Department of Computer Science,
University of Otago

Technical Report OUCS-2008-02

An optimal method for reasoning about actions

Authors:

Hans van Ditmarsch
University of Otago, New Zealand and IRIT, France

Andreas Herzig
IRIT, France

Tiago de Lima
Eindhoven University of Technology, the Netherlands

Department of Computer Science,
University of Otago, PO Box 56, Dunedin, Otago, New Zealand

http://www.cs.otago.ac.nz/research/techreports.html

An Optimal Method for Reasoning about Actions

Hans van Ditmarsch

Computer Science, University of Otago, P.O. Box 56, Dunedin 9054, New Zealand
and
Université Paul Sabatier, IRIT-LILaC
118 route de narbonne, F-31062 Toulouse Cedex 9,France

Andreas Herzig

Université Paul Sabatier, IRIT-LILaC
118 route de Narbonne, F-31062 Toulouse Cedex 9, France

Tiago de Lima *

Eindhoven University of Technology, Department of Technology Management
P.O. Box 513, 5600 MB EINDHOVEN, The Netherlands

Abstract

In this paper we propose a novel method for reasoning about actions and knowledge that has optimal computational complexity. In our approach, we use two different extensions of public announcement logic (PAL). The first, PALA, extends PAL with assignments, and the second, that we call PALAT, extends PAL with assignments and tests; both are equally expressive. First, we encode Scherl&Levesque's solution to the frame problem with knowledge into PALAT, and show that this is a polynomial transformation. Then, by extending Lutz' method for PAL satisfiability checking to PALA, we establish an optimal decision procedure for PALA. Our method runs in nondeterministic polynomial time for one agent, in polynomial space for multiple agents and in deterministic exponential time when common knowledge is involved.

Key words: Reasoning about actions, situation calculus, frame problem, successor state axioms, knowledge representation and reasoning, dynamic epistemic logic.

---

* Corresponding author.
Email addresses: `hans@cs.otago.ac.nz` and `hans@irit.fr` (Hans van Ditmarsch), `herzig@irit.fr` (Andreas Herzig), `T.d.Lima@tue.nl` (Tiago de Lima).
URLs: `http://www.cs.otago.ac.nz/staffpriv/hans` (Hans van Ditmarsch), `http://www.irit.fr/~Andreas.Herzig` (Andreas Herzig), `http://home.tm.tue.nl/tlima` (Tiago de Lima).

13 May 2008

1 Introduction

Thielscher (1999) 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 the relation between every action and fluent explicit: 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 framework that incorporates a solution to the representational version, it is the problem of designing an "efficient" decision procedure for reasoning in such a framework.

In this paper we consider the inferential version of the frame problem and investigate how it can be solved based on what is currently the most popular solution to the representational frame problem, viz. successor state axioms as introduced by Reiter (1991). Reiter's solution received a number of extensions such as for concurrent actions (Gelfond et al., 1991), for probabilistic actions (Bacchus et al., 1999), and for knowledge and knowledge-producing actions (Scherl and Levesque, 1993, 2003). We here also take into account the latter epistemic extension. We will start from the reformulation of Scherl&Levesque's solution in (Lakemeyer and Levesque, 2004, 2005).

We focus on decision procedures, and therefore only consider the propositional case. In this case and in Reiter's predicate logic formulation, the so-called propositional fluents only have situations as arguments, and $p(do(a,s))$ reads 'p holds in the situation $do(a,s)$', where the situation $do(a,s)$ results from the performance of action a in situation s. Then *successor state axioms* (SSAs) are of the form

$$\begin{aligned}\forall x \forall s\ ((p(do(x,s)) \leftrightarrow \\ &(x{=}a_1 \wedge \gamma^+(a_1,p,s)) \vee \cdots \vee (x{=}a_n \wedge \gamma^+(a_n,p,s)) \vee \\ &(p(s) \wedge \neg(x{=}a'_1 \wedge \gamma^-(a'_1,p,s)) \wedge \cdots \wedge \neg(x{=}a'_m \wedge \gamma^-(a'_m,p,s)))))\end{aligned}$$

The formula $\gamma^+(a_i,p,s)$ characterizes the condition under which a_i makes p true, and the formula $\gamma^-(a_i,p,s)$ characterizes the condition under which a_i makes p false. These formulas have to be *uniform in s*, which in particular means that the function symbol *do* does not occur in them.

The hypothesis underlying Reiter's solution is that due to inertia it is "rare" that actions change the truth value of fluents. This means that the formula on the right hand side of the equivalence can be expected to be short. It follows that the size of the set of all SSAs can be expected to be of the order of the number of fluents m, and that it is thus much smaller than twice the product of the number of actions and the number of fluents $2 \times n \times m$. Therefore, SSAs count as a solution to the representational frame problem.

Basic action theories essentially contain one successor state axiom for each fluent p. Given such a basic action theory one can reduce (or *regress*) any formula φ to an equivalent formula $\text{reg}(\varphi)$ not mentioning actions. This leads to a straightforward decision procedure for the propositional fragment of the language. However, the reduced formula can be exponentially larger than the original formula, and therefore *regression does not solve the inferential frame problem*.

In this paper we solve the inferential frame problem in the propositional case. For the extension to knowledge, among all epistemic actions, our method is optimal when epistemic actions are restricted to *observations*: all agents observe *that* some proposition holds in the world, and update their epistemic state accordingly. Note that observations are less general than sensing actions studied in (Scherl and Levesque, 2003). By performing the latter, the agents observe *whether* some proposition holds in the world.

Technically, our approach builds on recent progress in the field of dynamic epistemic logics. Precisely, we use two extensions of public announcement logic PAL (Plaza, 1989):

- public announcement logic with assignments PALA (van Ditmarsch et al., 2005; van Benthem et al., 2006; Kooi, 2007), and
- an extension of PALA by test actions that we call PALAT.

All three logics PAL, PALA and PALAT have the same expressivity. In these logics situation terms are left implicit, and one cannot quantify over actions as in the situation calculus. Thus the central device in Reiter's solution is not available. We show that nevertheless one can do without it: our first contribution is a polynomial transformation from Lakemeyer&Levesque's logic ES to PALAT. The logic PALA being an extension of PAL, we extend Lutz' procedure for PAL satisfiability checking (Lutz, 2006) to PALA, and show that we keep optimality. This provides an optimal decision procedure for reasoning about actions and knowledge: both in Reiter's case (without knowledge operators) and in the monoagent case, satisfiability checking can be done in nondeterministic polynomial time; in the multiagent case it can be done in polynomial space; and in the case of common knowledge it can be done in deterministic exponential time. All these results are optimal because they match the computational complexity of the underlying epistemic logic.

The remainder of the paper is organized as follows: Section 2 recalls Lakemeyer&Levesque's logic ES. Section 3 introduces public announcement logic with assignments and tests PALAT. Section 4 presents an encoding of ES basic action theories in PALAT; and Section 5 contains optimal decision procedures for satisfiability checking in PALA. Section 6 contains the results for the multiagent case, and section 7 concludes. [1]

[1] A first version of this paper entitled 'Optimal Regression for Reasoning about Knowledge and Actions' was presented at AAAI'2007 (van Ditmarsch et al., 2007a). The present

2 Lakemeyer&Levesque's logic ES

Reiter formulated SSAs within a dialect of second-order logic called *situation calculus*, which is the mainstream formalism in reasoning about actions (Reiter, 1991). However, because situation calculus is defined axiomatically, properties about action theories that are not direct entailments are very hard to prove. For instance, (Lakemeyer and Levesque, 2004, 2005) mentions the long proof in (Reiter, 2001b) for the fact that if $\mathbf{K}\varphi$ entails $\mathbf{K}\psi_1 \vee \mathbf{K}\psi_2$ in a theory Θ, then $\mathbf{K}\varphi$ entails $\mathbf{K}\psi_1$ in Θ, or $\mathbf{K}\varphi$ entails $\mathbf{K}\psi_2$ in Θ. Aiming at a "more workable" semantics for the situation calculus, they proposed a variant called ES. This logic is not as expressive as the entire situation calculus, but it handles Reiter's basic action theories and, thereby, also his solution to the frame problem. In this section we first give the syntax and semantics of ES, then show how basic action theories are defined, and finally explain how regression works.

2.1 *Syntax of* ES

The full language of ES provides modal operators of knowledge and action together with quantification over both actions and objects, and is thus a many-sorted modal language. In this work though, we focus our attention on decision procedures, and therefore consider only quantification over action symbols.

DEFINITION 1 Let U be a countable set of action variables, let P_0 be a countable set of fluents of arity 0, let A be a countable set of action constants, and let *Poss* ('possible') and *SF* ('sensed fluent') be two predicate symbols of arity 1.

A *Term* is an action constant $a \in A$ or an action variable $x \in U$.

The language $\mathcal{L}_{\mathsf{ES}}$ is the set of formulas φ defined by the following BNF:

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

where t ranges over the set of terms, p ranges over P_0, and x over U.

The predicate *Poss* is used to model executability preconditions of actions. If $Poss(a)$ holds, then the action a is executable. The predicate *SF* is used to model the result of sensing actions. The formula $SF(a)$ stands for the formula whose truth value is known by the agent after the execution of the action a. As in epistemic logics, the

---

version extends its results by establishing a precise formal relationship between a formal translation from ES to PALAT. These ideas were also presented at the 2007 Dagstuhl workshop 'Formal Models of Belief Change in Rational Agents' (van Ditmarsch et al., 2007b) and at the 2007 'Methods for Modalities' workshop.

operator $\mathbf{K}$ is used to model knowledge of the agent. 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 action a'. The formula $\Box\varphi$ is read 'φ holds after the execution of any sequence of actions'. [2]

We use the common abbreviations for the operators $\vee$, $\rightarrow$ and $\leftrightarrow$; and $\bot$ and $\top$ respectively abbreviate $p \wedge \neg p$ and $\neg(p \wedge \neg p)$, for some $p \in P_0$.

As usual a formula is called *ground* if no variable occurs in it. A *Boolean formula* is built from P_0 with the Boolean operators. Thus Boolean formulas neither contain *Poss*, *SF*, $\Box$, $[t]$, $\mathbf{K}$, $=$ nor variables.

DEFINITION 2 *Ground box-free formulas* are ground formulas without the operator $\Box$. The set of ground box-free formulas is noted $\mathcal{L}^0_{\mathsf{ES}}$.

2.2 *Semantics of* ES

Formulas of ES are interpreted at possible worlds after sequences of actions. First we need a definition.

Primitive formulas are ground formulas without any logical operator (neither equality nor Boolean nor modal operators). The set of primitive formulas is noted P_1, i.e.

$$P_1 = P_0 \cup \{Poss(a) : a \in A\} \cup \{SF(a) : a \in A\}$$

Let A^* be the set of all sequences of actions from A, where ε is the empty sequence. Let W_{ES} be the set of all mappings from $P_1 \times A^*$ to $\{0,1\}$. Formulas in $\mathcal{L}_{\mathsf{ES}}$ are evaluated in triples of the form $\langle e, w, \alpha\rangle$ such that: $e \subseteq W_{\mathsf{ES}}$ is the epistemic state of the agent, $w \in e$ is the actual state, and $\alpha \in A^*$ is the sequence of actions that has been performed.

To interpret what is known by the agent after a sequence of actions, we inductively define that two worlds are *indistinguishable* with respect to a sequence of actions α by:

- $w \sim_\varepsilon w'$, for all $w, w' \in W_{\mathsf{ES}}$; and
- $w \sim_{\alpha\cdot a} w'$ iff ($w \sim_\alpha w'$ and $w(SF(a),\alpha) = w'(SF(a),\alpha)$).

That is, w and w' are indistinguishable after action a if they were so before, and if a's sensed fluent has the same value at w and w'.

