p) = [\![\varphi]\!]_M$$

First, note that $M_{x_p}, w \Vdash \chi$ (because x_p does not appear in χ).

Second, note that $M_{x_p} \Vdash x_p \leftrightarrow \varphi$ (because $[\![x_p]\!]_{M_{x_p}} = [\![\varphi]\!]_{M_{x_p}}$).
Therefore $M_{x_p}, w \Vdash \mathrm{C}_N(x_p \leftrightarrow \varphi)$, i.e., $M_{x_p}, w \Vdash \mathrm{C}_N B$.

Third, note that for every $v \in W$ we have that:
$M_{x_p}, v \Vdash [p := \varphi]\psi$
iff $M_{x_p}^{p := \varphi}, v \Vdash \psi$
iff $M_{x_p}^{p := \varphi}, v \Vdash \psi'$ (because $V_{x_p}^{p := \varphi}(p) = V_{x_p}^{p := \varphi}(x_p)$).
Therefore $M_{x_p} \Vdash [p := \varphi]\psi \leftrightarrow \psi'$

Therefore $M_{x_p}, w \Vdash \chi' \wedge \mathrm{C}_N B$.

From the right to the left.
Suppose w.l.o.g. that M is generated from w.
Suppose that $M, w \Vdash \chi' \wedge \mathrm{C}_N(x_p \leftrightarrow \varphi)$.
Then $M \Vdash x_p \leftrightarrow \varphi$, i.e., $V(x_p) = [\![\varphi]\!]_M$.
Then, for all $v \in W$:
$M, v \Vdash \psi'$ iff $M^{p := \varphi}, v \Vdash \psi$ (because $V(x_p) = [\![\varphi]\!]_M = V^{p := \varphi}(p)$).
In other words, $M \Vdash \psi' \leftrightarrow [p := \varphi]\psi$.
Therefore $M, w \Vdash \chi$. □

Intuitively, the conjuncts $\mathrm{K}_i(x_{p_k} \leftrightarrow \varphi_k)$ set the value of the new propositional letter x_{p_k} to that of φ_k. To guarantee that the equivalences hold everywhere in the model we need to use a *master modality*. In the case of DEL we use the 'everybody knows' operator, that has to be iterated up to the modal depth of the formula.

Renaming avoids exponential blow-up. This allows the definition of reduction operators $\text{reg}_{\mathsf{DEL}}$, $\text{reg}_{\mathsf{DELC}}$ that iteratively eliminate all assignments.

For example, consider the formula $\neg[!\neg lady][lady := \neg lady]\text{K}_i lady$. Its reduction is $\neg[!\neg lady]\text{K}_i x_{lady} \wedge \text{K}_i(x_{lady} \leftrightarrow \neg lady)$.

Theorem 6. $\text{reg}_{\mathsf{DEL}}$ *and* $\text{reg}_{\mathsf{DELC}}$ *are polynomial transformations, and preserve satisfiability in the respective logics.*

Proof. Satisfiability-equivalence follows from Theorem 5.

For the common-knowledge case we prove that the size of the reduction of χ is at most $\text{len}(\chi) \times (\text{len}(\chi) + 6)$, and for the case of DEL we prove that the size of the reduction of χ is at most $\text{len}(\chi)^2 \times (\text{len}(\chi) + 6)$. Indeed, in Theorem 5 the size of χ' is at most $\text{len}(\chi)$, the size of each equivalence in B is at most $\text{len}(\chi) + 4$, and the number of these equivalences is bound by the number of (atomic) assignments in χ, which is at most $\text{len}(\chi)$. In the case of operators 'K' and 'E' the number of equivalences has to be multiplied by the modal depth of χ, which is at most $\text{len}(\chi)$.

