Logics of Communication and Change

Johan van Benthem[*], Jan van Eijck[†] and Barteld Kooi[‡]

March 4, 2005

Abstract

Current dynamic epistemic logics often become cumbersome and opaque when common knowledge is added for groups of agents. Still, postconditions regarding common knowledge express the essence of what communication achieves. We propose new systems that extend the underlying static epistemic languages in such a way that completeness proofs for the full dynamic systems can be obtained by perspicuous reduction axioms. Also, we include factual alteration, rather than just information change, which allows us to cover a much wider range of phenomena in the area of communication and change.

1 Introduction

Epistemic logic typically deals with what agents consider possible given their current information. This includes knowledge about facts, but also *higher-order information* about information that other agents have. A prime example is *common knowledge*. A formula φ is common knowledge if everybody knows φ, everybody knows that everybody knows that φ, and so on. Incidentally, although this paper is mainly written in 'knowledge' terminology, everything we say also holds, with minor technical modifications, when describing agents *beliefs*, including common belief in groups.

Dynamic epistemic logics analyze changes in both basic and higher-order information. One of the main attractions of such systems is their perspicuous analysis of effects of communicative actions in the format of an equivalence between epistemic postconditions and preconditions. A typical example concerns knowledge of an agent after and before a public announcement:

$$[\varphi]\Box_a\psi \leftrightarrow (\varphi \to \Box_a[\varphi]\psi).$$

We call such principles *reduction axioms*, because the announcement operator is 'pushed through' the epistemic operators. Such axioms make logical systems particularly perspicuous. For instance, the logic of public announcements without common knowledge has an easy completeness proof by way of a translation that follows the reduction axioms. Formulas with announcements are translated to provably equivalent ones without announcements. Then completeness follows from the known completeness of the epistemic base logic. This approach is also taken in [15] and [1] for more general epistemic updates.

But the heart of epistemic action concerns knowledge acquired by *groups*. In particular, we want to understand what groups and subgroups know in societies of communicating agents. Now, existing completeness proofs for dynamic epistemic logics with common knowledge do not follow the above perspicuous pattern. Reduction axioms are not available, as the logic with epistemic updates is more expressive than the logic without them [1]. We think this points at an infelicity in the design of current epistemic languages. In this paper, to restore the proper harmony between the static and dynamic features of the

[*]ILLC, University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands & Philosophy Department, Stanford University, `johan@science.uva.nl`

[†]CWI, P.O. Box 94079 1090 GB Amsterdam, `jve@cwi.nl`

[‡]Department of Philosophy, University of Groningen, Oude Boteringstraat 52, 9712 GL Groningen, The Netherlands, `barteld@philos.rug.nl`

system, we extend the epistemic base language with new operators in such a way that reduction axioms do work.

Before defining our most complex dynamic epistemic systems, we first look at a pilot case which demonstrates many issues in a simpler focus, the logic PAL of *public announcements*. Section 2 gives a new and complete set of reduction axioms for public announcement logic with common knowledge, obtained by strengthening the base language with an operator of relativized common knowledge, as first proposed in [4]. Moreover, since languages with model-shifting operators like $[\varphi]$ are of independent logical interest, we develop the model theory of PAL a bit further, using game techniques to investigate expressive power of several variants. Next, in Section 3 we move from public announcements to samples of the more general kinds of information change that we are interested in, including privacy, partial observation, and eventually also real physical actions changing the world. Section 4 proposes a new dynamic epistemic system dealing with all this, slightly generalizing the standard reference [1] to include factual change. More importantly, we strengthen the usual epistemic base language to a full-fledged PDL-style version including agents defined by complex program expressions. This much richer setting again allows for smooth reduction axioms for common knowledge after epistemic updates with factual change. A system of this sort was first proposed using finite automata techniques in [19], using a variant of propositional dynamic logic called 'automata PDL'. The new techniques used here (cf. [11]) work directly in epistemic PDL, using the idea behind Kleene's Theorem for regular languages and finite automata to find the relevant reduction axioms inductively. This analysis does not just yield the meta-theorems of completeness and decidability that we are after. It can also be used to actually compute valid axioms analyzing common knowledge following specific communicative or informational events. Section 5 analyzes some of our earlier communication types in just this fashion, obviating the need for earlier laborious calculations 'by hand' (cf. [27]). Section 6 draws conclusions and indicates some directions for further research.

The broader context for this paper are earlier systems of dynamic epistemic logic, with [26], [15], and [1] as key examples of progressively stronger systems, while [8] is a source of inspiring examples. Another major influence is the work of [13] on common knowledge in computational settings, using a more general temporal-epistemic framework allowing also for global protocol information about communicative processes. Connections between the two approaches are found, e.g., in [22]. Cf. [6] for further references to the literature on epistemic actions, and open problems in the general landscape of update logics.

Even though the main thrust of this paper is technical, we see our proposal as more than just a trick for smoothening completeness proofs. It also addresses a significant design issue of independent interest: what is the best epistemic language for describing information models of a group of agents? In particular, our main logic LCC in Section 4 is meant as a serious working for this purpose.

2 Logics of Public Announcement

Public announcement is the simplest form of communicative action. The corresponding update idea that announcing a proposition φ which is currently true removes all worlds where φ does not hold goes back far into the mists of logical folklore, and it has been stated explicitly since the 1970s by Stalnaker, Heim, and others. The same idea also served as a high-light in the work on epistemic logic in computer science (cf. [13]). Its first implementation as a dynamic-epistemic logic seems due to Plaza [26].

Section 2.1 is a brief introduction to public announcement logic (PAL) as usually stated. In Section 2.2 we give a new base logic of *relativized common knowledge*, EL-RC. This extension was first proposed in [4], which analyzed updates as a kind of *relativization* operator on models. Restricted or 'bounded' versions of logical operators like quantifiers or modalities are very common in semantics. The resulting epistemic logic is expressive enough to allow a reduction axiom for common knowledge. A proof system is defined in Section 2.3, and shown to be complete in Section 2.4. The system is extended with reduction axioms for public announcements in Section 2.5. Characteristic model comparison games for these logics are provided in Section 2.6. This technique is then used to investigate the expressive power of relativized common knowledge in Section 2.7, settling all issues of system comparison in this area. Finally, complexity issues are briefly discussed in Section 2.8.

2.1 Language and Semantics of PAL

Public announcement logic (PAL) was first defined by PLaza [26]. A public announcement is an epistemic update where all agents commonly know that they learn that a certain formula holds right now. This is modeled by a modal operator $[\varphi]$. A formula of the form $[\varphi]\psi$ is read as 'ψ holds after the announcement of φ'. If we also add an operator $C_B\varphi$ to express that φ is common knowledge among agents B, we get public announcement logic with common knowledge (PAL-C). The languages $\mathscr{L}_{\mathsf{PAL}}$ and $\mathscr{L}_{\mathsf{PAL\text{-}C}}$ are interpreted in standard models for epistemic logic.

Definition 1 (Epistemic models) Let a finite set of propositional variables P and a finite set of agents N be given. An epistemic model is a triple $M = (W, R, V)$ such that

- $W \neq \emptyset$ is a set of possible worlds,
- $R : N \to \wp(W \times W)$ assigns an accessibility relation $R(a)$ to each agent a,
- $V : P \to \wp(W)$ assigns a set of worlds to each propositional variable. □

In epistemic logic the relations $R(a)$ are usually equivalence relations. In this paper we treat the general modal case without such constraints - making 'knowledge' more like *belief*, as observed earlier. But our results also apply to the special modal S5-case of equivalence relations. The semantics are defined with respect to models with a distinguished 'actual world': M, w.

Definition 2 (Semantics of PAL and PAL-C) Let a model M, w with $M = (W, R, V)$ be given. Let $a \in N$, $B \subseteq N$, and $\varphi, \psi \in \mathscr{L}_{\mathsf{PAL}}$. For atomic propositions, negations, and conjunctions we take the usual definition:

$$\begin{array}{lll} M, w \models \Box_a\varphi & \text{iff} & M, v \models \varphi \text{ for all } v \text{ such that } (w, v) \in R(a) \\ M, w \models [\varphi]\psi & \text{iff} & M, w \models \varphi \text{ implies } M|\varphi, w \models \psi \\ M, w \models C_B\varphi & \text{iff} & M, v \models \varphi \text{ for all } v \text{ such that } (w, v) \in R(B)^* \end{array}$$

where $R(B) = \bigcup_{a \in B} R(a)$, and $R(B)^*$ is its reflexive transitive closure. The updated model $M|\varphi = (W', R', V')$ is defined by restricting M to those worlds where φ holds. Let

$$[\![\varphi]\!] = \{v \in W | M, v \models \varphi\}.$$

Now $W' = [\![\varphi]\!]$, $R'(a) = R(a) \cap [\![\varphi]\!]^2$, and $V'(p) = V(p) \cap [\![\varphi]\!]$. □

Here, mostly for convenience, we chose to define common knowledge as a *reflexive* transitive closure, as in [23]. In [13], common knowledge is defined as just *transitive* closure, which would be closer to the natural idea of *common belief*. Our results will work either way, with minimal adaptations to the case of common belief.

A completeness proof for public announcement logic without an operator for common knowledge (PAL) is straightforward.

Definition 3 (Proof system for PAL) The proof system for PAL is that for multi-modal S5 epistemic logic plus the following reduction axioms:

At	$[\varphi]p \leftrightarrow (\varphi \to p)$	(atoms)
PF	$[\varphi]\neg\psi \leftrightarrow (\varphi \to \neg[\varphi]\psi)$	(partial functionality)
Dist	$[\varphi](\psi \wedge \chi) \leftrightarrow ([\varphi]\psi \wedge [\varphi]\chi)$	(distribution)
KA	$[\varphi]\Box_a\psi \leftrightarrow (\varphi \to \Box_a[\varphi]\psi)$	(knowledge-announcement)

as well as an inference rule of necessitation for all announcement modalities. □

The formulas on the left of these equivalences are of the form $[\varphi]\psi$. In **At** the announcement operator no longer occurs on the right-hand side. In the other reduction axioms formulas within the scope of an announcement are of higher complexity on the left than on the right.

For public announcement logic including a common knowledge operator (PAL-C), a completeness proof with reduction axioms is impossible. There is no reduction axiom for formulas of the form $[\varphi]C_B\psi$, given the results in [1].

2.2 Relativized Common Knowledge: EL-RC

The semantic intuitions are clear, however. If φ is true in the old model, then every B-path in the new model ends in a ψ world. This implies that in the old model every B-path that consists exclusively of φ-worlds ends in a $[\varphi]\psi$ world. To facilitate this, we introduce a new operator $C_B(\varphi, \psi)$, which expresses that

> every B-path which consists exclusively of φ-worlds ends in a ψ world.

A natural language paraphrase of this notion might be 'if φ is announced it will become common knowledge among B that ψ was the case before the announcement.' So we consider only 'conditional common knowledge' of agents under the assumption that φ, just as one does in logics of conditionals, or in Bayesian update involving conditional probability. We call this operator *relativized common knowledge*. Putting this in a definition:

Definition 4 (Language and Semantics of EL-RC) The language of EL-RC is that of EL, together with the operator for relativized common knowledge, with semantics given by:

$$M, w \models C_B(\varphi, \psi) \text{ iff } M, v \models \psi \text{ for all } v \text{ such that } (w, v) \in (R(B) \cap [\![\varphi]\!]^2)^*$$

where $(R(B) \cap [\![\varphi]\!]^2)^*$ is the reflexive transitive closure of $R(B) \cap [\![\varphi]\!]^2$. □

Note that common knowledge relativized to φ is *not* what results from a public update with φ. E.g., $[p]C_B\Diamond_a\neg p$ is *not* equivalent to $C_B(p, \Diamond_a\neg p)$, for $[p]C_B\Diamond_a\neg p$ is always false, and $C_B(p, \Diamond_a\neg p)$ holds in models where every p path ends in a world with an a successor with $\neg p$. In Section 2.7 we will show that $C_B(p, \Diamond_a\neg p)$ cannot be expressed in PAL-C.