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

The *satisfaction relation* $\models$ between triples and sentences (formulas without free variables) is defined inductively by:

$$\begin{array}{rcl}
\langle e,w\rangle \models \varphi & \text{iff} & \langle e,w,\varepsilon\rangle \models \varphi \\
\langle e,w,\alpha\rangle \models p & \text{iff} & w(p,\alpha) = 1 \text{ for } p \in P_1 \\
\langle e,w,\alpha\rangle \models a_1{=}a_2 & \text{iff} & a_1 \text{ and } a_2 \text{ are (syntactically) identical} \\
\langle e,w,\alpha\rangle \models \neg\varphi & \text{iff} & \langle e,w,\alpha\rangle \not\models \varphi \\
\langle e,w,\alpha\rangle \models \varphi \wedge \psi & \text{iff} & \langle e,w,\alpha\rangle \models \varphi \text{ and } \langle e,w,\alpha\rangle \models \psi \\
\langle e,w,\alpha\rangle \models \forall x\varphi & \text{iff} & \text{for all } a \in A, \langle e,w,\alpha\rangle \models \varphi[x\backslash a] \\
\langle e,w,\alpha\rangle \models \mathbf{K}\varphi & \text{iff} & \text{for all } w' \in e, \text{ if } w \sim_\alpha w' \text{ then } \langle e,w',\alpha\rangle \models \varphi \\
\langle e,w,\alpha\rangle \models [a]\varphi & \text{iff} & \langle e,w,\alpha\cdot a\rangle \models \varphi \\
\langle e,w,\alpha\rangle \models \Box\varphi & \text{iff} & \text{for all } \alpha' \in A^*, \langle e,w,\alpha\cdot\alpha'\rangle \models \varphi
\end{array}$$

where $\varphi[x\backslash a]$ is the formula resulting from replacing all free occurrences of x in φ by a.

A formula $\varphi \in \mathcal{L}_{\mathsf{ES}}$ is a *valid* ES *consequence* of a set of formulas $\Psi \subseteq \mathcal{L}_{\mathsf{ES}}$, noted $\Psi \models_{\mathsf{ES}} \varphi$, if and only if for all e and w, if $\langle e,w\rangle \models \psi$ for all $\psi \in \Psi$ then $\langle e,w\rangle \models \varphi$. A formula φ is ES *valid*, noted $\models_{\mathsf{ES}} \varphi$, if and only if $\emptyset \models_{\mathsf{ES}} \varphi$.

For example we have $\models_{\mathsf{ES}} [a]\neg\varphi \leftrightarrow \neg[a]\varphi$. (Note that this equivalence is not valid in dynamic logic.) Lakemeyer&Levesque show that positive introspection $\Box(\mathbf{K}\varphi \to \mathbf{K}\mathbf{K}\varphi)$ is ES valid, as well as negative introspection $\Box(\neg\mathbf{K}\varphi \to \mathbf{K}\neg\mathbf{K}\varphi)$. They also show that the following *successor state axiom for knowledge* (SSAK) is valid:

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

It will be useful for our proofs that the rule of replacement of equivalences holds in ES: suppose ψ is a subformula of φ, and suppose $\Psi \models_{\mathsf{ES}} \psi \leftrightarrow \psi'$; then $\Psi \models_{\mathsf{ES}} \varphi \leftrightarrow \varphi'$, where φ' is obtained from φ by replacing subformula ψ by ψ'.

2.3 Basic action theories

Reiter's solution to the frame problem requires that action preconditions and effects be described by what he calls *basic action theories*. Such theories must contain in particular successor state axioms (SSAs) for each fluent $p \in P_0$.

DEFINITION 3 A *basic action theory* is a set of formulas $\Theta = \Theta_{\text{pre}} \cup \Theta_{\text{sense}} \cup \Theta_{\text{post}}$ such that:

- for each $a \in A$, Θ_{pre} contains a formula $\Theta_{\text{pre}}(a)$ of the form $\Box\ (Poss(a) \leftrightarrow \varphi_{Poss}(a))$, where $\varphi_{Poss}(a)$ is a Boolean formula;

- for each $a \in A$, Θ_{sense} contains a formula $\Theta_{\text{sense}}(a)$ of the form $\Box\,(SF(a) \leftrightarrow \varphi_{SF}(a))$, where $\varphi_{SF}(a)$ is a Boolean formula; and
- for each $p \in P_0$, Θ_{post} contains a formula of the form

$$\forall x \Box\,([x]p \leftrightarrow (x{=}a_1 \wedge \gamma^+(a_1,p)) \vee \cdots \vee (x{=}a_n \wedge \gamma^+(a_n,p)) \vee \\ (p \wedge \neg(x{=}a'_1 \wedge \gamma^-(a'_1,p)) \wedge \cdots \wedge \neg(x{=}a'_m \wedge \gamma^-(a'_m,p))))$$

  for each $p \in P_0$, where $\gamma^+(a_i,p)$ and $\gamma^-(a'_i,p)$ are Boolean formulas.

REMARK 4 Lakemeyer&Levesque's definition is slightly more general. First, Θ_{pre} consists of a single formula $\forall x \Box\,(Poss(x) \leftrightarrow \varphi_{Poss}(x))$, where $\varphi_{Poss}(x)$ may contain quantifiers and equalities; similar for Θ_{sense}.

Second, they use Reiter's generalization (Reiter, 2001a) and allow the SSA for a fluent p to take the form $\forall x \Box\,([x]p \leftrightarrow \gamma(x,p))$, where $\gamma(x,p)$ may again contain quantifiers and equalities. Our approach does not work for that generalization, as we will explain in Remark 8 in Section 2.3).

Given a basic action theory Θ and a formula φ, the *entailment problem* in ES is to decide whether $\Theta \models_{\mathsf{ES}} \varphi$.

When we use basic action theories we make some hypotheses. Reiter's non-epistemic solution relies on the following three. (1) All actions are deterministic. (2) Action precondition completeness: for each $a \in A$ there is a Boolean formula $\varphi_{Poss}(a)$ that characterizes the conditions under which a is executable. (3) Causal completeness: first, for each $a \in A$ there is a set $Eff^+(a)$ of fluents which may become true by the execution of a, and there is a set $Eff^-(a)$ of fluents which may become false by the execution of a; second, for each $p \in Eff^+(a)$ there is a Boolean formula $\gamma^+(a,p)$ characterizing the conditions under which p becomes true by the execution of a, and for each $p \in Eff^-(a)$ there is a Boolean formula $\gamma^-(a,p)$ characterizing the conditions under which p becomes false by the execution of a.

Scherl&Levesque's epistemic extension relies on the following supplementary hypotheses: (4) The agent knows the basic action theory Θ under concern. (5) The agent learns about all action occurrences. (6) For each action a, there is a formula $\varphi_{SF}(a)$ that characterizes what is perceived by the agent via the execution of a.

In the sequel we illustrate basic action theories by a running example inspired by a puzzle of Smullyan (1992).

EXAMPLE 5 The environment consists of an agent that dwells in 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 fluent $lady_1$ represents that the lady is behind door 1 and the tiger is behind door 2. Thus the formula $\neg lady_1$ expresses that the lady is behind door 2 and the tiger is behind door 1.

The available actions are $listen_1$ and $listen_2$ (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 $open_2$ (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).

A basic action theory for this example is made up of:

$$
\begin{aligned}
\Theta_{\text{pre}} = \{ & \Box\,(Poss(open_1) \leftrightarrow alive), \\
& \Box\,(Poss(open_2) \leftrightarrow alive), \\
& \Box\,(Poss(listen_1) \leftrightarrow alive), \\
& \Box\,(Poss(listen_2) \leftrightarrow alive))) \} \\
\Theta_{\text{sense}} = \{ & \Box\,(SF(open_1) \leftrightarrow \top), \\
& \Box\,(SF(open_2) \leftrightarrow \top), \\
& \Box\,(SF(listen_1) \leftrightarrow lady_1), \\
& \Box\,(SF(listen_1) \leftrightarrow \neg lady_1))) \} \\
\Theta_{\text{post}} = \{ & \forall x \Box\,([x]alive \leftrightarrow (alive \land \neg(x{=}open_1 \land \neg lady_1)\ \land \neg(x{=}open_2 \land lady_1))), \\
& \forall x \Box\,([x]married \leftrightarrow ((x{=}open_1 \land lady_1) \lor (x{=}open_2 \land \neg lady_1) \lor \\
& \qquad\qquad\qquad\qquad\qquad married)), \\
& \forall x \Box\,([x]lady_1 \leftrightarrow lady_1) \}
\end{aligned}
$$

We get e.g. the entailments $\Theta \models_{\mathsf{ES}} [listen_1](\mathbf{K} lady_1 \lor \mathbf{K} \neg lady_1)$, and

$$\Theta \models_{\mathsf{ES}} (lady_1 \land alive) \rightarrow [listen_1][open_1]\mathbf{K} married$$

We end this section by an interesting property of basic action theories.

Definition 6 Let Θ_{post} be the set of SSAs of a basic action theory Θ, and let

$$
\begin{aligned}
\forall x \Box\,([x]p \leftrightarrow\ & (x{=}a_1 \land \gamma^+(a_1,p)) \lor \cdots \lor (x{=}a_n \land \gamma^+(a_n,p)) \lor \\
& (p \land \neg(x{=}a'_1 \land \gamma^-(a'_1,p)) \land \cdots \land \neg(x{=}a'_m \land \gamma^-(a'_m,p))))
\end{aligned}
$$

be its SSA for p. An action a is *positively relevant* for p if and only if $a = a_i$ for some $1 \leq i \leq n$, and a is *negatively relevant* for p if and only if $a = a'_i$ for some $1 \leq i \leq m$.

The set of fluents p such that a is positively relevant for p is noted $Eff^+_\Theta(a)$, and the set of fluents p such that a is negatively relevant for p is noted $Eff^-_\Theta(a)$. Finally, $Eff_\Theta(a) = Eff^+_\Theta(a) \cup Eff^-_\Theta(a)$.

The set $\mathit{Eff}_{\Theta}(a)$ is the set of fluents $p \in P_0$ such that a occurs in the successor state axiom for p.

PROPOSITION 7 For every basic action theory Θ, if $p \not\in \mathit{Eff}_{\Theta}(a)$ then $\Theta \models_{\mathsf{ES}} p \leftrightarrow [a]p$.

REMARK 8 As already said in Remark 4, Lakemeyer&Levesque allow Θ_{post} to contain SSAs of the more general form $\forall x \Box\, ([x]p \leftrightarrow \gamma(x, p))$. An extreme example is $\forall x \Box\, ([x]p \leftrightarrow \top)$, stating that p is true after every action. The above Proposition 7 does not hold for such generalized action theories. It is for that reason that we will not be able to translate them into PALAT, and thus our optimal method does not apply to them.

Note that using $\mathit{Eff}^{+}_{\Theta}$ and $\mathit{Eff}^{-}_{\Theta}$, SSAs can be written as follows:

$$\forall x \Box\, ([x]p \;\leftrightarrow\; (\bigvee_{a \in \mathit{Eff}^{+}_{\Theta}(a)} (x{=}a \land \gamma^{+}(a,p))) \lor (p \land \neg \bigwedge_{a' \in \mathit{Eff}^{-}_{\Theta}(a)} (x{=}a' \land \gamma^{-}(a',p))))$$

2.4 Regression

Given a basic action theory Θ and a ground box-free formula φ of $\mathcal{L}^{0}_{\mathsf{ES}}$, there is an effective procedure that decides whether $\Theta \models_{\mathsf{ES}} \varphi$. It amounts to a simplification of φ by the equivalences of Θ: by iterating the application of these equivalences one obtains an equivalent formula without *Poss*, *SF*, $\Box$ or $[t]$. This procedure is called regression, and it allows to reduce the entailment problem to the validity problem in epistemic logic.

The two central equivalences are the SSAs for fluents of Θ_{post} and the successor state axiom for knowledge SSAK. These and the other equivalences of Θ can be turned into rewriting rules that allow to transform ground box-free formulas into epistemic formulas.

REMARK 9 Lakemeyer&Levesque do not require the box-free formula φ to be ground, and allow for sentences such as $\forall x[x]p$ and $\forall x \exists x' (x' \neq x \land (p \to [x][x']p))$. We will not be able to handle such formulas, the reason being that the target logic of our translation does not have quantifiers.

Instead of stating a formal definition we illustrate regression by our running example.