1. For $|N| = 1$. If $\chi \in \mathcal{L}_{\mathsf{DEL}}$, then χ is DEL-satisfiable if and only if $\text{reg}_{\mathsf{DEL}}(\chi)$ is PAL-satisfiable;
2. For $|N| \geq 2$. If $\chi \in \mathcal{L}_{\mathsf{DEL}}$, then χ is DEL-satisfiable if and only if $\text{reg}_{\mathsf{DEL}}(\chi)$ is PAL-satisfiable;
3. If $\chi \in \mathcal{L}_{\mathsf{DELC}}$, then χ is DELC-satisfiable if and only if $\text{reg}_{\mathsf{DELC}}(\chi)$ is PALC-satisfiable.

□

5.2 Eliminating announcements

Once assignments are eliminated, we can eliminate announcements by Lutz' procedure that we recall here. For simplicity we show only the case without common knowledge.

First we compute the set of contextual subformulas which are inductively defined as follows.

$$\begin{aligned}
S\text{ub}(p) &= \{(\epsilon, p)\} \\
S\text{ub}(\neg\varphi) &= S\text{ub}(\varphi) \cup \{(\epsilon, \neg\varphi)\} \\
S\text{ub}(\varphi \wedge \psi) &= S\text{ub}(\varphi) \cup S\text{ub}(\psi) \cup \{(\epsilon, \varphi \wedge \psi)\} \\
S\text{ub}(\text{K}_i\varphi) &= S\text{ub}(\varphi) \cup \{(\epsilon, \text{K}_i\varphi)\} \\
S\text{ub}([!\varphi]\psi) &= S\text{ub}(\varphi) \cup \{(\varphi \cdot \tau, \chi) \mid (\tau, \chi) \in S\text{ub}(\psi)\} \cup \{(\epsilon, [!\varphi]\psi)\}
\end{aligned}$$

where τ denotes lists, ϵ is the empty list, and '$\cdot$' is concatenation.

Intuitively, $S\text{ub}(\varphi)$ is the set of 'relevant' subformulas of φ together with the sequence of announcements in the scope of which they occur. $(\tau, \psi) \in S\text{ub}(\varphi)$ means that subformula ψ of φ is in the scope of the sequence τ of announcements. Now, let φ be a multi-agent formula whose DEL-satisfiability is to be decided.

We introduce a set of fresh propositional letters $P_\varphi = \{x^\tau_\psi \mid (\tau, \psi) \in S\mathrm{ub}(\varphi)\}$. Then the reduction of φ is:

$$\mathrm{reg}_{\mathsf{DEL}}(\varphi) = x^\epsilon_\varphi \wedge \bigwedge_{\ell \le \mathrm{md}(\varphi)} \bigwedge_{(\tau,\psi)\in S\mathrm{ub}(\varphi)} \mathrm{E}^\ell_N(B^\tau_\psi)$$

where $\mathrm{md}(\varphi)$ is the modal depth of φ, $\mathrm{E}^\ell_N\varphi$ abbreviates $\mathrm{E}_N \ldots \mathrm{E}_N\varphi$ (ℓ times), and the bi-implications B^τ_ψ are inductively defined as follows:

$$\begin{array}{ll} B^\tau_p & = x^\tau_p \leftrightarrow p \\ B^\tau_{\neg\varphi} & = x^\tau_{\neg\varphi} \leftrightarrow \neg x^\tau_\varphi \\ B^\tau_{\varphi\wedge\psi} & = x^\tau_{\varphi\wedge\psi} \leftrightarrow (x^\tau_\varphi \wedge x^\tau_\psi) \\ B^\tau_{\mathrm{K}_i\varphi} & = x^\tau_{\mathrm{K}_i\varphi} \leftrightarrow \mathrm{K}_i(\bigwedge_{\mu\in\mathrm{pre}(\tau)} x^\mu_{\mu/\tau} \to x^\tau_\varphi) \\ B^\tau_{[!\varphi]\psi} & = x^\tau_{[!\varphi]\psi} \leftrightarrow (x^\tau_\varphi \to x^{\tau\cdot\varphi}_\psi) \end{array}$$

where $\mathrm{pre}(\tau)$ is the set of true prefixes of τ, and μ/τ is the leftmost symbol of τ that is not in μ. When the sequence τ is empty, then the conjunction collapses to true. B^τ_ψ guarantees that x^τ_ψ is true exactly where ψ is true.