The semantics of the other operators is standard. Ordinary common knowledge can be defined with the new notion: $C_B\varphi \equiv C_B(\top, \varphi)$. The new operator is like the 'until' of temporal logic. A temporal sentence 'φ until ψ' is true iff there is some point in the future where ψ holds and φ is true up to that point.

2.3 Proof System for EL-RC

Relativized common knowledge still resembles common knowledge, and so we need just a slight adaptation of the usual axioms.

Definition 5 (Proof system for EL-RC) The proof system for EL-RC contains the following axioms and rules:

Taut	all instantiations of propositional tautologies	
Dist	$\Box_a(\varphi \to \psi) \to (\Box_a\varphi \to \Box_a\psi)$	(distribution)
Dist	$C_B(\varphi, \psi \to \chi) \to (C_B(\varphi, \psi) \to C_B(\varphi, \chi))$	(distribution)
Mix	$C_B(\varphi, \psi) \leftrightarrow (\varphi \to (\psi \wedge E_B(\varphi \to C_B(\varphi, \psi))))$	(mix)
Ind	$((\varphi \to \psi) \wedge C_B(\varphi, \psi \to E_B(\varphi \to \psi))) \to C_B(\varphi, \psi)$	(induction)
MP	$\dfrac{\varphi \qquad \varphi \to \psi}{\psi}$	(modus ponens)
Nec	$\dfrac{\varphi}{\Box_a\varphi}$	(necessitation)
Nec	$\dfrac{\varphi}{C_B(\psi, \varphi)}$	(necessitation)

In the mix axiom and the induction axiom $E_B\varphi$ is an abbreviation of $\bigwedge_{a\in B} \Box_a\varphi$ (*everybody* knows φ).

These axioms are all sound on the intended interpretation. In particular, understanding the validity of the relativized versions **Mix** and **Ind** provides the main idea of our analysis.

Next, as usual, a proof consists of a sequence of formulas such that each is either an instance of an axiom, or it can be obtained from formulas that appear earlier in the sequence by applying a rule. If there is a proof of φ, we write $\vdash \varphi$. $\Box$

Remark It may also be helpful to write $C_B(\varphi, \psi)$ as a sentence in propositional dunamic logic PDL: $[?\varphi; (\bigcup_{a\in B} a; ?\varphi)^*]\psi$. Our proof system essentially follows the usual PDL-axioms for this formula. This technical idea is the key to our more general system LCC in Section 4 below.

2.4 Completeness for EL-RC

To prove completeness for our extended static language EL-RC, we follow [17], [13]. The argument is standard, and our main point is that the usual proof actually yields information about a richer language than is commonly realized.

For a start, we take maximally consistent sets with respect to finite fragments of the language that form a canonical model for that fragment. In particular, for any given formula φ we work with a finite fragment called the *closure* of φ.

Definition 6 (Closure) The *closure* of φ is the minimal set Φ such that

1. $\varphi \in \Phi$,
2. Φ is closed under taking subformulas,
3. If $\psi \in \Phi$ and ψ is not a negation, then $\neg\psi \in \Phi$,
4. If $C_B(\psi, \chi) \in \Phi$, then $\Box_a(\psi \to C_B(\psi, \chi)) \in \Phi$ for all $a \in B$. $\Box$

Definition 7 (Canonical model) The *canonical model* M_φ for φ is the triple $(W_\varphi, R_\varphi, V_\varphi)$ where

- $W_\varphi = \{\Gamma \subseteq \Phi \mid \Gamma \text{ is maximally consistent in } \Phi\}$;
- $(\Gamma, \Delta) \in R_\varphi(a)$ iff $\psi \in \Delta$ for all ψ with $\Box_a\psi \in \Gamma$;
- $V_\varphi(p) = \{\Gamma \mid p \in \Gamma\}$. $\Box$

Next, we show that a formula in such a finite set is true in the canonical model where that set is taken to be a world, and vice versa.

Lemma 1 (Truth Lemma) For all $\psi \in \Phi$, $\psi \in \Gamma$ iff $M_\varphi, \Gamma \models \psi$. $\square$

Proof By induction on ψ. The cases for propositional variables, negations, conjunction, and individual epistemic operators are straightforward. Therefore we focus on the case for relativized common knowledge.

From left to right. Suppose $C_B(\psi, \chi) \in \Gamma$. If $\psi \notin \Gamma$, then by the induction hypothesis $M_\varphi, \Gamma \not\models \psi$, and by the semantics $M_\varphi, \Gamma \models C_B(\psi, \chi)$.

Otherwise, if $\psi \in \Gamma$, take any $\Delta \in W_\varphi$ such that $(\Gamma, \Delta) \in (R(B) \cap [\![\psi]\!]^2)^*$. We have to show that $\Delta \models \chi$, but we can show something stronger, namely that $\Delta \models \chi$ and $C_B(\psi, \chi) \in \Delta$. This is done by induction on the length of the path from Γ to Δ and the **Mix** axiom. We omit the details.

From right to left. Let $M_\varphi, \Gamma \models C_B(\psi, \chi)$. Now consider the set $\Lambda = \{\delta_\Delta | (\Gamma, \Delta) \in (R(B) \cap [\![\psi]\!]^2)^*\}$, where δ_Δ is $\bigwedge\{\psi | \psi \in \Delta\}$. Let $\delta_\Lambda = \bigvee \Lambda$ We can show that $\vdash \delta_\Lambda \to E_B(\psi \to \delta_\Lambda)$. By necessitation then $\vdash C_B(\psi, \delta_\Lambda \to E_B(\psi \to \delta_\Lambda))$. Applying the induction axiom, we get $\vdash (\psi \to \delta_\Lambda) \to C_B(\psi, \delta_\Lambda)$. Since $\vdash \delta_\Lambda \to \chi$, we also get $\vdash (\psi \to \delta_\Lambda) \to C_B(\psi, \chi)$. Now $\delta_\Gamma \to \delta_\Lambda$, and hence $\vdash \delta_\Gamma \to (\psi \to \delta_\Lambda)$. Therefore $C_B(\psi, \chi) \in \Gamma$. $\square$

This argument is an easy adaptation of the usual completeness proof for epistemic logic with just basic common knowledge, reinforcing our idea that our language extension is a natural one. It should also be noted that this argument would also go through for a relativized operator of common belief. The only changes in the proof system are in the axioms for **Mix** and **Ind**. Instead of those, we would get:

$$\begin{array}{ll} \textbf{Mix}' & C_B(\varphi, \psi) \leftrightarrow E_B(\varphi \to (\psi \wedge C_B(\varphi, \psi))), \\ \textbf{Ind}' & (E_B(\varphi \to \psi) \wedge C_B(\varphi, \psi \to E_B(\varphi \to \psi))) \to C_B(\varphi, \psi). \end{array}$$

Theorem 1 (Completeness for EL-RC) $\models \varphi$ iff $\vdash \varphi$. $\square$

Proof Let $\not\vdash \varphi$, i.e. $\neg\varphi$ is consistent. One easily finds a maximally consistent set Γ in the closure of $\neg\varphi$ with $\neg\varphi \in \Gamma$, as only finitely many formulas matter. By the Truth Lemma, $M_{\neg\varphi}, \Gamma \models \neg\varphi$, i.e., $M_{\neg\varphi}, \Gamma \not\models \varphi$.

The soundness of the proof system can easily be shown by induction on the length of proofs, and we do not provide it explicitly. $\square$

2.5 Reduction Axioms for PAL-RC

Next, let PAL-RC be the epistemic dynamic logic with both relativized common knowledge and public announcements. Its semantics combines those for PAL and EL-RC. We want to find a reduction axiom for $[\varphi]C_B(\psi, \chi)$, the formula that expresses that after public announcement of φ, every ψ path leads to a χ world. Note that this holds exactly in those worlds where every φ path where announcing φ makes ψ true ends in a world where announcing φ makes χ true. This observation yields the following proof system for PAL-RC:

Definition 8 (Proof system for PAL-RC) The proof system for PAL-RC is that for EL-RC plus the reduction axioms for PAL, together with:

$$\textbf{C-Red}\ [\varphi]C_B(\psi, \chi) \leftrightarrow C_B(\varphi \wedge [\varphi]\psi, [\varphi]\chi) \text{ (common knowledge reduction)}$$

as well as an inference rule of necessitation for all announcement modalities. $\square$

It turns out that PAL-RC is no more expressive than EL-RC by a direct translation, where the translation clause for $[\varphi]C_B(\psi, \chi)$ relies on the above insight:

Definition 9 (Translation from PAL-RC to EL-RC) The translation function t takes a formula from the language of PAL-RC and yields a formula in the language of EL-RC.

$$\begin{array}{lcllcl}
t(p) & = & p & t([\varphi]p) & = & t(\varphi) \to p \\
t(\neg\varphi) & = & \neg t(\varphi) & t([\varphi]\neg\psi) & = & t(\varphi) \to \neg t([\varphi]\psi) \\
t(\varphi \wedge \psi) & = & t(\varphi) \wedge t(\psi) & t([\varphi](\psi \wedge \chi)) & = & t([\varphi]\psi) \wedge t([\varphi]\chi) \\
t(\Box_a \varphi) & = & \Box_a t(\varphi) & t([\varphi]\Box_a \psi) & = & t(\varphi) \to \Box_a t([\varphi]\psi) \\
t(C_B(\varphi, \psi)) & = & C_B(t(\varphi), t(\psi)) & t([\varphi]C_B(\psi, \chi)) & = & C_B(t(\varphi) \wedge t([\varphi]\psi), t([\varphi]\chi)) \\
 & & & t([\varphi][\psi]\chi) & = & t([\varphi]t([\psi]\chi))
\end{array}$$

□

Theorem 2 (Translation Correctness) For all dynamic-epistemic formulas φ of PAL-RC and all models M, w,

$$M, w \models \varphi \text{ iff } M, w \models t(\varphi).$$

Theorem 3 ('PAL-RC = EL-RC') The languages PAL-RC and EL-RC have equal expressive power. □

Theorem 4 (Completeness for PAL-RC) $\models \varphi$ iff $\vdash \varphi$. □

Proof The proof system for EL-RC is complete (Theorem 1), and every formula in $\mathscr{L}_{\mathsf{PAL\text{-}RC}}$ is provably equivalent to one in $\mathscr{L}_{\mathsf{EL\text{-}RC}}$ (Lemma 2). □

2.6 Model Comparison Games for EL-RC

The notion of relativized common knowledge is of independent interest, just as irreducibly binary general quantifiers (such as *Most A are B*) lead to natural completions of logics with only unary quantifiers. We provide some more information through characteristic games.

Definition 10 (The EL-RC Game) Let two models $M = (W, R, V)$ and $M' = (W', R', V')$ be given. Starting from each $w \in W$ and $w' \in W'$, the *n-round* EL-RC *game* between Spoiler and Duplicator is given as follows. If $n = 0$ Spoiler wins if w and w' differ in their atomic properties, otherwise Duplicator wins. Otherwise Spoiler can initiate one of two scenarios in each round:

$\Box_a$-move Spoiler chooses a point x in one model which is an a-successor of the current w or w', and Duplicator responds with a matching successor y in the other model. The output of this move is x, y.

RC_B-**move** Spoiler chooses a B-path $x_0 \dots x_n$ in either of the models with x_0 the current w or w'. Duplicator responds with a B-path $y_0 \dots y_m$ in the other model, with $y_0 = w'$. Then Spoiler can (a) make the end points x_n, y_m the output of this round, or (b) he can choose a world z on Duplicators path, and Duplicator must respond by choosing a matching world u on Spoilers path, and z, u becomes the output.