EXAMPLE 10 Consider the basic action theory Θ of Example 5. We regress the formula $[listen_1][open_1]\mathbf{K}married$ by applying the equivalences in Θ:

$$\begin{aligned}
[listen_1][open_1]\mathbf{K}married &\leftrightarrow [listen_1]\mathbf{K}[open_1]married \\
&\leftrightarrow [listen_1]\mathbf{K}(lady_1 \vee married) \\
&\leftrightarrow (lady_1 \wedge \mathbf{K}(lady_1 \rightarrow [listen_1](lady_1 \vee married))) \vee \\
&\quad\; (\neg lady_1 \wedge \mathbf{K}(\neg lady_1 \rightarrow [listen_1](lady_1 \vee married))) \\
&\leftrightarrow (lady_1 \wedge \mathbf{K}(lady_1 \rightarrow (lady_1 \vee married))) \vee \\
&\quad\; (\neg lady_1 \wedge \mathbf{K}(\neg lady_1 \rightarrow (lady_1 \vee married))) \\
&\leftrightarrow lady_1 \vee \mathbf{K}(lady_1 \vee married)
\end{aligned}$$

The first and the third equivalences are valid by SSAK (the first is due to $\Theta \models_{\mathsf{ES}} SF(open_1) \leftrightarrow \top$, and the third is due to $\Theta \models_{\mathsf{ES}} SF(listen_1) \leftrightarrow lady_1$). The second and fourth equivalences are logical consequences of Θ_{post}: for the second, $\Theta \models_{\mathsf{ES}} [open_1]married \leftrightarrow lady_1 \vee married$; and for the fourth, first $\models_{\mathsf{ES}} [listen_1](lady_1 \vee married) \leftrightarrow ([listen_1]lady_1 \vee [listen_1]married)$, and then $\Theta \models_{\mathsf{ES}} ([listen_1]lady_1 \vee [listen_1]married) \leftrightarrow (lady_1 \vee married)$.

Let $\mathrm{reg}_\Theta(\varphi)$ denote the result of rewriting φ by means of the equivalences in Θ as formally defined in (Lakemeyer and Levesque, 2004). Their theorems 1 and 5 can be restated here as follows.

THEOREM 11 (Lakemeyer and Levesque, 2004) Let Θ be a basic action theory, and let $\varphi \in \mathcal{L}^0_{\mathsf{ES}}$ be a ground box-free formula. Then $\Theta \models_{\mathsf{ES}} \varphi$ if and only if $\models_{\mathsf{EL}} \mathrm{reg}_\Theta(\varphi)$. (Where EL stands for epistemic logic to be defined in Section 3.1.)

Regression has high computational complexity: $\mathrm{reg}_\Theta(\varphi)$ can be exponentially larger than φ. To see this, consider the application of SSAK in Example 10. Note that each time SSAK is applied to φ, the resulting formula may be twice as large as φ. In the rest of the paper we show that one can do better by applying recent techniques that were introduced in the field of dynamic epistemic logic.

3 Public announcement logic with assignments and tests PALAT

A different tradition in modelling dynamics of knowledge focusses on particular epistemic actions that make the agents expand their knowledge without changing the world itself, see for example work by Plaza (1989), Baltag et al. (1998), Gerbrandy (1999) and van Benthem (2006). It is only recently that ontic actions (actions changing the facts of the world) were introduced into these dynamic epistemic logics (van Ditmarsch et al., 2005; Kooi, 2007). All of them are extensions of Plaza's public announcement logic (PAL) (Plaza, 1989). In this section we recall this extension, that we baptize *public announcement logic with assignments* PALA, and that we augment by test actions. We call the result PALAT.

All these dynamic epistemic logics are based on standard epistemic logic, that we recall first.

3.1 Background: epistemic logic EL

Epistemic logics are a family of modal logics that use possible worlds semantics to represent agents' knowledge. This idea, originally due to Hintikka (1962), has known great development in more recent works such as (Fagin et al., 1995), (Meyer and van der Hoek, 1995) and (van Ditmarsch et al., 2007c). We here recall the monoagent case, postponing the multiagent case to Section 6.

DEFINITION 12 The language $\mathcal{L}_{\mathsf{EL}}$ of monoagent epistemic logic is the set of formulas φ defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{K}\varphi$$

where p ranges over the countable set of propositional letters P_0.

DEFINITION 13 An *epistemic model* (EL model) is a tuple $\langle W, R, V \rangle$ such that:

- W is a nonempty set of possible worlds;
- $R \subseteq (W \times W)$ is an equivalence relation;
- $V : P_0 \rightarrow \wp(W)$ associates an interpretation $V(p) \subseteq W$ to each $p \in P_0$.

For every $w \in W$, the pair (M, w) is a *pointed epistemic model*.

For convenience, we define $R(w) = \{w' : (w, w') \in R\}$. The elements of $R(w)$ are the worlds the agent considers possible at w.

DEFINITION 14 Let $(M, w) = (\langle W, R, V \rangle, w)$ be a pointed epistemic model. The *satisfaction relation* $\models$ between EL formulas and pointed epistemic models is inductively defined as follows:

$$\begin{array}{lll} M, w \models p & \text{iff} & w \in V(p) \\ M, w \models \neg\varphi & \text{iff} & M, w \not\models \varphi \\ M, w \models \varphi \wedge \psi & \text{iff} & M, w \models \varphi \text{ and } M, w \models \psi \\ M, w \models \mathbf{K}\varphi & \text{iff} & R(w) \subseteq [\![\varphi]\!]_M \end{array}$$

where $[\![\varphi]\!]_M \stackrel{\text{def}}{=} \{w : M, w \models \varphi\}$ is the extension of φ in M.

A formula $\varphi \in \mathcal{L}_{\mathsf{EL}}$ is EL *valid*, noted $\models_{\mathsf{EL}} \varphi$, if and only if for all pointed EL models (M,w), $(M,w) \models \varphi$; and it is EL *satisfiable* if and only if $\not\models_{\mathsf{EL}} \neg\varphi$.

Lakemeyer&Levesque's logic ES is a conservative extension of EL:

PROPOSITION 15 Let φ be a formula of $\mathcal{L}_{\mathsf{EL}}$. Then $\models_{\mathsf{ES}} \varphi$ if and only if $\models_{\mathsf{EL}} \varphi$.

Satisfiability checking in EL is NP-complete (Halpern and Moses, 1992).

3.2 Syntax of PALAT

DEFINITION 16 The *language of public announcement logic with assignment and test* $\mathcal{L}_{\mathsf{PALAT}}$ is the set of formulas φ defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{K}\varphi \mid [!\varphi]\varphi \mid [!!\varphi]\varphi \mid [\sigma]\varphi$$
$$\sigma ::= \varepsilon \mid p{:=}\varphi, \sigma$$

where p ranges over the countable set of propositional letters P_0 and ε is the *empty assignment*.

Let α be one of $!\varphi$, $!!\varphi$ or σ; the formula $[\alpha]\varphi$ reads 'φ holds after all possible executions of α'. The action $!\varphi$ is the public announcement of φ; the action $!!\varphi$ is the public test whether or not φ; and the action $p{:=}\varphi$ is the public assignment of φ to the atom p. For example, $p{:=}\bot$ is an assignment making p false, and $\mathbf{K}[p{:=}\bot]\neg p$ is a formula expressing that the agent knows this. A complex assignment $(p_1{:=}\varphi_1, \dots, p_n{:=}\varphi_n)$ is supposed to take place in parallel. We sometimes write this as the set $\{p_1{:=}\varphi_1, \dots, p_n{:=}\varphi_n\}$. Thus ε is identified with $\emptyset$. It is supposed that in parallel assignments, the same propositional letter can appear only once on the left hand side of the operator '$:=$'.

We shall show that announcements and tests are able to model epistemic actions, and that assignments are able to model ontic actions. For example, the epistemic action $listen_1$ of Example 5 can be modelled as $!!lady_1$, and the ontic action $open_1$ can be modelled as the complex assignment

$$\{alive{:=}(lady_1 \wedge alive), married{:=}(lady_1 \vee married)\}.$$

The *language of public announcement logic with assignment* $\mathcal{L}_{\mathsf{PALA}}$ is $\mathcal{L}_{\mathsf{PALAT}}$ without tests. The *language of public announcement logic* $\mathcal{L}_{\mathsf{PAL}}$ is $\mathcal{L}_{\mathsf{PALA}}$ without assignments.

Just as formulas of epistemic logic, formulas of $\mathcal{L}_{\mathsf{PALAT}}$ are interpreted in pointed epistemic models.

DEFINITION 17 Let $(M,w) = (\langle W,R,V\rangle,w)$ be a pointed epistemic model. The *satisfaction relation* $\models$ between PALAT formulas and pointed epistemic models is that of Definition 14 extended with the following three clauses:

$$\begin{array}{lll} M,w \models [!\varphi]\psi & \text{iff} & M,w \models \varphi \text{ implies } M^{!\varphi},w \models \psi \\ M,w \models [!!\varphi]\psi & \text{iff} & M,w \models [!\varphi]\psi \text{ and } M,w \models [!\neg\varphi]\psi \\ M,w \models [\sigma]\varphi & \text{iff} & M^{\sigma},w \models \varphi \end{array}$$

The models $M^{!\varphi}$ and M^{σ} are updates of the epistemic model M, respectively defined as:

$$\begin{array}{rlrl} M^{!\varphi} &= \langle W^{!\varphi},R^{!\varphi},V^{!\varphi}\rangle & M^{\sigma} &= \langle W,R,V^{\sigma}\rangle \\ W^{!\varphi} &= W \cap [\![\varphi]\!]_M & V^{\sigma}(p) &= [\![\sigma(p)]\!]_M \\ R^{!\varphi} &= R \cap ([\![\varphi]\!]_M \times [\![\varphi]\!]_M) & & \\ V^{!\varphi}(p) &= V(p) \cap [\![\varphi]\!]_M & & \end{array}$$

where $\sigma(p)$ is the formula assigned to p by σ. If there is no such formula, i.e., if there is no $(p{:=}\varphi) \in \sigma$, then $\sigma(p) = p$. For example $\emptyset(p) = \varepsilon(p) = p$ for all p, and $\{p{:=}\neg p\}(p) = \neg p$.

To illustrate this let (M,w) be any pointed epistemic model. We have $M,w \models [p{:=}\bot]\neg p$ because $V^{p:=\bot}(p) = [\![(p{:=}\bot)(p)]\!]_M = [\![\bot]\!]_M = \emptyset$; and we have $M,w \models [!p]\mathbf{K}p$ because $V^{!p}(p) = W^{!p}$.

A formula $\varphi \in \mathcal{L}_{\mathsf{PALAT}}$ is PALAT *valid*, noted $\models_{\mathsf{PALAT}} \varphi$, if and only if for all pointed epistemic models (M,w), $(M,w) \models \varphi$; and it is PALAT *satisfiable* if and only if $\not\models_{\mathsf{PALAT}} \neg\varphi$.

For example, $[p{:=}\bot]\neg p$, $[!p]\mathbf{K}p$ and $p \to [!!p]\mathbf{K}p$ are all PALAT valid (for atomic p). Note that neither $[!\varphi]\varphi$ nor the stronger $[!\varphi]\mathbf{K}\varphi$ are PALAT valid.

PROPOSITION 18 The following equivalences are PALAT valid.

- Announcements and tests are interdefinable:

$$\models_{\mathsf{PALAT}} [!!\varphi]\psi \leftrightarrow [!\varphi]\psi \wedge [!\neg\varphi]\psi \quad \text{and} \quad \models_{\mathsf{PALAT}} [!\varphi]\psi \leftrightarrow (\varphi \to [!!\varphi]\psi)$$

- Assignments and tests are deterministic and executable:

$$\models_{\mathsf{PALAT}} [!!\varphi]\neg\psi \leftrightarrow \neg[!!\varphi]\psi \quad \text{and} \quad \models_{\mathsf{PALAT}} [p{:=}\varphi]\neg\psi \leftrightarrow \neg[p{:=}\varphi]\psi$$

- Tests do not modify Boolean formulas:

$$\models_{\mathsf{PALAT}} [!!\varphi]\psi \leftrightarrow \psi \text{ if } \psi \text{ is Boolean}$$

Therefore our test operator can be defined in terms of the announcement operator, and thus does not increase the expressivity of PALAT. Nevertheless, its definition as a primitive operator allows us to provide a polynomial translation of ES sensing actions into PALAT. If tests were defined as abbreviations, then the translation of an ES formula with sensing actions of Section 4 would be exponentially larger than the original formula in the worst case.