When applied to a formula in $\mathcal{L}_{\mathsf{DEL}}$ without assignments, $\mathrm{reg}_{\mathsf{DEL}}$ returns an EL-formula. For example, consider the formula $\neg[!p]\mathrm{K}_i p$. The set of relevant bi-implications is $B = \{x^\epsilon_{\neg[!p]\mathrm{K}_i p} \leftrightarrow \neg x^\epsilon_{[!p]\mathrm{K}_i p}, x^\epsilon_{[!p]\mathrm{K}_i p} \leftrightarrow (x^\epsilon_p \to x^{\epsilon\cdot p}_{\mathrm{K}_i p}), x^{\epsilon\cdot p}_{\mathrm{K}_i p} \leftrightarrow \mathrm{K}_i(x^\epsilon_p \to x^{\epsilon\cdot p}_p), x^{\epsilon\cdot p} \leftrightarrow p, x^\epsilon \leftrightarrow p\}$. Then $\mathrm{reg}_{\mathsf{DEL}}(\neg[!p]\mathrm{K}_i p) = x^\epsilon_{\neg[!p]\mathrm{K}_i p} \wedge \mathrm{K}_i \bigwedge B$, which successively implies x^ϵ_p, $\neg x^{\epsilon\cdot p}_{\mathrm{K}_i p}$, and $\neg\mathrm{K}_i(x^\epsilon_p \to x^{\epsilon\cdot p}_p)$. The latter is inconsistent with $\mathrm{K}_i(x^{\epsilon\cdot p} \leftrightarrow p)$ and $\mathrm{K}_i(x^\epsilon \leftrightarrow p)$ which are the last two bi-implications prefixed by K_i.

Theorem 7. *[8]* PAL*-satisfiability has the same computational complexity of* EL*-satisfiability.*

5.3 Eliminating both

Via Theorem 4 we obtain the result below.

Corollary 1. *Satisfiability checking of formulas in* $\mathcal{L}^0_{\mathsf{ES}}$ *has the same computational complexity of satisfiability checking in* EL*.*

That is the satisfiability problem for formulas in $\mathcal{L}^0_{\mathsf{ES}}$ is:

1. NP-complete if $|N| = 1$;
2. PSPACE-complete if $|N| \geq 2$; and
3. EXPTIME-complete if common knowledge is involved.

These results apply to the plan verification problem:[5] given the basic action theory Θ, an $\mathcal{L}_{\mathsf{ELC}}$-formula φ_0 describing the initial situation, an $\mathcal{L}_{\mathsf{ELC}}$-formula ψ_ω

[5] Called "projection problem" in [3, p.22].

describing the goal, and a sequence of actions (or plan) made up of $a_1, \dots, a_n \in A$, we have to decide whether:

$$\Theta \models_{\mathsf{ES}} \varphi_0 \to \langle a_1 \rangle \dots \langle a_n \rangle \varphi_\omega$$

Upper bounds follow from Corollary 1. Lower bounds obtain because satisfiability of χ can be checked by putting $\Theta_{\text{una}} = \Theta_{\text{pre}} = \Theta_{\text{sense}} = \Theta_{\text{post}} = \emptyset$, $\varphi = \neg\chi$, $n = 0$ and $\psi = \bot$.Therefore, the plan verification problem inherits the complexity of the underlying logic.

6 Discussion and conclusion

We have modelled the frame problem in dynamic epistemic logic by providing counterparts for situation calculus style ontic and sensing actions, and we have given complexity results using that translation. As far as we know, this is the first optimal decision procedure for a Reiter-style solution to the frame problem.

A similar approach for epistemic actions has been proposed in [33]. The logic for epistemic tests therein has an operator called 'test that', which corresponds to public announcement. However, that logic has no ontic actions, and the regression procedure is suboptimal. In addition, the complexity result given there is restricted to non-nested tests, while here we permit any formula under the scope of the dynamic operators.