The game continues with the new output states. If these differ in their atomic properties, Spoiler wins – otherwise, a player loses whenever he cannot perform a move while it is his turn. If Spoiler has not won after all n rounds, Duplicator wins the whole game. □

Definition 11 (Modal depth) The *modal depth* of a formula is defined by:

$$\begin{aligned}
&d(\bot) = d(p) = 1\\
&d(\neg\varphi) = d(\varphi)\\
&d(\varphi \wedge \psi) = \max(d(\varphi), d(\psi))\\
&d(\Box_a \varphi) = d(\varphi) + 1\\
&d(C_B(\varphi, \psi)) = \max(d(\varphi), d(\psi)) + 1
\end{aligned}$$

If two models M, w and M', w' have the same theory up to depth n, we write $M, w \equiv_n M', w'$. □

The following result holds for all logical languages that we use in this paper.

Lemma 2 (Propositional finiteness) For every n, up to modal depth n, there are only finitely logically non-equivalent propositions. □

Theorem 5 (Adequacy of EL-RC Games) Duplicator has a winning strategy for the n-round game from M, w, M', w' iff $M, w \equiv_n M', w'$. □

Proof The proof is by induction on n. The base case is obvious, and all inductive cases are also standard in modal logic, except that for relativized common knowledge. As usual, perspicuity is increased somewhat by using the dual existential modality $\hat{C}_B(\varphi, \psi)$. From left to right the proof is straightforward.

From right to left. Suppose that $M, w \equiv_{n+1} M', w'$. A winning strategy for Duplicator in the $(n+1)$-round game can be described as follows. If Spoiler makes an opening move of type [$\Box_a$-move], then the usual modal argument works. Next, suppose that Spoiler opens with a finite sequence in one of the models, say M, without loss of generality. By the Lemma 2, we know that there is only a finite number of complete descriptions of points up to logical depth n, and each point s in the sequence satisfies one of these: say $\Delta(s, n)$. In particular, the end point v satisfies $\Delta(v, n)$. Let $\Delta(n)$ be the disjunction of all formulas $\Delta(s, n)$ occurring on the path. Then, the initial world w satisfies the following formula of modal depth $n+1$: $\hat{C}_B(\Delta(n), \Delta(v, n))$. By our assumption, we also have $M', w' \models \hat{C}_B(\Delta(n), \Delta(v, n))$. But any sequence witnessing this by the truth definition is a response that Duplicator can use for her winning strategy. Whatever Spoiler does in the rest of this round, Duplicator always has a matching point that is n-equivalent in the language. □

Thus, games for $\mathscr{L}_{\mathsf{EL\text{-}RC}}$ are straightforward. But it is also of interest to look at the language $\mathscr{L}_{\mathsf{PAL\text{-}C}}$. Here, the shift modality $[\varphi]$ passing to definable submodels requires a new type of move, where players can decide to change the current model. The following description of what happens is 'modular': a model changing move can be added to model comparison games for ordinary epistemic logic (perhaps with common knowledge), or for our EL-RC game. By way of explanation: we let Spoiler propose a model shift. Players first discuss the 'quality' of that shift, and Duplicator can win if it is deficient; otherwise, the shift really takes place, and play continues within the new models. This involves a somewhat unusual sequential composition of games, but perhaps one of independent interest.

Definition 12 (The PAL-C Game) Let the setting be the same as for the n-round game in Definition 10. Now Spoiler can initiate one of the following scenario's each round

$\Box_a$**-move** Spoiler chooses a point x in one model which is an a-successor of the current w or w', and Duplicator responds with a matching successor y in the other model. The output of this move is x, y.

C_B**-move** Spoiler chooses a point x in one model which is reachable by a B-path from w or w', and Duplicator responds by choosing a matching world y in the other model. The output of this move is x, y.

$[\varphi]$**-move** Spoiler chooses a number $r < n$, and sets $S \subseteq W$ and $S' \subseteq W'$, with the current $w \in S$ and likewise $w' \in S'$. *Stage 1*: Duplicator chooses states s in $S \cup S'$, $\overline{s}$ in $\overline{S} \cup \overline{S'}$. Then Spoiler and Duplicator play the r-round game for these worlds. If Duplicator wins this subgame, she wins the n-round game. *Stage 2*: Otherwise, the game continues in the relativized models $M|S, w$ and $M'|S', w'$ over $n - r$ rounds. □

The definition of depth is easily extended to formulas $[\varphi]\psi$ as $d([\varphi]\psi) = d(\varphi) + d(\psi)$.

Theorem 6 (Adequacy of PAL-RC Games) Duplicator has a winning strategy for the n-round game on M, w and M', w' iff $M, w \equiv_n M', w'$ in $\mathscr{L}_{\mathsf{PAL\text{-}RC}}$. □

Proof We only discuss the inductive case demonstrating the match between announcement modalities and model-changing steps. From left to right the proof is straightforward.

From right to left. Suppose that M, w and M', w' are equivalent up to modal depth $n+1$. We need to show that Duplicator has a winning strategy. Consider any opening choice of S, S' and $r < n+1$ made by Spoiler. *Case 1*: Suppose there are two points s, and $\overline{s}$ that are equivalent up to depth r. By the induction hypothesis, this is the case if and only if Duplicator has a winning strategy for the r-round game starting from these worlds and so has a winning strategy in Stage 1. *Case 2*: Duplicator has no such winning strategy, which means that Spoiler has one – or equivalently by the inductive hypothesis, every pair $s, \overline{s}$ is distinguished by some formula $\varphi_{s\overline{s}}$ of depth at most r which is true in s and false in $\overline{s}$.

Observe that $\delta_s = \bigwedge_{\overline{s} \in \overline{S} \cup \overline{S'}} \varphi_{s\overline{s}}$ is true in s and false in $\overline{S} \cup \overline{S'}$. Note that there can be infinitely many worlds involved in the comparison, but finitely many different formulas will suffice by the Lemma 2, which also holds for this extended language. Further, the formula $\Delta_S = \bigvee_{s \in S} \delta_s$ is true in S and false in $\overline{S} \cup \overline{S'}$. A formula $\Delta_{S'}$ is found likewise, and we let Δ be $\Delta_{S'} \vee \Delta_S$. It is easy to see that Δ is of depth r and defines S in M and S' in M'. Now we use the given language equivalence between M, w and M', w' with respect to all depth $(n+1)$-formulas $\langle \Delta \rangle \psi$ where ψ runs over all formulas of depth $(n+1) - r$. We can conclude that $M|\Delta, w$ and $M'|\Delta, w'$ are equivalent up to depth $(n+1) - r$, and hence Duplicator has a winning strategy for the remaining game, by the inductive hypothesis. So in this case Duplicator has a winning strategy in Stage 2. □

In the next section we will use this game to show that EL-RC is more expressive than PAL-C. Now we will give an example of how this game can be played.

Definition 13 Let the model $M(n) = (W, R, V)$ be defined by

- $W = \{x \in \mathbb{N} \mid 0 \leq x \leq n\}$
- $R = \{(x, (x-1)) \mid 1 \leq x \leq n\}$
- $V(p) = W$ □

These models are simply lines of worlds. They can all be seen as submodels of the entire line of natural numbers (where $W = \mathbb{N}$). The idea is that Spoiler cannot distinguish two of these models if the line is long enough. The only hope that Spoiler has is to force one of the current worlds to an endpoint and the other not to be an endpoint. In that case Spoiler can make a $\Box$-move in the world that is not an endpoint and Duplicator is stuck. This will not succeed if the lines are long enough. Note that a C-move does not help Spoiler. Also a $[\varphi]$-move will not help Spoiler. Such a move will shorten the lines, but that will cost as many rounds as it shortens them, so Spoiler still loses if they are long enough. The following Lemma captures this idea.

Lemma 3 For all m, n, and all $x \leq m$ and $y \leq n$ Duplicator has a winning for the PAL-C game for $M(m), x$ and $M(n), y$ with at most $\min(x, y)$ rounds. □

Proof If $x = y$, the proof is trivial. We proceed by induction on the number of rounds. Suppose the number of rounds is 0. Then x and y only have to agree on propositional variables. They must agree, since p is true everywhere.

Suppose that the number of rounds is $k+1$ (i.e. $\min(x, y) = k+1$). Duplicator's strategy is the following. If Spoiler chooses to play a $\Box$-move, he moves to $x-1$ (or to $y-1$). Duplicator responds by choosing $y-1$ (or $x-1$). Duplicator has a winning strategy for the resulting subgame by the induction hypothesis.

Suppose Spoiler chooses to play a C-move. If Spoiler chooses a $z \leq \min(x, y)$, then Duplicator also chooses z. Otherwise, Duplicator does not move at all. Duplicator has a winning strategy for the resulting subgame by the induction hypothesis.

Suppose Spoiler chooses to play a $[\varphi]$-move. Spoiler chooses a number of rounds r and some S and S'. Observe that for all $z < \min(x, y)$ it must be the case that $z \in S$ iff $z \in S'$. Otherwise, Duplicator has a winning strategy by the induction hypothesis by choosing z and z. Moreover for all $z \geq r$ it must be the case that $z \in S \cup S'$. Otherwise, Duplicator has a winning strategy by the induction hypothesis for $\min(x, y)$ and z. In Stage 2 the resulting subgame will be for two models bisimilar to models to which the induction hypothesis applies. The number of rounds will be $(k+1) - r$, and the lines will be at least $\min(x, y) - r$ long (and $(\min(x, y) = k+1)$. This situation is sketched in Figure 1. □

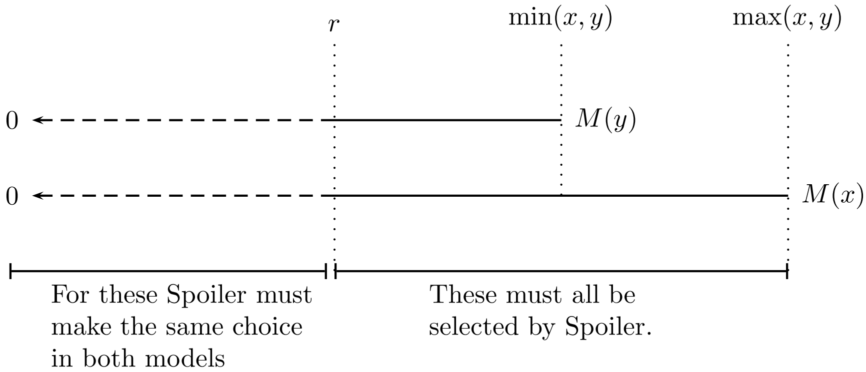


Figure 1: Illustration of the proof of Lemma 3.

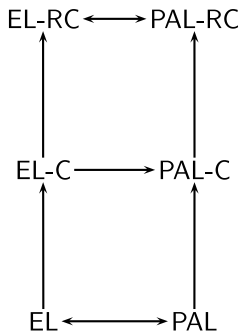


Figure 2: Expressive power of static and dynamic epistemic logics.

2.7 Expressivity Results

In this section we investigate the expressive power of the various logics under consideration here. Reduction axioms and the accompanying translation tell us that two logics have equal expressive power. But our inability to find a compositional translation from one logic to another does not imply that those logics have different expressive power. Here are some known facts. Epistemic logic with common knowledge is more expressive than epistemic logic without common knowledge. In [1] it was shown that public announcement logic with common knowledge is more expressive than epistemic logic with common knowledge. This can also be shown using the results on PDL without tests in [16]. In this section we show that relativized common knowledge logic is more expressive than public announcement logic with common knowledge. The landscape of expressive power is summarized in Figure 2. All arrows are strict.