Just as Lakemeyer&Levesque's ES, our PALAT is a conservative extension of EL:

PROPOSITION 19 Let φ be an EL formula. Then $\models_{\mathsf{PALAT}} \varphi$ if and only if $\models_{\mathsf{EL}} \varphi$.

REMARK 20 The semantics of both the announcement and the test operator is different from that of the dynamic logic test operator '?'. First, in dynamic logic $[\varphi?]\psi$ is equivalent to $\varphi \to \psi$, while $[!\varphi]\psi$ is not equivalent to $\varphi \to \psi$ in PALAT. Second, $[!!\varphi]\bot$ is unsatisfiable in PALAT, while the dynamic logic formula $[\varphi?]\bot$ is not. Also note the difference between our reading of $!!\varphi$ as 'test *whether or not* φ', and the dynamic logic reading of $\varphi?$ as 'test *that* φ'.

3.4 Reduction to epistemic logic

PALAT is axiomatized by the axioms and inference rules of the logic S5 plus the following axioms (van Benthem et al., 2006; Kooi, 2007) (the axiom for !! is our addition):

$$
\begin{aligned}
[\sigma]p &\leftrightarrow \sigma(p)\\
[\sigma]\neg\varphi &\leftrightarrow \neg[\sigma]\varphi\\
[\sigma](\varphi_1 \wedge \varphi_2) &\leftrightarrow ([\sigma]\varphi_1 \wedge [\sigma]\varphi_2)\\
[\sigma]\mathbf{K}\varphi &\leftrightarrow \mathbf{K}[\sigma]\varphi\\
[!\psi]p &\leftrightarrow (\psi \to p)\\
[!\psi]\neg\varphi &\leftrightarrow (\psi \to \neg[!\psi]\varphi)\\
[!\psi](\varphi_1 \wedge \varphi_2) &\leftrightarrow ([!\psi]\varphi_1 \wedge [!\psi]\varphi_2)\\
[!\psi]\mathbf{K}\varphi &\leftrightarrow (\psi \to \mathbf{K}[!\psi]\varphi)\\
[!!\psi]\varphi &\leftrightarrow ([!\psi]\varphi \wedge [!\neg\psi]\varphi)
\end{aligned}
$$

These axioms provide equivalences for all possible combinations of the logical connectives with dynamic modal operators. The right hand side of these equivalences is simpler than the left hand side, and for that reason they are called reduction axioms (see e.g. Kooi (2007) for a precise definition of what it means to be 'simpler'). Such axioms provide at the same time a proof method: they allow to rewrite every PALAT-formula into an equivalent EL-formula, which can then be checked by independent means.

Let $\text{red}(\varphi)$ be the formula that is obtained in this way.

THEOREM 21 (van Benthem et al., 2006; Kooi, 2007) Let φ be a $\mathcal{L}_{\mathsf{PALAT}}$-formula. Then $\models_{\mathsf{PALAT}} \varphi$ if and only if $\models_{\mathsf{EL}} \text{red}(\varphi)$.

However, this method has the same problem as ES regression of Section 2.4: in the worst case $\text{red}(\varphi)$ is exponentially larger than φ. This cannot be avoided: at least in the multiagent case it can be shown that PAL is more succinct than EL (Lutz, 2006). This means that there are PAL formulas such that every equivalent EL formula is exponentially longer, see Example 33 in Section 6.

In Section 5 we provide a better method that performs a reduction from PALA to EL in polynomial time. But first we establish the link between ES and PALAT.

4 Translation from ES to PALAT

Reiter's regression of Section 2 is similar in spirit to PALAT reduction of Section 3. We will show in this section that the problem of entailment in ES can be translated to a validity problem in PALAT: the changes brought about by an action a described by a basic action theory Θ can be modelled as a PALAT test of $\varphi_{SF}(a)$ followed by a set of assignments simulating Reiter's SSAs.

4.1 Finite change constraint

In order to make our proof method work we moreover have to require basic action theories Θ to satisfy a constraint of 'finite potential change': for every action a, the set of fluent constants whose truth value may be flipped by the execution of a is *finite*.

Remember that $\mathit{Eff}_{\Theta}(a)$ is the set of fluents for which a is relevant (Definition 6 of Section 2.3).

DEFINITION 22 Let Θ be a basic action theory. Θ satisfies the *finite change constraint* if and only $\mathit{Eff}_{\Theta}(a)$ is finite for every action $a \in A$.

EXAMPLE 23 Here is a basic action theory that does not satisfy the finite change constraint. Consider the set of fluents $P = \{at_i : i \in \mathbb{Z}\}$ where $\mathbb{Z}$ is the set of integers, and the singleton set of actions $A = \{inc\}$. at_i means that the value of the counter is i, and inc increments the value of the counter. Then $\gamma^+(inc, at_i) = at_{i-1}$, and $\gamma^-(inc, at_i) = at_i$. Hence the set of fluents that are possibly changed by inc is the entire set of fluents: $\mathit{Eff}_\Theta(inc) = P_0$. Therefore Θ does not satisfy the finite change constraint.

We nevertheless believe that finite change action theories are sufficiently expressive to be of interest. For such theories Θ we are going to define a translation tra_Θ such that for every ground box-free sentence φ of $\mathcal{L}^0_{\mathsf{ES}}$, Θ entails φ in ES if and only if $\mathrm{tra}_\Theta(\varphi)$ is PALAT valid.

4.2 Polynomial transformation

DEFINITION 24 Let Θ be a basic action theory satisfying the finite change constraint (Definition 22). We inductively define a mapping tra_Θ from the set of ground box-free formulas $\mathcal{L}^0_{\mathsf{ES}}$ to $\mathcal{L}_{\mathsf{PALAT}}$:

(1) $\mathrm{tra}_\Theta(p) = p$ for $p \in P_0$

(2) $\mathrm{tra}_\Theta(a_1{=}a_2) = \begin{cases} \top \text{ if } a_1 \text{ and } a_2 \text{ are (syntactically) identical} \\ \bot \text{ otherwise} \end{cases}$

(3) $\mathrm{tra}_\Theta(Poss(a)) = \varphi_{Poss}(a)$

(4) $\mathrm{tra}_\Theta(SF(a)) = \varphi_{SF}(a)$

(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) $\mathrm{tra}_\Theta(\mathbf{K}\varphi) = \mathbf{K}\,\mathrm{tra}_\Theta(\varphi)$

(8) $\mathrm{tra}_\Theta([a]\varphi) = [!!\varphi_{SF}(a)][\sigma_a]\,\mathrm{tra}_\Theta(\varphi)$,
where σ_a is the complex assignment defined by:

$$\begin{aligned}
&\{p{:=}(\gamma^+(a,p) \vee p) : p \in \mathit{Eff}^+_\Theta(a) \text{ and } p \notin \mathit{Eff}^-_\Theta(a)\} \quad \cup \\
&\{p{:=}(p \wedge \neg\gamma^-(a,p)) : p \notin \mathit{Eff}^+_\Theta(a) \text{ and } p \in \mathit{Eff}^-_\Theta(a)\} \quad \cup \\
&\{p{:=}(\gamma^+(a,p) \vee (p \wedge \neg\gamma^-(a,p))) : p \in \mathit{Eff}^+_\Theta(a) \cap \mathit{Eff}^-_\Theta(a)\}
\end{aligned}$$

The formulas $\varphi_{Poss}(a)$, $\varphi_{SF}(a)$, $\gamma^+(a,p)$ and $\gamma^-(a,p)$ are those from Θ (Definition 3).

Note that in the last item the set σ_a is finite because Θ satisfies the finite change constraint. Therefore the mapping tra_Θ is well-defined. For the theory of Example 23 which does not have that property, the set σ_{inc} would be infinite and thus $\mathrm{tra}_\Theta([inc]at_i)$ would not be a well-formed formula.

EXAMPLE 25 Consider again our running example. For the epistemic action $listen_1$

we get:

$$\begin{aligned}\mathrm{tra}_\Theta([\mathit{listen}_1]\mathbf{K}\mathit{lady}_1) &= [!!\mathit{lady}_1][\varepsilon]\,\mathrm{tra}_\Theta(\mathbf{K}\mathit{lady}_1)\\ &= [!!\mathit{lady}_1][\varepsilon]\mathbf{K}\mathit{lady}_1\end{aligned}$$

that is equivalent to $[!!\mathit{lady}_1]\mathbf{K}\mathit{lady}_1$. For the ontic action open_1 we get:

$$\mathrm{tra}_\Theta([\mathit{open}_1]\mathit{alive}) = [!!\top][\mathit{alive}:=\mathit{lady}_1 \wedge \mathit{alive}, \mathit{married}:=\mathit{lady}_1 \vee \mathit{married}]\mathit{alive}$$

Since $[!!\top]\varphi$ is equivalent to φ by Proposition 18, and since

$$[\mathit{alive}:=\mathit{lady}_1 \wedge \mathit{alive}, \mathit{married}:=\mathit{lady}_1 \vee \mathit{married}]\mathit{alive}$$

is equivalent to $\mathit{lady}_1 \wedge \mathit{alive}$, the formula $\mathrm{tra}_\Theta([\mathit{open}_1]\mathit{alive})$ is equivalent to $\mathit{lady}_1 \wedge \mathit{alive}$.

THEOREM 26 Let Θ be a basic action theory satisfying the finite change constraint, and let φ be a ground box-free formula of $\mathcal{L}^0_{\mathsf{ES}}$. Then

$$\Theta \models_{\mathsf{ES}} \varphi \text{ if and only if } \models_{\mathsf{PALAT}} \mathrm{tra}_\Theta(\varphi).$$

PROOF. We take advantage of both regression and reduction: by Theorem 11, $\Theta \models_{\mathsf{ES}} \varphi$ iff $\models_{\mathsf{EL}} \mathrm{reg}_\Theta(\varphi)$; by Theorem 21, $\models_{\mathsf{PALAT}} \mathrm{tra}_\Theta(\varphi)$ iff $\models_{\mathsf{EL}} \mathrm{red}(\mathrm{tra}_\Theta(\varphi))$. It therefore suffices to prove that $\models_{\mathsf{EL}} \mathrm{reg}_\Theta(\varphi)$ iff $\models_{\mathsf{EL}} \mathrm{red}(\mathrm{tra}_\Theta(\varphi))$. The details are in Appendix A. ∎

In order to prove that this transformation 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 set X is $\mathrm{len}(X) = 1 + \sum_{x\in X}(1+\mathrm{len}(x))$, while for a tuple $Y = \langle y_1, \ldots, y_n\rangle$, it is $\mathrm{len}(Y) = 1 + \sum_{k=1}^{n}(1+\mathrm{len}(y_k))$. Note that $\mathrm{len}(X) \geq 1$ for every set X; in particular $\mathrm{len}(\emptyset) = 1$.

For formulas in $\mathcal{L}_{\mathsf{EL}}$, we inductively define:

$$\begin{aligned}\mathrm{len}(p) &= 1\\ \mathrm{len}(\neg\varphi) &= 1 + \mathrm{len}(\varphi)\\ \mathrm{len}(\varphi\wedge\psi) &= 1 + \mathrm{len}(\varphi) + \mathrm{len}(\psi)\\ \mathrm{len}(\mathbf{K}\varphi) &= 1 + \mathrm{len}(\varphi)\end{aligned}$$

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

$$\begin{aligned}\mathrm{len}(a_1{=}a_2) &= 3\\ \mathrm{len}([a]\varphi) &= 2 + \mathrm{len}(\varphi)\end{aligned}$$

and for formulas of $\mathcal{L}_{\mathsf{PALAT}}$ we also use:

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

where we consider σ as a set of assignments.

For example, $\text{len}(\bot) = 4$, $\text{len}(\top) = 5$, and

$$\begin{aligned} \text{len}([\{p := q, q := p \wedge q\}]\mathbf{K}p &= 1 + \text{len}(\{p := q, q := p \wedge q\}) + \text{len}(\mathbf{K}p) \\ &= 1 + (1 + (1 + 3) + (1 + 5)) + 2 \\ &= 14 \end{aligned}$$

THEOREM 27 Let Θ be a basic action theory satisfying the finite change constraint, and let $\varphi \in \mathcal{L}^0_{\mathsf{ES}}$. Then $\text{len}(\text{tra}_\Theta(\varphi)) \leq O(\text{len}(\Theta) \times \text{len}(\varphi))$.