Scherl & Levesque's epistemic extension of Reiter's solution allows for sensing actions $!!\varphi$, which test whether some formula φ is true. Such sensing actions can be viewed as abbreviating the nondeterministic composition of two announcements. We could have defined them as: $!!\varphi = (!\varphi \cup !\neg\varphi)$, where '$\cup$' is nondeterministic composition. The expansion of such abbreviations however leads to exponential blowing-up, which does not allow to extend our approach and integrate primitive sensing actions: it is not clear how the associated successor state axiom (cf. axiom SSAK in Section 2)

$$[!!\varphi]\mathrm{K}_i\psi \leftrightarrow ((\varphi \to \mathrm{K}_i(\varphi \to [!!\varphi]\psi)) \wedge (\neg\varphi \to \mathrm{K}_i(\neg\varphi \to [!!\varphi]\psi)))$$

could be transformed into a polynomial transformation. Further evidence that the presence of sensing actions increases complexity is provided by the result in [34] that plan verification in this case is Π_2^p-complete. We therefore leave integration of sensing actions to future work.

The present paper also tries to build a bridge between situation calculus and dynamic epistemic logics research communities. This bridge should aid to bring about advancements on both sides. For instance, we believe that it is possible to integrate non-public actions to Scherl & Levesque's approach. For doing so, one could do as in [35] (see also [36]) and use an equivalent approach that replaces basic action theories by structures of the form $\langle Poss(a), \gamma^+(a), \gamma^-(a) \rangle$. The components of this structure play the same role as their counterparts in the present approach. Then, one can follow [23] (see also [37]) and add the component $R_i(a)$ to this structure. It is the set of actions that agent i cannot distinguish from the action a when this action is executed. This representation is parsimonious and [23] also provide a reduction method that extends the one in Theorem 2.

From the other side of the bridge we can cite the high expressivity of the entire language of situation calculus (and also ES). With the argument of keeping decidability and elegance, the dynamic epistemic logics community generally avoids adding quantifiers, predicates, functions, etc, to their formalisms. Reiter's, Scherl & Levesque's and Lakemeyer & Levesques's approaches show that, under reasonable restrictions, these components can be added and even be used in practice, as done in the GOLOG programming language.

References

1. Thielscher, M.: From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. Artificial Intelligence **111** (1999) 277–299
2. Reiter, R.: The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In Lifschitz, V., ed.: Papers in Honor of John McCarthy. Academic Press Professional Inc. (1991) 359–380
3. Scherl, R., Levesque, H.: Knowledge, action and the frame problem. Artificial Intelligence **144** (2003) 1–39
4. Levesque, H., Reiter, R., Lespérance, Y., Lin, F., Scherl, R.: GOLOG: A logic programming language for dynamic domains. Journal of Logic Programming **31** (1997) 59–83
5. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic epistemic logic with assignment. In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M., Wooldridge, M., eds.: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), ACM (2005) 141–148
6. Kooi, B.: Expressivity and completeness for public update logic via reduction axioms. Journal of Applied Non-Classical Logics **17** (2007) 231–253
7. van Benthem, J.: Modal logic meets situation calculus. Manuscript (2007)
8. Lutz, C.: Complexity and succintness of public announcement logic. In Stone, P., Weiss, G., eds.: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS). (2006) 137–144
9. Lakemeyer, G., Levesque, H.: Semantics for a useful fragment of the situation calculus. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Professional Book Center (2005) 490,496
10. McCarthy, J.: Situations, actions and causual laws. In Minsky, M., ed.: Semantic Information Processing. The MIT Press (1968) 410–417
11. Gelfond, M., Lifschitz, V., Rabinov, A.: What are the limitations of situation calculus. In Boyer, R., ed.: Essays in Honor of Woody Bledsoe. Kluwer Academic Publishers Group (1991) 167–180
12. Scherl, R., Levesque, H.: The frame problem and knowledge-producing actions. In: Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI), The AAAI Press (1993) 689–695
13. Bacchus, F., Halpern, J., Levesque, H.: Reasoning about noisy sensors and effectors in the situation calculus. Artificial Intelligence **111** (1999) 171–208
14. Reiter, R.: On knowledge-based programming with sensing in the situation calculus. ACM Transactions on Computational Logic (2001) 433–437