In general one logic L is more expressive than another logic (L' ($\mathsf{L} \longrightarrow \mathsf{L}'$ in Figure 2) if there is a formula in the language of L which is not equivalent to any formula in the language of L' (and every formula in the language of L' is equivalent to some formula in the language of L). So, in order to show that $\mathsf{EL\text{-}RC}$ is more expressive than $\mathsf{PAL\text{-}C}$ we need to find a formula in $\mathscr{L}_{\mathsf{EL\text{-}RC}}$ which is not equivalent to any formula in $\mathscr{L}_{\mathsf{PAL\text{-}C}}$. The formula

$$C(p, \neg\Box p)$$

fits this purpose. This will be shown in Theorem 7.

We can show that this formula cannot be expressed in $\mathscr{L}_{\mathsf{PAL\text{-}C}}$ by using model comparison games.

We will show that for any number of rounds there are two models such that Duplicator has a winning strategy for the model comparison game, but $C(p, \neg\Box p)$ is true in one of these models and false in the other.

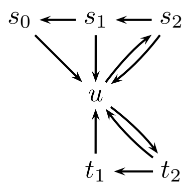
In definition 15 we provide the models that EL-RC can distinguish, but PAL-C cannot. Since the model comparison game for PAL-C contains the $[\varphi]$-move we also need to prove that the relevant submodels cannot be distinguished by PAL-C. We deal with these submodels first in the next definition and lemma.

Consider the following set of models.

Definition 14 Let the model $M(m, n) = (W, R, V)$ where $0 < n \leq m$ be defined by

- $W = \{s_x \mid 0 \leq x \leq m\} \cup \{t_x \mid n \leq x \leq m\} \cup \{u\}$
- $R = \{(s_x, s_{x-1}) \mid 1 \leq x \leq m\} \cup \{(t_x, t_{x-1}) \mid n+1 \leq x \leq m\} \cup \{(w, u) \mid w \in W \setminus \{u\}\}$
- $V(p) = W \setminus \{u\}$ □

The picture below represents $M(2, 1)$.



Let us call these models hourglasses. The idea is that Spoiler cannot distinguish the top line from the bottom line of these models if they are long enough. Note that apart from u this model consists of two lines. So if Spoiler plays $\Box$-moves on these lines, Duplicator's strategy is the same as for the line models described above. If he moves to u, Duplicator also moves to u, and surely Duplicator cannot lose the subsequent game in that case. In these models a C-move is very bad for Spoiler, since all worlds are connected by the reflexive transitive closure of R. A $[\varphi]$-move will either yield two lines which are too long, or it will be a smaller hourglass model, which will still be too large, since the $[\varphi]$-move reduces the number of available moves. The Lemma below captures this idea.

In the Lemma below we refer to w_x as a variable ranging over s_x and t_x. And if t_x does not exist it refers to s_x.

Lemma 4 For all m, n and all $x \leq m$ and $y \leq m$ Duplicator has a winning strategy for the public announcement game for $M(m, n), w_x$ and $M(m, n), w_y$ with at most $\min(x, y) - m$ rounds. □

Proof We prove the case when $w_x = s_x$ and $w_y = t_y$ (the other cases are completely analogous). By induction on the number of rounds. Suppose the number of rounds is 0. Then s_x and t_y only have to agree on propositional variables. They do agree, since p is true in both.

Suppose that the number of rounds is $k + 1$. Duplicator's winning strategy is the following. If Spoiler chooses to play a $\Box$-move, he moves to s_{x-1} (or to t_{y-1}), or to u. In the last case Duplicator responds by also choosing u, and has a winning strategy for the resulting subgame. Otherwise Duplicator moves to the t_{y-1} (or s_{x-1}). Duplicator has a winning strategy for the resulting subgame by the induction hypothesis.

Suppose Spoiler chooses to play a C-move. If Spoiler moves to w, then Duplicator moves to the same w, and has a winning strategy for the resulting subgame.

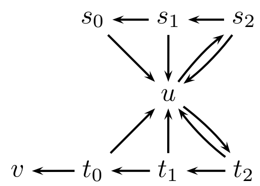
Suppose Spoiler chooses to play a $[\varphi]$-move. Spoiler chooses a number of rounds r and some S. Since there is only one model, Spoiler only chooses one subset of W. Moreover for all $z \geq \min(x, y) - m - r$ it must be the case that $w_z \in S$. Otherwise, Duplicator has a winning strategy by the induction hypothesis for s_x and w_z. In Stage 2 the result will be two models bisimilar to models to which the induction hypothesis applies, or to which Lemma 3 applies. □

Lastly consider the following class of models.

Definition 15 Let the model $M(m, n) = (W, R, V)$ where $0 < n \leq m$ be defined by

- $W = \{s_x \mid n \leq x \leq m\} \cup \{t_x \mid 0 \leq x \leq m\} \cup \{v, u\}$
- $R = \{(s_x, s_{x-1}) \mid n+1 \leq x \leq m\} \cup \{(t_x, t_{x-1}) \mid 1 \leq x \leq m\} \cup$
  $\{(w, u) \mid w \in W \setminus \{v, u\}\} \cup \{(t_0, v)\}$
- $V(p) = W \setminus \{u\}$ □

The picture below represents $M(2, 0)$.



In these 'hourglasses with an appendage', the idea is that Duplicator cannot distinguish the top line from the bottom line of these models when they are long enough. Apart from v the model is just like a hourglass. So the only new option for Spoiler is to force one of the current worlds to v, and the other to another world. Then Spoiler chooses a $\Box$-move and takes a step from the non-v world and Duplicator is stuck at v. However if the model is large enough v is too far away. Again a C-move does not help Spoiler, because it can be matched exactly by Duplicator. Reducing the model with a $[\varphi]$-move will yield either a hourglass (with or without an appendage) or two lines, for which Spoiler does not have a winning strategy. The idea leads to the following Lemma.

Lemma 5 For all m, n and all $x \leq m$ and $y \leq m$ Duplicator has a winning strategy for the public announcement game for $M(m, n), w_x$ and $M(m, n), w_y$ with at most $\min(x, y) - m$ rounds. □

Proof The proof is analogous to the proof of Lemma 4. □

This Lemma leads to the following theorem.

Theorem 7 EL-RC is more expressive than PAL-C. □

Proof Suppose PAL-C is just as expressive as EL-RC. Then there is a formula $\varphi \in \mathscr{L}_{\textsf{PAL-C}}$ with $\varphi \equiv C(p, \neg\Box p)$. Suppose $d(\varphi) = n$. In that case we would have $M(n, n), s_n \models \varphi$ and $M(n, n), t_n \not\models \varphi$, contradicting Lemma 5. Hence, EL-RC is more expressive. □

2.8 Complexity Results

Update logics are about processes that manipulate information, and hence they raise natural questions of *complexity*, as the counterpoint to the above feature of expressive power. In particular, all of the usual complexity questions concerning a logical system make sense. *Model checking* asks where a given formula is true in a model, and this is obviously crucial to computing updates. *Satisfiability testing* asks when a given formula has a model, which corresponds to consistency of conversational scenarios in our dynamic epistemic setting. Or, stating the issue in terms of *validity*: when will a given epistemic update always produce some global specified effect? Finally, just as in basic modal logic, there is a non-trivial issue of *model comparison*: when do two given models satisfy the same formulas in our language, i.e., when are two group information states 'the same' for our purposes? As usual, this is related with checking for *bisimulation*, or in a more finely-grained version, the existence of winning strategies for Duplicator in the above model comparison games.

Now technically, the translation of definition 9 combined with known algorithms for model checking, satisfiability, validity, or model comparison for epistemic logic yield similar algorithms for public announcement logic. But, in a worst case, the length of the translation of a formula is exponential in the length of the formula. E.g., the translation of φ occurs three times in that of $[\varphi]C_B(\psi, \chi)$. Therefore, a direct complexity analysis is worth-while. We provide two results plus some references.

Lemma 6 Deciding whether a finite model M, w satisfies a formula $\varphi \in \mathscr{L}_{\mathsf{EL\text{-}RC}}$ is computable in polynomial time in the length of φ and the size of M. □

Proof The argument is an easy adaptation of the usual proof for PDL. □

This algorithm does not suffice for the case with public announcements. The truth values of φ and ψ in the given model do not fix that of $[\varphi]\psi$. We must also know the value of ψ in the model restricted to φ worlds.

Lemma 7 Deciding whether a finite model M, w satisfies a formula $\varphi \in \mathscr{L}_{\mathsf{PAL\text{-}RC}}$ is computable in polynomial time in the length of φ and the size of M. □

Proof Again there are at most $|\,\varphi\,|$ subformulas of φ. Now we make a binary tree of these formulas which splits with formulas of the form $[\psi]\chi$. On the left subtree all subformulas of ψ occur, on the right all those of χ. This tree can be constructed in time $\mathcal{O}(|\,\varphi\,|)$. Labeling the model is done by processing this tree from bottom to top from left to right. The only new case is when we encounter a formula $[\psi]\chi$. In that case we have already processed the left subtree for ψ. Now we first label those worlds where ψ does not hold as worlds where $[\psi]\chi$ holds, then we process the right subtree under $[\psi]\chi$ where we restrict the model to worlds labeled as ψ-worlds. After this process we label those worlds that were labeled with χ as worlds where $[\psi]\chi$ holds and the remaining as worlds where it does not hold. We can see by induction on formula complexity that this algorithm is correct.

Also by induction on φ, this algorithm takes time $\mathcal{O}(|\,\varphi\,| \times \|M\|^2)$. The only difficult step is labeling the model with $[\psi]\chi$. By the induction hypothesis, restricting the model to ψ takes time $\mathcal{O}(|\,\psi\,| \times \|M\|^2)$. We simply remove (temporarily) all worlds labelled $\neg\varphi$ and all arrows pointing to such worlds. Again by the induction hypothesis, checking χ in this new model takes $\mathcal{O}(|\,\psi\,| \times \|M\|^2)$ steps. The rest of the process takes $\|M\|$ steps. So, this step takes over-all time $\mathcal{O}(|\,[\psi]\chi\,| \times \|M\|^2)$. □

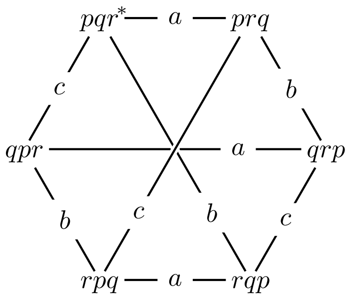
Moving on from model checking, the satisfiability and the validity problem of epistemic logic with common knowledge are both known to be EXPTIME-complete. In fact, this is true for almost any logic that contains a transitive closure modality. Satisfiability and validity for PDL are also EXPTIME-complete. Now there is a linear time translation of the language of EL-RC to that of PDL. Therefore the satisfiability and validity problems for EL-RC are also EXPTIME-complete. For PAL-RC and even PAL-C, however, the complexity of satisfiability and validity is not settled by this. Lutz [21] shows that satisfiability in PAL is PSPACE-complete, using a polynomial-time translation from dynamic-epistemic to purely epistemic formulas. The latter is unlike the translation underpinning our reduction axioms, in that it is not meaning-preserving. The same method probably extends to PAL-C and PAL-RC.

Finally, the complexity of model comparison for finite models is the same as that for ordinary epistemic logic, viz. PTIME. The reason is that even basic modal equivalence on finite models implies the existence of a bisimulation, while all our extended languages are bisimulation-invariant.