PROOF. Please, see Appendix B. ∎

Hence for finite change basic action theories the problem of deciding whether $\Theta \models_{\mathsf{ES}} \varphi$ can be polynomially reduced to a validity problem in PALAT. This is our first main result.

It remains to define a proof method for PALAT. In the next section we give a method that is optimal for the fragment PALA of PALAT.

5 Optimal reduction for PALA

Our optimal reduction is based on a recent method that allows to eliminate announcements from PAL formulas by means of subformula renaming (Lutz, 2006). The transformation is polynomial, and provides an optimal decision procedure for PAL.

First note that the reduction axioms for assignment operators also cause a combinatorial explosion. Indeed, consider the family of formulas inductively defined by: $\psi_1 = p_1$, and $\psi_n = [p_{n-1} := p_n \wedge p_n] \ldots [p_1 := p_2 \wedge p_2] p_1$. By the reduction axioms we get:

$$\begin{aligned}
&[p_{n-1} := p_n \wedge p_n] \dots [p_2 := p_3 \wedge p_3][p_1 := p_2 \wedge p_2] p_1 \\
&\leftrightarrow [p_{n-1} := p_n \wedge p_n] \dots [p_2 := p_3 \wedge p_3](p_2 \wedge p_2) \\
&\leftrightarrow [p_{n-1} := p_n \wedge p_n] \dots ((p_3 \wedge p_3) \wedge (p_3 \wedge p_3)) \\
&\leftrightarrow (\dots (p_n \wedge p_n) \wedge \dots) \dots)
\end{aligned}$$

The last formula is $\mathrm{red}(\psi_n)$. According to the definition of the length of a formula we have that $\mathrm{len}(\psi_n) = 6n - 5$ and $\mathrm{len}(\mathrm{red}(\psi_n)) = 2^n - 1$. Therefore, the length of $\mathrm{red}(\psi_n)$ is exponential in the length of ψ_n.

By using a subformula renaming technique similar to Lutz's we define a polynomial reduction that allows to eliminate assignments from PALA formulas. The combination of these two polynomial transformations is again polynomial.

In this section we only consider formulas of $\mathcal{L}_{\mathsf{PALA}}$, i.e. $\mathcal{L}_{\mathsf{PALAT}}$ without the test operator '!!'. The reason for this restriction is that we did not succeed in finding a polynomial reduction when formulas contain that operator: in the worst case the reduction method proposed here is not polynomial any more. However, note that by Proposition 18, tests can be decomposed into two announcements: the formula $[!!\varphi]\psi$ is equivalent to $[!\varphi]\psi \wedge [!\neg\varphi]\psi$. Therefore this syntactic restriction does not restrict the expressivity of the logic.

Our method takes a formula in $\mathcal{L}_{\mathsf{PALA}}$ as input and returns a satisfiability-equivalent formula in $\mathcal{L}_{\mathsf{EL}}$. The first step eliminates assignments and returns a formula in $\mathcal{L}_{\mathsf{PAL}}$; the second step eliminates announcements and returns a satisfiability-equivalent formula in $\mathcal{L}_{\mathsf{EL}}$.

5.1 Eliminating assignments

In order to eliminate assignments from PALAT formulas we apply a technique that is fairly standard in automated theorem proving, see e.g. (Nonnengart and Weidenbach, 2001), and that is based on the theorem below.

THEOREM 28 Let $\varphi \in \mathcal{L}_{\mathsf{PALA}}$, and let $[p_1 := \varphi_1, \dots, p_n := \varphi_n]\varphi_{n+1}$ be a subformula of φ. Let ψ_{n+1} be obtained from φ_{n+1} by substituting every occurrence of p_k by x_{p_k}, where x_{p_k} is a new propositional letter not occurring in φ. Let ψ be obtained from φ by replacing $[p_1 := \varphi_1, \dots, p_n := \varphi_n]\varphi_{n+1}$ by ψ_{n+1}. Then, φ is PALA satisfiable if and only if:

$$\mathbf{K}\left(\bigwedge_{1 \leq k \leq n} x_{p_k} \leftrightarrow \varphi_k\right) \wedge \psi$$

is PALA satisfiable.

PROOF. To simplify the exposition let us suppose singleton assignments, i.e., the subformula of φ is $[p{:=}\varphi_1]\varphi_2$.

From the left to the right suppose that $M = \langle W, R, V\rangle$ is an epistemic model such that for some $w \in W$, $(M,w) \models \varphi$. Now consider the epistemic model $M_{x_p} = \langle W, R, V_{x_p}\rangle$ such that $V_{x_p}(p) = V(p)$ for all $p \neq x_p$, and $V_{x_p}(x_p) = [\![\varphi_1]\!]_M$. First, note that $(M_{x_p}, w) \models \varphi$ (because x_p does not appear in φ). Second, note that $M_{x_p} \models x_p \leftrightarrow \varphi_1$ (because $[\![x_p]\!]_{M_{x_p}} = [\![\varphi_1]\!]_{M_{x_p}}$). Therefore $(M_{x_p}, w) \models \mathbf{K}(x_p \leftrightarrow \varphi_1)$. Third, note that for every $v \in W$, $(M_{x_p}, v) \models [p{:=}\varphi_1]\varphi_2$ iff $(M_{x_p}^{p:=\varphi_1}, v) \models \varphi_2$ iff $(M_{x_p}^{p:=\varphi_1}, v) \models \psi_2$ (because $V_{x_p}^{p:=\varphi_1}(p) = V_{x_p}^{p:=\varphi_1}(x_p)$). Therefore $M_{x_p} \models [p{:=}\varphi_1]\varphi_2 \leftrightarrow \psi_2$, and therefore $(M_{x_p}, w) \models \mathbf{K}(x_p \leftrightarrow \varphi_1) \wedge \psi$.

From the right to the left suppose w.l.o.g. that the epistemic model $M = \langle W, R, V\rangle$ is point-generated from the world $w \in W$. Now suppose that $(M,w) \models \mathbf{K}(x_p \leftrightarrow \varphi_1) \wedge \psi$. Then $M \models x_p \leftrightarrow \varphi_1$, i.e., $V(x_p) = [\![\varphi_1]\!]_M$. Hence for all $v \in W$, $(M, v) \models \psi_2$ iff $(M^{p:=\varphi_1}, v) \models \varphi_2$ (because $V(x_p) = [\![\varphi_1]\!]_M = V^{p:=\varphi_1}(p)$). In other words, $M \models \psi_2 \leftrightarrow [p{:=}\varphi_1]\varphi_2$. Therefore $M, w \models \varphi$. ∎

Intuitively, the formula $\mathbf{K}\big(\bigwedge_{1\le k\le n}(x_{p_k} \leftrightarrow \varphi_k)\big)$ sets the value of each new propositional letter x_{p_k} to that of φ_k in all accessible worlds.

We are ready to define the first step of our reduction method. In order to improve readability the definition below only considers singleton assignments. The extension to complex assignments is straightforward. The below mapping $\mathrm{reg}_{\mathsf{PALA}}$ collects the announcements occurring in a given formula φ and relates them to new propositional letters.

DEFINITION 29 Let $\varphi \in \mathcal{L}_{\mathsf{PALA}}$, and let α, α_1 and α_2 denote lists of formulas. Let ε be the empty list, and let '$\cdot$' be concatenation. The assignment elimination operator $\mathrm{reg}_{\mathsf{PALA}}$ is defined inductively as follows:

(1) $\mathrm{reg}_{\mathsf{PALA}}(\varphi) = \big(\bigwedge_{\chi\in\alpha}\chi\big) \wedge \psi$,
where $\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \varphi) = (\alpha, \psi)$
(2) $\mathrm{reg}_{\mathsf{PALA}}(\alpha, p) = (\alpha, p)$
(3) $\mathrm{reg}_{\mathsf{PALA}}(\alpha, \neg\varphi) = (\alpha\cdot\alpha_1, \neg\psi)$,
where $\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \varphi) = (\alpha_1, \psi)$
(4) $\mathrm{reg}_{\mathsf{PALA}}(\alpha, \varphi_1 \wedge \varphi_2) = (\alpha\cdot\alpha_1\cdot\alpha_2, \psi_1 \wedge \psi_2)$,
where $\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \varphi_1) = (\alpha_1, \psi_1)$ and $\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \varphi_2) = (\alpha_2, \psi_2)$
(5) $\mathrm{reg}_{\mathsf{PALA}}(\alpha, \mathbf{K}\varphi) = (\alpha\cdot\alpha_1, \mathbf{K}\psi)$,
where $\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \varphi) = (\alpha_1, \psi)$
(6) $\mathrm{reg}_{\mathsf{PALA}}(\alpha, [!\varphi_1]\varphi_2) = (\alpha\cdot\alpha_1\cdot\alpha_2, [!\psi_1]\psi_2)$,
where $\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \varphi_1) = (\alpha_1, \psi_1)$ and $\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \varphi_2) = (\alpha_2, \psi_2)$
(7) $\mathrm{reg}_{\mathsf{PALA}}(\alpha, [p{:=}\varphi_1]\varphi_2) = (\alpha\cdot\alpha_1\cdot\alpha_2\cdot\mathbf{K}(x_p \leftrightarrow \psi_1), \psi_2[x_p\backslash p])$,
where $\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \varphi_1) = (\alpha_1, \psi_1)$ and $\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \varphi_2) = (\alpha_2, \psi_2)$

The crucial point is Clause 7, that applies Theorem 28. Note that there, and in the other clauses, exponential blowup is avoided by starting on the innermost assignment and then simply concatenating the conjunctions of bi-implications one after another. Also note that the bi-implications do not need to be reduced since they link formulas in $\mathcal{L}_{\mathsf{PAL}}$.

For example, consider the valid formula $[!p][p:=\neg p]\mathbf{K}\neg p$, which means that after the announcement of p and then toggling its truth value, the agent knows that p is false. We have

$$\mathrm{reg}_{\mathsf{PALA}}([!p][p:=\neg p]\mathbf{K}\neg p) = \mathbf{K}(x_p \leftrightarrow \neg p) \wedge [!p]\mathbf{K}\neg x_p.$$

The steps are below.

$$
\begin{aligned}
&\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, p) = (\varepsilon, p)\\
&\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \neg p) = (\varepsilon, \neg p)\\
&\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \mathbf{K}\neg p) = (\varepsilon, \mathbf{K}\neg p)\\
&\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, [p:=\neg p]\mathbf{K}\neg p) = (\mathbf{K}(x_p \leftrightarrow \neg p), \mathbf{K}\neg p[x_p \backslash p])\\
&\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, [!p][p:=\neg p]\mathbf{K}\neg p) = (\mathbf{K}(x_p \leftrightarrow \neg p), [!p]\mathbf{K}\neg x_p)
\end{aligned}
$$

THEOREM 30 $\mathrm{reg}_{\mathsf{PALA}}$ is a polynomial transformation that preserves satisfiability of formulas.

PROOF. Satisfiability-equivalence follows from Theorem 28.

Concerning the length of the translated formulas: let φ be a formula of $\mathcal{L}_{\mathsf{PALA}}$, and let $\mathrm{reg}_{\mathsf{PALA}}(\varepsilon, \varphi) = (\alpha, \psi)$. Note that the length of ψ is bounded by $\mathrm{len}(\varphi)$, because ψ is obtained from φ by dropping its assignments and substituting some of its subformulas by propositional letters. To simplify the presentation suppose that the assignments in φ are singletons, i.e., there are no assignments in parallel. In the sequel we show that $\mathrm{len}(\alpha) \leq 2 \times \mathrm{len}(\varphi)^2$, which implies that $\mathrm{reg}_{\mathsf{PALA}}(\varphi) = O(\mathrm{len}(\varphi)^2)$.