15. Lakemeyer, G., Levesque, H.: Situations, si! Situation terms, no! In: Proceedings of the International Conference on Knowledge Representation and Reasoning (KR), AAAI Press (2004) 516–526
16. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., Michie, D., eds.: Machine Intelligence 4. Edinburgh University Press (1969) 463–502
17. Haas, A.: The case for domain-specific frame axioms. In Brown, F.M., ed.: The Frame Problem in Artificial Intelligence. Morgan Kaufmann (1987)
18. Schubert, L.: Monotonic solution of the frame problem in the situation calculus: An efficient method for worlds with fully specified actions. In Kynburg, H., Loui, R., Carlson, G., eds.: Knowledge Representation and Defeasible Reasoning. Kluwer Academic Publishing (1990) 23–67
19. Reiter, R.: Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. The MIT Press (2001)
20. Smullyan, R.: The Lady or the Tiger? and Other Logic Puzzles Including a Mathematical Novel That Features Godel's Great Discovery. Random House Puzzles & Games (1992)
21. Waldinger, R.: Achieving several goals simutaneously. In Elock, E., Michie, D., eds.: Machine Intelligence. Volume 8. Ellis Harwood (1977) 94–136
22. Plaza, J.: Logics of public communications. In Emrich, M.L., Hadzikadic, M., Pfeifer, M.S., Ras, Z.W., eds.: Proceedings of the Fourth International Symposium on Methodologies for Intelligent Systems (ISMIS). (1989) 201–216
23. Baltag, A., Moss, L., Solecki, S.: The logic of common knowledge, public announcements, and private suspicions. In: Proceedings of the seventh Theoretical Aspects of Rationality and Knowledge conferene (TARK), Morgan Kaufmann Publishers Inc. (1998) 43–46
24. van Benthem, J.: "One is a Lonely Number": logic and communication. In Chatzidakis, Z., Koepke, P., Pohlers, W., eds.: Logic Colloquium'02. Volume 27 of Lecture Notes in Logic. ASL & A.K. Peters (2006) 96–129
25. Hintikka, J.: Knowledge and Belief. Cornell University Press (1962)
26. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about Knowledge. The MIT Press (1995)
27. Meyer, J., van der Hoek, W.: Epistemic Logic for AI and Computer Science. Number 41 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (1995)
28. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Volume 337 of Synthese Library. Springer (2007)
29. von Wright, G.: An Essay in Modal Logic. North-Holland (1951)
30. Lewis, D.: Convention, A Philosophical Study. Harvard University Press (1969)
31. Halpern, J., Moses, Y.: A guide to completeness and complexity for modal logics of knowledge and belief. Artificial Intelligence **54** (1992) 311–379
32. Nonnengart, A., Weidenbach, C.: Computing small clause normal forms. In: Handbook of Automated Reasoning. North Holland (2001) 335–367
33. Herzig, A., Lang, J., Polacsek, T.: A modal logic for epistemic tests. In Horn, W., ed.: Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI), IOS Pres (2000) 553–557
34. Herzig, A., Lang, J., Longin, D., Polacsek, T.: A logic for planning under partial observability. In: Proceedings of the Seventeenth Conference on Artificial Intelligence (AAAI) and the Twelfth Conference on Innovative Applications of Artificial Intelligence (IAAI), The AAAI Press (2000) 768–773

35. van Ditmarsch, H., Herzig, A., de Lima, T.: Optimal regression for reasoning about knowledge and actions. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, AAAI Press (2007) 1070–1075
36. Demolombe, R., Herzig, A., Varzinczak, I.: Regression in modal logic. Journal of Applied Non-Classical Logics **13** (2003) 165–185
37. Baltag, A., Moss, L.: Logics for epistemic programs. Synthese **139** (2004) 165–224