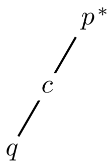
3 Modelling Effects of Communication and Change

Epistemic update logics are about the effects of general communication, and even any kind of information-bearing event. But in practice, it is helpful to look at more constrained scenarios. A good source of examples are basic actions in *card games*. Game moves then involve looking at a card, showing a card to someone with other players looking on), exchanging cards with someone (with other players looking on), and so on (cf. [8]), and perhaps even changing the setting in more drastic ways. In this section we list examples involving combinations of epistemic and actual change that we think any full-fledged dynamic-epistemic logic should be able to deal with. This requires updates of the initial information model which are much more involved than elimination of worlds. Some of them can even make the current model bigger, as alternatives can multiply. Since all these actions involve groups of agents, reasoning about the common knowledge resulting from them is again essential. In Section 4 we propose a new logic LCC that can deal with the relevant richer class of updates, as well as arbitrary postconditions involving common knowledge, whether pure or relativized.

Card Showing A simple card showing situation goes as follows. Alice, Bob and Carol each hold one of cards p q, r. The actual deal is: Alice holds p, Bob holds q, Carol holds r. Assuming that all players have looked at their own cards, but have kept them hidden from the others, this situation is modelled as follows (xyz represents the situation where Alice holds x, Bob holds y and Carol holds z, xyz—a—$x'y'z'$ represents the fact that Alice cannot distinguish xyz from $x'y'z'$, and xyz^* indicates that xyz is the situation that actually is the case):

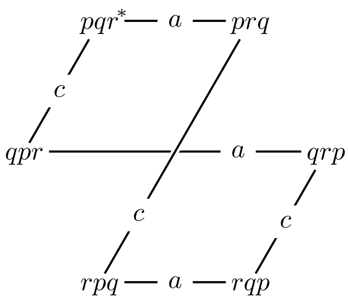


Now here is a major further idea, due to [1]. The action of Alice showing her card to Bob, with Carol looking on (Carol sees that a card is shown, but does not see which card), can itself be pictured as an *update model* whose structure is similar to that of epistemic models in general:



The card that Alice shows to Bob is card p, but for all Carol knows, it might have been q (it cannot be r, as that is the card Carol holds and has inspected).

Intuitively, we now want a new information model arising from the initial one that the agents were in plus the update model containing the relevant actions. Moreover, our intuitions tell us what the desired result of this update should look like, viz.:



Card Inspection Next, suppose the three cards are dealt to Alice, Bob and Carol, but are still face down on the table. The following picture describes an update model for the action of Alice inspecting her own card and discovering it to be p, with the others merely looking on.

$$\begin{tikzcd} & p_{--}^{*} \arrow[dl, dash, "bc"'] \arrow[dr, dash, "bc"] & \\ r_{--} \arrow[rr, dash, "bc"'] & & q_{--} \end{tikzcd}$$

And here is the related update model of Alice picking up her card and showing it to the others, without taking a look herself:

$$\begin{tikzcd} & p_{--}^{*} \arrow[dl, dash, "a"'] \arrow[dr, dash, "a"] & \\ r_{--} \arrow[rr, dash, "a"'] & & q_{--} \end{tikzcd}$$

In all these cases we have clear intuitions about what the outcomes of the updates should be.

Card Exchange Now we add a case where real physical action takes place, which is not of a purely informational nature. Suppose in the initial situation, where Alice holds p, Bob holds q and Carol holds r, Alice and Bob exchange cards, without showing the cards to Carol. To model this, we need an update that also changes the state of the world. For that, we need to change the valuation for atomic facts. This may be done using 'substitutions' as follows:

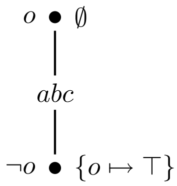
$$\begin{tikzcd} pq_{-} \bullet \; \{pq_{-} \mapsto qp_{-}\} \arrow[d, dash, "c"] \\ qp_{-} \bullet \; \{qp_{-} \mapsto pq_{-}\} \end{tikzcd}$$

Note that the diagram now indicates both 'preconditions' and 'postconditions' for successful actions or events. The result of this update model in the initial situation, where all players have looked at their cards and the actual deal is pqr:

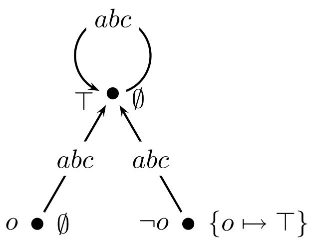
$$\begin{tikzcd} & qpr^{*} \arrow[dl, dash, "c"'] \\ pqr & \end{tikzcd}$$

These were card examples, where actions are meant to be communicative, with some intentional agent. But it is important to realize that dynamic epistemic logic can also serve as a system for analyzing *observations* of events that carry some information to observers, whether intended or not. That is, it is a logic of perception as much as of communication, and it is useful for modelling the essence of what goes on in very common everyday actions. We conclude with two illustrations in the latter mode.

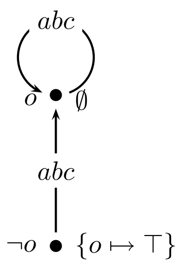
Opening a window The precondition for opening a window is that the window is closed. To make this into an update that can be performed no matter what, we must specify different actions depending on whether the window is open or not: if it is open, then nothing needs to be done. If it is closed, then open it. Assuming the update is invisible to all (Alice, Bob and Carol), it is modelled as follows:



Again, we see both pre- and postconditions appear, with postconditions depending on the preconditions. Note that the effect of this action can be that agents get 'out of touch with reality.' Getting out of touch with reality can also be the result of being misled. If the window is opened in secret, its update model looks as follows (o denotes an open window; $o \mapsto \top$ is the substitution that makes o true).



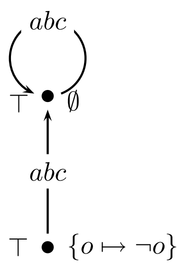
Further variations on this update model are possible. E.g., the window is in fact opened, while everyone is told that it was already open. Here is the corresponding update:



Fiddling with a light switch Fiddling with a light switch is an update that depends on the actual situation as follows: if the light is on, then switch it off, if it is off, then switch it on. If this fiddling is done in a way such that the result is visible to all, then here is its update model:

$$\top \bullet \{o \mapsto \neg o\}$$

If the fiddling (and its result) is done in secret, its update model looks as follows:



Remark The examples may suggest that we can view an update model with several actions as a kind of disjunction of actions under conditions, i.e., as a kind of program 'if the window is open then do A or if the window is ajar then do B or if the window is wide open then do C ....' This may lead to a set-up where action models are built from simple actions by means of choice, sequence and iteration (the regular operations). We will not adopt this approach, however, as it is known to result in ill-behaved action models. Miller and Moss [24] show that adding iteration to PAL without common knowledge already leads to an undecidable logic.

4 A New Logic of Communication and Change

The examples in the preceding section set a higher ambition level for a dynamic epistemic logic updating with events involving combinations of communication and actual change. Systems of this general sort were proposed in [1, 3, 2], but without reduction axioms for common knowledge. We now proceed to a version which can deal with common knowledge, generalizing the methodology for epistemic logic with announcements from Section 2.

In Section 4.1 we introduce *update models* (U, e) and specify their execution on epistemic models. In Section 4.2 we review propositional dynamic logic (PDL) under its epistemic/doxastic interpretation, written henceforth as E-PDL. In Section 4.3 we then present our dynamic epistemic *logic of communication and change* LCC as an extension of E-PDL with dynamic modalities $[\mathsf{U}, \mathsf{e}]$ for update models. In Section 4.4 we show that the full system LCC can be reduced to epistemic E-PDL, and in Section 4.5 we present a proof system for LCC in terms of perspicuous reduction axioms based on this insight [11, 10].

From a technical perspective, the proofs to follow are not just simple generalizations of those for public announcements. In [19] a correspondence between update models and *finite automata* is used to obtain reduction axioms in a dynamic epistemic logic based on so-called 'automata PDL', a variant of E-PDL. But this can be avoided by analyzing the automata inductively inside E-PDL itself [11], using the well-known proof of Kleene's theorem, stating that languages generated by nondeterministic finite automata are regular [18]. The main theorems to follow use an inductive 'program transformation' approach to epistemic updates whose structure resembles that of Kleene's translation from finite automata to regular expressions. This technical method may be of independent interest.

4.1 Update Models and their Execution

Dynamic *updates* with epistemic aspects, such as communication or other information-bearing events, are quite similar to static epistemic *situations*. In [1] this analogy is used as the engine for general update of epistemic models under epistemic actions. In particular, individual events come with *preconditions* holding only at those worlds where they can occur. Here we extend this with 'substitutions' that effect changes in valuations at particular worlds, representing real change as a result of the event. In other words, substitutions may be viewed as *postconditions* for these events to occur.

Definition 16 (Substitutions) $\mathscr{L}$ substitutions are functions of type $\mathscr{L} \to \mathscr{L}$ that distribute over all language constructs, and that map all but a finite number of basic propositions to themselves. $\mathscr{L}$ substitutions can be represented as sets of bindings

$$\{p_1 \mapsto \varphi_1, \dots, p_n \mapsto \varphi_n\}$$

where all the p_i are different . If σ is a $\mathscr{L}$ substitution, then the set $\{p \in P \mid \sigma(p) \neq p\}$ is called its *domain*, notation $\text{dom}(\sigma)$. Use ϵ for the identity substitution. Let $\text{SUB}_{\mathscr{L}}$ be the set of all $\mathscr{L}$ substitutions. □

Definition 17 (Epistemic Models under a Substitution) If $M = (W, V, R)$ is an epistemic model and σ is a $\mathscr{L}$ substitution (for an appropriate epistemic language $\mathscr{L}$), then V_M^σ is the valuation given by $\lambda w \lambda p \cdot w \in [\![\sigma(p)]\!]^M$. In other words, V_M^σ assigns to w the set of basic propositions p such that $\sigma(p)$ is true in world w in model M. For $M = (W, V, R)$, call M^σ the model given by (W, V_M^σ, R). □

Definition 18 (Update models) An update model for a finite set of agents N with a language $\mathscr{L}$ is a quadruple $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{sub})$ where

- $\mathsf{E} = \{\mathsf{e}_0, \dots, \mathsf{e}_{n-1}\}$ is a finite non-empty set of events,
- $\mathsf{R} : N \to \wp(\mathsf{E}^2)$ assigns an accessibility relation $\mathsf{R}(a)$ to each agent $a \in N$.
- $\mathsf{pre} : \mathsf{E} \to \mathscr{L}$ assigns a precondition to each event,
- $\mathsf{sub} : \mathsf{E} \to \mathrm{SUB}_{\mathscr{L}}$ assigns a $\mathscr{L}$ substitution to each event.

A pair U, e is an update model with a distinguished actual event $\mathsf{e} \in \mathsf{E}$. □

Here $\mathscr{L}$ can be any language that can be interpreted in the models of definition 1. Note that an 'action model' in the sense of [1] is a special case of an update model in our sense, where sub assigns the identity substitution ϵ to every event. Our substitutions are a natural way of taking the original action model philosophy one step further: events can have both preconditions plus *postconditions*. Later on, this allows us, in update execution, to naturally reset both basic features of information models: the epistemic accessibility relations, and the propositional valuation.

Definition 19 (Update bisimulation) An update bisimulation is like an ordinary bisimulation, except for the fact that the requirement of 'same valuations' is replaced by a requirement of 'equivalent preconditions and equivalent substitutions'. □

The effect of executing an update is modeled by the following *product construction*; our definition extends that of [1] by taking the effects of the substitutions into account.