The proof is by induction on the the maximal number of nested assignment operators in φ, i.e., the *assignment depth* of φ. For the induction base, suppose that the assignment depth of φ is 0. That is, $\varphi \in \mathcal{L}_{\mathsf{PAL}}$. Then clearly $\mathrm{len}(\alpha) = 0$, because Clause 7 is never triggered. The induction hypothesis is: if the assignment depth of φ is at most n, then $\mathrm{len}(\alpha) \leq 2n \times \mathrm{len}(\varphi)$. For the induction step, suppose that the assignment depth of φ is $n+1$, and let the formula $[p:=\varphi_1]\varphi_2$ be a subformula of φ such that φ_1 or φ_2 have assignment depth equal to n, and such that $\mathrm{reg}_{\mathsf{PALA}}(\varphi_1, \varepsilon) = (\alpha_1, \psi_1)$ and $\mathrm{reg}_{\mathsf{PALA}}(\varphi_2, \varepsilon) = (\alpha_2, \psi_2)$. By induction hypothesis $\mathrm{len}(\alpha_1) \leq 2n \times \mathrm{len}(\varphi_1)$ and analogously for $\mathrm{len}(\alpha_2)$. Then $\mathrm{len}(\alpha_1) + \mathrm{len}(\alpha_2) \leq$

$2n \times \mathrm{len}(\varphi)$, because $\mathrm{len}(\varphi_1) + \mathrm{len}(\varphi_2) \leq \mathrm{len}(\varphi)$. Therefore:

$$
\begin{array}{lr}
\mathrm{len}(\alpha_1) + \mathrm{len}(\alpha_2) + \mathrm{len}(\mathbf{K}(x_p \leftrightarrow \psi_1)) & \\
\leq \mathrm{len}(\alpha_1) + \mathrm{len}(\alpha_2) + 1 + 2 \times (3 + \mathrm{len}(\psi_1)))) & \text{(because } \leftrightarrow \text{ is an abbreviation)} \\
\leq \mathrm{len}(\alpha_1) + \mathrm{len}(\alpha_2) + 1 + 2 \times \mathrm{len}(\varphi_1)) & \text{(because } \mathrm{len}(\varphi_1) \geq 4 + \mathrm{len}(\psi_1)\text{)} \\
\leq \mathrm{len}(\alpha_1) + \mathrm{len}(\alpha_2) + 2 \times \mathrm{len}(\varphi) & \text{(because } \mathrm{len}(\varphi) > 1 + \mathrm{len}(\varphi_1)\text{)} \\
= 2(n+1) \times \mathrm{len}(\varphi) & \text{(by the observation above).}
\end{array}
$$

And therefore $\mathrm{len}(\alpha) \leq 2 \times \mathrm{len}(\varphi)^2$, because φ has at most $\mathrm{len}(\varphi)$ subformulas containing assignments. ∎

5.2 Eliminating announcements

Once assignments are eliminated, we can eliminate announcements by Lutz' procedure that we recall here. First we compute the set of contextual subformulas which is inductively defined as follows:

$$
\begin{aligned}
S\mathrm{ub}(p) &= \{(\varepsilon, p)\} \\
S\mathrm{ub}(\neg\varphi) &= S\mathrm{ub}(\varphi) \cup \{(\varepsilon, \neg\varphi)\} \\
S\mathrm{ub}(\varphi \wedge \psi) &= S\mathrm{ub}(\varphi) \cup S\mathrm{ub}(\psi) \cup \{(\varepsilon, \varphi \wedge \psi)\} \\
S\mathrm{ub}(\mathbf{K}\varphi) &= S\mathrm{ub}(\varphi) \cup \{(\varepsilon, \mathbf{K}\varphi)\} \\
S\mathrm{ub}([!\varphi]\psi) &= S\mathrm{ub}(\varphi) \cup \{(\varphi \cdot \alpha, \chi) \mid (\alpha, \chi) \in S\mathrm{ub}(\psi)\} \cup \{(\varepsilon, [!\varphi]\psi)\}
\end{aligned}
$$

Intuitively, $S\mathrm{ub}(\varphi)$ is the set of "relevant" subformulas of φ together with the sequence of announcements in the scope of which they occur. $(\alpha, \psi) \in S\mathrm{ub}(\varphi)$ means that the subformula ψ of φ is in the scope of the sequence α of announcements.

Let φ be formula whose PAL satisfiability is to be decided. We introduce a set of fresh propositional letters $P_0^{\varphi} = \{x_{\psi}^{\alpha} : (\alpha, \psi) \in S\mathrm{ub}(\varphi)\}$. Then the reduction of φ is:

$$\mathrm{reg}_{\mathsf{PAL}}(\varphi) = \left(\bigwedge_{(\alpha,\psi) \in S\mathrm{ub}(\varphi)} \mathbf{K}\beta_{\psi}^{\alpha} \right) \wedge x_{\varphi}^{\varepsilon}$$

where the bi-implications β_{ψ}^{α} are inductively defined as follows:

$$
\begin{aligned}
\beta_p^{\alpha} &= x_p^{\alpha} \leftrightarrow p \\
\beta_{\neg\varphi}^{\alpha} &= x_{\neg\varphi}^{\alpha} \leftrightarrow \neg x_{\varphi}^{\alpha} \\
\beta_{\varphi\wedge\psi}^{\alpha} &= x_{\varphi\wedge\psi}^{\alpha} \leftrightarrow (x_{\varphi}^{\alpha} \wedge x_{\psi}^{\alpha}) \\
\beta_{\mathbf{K}\varphi}^{\alpha} &= x_{\mathbf{K}\varphi}^{\alpha} \leftrightarrow \mathbf{K}(\textstyle\bigwedge_{\mu \in \mathrm{pre}(\alpha)} x_{\mu/\alpha}^{\mu} \rightarrow x_{\varphi}^{\alpha}) \\
\beta_{[!\varphi]\psi}^{\alpha} &= x_{[!\varphi]\psi}^{\alpha} \leftrightarrow (x_{\varphi}^{\alpha} \rightarrow x_{\psi}^{\alpha\cdot\varphi})
\end{aligned}
$$

and where pre(α) is the set of true prefixes of α, and μ/α is the leftmost symbol of α that is not in μ. When the sequence α is empty, then the conjunction collapses to $\top$.

Intuitively β^{α}_{ψ} guarantees that x^{α}_{ψ} is true exactly where ψ is true. For example, consider the inconsistent formula $\neg[!p]\mathbf{K}p$. The set of its relevant bi-implications is:

$$\begin{aligned} B = \{ & x^{\varepsilon}_{\neg[!p]\mathbf{K}p} \leftrightarrow \neg x^{\varepsilon}_{[!p]\mathbf{K}p}, \\ & x^{\varepsilon}_{[!p]\mathbf{K}p} \leftrightarrow (x^{\varepsilon}_{p} \to x^{\varepsilon\cdot p}_{\mathbf{K}p}), \\ & x^{\varepsilon\cdot p}_{\mathbf{K}p} \leftrightarrow \mathbf{K}(x^{\varepsilon}_{p} \to x^{\varepsilon\cdot p}_{p}), \\ & x^{\varepsilon\cdot p} \leftrightarrow p, \\ & x^{\varepsilon} \leftrightarrow p \ \} \end{aligned}$$

Then $\mathrm{reg}_{\mathsf{PAL}}(\neg[!p]\mathbf{K}p) = \left(\bigwedge_{\chi\in B}\chi\right) \wedge \mathbf{K}\left(\bigwedge_{\chi\in B}\chi\right) \wedge x^{\varepsilon}_{\neg[!p]\mathbf{K}p}$, which successively implies x^{ε}_{p}, $\neg x^{\varepsilon\cdot p}_{\mathbf{K}p}$, and $\neg\mathbf{K}(x^{\varepsilon}_{p} \to x^{\varepsilon\cdot p}_{p})$. The latter is inconsistent with $\mathbf{K}(x^{\varepsilon\cdot p} \leftrightarrow p)$ and $\mathbf{K}(x^{\varepsilon} \leftrightarrow p)$ which are the last two bi-implications prefixed by $\mathbf{K}$.

THEOREM 31 (Lutz, 2006) $\mathrm{reg}_{\mathsf{PAL}}$ is a polynomial transformation that preserves satisfiability of formulas.

5.3 Complexity results

Via Theorems 26, 27, 30 and 31 we obtain our second main result.

THEOREM 32 Let Θ be a basic action theory satisfying the finite change constraint, and let $\varphi \in \mathcal{L}^{0}_{\mathsf{ES}}$ be a ground box-free formula. The problem of checking entailments $\Theta \models_{\mathsf{ES}} \varphi$ is NP-complete.

We thus do much better than the regression method of Theorem 11 and the reduction method of Theorem 21, which both may cause exponential blowup.

This results also apply to the *plan verification problem* (called projection problem in (Scherl and Levesque, 2003, p.22)).

6 Multiagent extensions

We now show that the results of the previous section can be extended straightforwardly to the multiagent case.

As for the monoagent case, let P_0 be a countable set of propositional letters, and let N be a nonempty finite set of agents. So far we have investigated the case where N is a singleton.

The language of multiagent epistemic logic with common knowledge $\mathcal{L}_{\mathsf{EL}^C}$ is the set of formulas φ defined by the following BNF:

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

where p ranges over P_0, i ranges over N, and G ranges over $\wp(N)$. The *language of multiagent epistemic logic without common knowledge* $\mathcal{L}_{\mathsf{EL}}$ is obtained from $\mathcal{L}_{\mathsf{EL}^C}$ by dropping operators $\mathbf{C}_G$. (We use EL in both the monoagent and the multiagent case in order to simplify notation.)

The formula $\mathbf{K}_i\varphi$ reads 'agent i knows that φ', The formula $\mathbf{E}_G\varphi$ reads 'all agents in group G know that φ', and the formula $\mathbf{C}_G\varphi$ reads 'all agents in group G commonly know that φ'.

$\mathbf{E}_G^\ell\varphi$ abbreviates the ℓ-fold nesting $\mathbf{E}_G \ldots \mathbf{E}_G\varphi$.

A *multiagent epistemic model* is a tuple $\langle W, R, V\rangle$ where W and V are as in Definition 13, and:

- $R : N \to \wp(W \times W)$ associates an equivalence relation $R(i)$ to each $i \in N$.

For convenience, instead of $R(i)$ we write R_i.

We have the usual truth conditions for EL, plus

$$\begin{array}{lll}
M, w \models \mathbf{K}_i\varphi & \text{iff} & R_i(w) \subseteq [\![\varphi]\!]_M \\
M, w \models \mathbf{E}_G\varphi & \text{iff} & \bigcup_{i\in G} R_i(w) \subseteq [\![\varphi]\!]_M \\
M, w \models \mathbf{C}_G\varphi & \text{iff} & \left(\bigcup_{i\in G} R_i\right)^+ (w) \subseteq [\![\varphi]\!]_M
\end{array}$$

where '$+$' is transitive closure. Truth in a model, validity and satisfiability are defined as usual.

If the set of agents N contains at least two elements then the problem of deciding satisfiability is PSPACE-complete for $\mathcal{L}_{\mathsf{EL}}$-formulas, and EXPTIME-complete for $\mathcal{L}_{\mathsf{EL}^C}$-formulas (Halpern and Moses, 1992).

6.2 *Syntax and semantics of multiagent* PALAT *and* PALAT^C

The language of multiagent PALAT extends that of multiagent EL by announcements, assignments and test operators.

Models for multiagent PALAT are just multiagent epistemic models of the previous section. The truth conditions for the epistemic operators are those for multiagent epistemic logic, and the truth conditions for the dynamic operators are those for monoagent PALAT, the only difference being that we have to manage the agent-subscripts of accessibility relations. Thus for example the accessibility relation $R_i^{!\varphi}$ of $M^{!\varphi}$ is $R_i \cap ([\![\varphi]\!]_M \times [\![\varphi]\!]_M)$. Validity and satisfiability are defined as before.

Multiagent PALAT can be axiomatized by means of reduction axioms just as monoagent PALAT. As we have announced in Section 3, there are formulas φ whose multiagent EL-equivalent is exponentially longer than φ (Lutz, 2006, Theorem 2).