Definition 20 (Update Execution) Given a static epistemic model $M = (W, R, V)$, a world $w \in W$, an action model $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{sub})$ and an action state $\mathsf{e} \in \mathsf{E}$ with $M, w \models \mathsf{pre}(\mathsf{e})$, we say that the result of *executing* U, e *in* M, w is the model $M \circ \mathsf{U}, (w, \mathsf{e}) = (W', R', V'), (w, \mathsf{e})$ where

- $W' = \{(v, \mathsf{f}) \mid M, v \models \mathsf{pre}(\mathsf{f})\}$,
- $R'(a) = \{((v, \mathsf{f}), (u, \mathsf{g})) \mid (v, u) \in R(a) \text{ and } (\mathsf{f}, \mathsf{g}) \in \mathsf{R}(a)\}$,
- $V'(p) = \{(v, \mathsf{f}) \mid M, v \models \mathsf{sub}(\mathsf{f})(p)\}$ □

Definitions 18 (with all substitutions equal to ϵ) and 20 provide a semantics for the logic of epistemic actions LEA of [1]. The basic epistemic language $\mathscr{L}_{\mathsf{LEA}}$ is extended with dynamic modalities $[\mathsf{U}, \mathsf{e}]\varphi$, where a U is any *finite update model* for $\mathscr{L}_{\mathsf{LEA}}$. These say that 'every execution of U, e yields a model where φ holds':

$$M, w \models [\mathsf{U}, \mathsf{e}]\varphi \quad \text{iff} \quad M, w \models \mathsf{pre}(\mathsf{e}) \text{ implies that } M \circ \mathsf{U}, (w, \mathsf{e}) \models \varphi$$

In [1] a proof system for LEA is presented with a complicated completeness proof, and without reduction axioms for common knowledge (which were already lacking for public announcement updates). So, we will extend this language to get reduction axioms after all.

Again, the semantic intuition about the crucial case $M, w \models [\mathsf{U}, \mathsf{e}]C_B\varphi$ is clear. It says that, if there is a B-path $w_0, \dots, w_n$ (with $w_0 = w$) in the static model and a matching B-path $\mathsf{e}_0, \dots, \mathsf{e}_n$ (with $\mathsf{e}_0 = \mathsf{e}$) in the update model with $M, w_i \models \mathsf{pre}(\mathsf{e}_i)$ for all $i \leq n$, then $M, w_n \models \varphi$. To express all this in the initial static model, it turns out to be convenient to choose a representation of complex epistemic assertions that meshes well with update models. Now, the relevant finite paths in static models involve strings of agent accessibility steps and tests on formulas. These are the sort of object that *propositional dynamic logic* (PDL) is very well suited for.

4.2 Epistemic PDL

The language of propositional dynamic logic and all further information about its semantics and proof theory may be found in [17], which also has references to the history of this calculus, and its original motivations in computer science.

Definition 21 (PDL, Language) Let a set of propositional variables P and a set of relational atoms N be given, with p ranging over P and a over N. The language of PDL is given by:

$$\begin{array}{lcl} \varphi & ::= & \top \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid [\pi]\varphi \\ \pi & ::= & a \mid ?\varphi \mid \pi_1;\pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^* \end{array} \qquad \square$$

In our epistemic perspective, relational atoms will be viewed as (the epistemic accessibilities of) single agents. We employ the usual abbreviations: $\perp$ is shorthand for $\neg\top$, $\varphi_1 \vee \varphi_2$ is shorthand for $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \rightarrow \varphi_2$ is shorthand for $\neg(\varphi_1 \wedge \varphi_2)$, $\varphi_1 \leftrightarrow \varphi_2$ is shorthand for $(\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$, and $\langle\pi\rangle\varphi$ is shorthand for $\neg[\pi]\neg\varphi$. Also, if $B \subseteq N$ and B is finite, use B as shorthand for $b_1 \cup b_2 \cup \cdots$. Under this convention, the general knowledge operator $E_B\varphi$ takes the shape $[B]\varphi$, while the common knowledge operator $C_B\varphi$ appears as $[B^*]\varphi$, i.e., $[B]\varphi$ expresses that it is general knowledge among agents B that φ, and $[B^*]\varphi$ expresses that it is common knowledge among agents B that φ. In the special case where $B = \emptyset$, B turns out equivalent to $?\perp$, the program that always fails. Note that common *belief* among agents B that φ can be expressed as $[B; B^*]\varphi$.

Definition 22 (PDL, Semantics) The semantics of PDL over P, N is given relative to Kripke models $M = (W, R, V)$ for signature P, N. The formulas of PDL are interpreted as subsets of W, the relational atoms a of PDL as binary relations on W (with the interpretation of relational atoms a given as $R(a)$), as follows:

$$\begin{array}{rcl} [\![\top]\!]^M & = & W \\ [\![p]\!]^M & = & \{w \in W \mid p \in V(w)\} \\ [\![\neg\varphi]\!]^M & = & W \setminus [\![\varphi]\!]^M \\ [\![\varphi_1 \wedge \varphi_2]\!]^M & = & [\![\varphi_1]\!]^M \cap [\![\varphi_2]\!]^M \\ [\![\,[\pi]\varphi]\!]^M & = & \{w \in W \mid \forall v \text{ if } (w,v) \in [\![\pi]\!]^M \text{ the n } v \in [\![\varphi]\!]^M\} \\ \\ [\![a]\!]^M & = & R(a) \\ [\![?\varphi]\!]^M & = & \{(w,w) \in W \times W \mid w \in [\![\varphi]\!]^M\} \\ [\![\pi_1;\pi_2]\!]^M & = & [\![\pi_1]\!]^M \circ [\![\pi_2]\!]^M \\ [\![\pi_1 \cup \pi_2]\!]^M & = & [\![\pi_1]\!]^M \cup [\![\pi_2]\!]^M \\ [\![\pi^*]\!]^M & = & ([\![\pi]\!]^M)^* \end{array} \qquad \square$$

Here $([\![\pi]\!]^M)^*$ is the reflexive transitive closure of binary relation $[\![\pi]\!]^M$.

If $w \in W$ then we use $M, w \models \varphi$ for $w \in [\![\varphi]\!]^M$, and we say that φ is true at w. A PDL formula φ is *true* in a model if it holds at every state in that model, i.e., if $[\![\varphi]\!]^M = W_M$.

These definitions specify how formulas of PDL can be used to make assertions about PDL models. The formula $\langle a\rangle\top$, when interpreted at some state in a PDL model, expresses that that state has a successor in the $R(a)$ relation in that model. Truth of the formula $\langle a\rangle\top$ in a model expresses that $R(a)$ is serial in that model.

Note that ? is an operation for mapping formulas to programs. Programs of the form $?\varphi$ are called *tests*; they are interpreted as the identity relation, restricted to the states s satisfying the formula φ.

If $\sigma = \{p_1 \mapsto \varphi_1, \dots, p_n \mapsto \varphi_n\}$ is a PDL substitution, we use φ^σ for $\sigma(\varphi)$ and π^σ for $\sigma(\pi)$. We can spell out φ^σ and π^σ, as follows:

$$\begin{array}{lllllll}
\top^\sigma & = & \top & a^\sigma & = & a \\
p^\sigma & = & \sigma(p) & (?\varphi)^\sigma & = & ?\varphi^\sigma \\
(\neg\varphi)^\sigma & = & \neg\varphi^\sigma & (\pi_1;\pi_2)^\sigma & = & \pi_1^\sigma;\pi_2^\sigma \\
(\varphi_1 \wedge \varphi_2)^\sigma & = & \varphi_1^\sigma \wedge \varphi_2^\sigma & (\pi_1 \cup \pi_2)^\sigma & = & \pi_1^\sigma \cup \pi_2^\sigma \\
([\pi]\varphi)^\sigma & = & [\pi^\sigma]\varphi^\sigma & (\pi^*)^\sigma & = & (\pi^\sigma)^*.
\end{array}$$

The following holds by simultaneous induction on the structure of formulas and programs:

Lemma 8 (Substitution) For all PDL models M, all PDL formulas σ, all PDL programs π, all PDL substitutions σ:

$$M, w \models \varphi^\sigma \text{ iff } M^\sigma, w \models \varphi.$$

$$(w, w') \in [\![\pi^\sigma]\!]^M \text{ iff } (w, w') \in [\![\pi]\!]^{M^\sigma}.$$

This is the just the beginning of a more general model-theory for PDL, which is bisimulation-based just like basic modal logic.

4.3 LCC, a Dynamic Logic of Communication and Change

Now we have all the ingredients for the definition of the logic of communication and change.

Definition 23 (LCC, Language) The language $\mathscr{L}_{\mathsf{LCC}}$ is the result of adding a clause $[\mathsf{U}, \mathsf{e}]\varphi$ for update execution to the language of PDL, where U is an update model for $\mathscr{L}_{\mathsf{LCC}}$. □

Definition 24 (LCC, Semantics) The semantics $[\![\varphi]\!]^M$ is the standard semantics of PDL, with the meaning of $[\mathsf{U}, \mathsf{e}]\varphi$ in $M = (W, R, V)$ given by:

$$[\![[\mathsf{U}, \mathsf{e}]\varphi]\!]^M = \{w \in W \mid \text{if } M, w \models \mathsf{pre}(\mathsf{e}) \text{ then } (w, \mathsf{e}) \in [\![\varphi]\!]^{M \circ \mathsf{U}}\}.$$

□

We have to check that the definition of execution of update models is well behaved. The following theorems state that it is, in the sense that it preserves epistemic model bisimulation and update model bisimulation (the corresponding theorems for LEA are proved in [1]).

Theorem 8 For all PDL models M, w and N, v and all formulas $\varphi \in \mathscr{L}_{\mathsf{LCC}}$

$$\text{If } M, w \leftrightarrows N, v \text{ then } w \in [\![\varphi]\!]^M \text{ iff } v \in [\![\varphi]\!]^N$$

□

This theorem must be proved simultaneously with the following theorem.

Theorem 9 For all PDL models M, w and N, v, all update models U, e:

$$\text{If } M, w \leftrightarrows N, v \text{ then } M \circ \mathsf{U}, (w, \mathsf{e}) \leftrightarrows N \circ \mathsf{U}, (v, \mathsf{U}).$$

□

Proof We prove both theorems simultaneously by induction on φ and the preconditions of the update models.

- proof of Theorem 8: the base case for propositional variables and the cases for negation, conjunction, and program modalities is standard. The only interesting case is for formulas of the form $[\mathsf{U},\mathsf{e}]\varphi$. Suppose $w \in [\![[\mathsf{U},\mathsf{e}]\varphi]\!]^M$. Therefore $w \in [\![\mathsf{pre}(\mathsf{e})]\!]^M$ implies $(w,\mathsf{e}) \in [\![\varphi]\!]^{M\circ\mathsf{U}}$. By the induction hypothesis $w \in [\![\mathsf{pre}(\mathsf{e})]\!]^M$ iff $v \in [\![\mathsf{pre}(\mathsf{e})]\!]^N$ and $M \circ \mathsf{U}, (w,\mathsf{e}) \leftrightarrows N \circ \mathsf{U}, (v,\mathsf{e})$. Therefore, by applying the induction hypothesis to $M \circ \mathsf{U}, (w, a)$ and $N \circ \mathsf{U}, (v,\mathsf{f})$, we infer $v \in [\![\mathsf{pre}(\mathsf{e}]\!]^N$ implies $(v,\mathsf{e}) \in [\![\varphi]\!]^{N\circ\mathsf{U}}$. By the semantics this is equivalent to $v \in [\![[\mathsf{U},\mathsf{e}]\varphi]\!]^N$. The other way around is completely analogous.

- proof of Theorem 9: Let R be a bisimulation witnessing $M, w \leftrightarrows N, v$. Then the relation C between $W_M \times \mathsf{E}_\mathsf{U}$ and $W_N \times \mathsf{E}_\mathsf{U}$ given by

  $$(w,\mathsf{e})C(v,\mathsf{f}) \text{ iff } wRv \text{ and } \mathsf{e} = \mathsf{f}$$

  is a bisimulation.

  The induction hypothesis guarantees that (w,e) exists iff (v,f) exists.

  Suppose $(w,\mathsf{e})C(v,\mathsf{f})$. Then wRv and $\mathsf{e} = \mathsf{f}$. The only non-trivial check is the check for sameness of valuation. By wRv, w and v satisfy $V_M(w) = V_N(v)$. By $\mathsf{e} = \mathsf{f}$, e and f have the same substitution σ. By the fact that w and v are bisimilar, by the induction hypothesis we have that $w \in [\![\varphi]\!]^M$ iff $v \in [\![\varphi]\!]^N$. Thus, by $V_M(w) = V_N(s)$ and the definition of V_M^σ and V_N^σ, we get $V_M^\sigma(w) = V_N^\sigma(v)$.

□

Theorem 10 For all PDL models M, w, all update models U_1, e and U_2, f:

$$\text{If } \mathsf{U}_1, \mathsf{e} \leftrightarrows \mathsf{U}_2, \mathsf{f} \text{ then } M \circ \mathsf{U}_1, (w,\mathsf{e}) \leftrightarrows M \circ \mathsf{U}_2, (w,\mathsf{f}).$$

Proof Let R be a bisimulation witnessing $\mathsf{U}_1, \mathsf{e} \leftrightarrows \mathsf{U}_2, \mathsf{f}$. Then the relation C between $W_M \times W_{\mathsf{U}_1}$ and $W_M \times W_{\mathsf{U}_2}$ given by

$$(w,\mathsf{e})C(v,\mathsf{f}) \text{ iff } w = v \text{ and } \mathsf{eRf}$$

is a bisimulation.

Suppose $(w,\mathsf{e})C(v,\mathsf{f})$. Then $w = v$ and $\mathsf{e}R\mathsf{f}$. Again, the only non-trivial check is the check for sameness of valuation. By $\mathsf{e}R\mathsf{f}$, the substitutions σ of e and τ of f are equivalent. By $w = v$, $V_M(w) = V_M(v)$. It follows that $V_M^\sigma(w) = V_M^\sigma(v)$, i.e., (w,e) and (v,f) have the same valuation. □

4.4 Expressive Power: Reducing LCC to Epistemic PDL

In order to reduce LCC to PDL we need a reduction axiom for formulas of the form $[\mathsf{U},\mathsf{e}][\pi]\varphi$. As before, the quest for reduction axioms starts with the attempt to describe what is the case after the update in terms of what is the case before the update. In the case of LCC, epistemic relations can take the shape of arbitrary PDL programs. So we must ask ourselves how we can find, for a given relation $[\![\pi]\!]^{M\circ\mathsf{U}}$ a corresponding relation in the original model M, w.

A formula of the form $\langle\mathsf{U},\mathsf{e}_i\rangle\langle\pi\rangle\varphi$ is true in some model M, w iff there is a π-path in $M \circ \mathsf{U}$ leading from (w,e_i) to a φ world (v,e_j). That means there is some path $w \dots v$ in M and some path $\mathsf{e}_i \dots \mathsf{e}_j$ in U such that $(M,w) \models \mathsf{pre}(\mathsf{e}_i)$ and … and $(M,v) \models \mathsf{pre}(\mathsf{e}_j)$ and of course $(M,v) \models \langle\mathsf{U},\mathsf{e}_j\rangle\varphi$. The program $T_{ij}^{\mathsf{U}}(\pi)$ captures this. A $T_{ij}^{\mathsf{U}}(\pi)$-path in the original model corresponds to a π-path in the updated model. But in defining $T_{ij}^{\mathsf{U}}(\pi)$ we cannot refer to a model M. The definition of $T_{ij}^{\mathsf{U}}(\pi)$ only depends on π, U, e_i and e_j. These transformers are used in the reduction axiom, which can be formulated as follows:

$$[\mathsf{U},\mathsf{e}_i][\pi]\varphi \quad \leftrightarrow \quad \bigwedge_{j=0}^{n-1} [T_{ij}^{\mathsf{U}}(\pi)][\mathsf{U},\mathsf{e}_j]\varphi.$$

The remainder of this section is directed towards showing that this axiom is sound (Theorem 12).

The program transformer T_{ij}^{U} is defined as follows:

Definition 25 (T^{U}_{ij} Program Transformers)

$$
\begin{array}{rcl}
T^{\mathsf{U}}_{ij}(a) & = & \begin{cases} ?\mathsf{pre}(\mathsf{e}_i); a & \text{if } \mathsf{e}_i R(a) \mathsf{e}_j, \\ ?\bot & \text{otherwise} \end{cases} \\
T^{\mathsf{U}}_{ij}(?\varphi) & = & \begin{cases} ?(\mathsf{pre}(\mathsf{e}_i) \wedge [\mathsf{U}, \mathsf{e}_i]\varphi) & \text{if } i = j, \\ ?\bot & \text{otherwise} \end{cases} \\
T^{\mathsf{U}}_{ij}(\pi_1; \pi_2) & = & \bigcup_{k=0}^{n-1} (T^{\mathsf{U}}_{ik}(\pi_1); T^{\mathsf{U}}_{kj}(\pi_2)) \\
T^{\mathsf{U}}_{ij}(\pi_1 \cup \pi_2) & = & T^{\mathsf{U}}_{ij}(\pi_1) \cup T^{\mathsf{U}}_{ij}(\pi_2) \\
T^{\mathsf{U}}_{ij}(\pi^*) & = & K^{\mathsf{U}}_{ijn}(\pi)
\end{array}
$$

where $K^{\mathsf{U}}_{ijn}(\pi)$ is given by Definition 26. □

We need the program transformer K^{U}_{ijn} in order to build the paths corresponding to the transitive closure of π in the updated model step by step, where we take more and more world of the update model into account. Intuitively, $K^{\mathsf{U}}_{ijk}(\pi)$ is a (transformed) program for all the π paths from (w_i, e_i) to (w_j, e_j) with that can be traced through $M \circ \mathsf{U}$ while avoiding a pass through intermediate states with events e_k and higher (Definition 26). Here, a π path from (w_i, e_i) to (w_j, e_j) is a path of the form $(w_i, \mathsf{e}_i), (w_j, \mathsf{e}_j)$ (in case $i = j$), or $(w_i, \mathsf{e}_i) \xrightarrow{\pi} \cdots \xrightarrow{\pi} (w_j, \mathsf{e}_j)$. Intermediate states are the states at positions $\cdots$ where a π step ends and a π step starts. Note that the restriction only applies to intermediate states. States that are passed during the execution of π are not restricted (they may involve events e_m with $m > k$). A given intermediate state e_r may occur more than once in a π path. Just as the definition of $T^{\mathsf{U}}_i j(\pi)$ does not refer to a concrete model M, also $K^{\mathsf{U}}_{ijn}(\pi)$ does not depend on a concrete model M. We only need to be concerned about the paths from e_i to e_j that could be the event components in a π-path in the updated model. Thus, $K^{\mathsf{U}}_{ij0}(\pi)$ is a program for all the paths from e_i to e_j that can be traced through U without stopovers at intermediate states that could yield a π path in an updated model. If $i = j$ it either is the skip action or a direct π loop, and otherwise it is a direct $T^{\mathsf{U}}_{ij}(\pi)$ step. This explains the base case in Definition 26.

Definition 26 (K^{U}_{ijk} Path Transformers) $K^{\mathsf{U}}_{ijk}(\pi)$ is defined by recursing on k, as follows:

$$
K^{\mathsf{U}}_{ij0}(\pi) = \begin{cases} ?\top \cup T^{\mathsf{U}}_{ij}(\pi) & \text{if } i = j, \\ T^{\mathsf{U}}_{ij}(\pi) & \text{otherwise} \end{cases}
$$

$$
K^{\mathsf{U}}_{ij(k+1)}(\pi) = \begin{cases} (K^{\mathsf{U}}_{kkk}(\pi))^* & \text{if } i = k = j, \\ (K^{\mathsf{U}}_{kkk}(\pi))^*; K^{\mathsf{U}}_{kjk}(\pi) & \text{if } i = k \neq j, \\ K^{\mathsf{U}}_{ikk}(\pi); (K^{\mathsf{U}}_{kkk}(\pi))^* & \text{if } i \neq k = j, \\ K^{\mathsf{U}}_{ijk}(\pi) \cup (K^{\mathsf{U}}_{ikk}(\pi); (K^{\mathsf{U}}_{kkk}(\pi))^*; K^{\mathsf{U}}_{kjk}(\pi)) & \text{otherwise } (i \neq k \neq j). \end{cases}
$$

□

Example applications of Definitions 25 and 26 are in Section 5. The next theorem states that the program transformation of a program π yields all the paths in the original model that correspond to paths in the updated model.

Theorem 11 (Program Transformation into E-PDL) For all update models U, all PDL programs π:

$$
(w, v) \in [\![T^{\mathsf{U}}_{ij}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \text{ iff } ((w, \mathsf{e}_i), (v, \mathsf{e}_j)) \in [\![\pi]\!]^{M \circ \mathsf{U}}.
$$

To prove Theorem 11 we need two lemmas.

Lemma 9 (Constrained Kleene Path) Suppose

$$(w,v) \in [\![T^{\mathsf{U}}_{ij}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \text{ iff } ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [\![\pi]\!]^{M\circ\mathsf{U}}.$$

Then

$$(w,v) \in [\![K^{\mathsf{U}}_{ijk}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \text{ iff there is a } \pi \text{ path from } (w,\mathsf{e}_i) \text{ to } (v,\mathsf{e}_j)) \text{ in } M\circ\mathsf{U}$$
$$\text{that does not have intermediate states } \cdots \xrightarrow{\pi} (u,\mathsf{e}_r) \xrightarrow{\pi} \cdots \text{ with } r \geq k.$$

Proof Induction on k, using the definition of K^{U}_{ijk}.

Base case $k = 0$, subcase $i = j$: A π path from (w,e_i) to (v,e_j) in $M\circ\mathsf{U}$ that does not visit any intermediate states is either the empty path or a single π step from (w,e_i) to (v,e_j). Such a path exists iff

$$\begin{array}{rl}
 & ((w,\mathsf{e}_i).(v,\mathsf{e}_j)) \in [\![?\top \cup \pi]\!]^{M\circ\mathsf{U}} \\
\text{iff (assumption)} & (w,v) \in [\![T^{\mathsf{U}}_{ij}(?\top \cup \pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff (definition } T^{\mathsf{U}}_{ij}) & (w,v) \in [\![?(\mathsf{pre}(\mathsf{e}_i) \wedge [\mathsf{U},\mathsf{e}_i]\top) \cup T^{\mathsf{U}}_{ij}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff } (i = j, \text{ so } \mathsf{pre}(\mathsf{e}_i) = \mathsf{pre}(\mathsf{e}_j) & (w,v) \in [\![?\top \cup T^{\mathsf{U}}_{ij}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff (definition } K^{\mathsf{U}}_{ij0}) & (w,v) \in [\![K^{\mathsf{U}}_{ij0}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M.
\end{array}$$

Base case $k = 0$, subcase $i \neq j$: A π path from (w,e_i) to (v,e_j) in $M\circ\mathsf{U}$ that does not visit any intermediate states is a single π step from (w,e_i) to (v,e_j). Such a path exists iff

$$\begin{array}{rl}
 & ((w,\mathsf{e}_i).(v,\mathsf{e}_j)) \in [\![\pi]\!]^{M\circ\mathsf{U}} \\
\text{iff (assumption)} & (w,v) \in [\![T^{\mathsf{U}}_{ij}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff (definition } K^{\mathsf{U}}_{ij0}) & (w,v) \in [\![K^{\mathsf{U}}_{ij0}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M.
\end{array}$$

Induction step. Assume that $(w,v) \in [\![K^{\mathsf{U}}_{ijk}(\pi); ?\mathsf{pre}(\mathsf{e}_j]\!]^M$ iff there is a π path from (w,e_i) to (v,e_j) in $M\circ\mathsf{U}$ that does not pass through any pairs (u,e) with $\mathsf{e} \in \{\mathsf{e}_k,\ldots,\mathsf{e}_{n-1}\}$.