EXAMPLE 33 (Theorem 2 of Lutz, 2006) Suppose the underlying epistemic logic is not S5 but K, i.e. accessibility relations are not necessarily equivalence relations. Consider the following family of formulas of multiagent PAL.

$$\begin{aligned} \varphi_0 &= \top \\ \varphi_{n+1} &= \neg[!\neg[!\varphi_n]\mathbf{K}_i\neg\top]\mathbf{K}_j\bot \end{aligned}$$

Every EL-formula that is equivalent to φ_n has length at least exponential in n.

It follows that PAL is more succinct than EL.

6.3 *Complexity results*

In the monoagent case we had eliminated assignments by proving that a $\mathcal{L}_{\mathsf{PALA}}$ formula φ is PALA satisfiable if and only if:

$$\mathbf{K}\Bigg(\bigwedge_{1\le k\le n}(x_{p_k} \leftrightarrow \varphi_k)\Bigg) \wedge \psi$$

is PALA satisfiable. In multiagent PALA, the same result is obtained by replacing the operator $\mathbf{K}$ by the 'everybody knows' operator $\mathbf{E}_N$. In this case, however, we need to iterate the operator up to the horizon of the formula.

THEOREM 34 Let φ be a PALA formula, and let x_{p_k}, φ_k and ψ be as in Theorem 28. Then φ is PALA satisfiable if and only if

$$\Bigg(\bigwedge_{\ell\le \mathrm{md}(\varphi)} \mathbf{E}_N^{\ell}\Bigg(\bigwedge_{1\le k\le n}(x_{p_k} \leftrightarrow \varphi_k)\Bigg)\Bigg) \wedge \psi$$

is EL satisfiable, where $\mathrm{md}(\varphi)$ is the modal depth of φ (the maximal number of nested modal operators in φ).

If the common knowledge operator is available then a single conjunct suffices.

THEOREM 35 Let φ be a PALA^C formula, and let x_{p_k}, φ_k and ψ be as in Theorem 28. Then φ is PALA^C satisfiable if and only if

$$\mathbf{C}_N\left(\bigwedge_{1\leq k\leq n}(x_{p_k} \leftrightarrow \varphi_k)\right) \wedge \psi$$

is EL^C satisfiable.

Both equivalences lead to polynomial transformations.[3] Then Lutz's reduction method for multi-agent PAL can be applied. In other words, we again obtain an optimal theorem proving method.

7 Discussion and conclusion

We have modelled the frame problem in a 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 (Herzig et al., 2000b). The logic for epistemic tests therein has an operator that corresponds to the public announcement operator. 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, and we could have defined them as: $!!\varphi = (!\varphi \cup !\neg\varphi)$, where $\cup$ is nondeterministic choice. The expansion of such abbreviations however leads to an exponential blowup, which does not allow to extend our approach to integrate primitive sensing

[3] Note that it is crucial that the 'everbody knows' operator is primitive in the language $\mathcal{L}_{\mathsf{PALA}}$: if we had defined it as an abbreviation

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

then we would get an exponential blowup in the reduction. (We are grateful to Balder ten Cate for pointing this out to us.)

actions. It is not clear for us how the associated successor state axiom (cf. axiom SSAK in Section 2):

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

could be integrated into the polynomial transformations of sections 5 and 6. Further evidence that the presence of sensing actions increases complexity is provided by the result in (Herzig et al., 2000a) that plan verification in this case is Π_2^p-complete. We therefore leave integration of sensing actions to future work.

The present article also shows that research carried out by situation calculus and dynamic epistemic logic communities go into the same direction. Close similarities between situation calculus and dynamic epistemic logics are also outlined by van Benthem (2007). We believe that this kind of work can aid to bring about advancements on both sides. For example, Scherl&Levesque do not allow for epistemic operators in the formulas $\varphi_{Poss}(a)$, $\varphi_{SF}(a)$, $\gamma^+(a_i, p)$ and $\gamma^-(a'_i, p)$ of basic action theories, while both the announcement $!\varphi$ and the assignment $p{:=}\varphi$ may contain such operators. Another example are non-public actions: they were studied extensively in dynamic epistemic logics, while there is only little work in the situation calculus framework. For integrating such actions one could proceed as in (Baltag et al., 1998; Baltag and Moss, 2004) and add so-called event models that represent the agents' perception of events. The resulting logic comes with a reduction method that extends the one in Theorem 21. One could also use existing model checkers for PALA such as DEMO (van Eijck, 2004) or MCK (Gammie and van der Meyden, 2004).

Going into the other direction, 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 frequently avoids adding quantifiers, predicates, functions, etc, to its 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 (Levesque et al., 1997). Our optimality results could be implemented in order to improve GOLOG's efficiency.

References

Bacchus, F., Halpern, J., Levesque, H., 1999. Reasoning about noisy sensors and effectors in the situation calculus. Artificial Intelligence 111 (1–2), 171–208.

Baltag, A., Moss, L., 2004. Logics for epistemic programs. Synthese 139 (2), 165–224.

Baltag, A., Moss, L., Solecki, S., 1998. 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., pp. 43–46.
Fagin, R., Halpern, J., Moses, Y., Vardi, M., 1995. Reasoning about Knowledge. The MIT Press.
Gammie, P., van der Meyden, R., 2004. MCK: Model checking the logic of knowledge. In: Alur, R., Peled, D. (Eds.), Proceedings of the 16th International Conference on Computer Aided Verification (CAV 2004). Springer, pp. 479–483.
Gelfond, M., Lifschitz, V., Rabinov, A., 1991. What are the limitations of situation calculus. In: Boyer, R. (Ed.), Essays in Honor of Woody Bledsoe. Kluwer Academic Publishers, pp. 167–180.
Gerbrandy, J., 1999. Bisimulations on planet kripke. Ph.D. thesis, ILLC, University of Amsterdam.
Halpern, J., Moses, Y., 1992. A guide to completeness and complexity for modal logics of knowledge and belief. Artificial Intelligence 54, 311–379.
Herzig, A., Lang, J., Longin, D., Polacsek, T., 2000a. 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, pp. 768–773.
Herzig, A., Lang, J., Polacsek, T., 2000b. A modal logic for epistemic tests. In: Horn, W. (Ed.), Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI). IOS Pres, pp. 553–557.
Hintikka, J., 1962. Knowledge and Belief. Cornell University Press.
Kooi, B., 2007. Expressivity and completeness for public update logic via reduction axioms. Journal of Applied Non-Classical Logics 17 (2), 231–253.
Lakemeyer, G., Levesque, H., 2004. Situations, si! Situation terms, no! In: Proceedings of the International Conference on Knowledge Representation and Reasoning (KR). AAAI Press, pp. 516–526.
Lakemeyer, G., Levesque, H., 2005. Semantics for a useful fragment of the situation calculus. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI). Professional Book Center, pp. 490–496.
Levesque, H., Reiter, R., Lespérance, Y., Lin, F., Scherl, R., 1997. GOLOG: A logic programming language for dynamic domains. Journal of Logic Programming 31 (1–2), 59–83.
Lutz, C., 2006. 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). pp. 137–144.
Meyer, J., van der Hoek, W., 1995. Epistemic Logic for AI and Computer Science. No. 41 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.
Nonnengart, A., Weidenbach, C., 2001. Computing small clause normal forms. In: Handbook of Automated Reasoning. North Holland, pp. 335–367.
Plaza, J., 1989. 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). pp. 201–216.
Reiter, R., 1991. 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., pp. 359–380.

Reiter, R., 2001a. Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. The MIT Press.

Reiter, R., 2001b. On knowledge-based programming with sensing in the situation calculus. ACM Transactions on Computational Logic, 433–437.

Scherl, R., Levesque, H., 1993. The frame problem and knowledge-producing actions. In: Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI). The AAAI Press, pp. 689–695.

Scherl, R., Levesque, H., 2003. Knowledge, action and the frame problem. Artificial Intelligence 144 (1–2), 1–39.

Smullyan, R., 1992. The Lady or the Tiger? and Other Logic Puzzles Including a Mathematical Novel That Features Gödel's Great Discovery. Random House Puzzles & Games.

Thielscher, M., 1999. From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. Artificial Intelligence 111 (1-2), 277–299.

van Benthem, J., 2006. "one is a lonely number": logic and communication. In: Chatzidakis, Z., Koepke, P., Pohlers, W. (Eds.), Logic Colloquium'02. Vol. 27 of Lecture Notes in Logic. ASL & A.K. Peters, pp. 96–129.

van Benthem, J., 2007. Modal logic meets situation calculus, manuscript.

van Benthem, J., van Eijck, J., Kooi, B., 2006. Logics of communication and change. Information and Computation 204(11), 1620–1662.

van Ditmarsch, H., Herzig, A., de Lima, T., 2007a. Optimal regression for reasoning about knowledge and actions. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence. AAAI Press, pp. 1070–1075.

van Ditmarsch, H., Herzig, A., de Lima, T., 2007b. Optimal regression for reasoning about knowledge and actions. In: Bonanno, G., Delgrande, J., Lang, J., Rott, H. (Eds.), Formal Models of Belief Change in Rational Agents. Dagstuhl Seminar Proceedings, Dagstuhl (Germany).
URL `http://drops.dagstuhl.de/portals/07351`

van Ditmarsch, H., van der Hoek, W., Kooi, B., 2005. 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, pp. 141–148.

van Ditmarsch, H., van der Hoek, W., Kooi, B., 2007c. Dynamic Epistemic Logic. Vol. 337 of Synthese Library. Springer.

van Eijck, J., 2004. Dynamic epistemic modelling. Tech. rep., Centrum voor Wiskunde en Informatica, Amsterdam, cWI Report SEN-E0424.

A Proof of Theorem 26

THEOREM 26 Let Θ be a basic action theory satisfying the finite change constraint, and let $\varphi \in \mathcal{L}^0_{\mathsf{ES}}$. Then $\Theta \models_{\mathsf{ES}} \varphi$ if and only if $\models_{\mathsf{PALAT}} \mathrm{tra}_\Theta(\varphi)$.

PROOF. We take advantage of both regression and reduction: by Theorem 11, $\Theta \models_{\mathsf{ES}} \varphi$ iff $\models_{\mathsf{EL}} \mathrm{reg}_\Theta(\varphi)$; by Theorem 21, $\models_{\mathsf{PALAT}} \mathrm{tra}_\Theta(\varphi)$ iff $\models_{\mathsf{EL}} \mathrm{red}(\mathrm{tra}_\Theta(\varphi))$.

It remains to prove that $\models_{\mathsf{EL}} \mathrm{reg}_\Theta(\varphi)$ iff $\models_{\mathsf{EL}} \mathrm{red}(\mathrm{tra}_\Theta(\varphi))$. To that end we prove that $\models_{\mathsf{ES}} \mathrm{reg}_\Theta(\varphi) \leftrightarrow \mathrm{red}(\mathrm{tra}_\Theta(\varphi))$ by induction on the length of φ, where the case $[a]\varphi$ is decomposed into subcases. Our proof extensively uses the following lemma.

LEMMA 36 If $\models_{\mathsf{PALAT}} \varphi_1 \leftrightarrow \varphi_2$ then $\models_{\mathsf{ES}} \mathrm{red}(\varphi_1) \leftrightarrow \mathrm{red}(\varphi_2)$.

PROOF. Suppose that $\models_{\mathsf{PALAT}} \varphi_1 \leftrightarrow \varphi_2$. By the Reduction Theorem 21, $\models_{\mathsf{PALAT}} \varphi_1 \leftrightarrow \mathrm{red}(\varphi_1)$ and $\models_{\mathsf{PALAT}} \varphi_2 \leftrightarrow \mathrm{red}(\varphi_2)$. Therefore $\models_{\mathsf{PALAT}} \mathrm{red}(\varphi_1) \leftrightarrow \mathrm{red}(\varphi_2)$.

The formula $\mathrm{red}(\varphi_1) \leftrightarrow \mathrm{red}(\varphi_2)$ being in the language of EL we use that both PALAT and ES are conservative extensions of EL: first, $\models_{\mathsf{EL}} \mathrm{red}(\varphi_1) \leftrightarrow \mathrm{red}(\varphi_2)$ by Proposition 19, and second, $\models_{\mathsf{ES}} \mathrm{red}(\varphi_1) \leftrightarrow \mathrm{red}(\varphi_2)$ by Proposition 15. ∎

Let us now prove that $\models_{\mathsf{ES}} \mathrm{reg}_\Theta(\varphi) \leftrightarrow \mathrm{red}(\mathrm{tra}_\Theta(\varphi))$ by induction on the length of φ. We analyze the possible cases and subcases concerning the form of φ.

(1) $\models_{\mathsf{ES}} \mathrm{reg}_\Theta(p) \leftrightarrow \mathrm{red}(\mathrm{tra}_\Theta(p))$
This clearly holds, given that $\mathrm{reg}_\Theta(p) = p = \mathrm{tra}_\Theta(p) = \mathrm{red}(\mathrm{tra}_\Theta(p))$.

(2) $\models_{\mathsf{ES}} \mathrm{reg}_\Theta(a_1 = a_2) \leftrightarrow \mathrm{red}(\mathrm{tra}_\Theta(a_1 = a_2))$
This can be proved by checking the cases where a_1 and a_2 are (syntactically) equal, and where they are different.

(3) $\models_{\mathsf{ES}} \mathrm{reg}_\Theta(\neg\psi) \leftrightarrow \mathrm{red}(\mathrm{tra}_\Theta(\neg\psi))$
We have $\mathrm{red}(\mathrm{tra}_\Theta(\neg\psi)) = \mathrm{red}(\neg\,\mathrm{tra}_\Theta(\psi)) = \neg\,\mathrm{red}(\mathrm{tra}_\Theta(\psi))$. By induction hypothesis (and by the rule of substitution of equivalents of ES) the latter is ES equivalent to $\neg\,\mathrm{reg}_\Theta(\psi)$. Finally, $\neg\,\mathrm{reg}_\Theta(\psi) = \mathrm{reg}_\Theta(\neg\psi)$.

(4) $\models_{\mathsf{ES}} \mathrm{reg}_\Theta(\psi \wedge \psi') \leftrightarrow \mathrm{red}(\mathrm{tra}_\Theta(\psi \wedge \psi'))$
Similar to the case of negation.

(5) $\models_{\mathsf{ES}} \mathrm{reg}_\Theta([a]p) \leftrightarrow \mathrm{red}(\mathrm{tra}_\Theta([a]p))$
We have

$$\begin{aligned}\mathrm{reg}_\Theta([a]p) = {} & (a{=}a_1 \wedge \gamma^+(a_1,p)) \vee \cdots \vee (a{=}a_n \wedge \gamma^+(a_n,p)) \vee \\ & (p \wedge \neg(a{=}a'_1 \wedge \gamma^-(a'_1,p)) \wedge \cdots \wedge \neg(a{=}a'_m \wedge \gamma^-(a'_m,p)))\end{aligned}$$

The latter is ES equivalent to:

- p if $p \notin \mathit{Eff}^-(a) \cup \mathit{Eff}^+(a)$;
- $\gamma^+(a,p) \vee p$ if $p \in \mathit{Eff}^+(a) \setminus \mathit{Eff}^-(a)$;
- $p \wedge \neg\gamma^-(a,p)$ if $p \in \mathit{Eff}^-(a) \setminus \mathit{Eff}^+(a)$;
- $\gamma^+(a,p) \vee (p \wedge \neg\gamma^-(a,p))$ if $p \in \mathit{Eff}^-(a) \cap \mathit{Eff}^+(a)$.

The latter are syntactically equal to $\sigma_a(p)$, where σ_a is as in Definition 24. Now $\sigma_a(p) = \mathrm{red}(\sigma_a(p))$ because $\sigma_a(p)$ is a Boolean formula. For the same reason, by Proposition 18 we have $\models_{\mathsf{PALAT}} \sigma_a(p) \leftrightarrow [!!\varphi_{SF}(a)]\sigma_a(p)$. Therefore $\models_{\mathsf{PALAT}} \mathrm{red}(\sigma_a(p)) \leftrightarrow \mathrm{red}([!!\varphi_{SF}(a)][\sigma_a]p)$. Finally, the latter is nothing but $\mathrm{red}(\mathrm{tra}_\Theta([a]p))$.

(6) $\models_{\mathsf{ES}} \mathrm{reg}_\Theta([a]a_1 = a_2) \leftrightarrow \mathrm{red}(\mathrm{tra}_\Theta([a]a_1 = a_2))$
Straightforward by checking the cases.

(7) $\models_{\mathsf{ES}} \mathrm{reg}_\Theta([a]\neg\psi) \leftrightarrow \mathrm{red}(\mathrm{tra}_\Theta([a]\neg\psi))$
Straightforward by the induction hypothesis and using that $\mathrm{tra}_\Theta([a]\neg\psi) \leftrightarrow \mathrm{tra}_\Theta(\neg[a]\psi)$ holds (because tests and assignments are both deterministic and executable by Proposition 18).

(8) $\models_{\mathsf{ES}} \mathrm{reg}_\Theta([a](\psi \wedge \psi')) \leftrightarrow \mathrm{red}(\mathrm{tra}_\Theta([a](\psi \wedge \psi')))$
Straightforward by applying the induction hypothesis and the equivalence $[a](\psi \wedge \psi') \leftrightarrow ([a]\psi \wedge [a]\psi')$ that is both ES and PALAT valid.

(9) $\models_{\mathsf{ES}} \mathrm{reg}_\Theta([a]\mathbf{K}\psi) \leftrightarrow \mathrm{red}(\mathrm{tra}_\Theta([a]\mathbf{K}\psi))$
We regress the left hand side: $\mathrm{reg}_\Theta([a]\mathbf{K}\psi)$ is ES equivalent to

$$\mathrm{reg}_\Theta((\varphi_{SF}(a) \to \mathbf{K}(\varphi_{SF}(a) \to [a]\psi)) \wedge (\neg\varphi_{SF}(a) \to \mathbf{K}(\neg\varphi_{SF}(a) \to [a]\psi)))$$

which is ES equivalent to

$$\begin{aligned}&(\varphi_{SF}(a) \to \mathbf{K}(\varphi_{SF}(a) \to \mathrm{reg}_\Theta([a]\psi)))\wedge \\ &\qquad\qquad (\neg\varphi_{SF}(a) \to \mathbf{K}(\neg\varphi_{SF}(a) \to \mathrm{reg}_\Theta([a]\psi)))\end{aligned}$$

because $\varphi_{SF}(a)$ is Boolean.

We reduce the right hand side by means of ES equivalences, using the above Lemma 36:

$$\begin{aligned}\mathrm{red}(\mathrm{tra}_\Theta([a]\mathbf{K}\psi)) &\leftrightarrow \mathrm{red}([!!\varphi_{SF}(a)][\sigma_a]\mathbf{K}\psi) \\ &\leftrightarrow \mathrm{red}([!!\varphi_{SF}(a)]\mathbf{K}[\sigma_a]\psi) \\ &\leftrightarrow \mathrm{red}([!\varphi_{SF}(a)]\mathbf{K}[\sigma_a]\psi \wedge [!\neg\varphi_{SF}(a)]\mathbf{K}[\sigma_a]\psi) \\ &\leftrightarrow \varphi_{SF}(a) \to \mathbf{K}\,\mathrm{red}([!\varphi_{SF}(a)][\sigma_a]\psi) \wedge \\ &\qquad \neg\varphi_{SF}(a) \to \mathbf{K}\,\mathrm{red}([!\neg\varphi_{SF}(a)][\sigma_a]\psi) \\ &\leftrightarrow \varphi_{SF}(a) \to \mathbf{K}(\varphi_{SF}(a) \to [!!\varphi_{SF}(a)][\sigma_a]\psi) \wedge \\ &\qquad \neg\varphi_{SF}(a) \to \mathbf{K}(\neg\varphi_{SF}(a) \to [!!\varphi_{SF}(a)][\sigma_a]\psi) \\ &\leftrightarrow \varphi_{SF}(a) \to \mathbf{K}(\varphi_{SF}(a) \to \mathrm{tra}_\Theta([a]\psi)) \wedge \\ &\qquad \neg\varphi_{SF}(a) \to \mathbf{K}(\neg\varphi_{SF}(a) \to \mathrm{tra}_\Theta([a]\psi))\end{aligned}$$

The last but one step uses that by Proposition 18 announcements can be defined by means of tests.

Finally, by induction hypothesis the regressed left hand side and the reduced right hand side are equivalent.

This ends the proof. ∎

B Proof of Theorem 27

THEOREM 27 Let Θ be a basic action theory satisfying the final change constraint, and let $\varphi \in \mathcal{L}^0_{\mathsf{ES}}$. Then $\text{len}(\text{tra}_\Theta(\varphi)) \leq O(\text{len}(\Theta) \times \text{len}(\varphi))$.

PROOF. We prove by induction on the structure of φ that $\text{len}(\text{tra}_\Theta(\varphi)) \leq 3 \times \text{len}(\Theta) \times \text{len}(\varphi)$.

(1) $\text{len}(\text{tra}_\Theta(p)) = \text{len}(p) = 1 \leq 3 \times \text{len}(\Theta)$

(2) $\text{len}(\text{tra}_\Theta(a_1 = a_2)) \leq 5 \leq 3 \times \text{len}(\Theta) \times 3$
This can be proved by analyzing the two possible cases $\text{tra}_\Theta(\varphi) = \top$ and $\text{tra}_\Theta(\varphi) = \bot$. (Note that $\text{len}(\Theta) \geq 1$, and remember that $\top$ and $\bot$ are abbreviations.)

(3) $\text{len}(\text{tra}_\Theta(Poss(a)) = \text{len}(\varphi_{Poss}(a)) \leq \text{len}(\Theta)$

(4) $\text{len}(\text{tra}_\Theta(SF(a)) = \text{len}(\varphi_{SF}(a)) \leq \text{len}(\Theta)$

(5) $\text{len}(\text{tra}_\Theta(\neg\varphi_1)$
$= \text{len}(\neg\,\text{tra}_\Theta(\varphi_1))$
$= 1 + \text{len}(\text{tra}_\Theta(\varphi_1))$
$< 3 \times \text{len}(\Theta) + \text{len}(\text{tra}_\Theta(\varphi_1))$ (because $1 < 3 \times \text{len}(\Theta)$)
$\leq 3 \times \text{len}(\Theta) + 3 \times \text{len}(\Theta) \times \text{len}(\varphi_1)$ (by induction hypothesis)
$= 3 \times \text{len}(\Theta) \times (1 + \text{len}(\varphi_1))$
$= 3 \times \text{len}(\Theta) \times \text{len}(\neg\varphi_1)$.

(6) $\text{len}(\text{tra}_\Theta(\varphi_1 \wedge \varphi_2)) \leq 3 \times \text{len}(\Theta) \times \text{len}(\varphi_1 \wedge \varphi_2)$
Similar to the case of negation.

(7) $\text{len}(\text{tra}_\Theta(\mathbf{K}\varphi_1)) \leq 3 \times \text{len}(\Theta) \times \text{len}(\mathbf{K}\varphi_1)$
Similar to the case of negation.

(8) $\text{len}(\text{tra}_\Theta([a]\varphi_1))$
$= \text{len}([!!\varphi_{SF}(a)][\sigma_a]\,\text{tra}_\Theta(\varphi_1))$
$= 2 + \text{len}(\varphi_{SF}(a)) + \text{len}(\sigma_a) + \text{len}(\text{tra}_\Theta(\varphi_1))$
$\leq 2 + \text{len}(\Theta) + \text{len}(\Theta) + \text{len}(\text{tra}_\Theta(\varphi_1))$
$\leq 3 \times \text{len}(\Theta) + \text{len}(\text{tra}_\Theta(\varphi_1))$ (because $\text{len}(\Theta) \geq 3$ since Θ is not empty)
$\leq 3 \times \text{len}(\Theta) + 3 \times \text{len}(\Theta) \times \text{len}(\varphi_1)$ (by induction hypothesis)
$= 3 \times \text{len}(\Theta) \times (1 + \text{len}(\varphi_1))$
$\leq 3 \times \text{len}(\Theta) \times (2 + \text{len}(\varphi_1))$
$= 3 \times \text{len}(\Theta) \times \text{len}([a]\varphi_1)$.

This ends the proof. ∎

Contents