We have to show that $(w,v) \in [\![K^{\mathsf{U}}_{ij(k+1)}(\pi); ?\mathsf{pre}(\mathsf{e}_j]\!]^M$ iff there is a π path from (w,e_i) to (v,e_j) in $M\circ\mathsf{U}$ that does not pass through any pairs (u,e) with $\mathsf{e} \in \{\mathsf{e}_{k+1},\ldots,\mathsf{e}_{n-1}\}$.

Case $i = k = j$. A π path from (w,e_i) to (v,e_j) in $M\circ\mathsf{U}$ that does not pass through any pairs (u,e) with $\mathsf{e} \in \{\mathsf{e}_{k+1},\ldots,\mathsf{e}_{n-1}\}$ now consists of an arbitrary composition of π paths from e_k to e_k that do not visit any intermediate states with event component e_k or higher. By the induction hypothesis, such a path exists iff $(w,v) \in [\![(K^{\mathsf{U}}_{kkk}(\pi))^*; ?\mathsf{pre}(\mathsf{e}_j]\!]^M$ iff (definition of $K^{\mathsf{U}}_{ij(k+1)}$) $(w,v) \in [\![K^{\mathsf{U}}_{ij(k+1)}(\pi); ?\mathsf{pre}(\mathsf{e}_j]\!]^M$.

Case $i = k \neq j$. A π path from (w,e_i) to (v,e_j) in $M\circ\mathsf{U}$ that does pass through any pairs (u,e) with $\mathsf{e} \in \{\mathsf{e}_{k+1},\ldots,\mathsf{e}_{n-1}\}$ now consists of a π path starting in (w,e_k) visiting states of the form (u,e_k) an arbitrary number of times, but never visiting states with event component e_k or higher in between, and ending in (v,e_k), followed by a π path from (u,e_k) to (v,e_j) that does not visit any pairs with event component $\mathsf{e} \in \{\mathsf{e}_k,\ldots,\mathsf{e}_{n-1}\}$. By the induction hypothesis, a π path from (w,e_k) to (u,e_k) of the first kind exists iff $(w,u) \in [\![(K^{\mathsf{U}}_{kkk}(\pi))^*; ?\mathsf{pre}(\mathsf{e}_k)]\!]^M$. Again by the induction hypothesis, a path from (u,e_k) to (v,e_j) of the second kind exists iff $(u,v) \in [\![K^{\mathsf{U}}_{kjk}; ?\mathsf{pre}(\mathsf{e}_j)]\!]^M$. Thus, the required path from (w,e_i) to (v,e_j) in $M\circ\mathsf{U}$ exists iff $(w,v) \in [\![(K^{\mathsf{U}}_{kkk}(\pi))^*; K^{\mathsf{U}}_{kjk}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M$ iff (definition of $K^{\mathsf{U}}_{ij(k+1)}$) $(w,v) \in [\![K^{\mathsf{U}}_{ij(k+1)}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M$.

The other two cases are similar. □

Lemma 10 (General Kleene Path) Suppose $(w,v) \in [\![T^{\mathsf{U}}_{ij}(\pi)); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M$ iff there is a π step from (w,e_i) to (v,e_j) in $M\circ\mathsf{U}$. Then $(w,v) \in [\![K^{\mathsf{U}}_{ijn}(\pi)); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M$ iff there is a π path from (w,e_i) to (v,e_j) in $M\circ\mathsf{U}$. □

Proof Suppose $(w,v) \in [\![T^{\mathsf{U}}_{ij}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M$ iff there is a π path from (w,e_i) to (v,e_j) in $M \circ \mathsf{U}$. Then, assuming that U has states $\mathsf{e}_0,\ldots,\mathsf{e}_{n-1}$, an application of Lemma 9 yields that $K^{\mathsf{U}}_{ijn}(\pi)$ is a program for all the π paths from (w,e_i) to (v,e_j) that can be traced through $M \circ \mathsf{U}$, for stopovers at any (u,e_k) with $0 \leq k \leq n-1$ are allowed. $\Box$

Lemma 10 explains the use of K^{U}_{ijn} in the clause for π^* in Definition 25. The lemma also allows us to prove Theorem 11.

Proof of Theorem 11. Induction on the structure of π.

Base case a:

$$\begin{array}{ll} & (w,v) \in [\![T^{\mathsf{U}}_{ij}(a); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\ \text{iff} & (w,v) \in [\![?\mathsf{pre}(\mathsf{e}_i); a; ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \text{ and } \mathsf{e}_i R(a) \mathsf{e}_j \\ \text{iff} & M,w \models \mathsf{pre}(\mathsf{e}_i), (w,v) \in [\![a]\!]^M, \mathsf{e}_i R(a) \mathsf{e}_j \text{ and } M,v \models \mathsf{pre}(\mathsf{e}_j) \\ \text{iff} & ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [\![\pi]\!]^{M\circ\mathsf{U}}. \end{array}$$

Base case $?\varphi$, subcase $i = j$:

$$\begin{array}{ll} & (w,v) \in [\![T^{\mathsf{U}}_{ij}(?\varphi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\ \text{iff} & (w,v) \in [\![?(\mathsf{pre}(\mathsf{e}_i) \wedge [\mathsf{U},\mathsf{e}_i]\varphi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\ \text{iff} & w = v \text{ and } M,w \models \mathsf{pre}(\mathsf{e}_i) \text{ and } M,w \models [\mathsf{U},\mathsf{e}_i]\varphi \\ \text{iff} & w = v \text{ and } M,w \models \mathsf{pre}(\mathsf{e}_i) \text{ and } M,w \models \mathsf{pre}(\mathsf{e}_i) \text{ implies } M \circ \mathsf{U},(w,\mathsf{e}_i) \models \varphi \\ \text{iff} & w = v \text{ and } M \circ \mathsf{U},(w,\mathsf{e}_i) \models \varphi \\ \text{iff} & ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [\![?\varphi]\!]^{M\circ\mathsf{U}}. \end{array}$$

Base case $?\varphi$, subcase $i \neq j$:

$$\begin{array}{ll} & (w,v) \in [\![T^{\mathsf{U}}_{ij}(?\varphi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\ \text{iff} & (w,v) \in [\![?\bot]\!]^M \\ \text{iff} & ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [\![?\varphi]\!]^{M\circ\mathsf{U}}. \end{array}$$

Induction step. Now consider any complex program π and assume for all components π' of π it holds that:

$$(w,v) \in [\![T^{\mathsf{U}}_{ij}(\pi'); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \text{ iff } ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [\![\pi']\!]^{M\circ\mathsf{U}}.$$

We have to show:

$$(w,v) \in [\![T^{\mathsf{U}}_{ij}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \text{ iff } ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [\![\pi]\!]^{M\circ\mathsf{U}}.$$

$\pi = \pi_1; \pi_2$:

$$\begin{array}{ll}
 & (w,v) \in [\![T^{\mathsf{U}}_{ij}(\pi_1;\pi_2); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff} & (w,v) \in [\![\bigcup_{k=0}^{n-1} (T^{\mathsf{U}}_{ik}(\pi_1); T^{\mathsf{U}}_{kj}(\pi_2)); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff} & \text{for some } k \in \{0,\dots,n-1\} (w,v) \in [\![T^{\mathsf{U}}_{ik}(\pi_1); T^{\mathsf{U}}_{kj}(\pi_2); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff} & \text{for some } k \in \{0,\dots,n-1\} \text{ and some } u \in W \\
 & (w,u) \in [\![T^{\mathsf{U}}_{ik}(\pi_1)]\!]^M \text{ and } (u,v) \in [\![T^{\mathsf{U}}_{kj}(\pi_2); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff (ih)} & \text{for some } k \in \{0,\dots,n-1\} \text{ and some } u \in W \\
 & (w,u) \in [\![T^{\mathsf{U}}_{ik}(\pi_1)]\!]^M \text{ and } ((u,\mathsf{e}_k),(v,\mathsf{e}_j)) \in [\![\pi_2]\!]^{M \circ \mathsf{U}} \\
\text{iff} & \text{for some } k \in \{0,\dots,n-1\} \text{ and some } u \in W \\
 & (w,u) \in [\![T^{\mathsf{U}}_{ik}(\pi_1)]\!]^M, M,u \models \mathsf{pre}(\mathsf{e}_k) \text{ and } ((u,\mathsf{e}_k),(v,\mathsf{e}_j)) \in [\![\pi_2]\!]^{M \circ \mathsf{U}} \\
\text{iff} & \text{for some } k \in \{0,\dots,n-1\} \text{ and some } u \in W \\
 & (w,u) \in [\![T^{\mathsf{U}}_{ik}(\pi_1); \mathsf{pre}(\mathsf{e}_k)]\!]^M \text{ and } ((u,\mathsf{e}_k),(v,\mathsf{e}_j)) \in [\![\pi_2]\!]^{M \circ \mathsf{U}} \\
\text{iff (ih again)} & \text{for some } k \in \{0,\dots,n-1\} \text{ and some } u \in W \\
 & ((w,\mathsf{e}_i),(u,\mathsf{e}_k)) \in [\![\pi_1]\!]^{M \circ \mathsf{U}} \text{ and } ((u,\mathsf{e}_k),(v,\mathsf{e}_j)) \in [\![\pi_2]\!]^{M \circ \mathsf{U}} \\
\text{iff} & ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [\![\pi]\!]^{M \circ \mathsf{U}}.
\end{array}$$

$\pi = \pi_1 \cup \pi_2$:

$$\begin{array}{ll}
 & (w,v) \in [\![T^{\mathsf{U}}_{ij}(\pi_1 \cup \pi_2); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff} & (w,v) \in [\![(T^{\mathsf{U}}_{ij}(\pi_1) \cup T^{\mathsf{U}}_{ij}(\pi_2)); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff} & (w,v) \in [\![(T^{\mathsf{U}}_{ij}(\pi_1); ?\mathsf{pre}(\mathsf{e}_j)) \cup (T^{\mathsf{U}}_{ij}(\pi_2) ?\mathsf{pre}(\mathsf{e}_j))]\!]^M \\
\text{iff} & (w,v) \in [\![T^{\mathsf{U}}_{ij}(\pi_1); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \text{ or } (w,v) \in [\![T^{\mathsf{U}}_{ij}(\pi_2); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff (ih)} & ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [\![\pi_1]\!]^{M \circ \mathsf{U}} \text{ or } ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [\![\pi_2]\!]^{M \circ \mathsf{U}} \\
\text{iff} & ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [\![\pi]\!]^{M \circ \mathsf{U}}.
\end{array}$$

$\pi = \pi^*$:

$$\begin{array}{ll}
 & (w,v) \in [\![T^{\mathsf{U}}_{ij}(\pi^*); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff (definition } T^{\mathsf{U}}_{ij}) & (w,v) \in [\![K^{\mathsf{U}}_{ijn}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^M \\
\text{iff (ih, Lemma 10)} & \text{there is a } \pi \text{ path from } (w,\mathsf{e}_i) \text{ to } (v,\mathsf{e}_j) \text{ in } M \circ \mathsf{U} \\
\text{iff} & ((w,\mathsf{e}_i),(v,\mathsf{e}_j)) \in [\![\pi^*]\!]^{M \circ \mathsf{U}}.
\end{array}$$

□

Theorem 12 (Reduction Equivalence) Assume U has n states $\mathsf{e}_0, \dots, \mathsf{e}_{n-1}$. Then:

$$M,w \models [\mathsf{U},\mathsf{e}_i][\pi]\varphi \text{ iff } M,w \models \bigwedge_{j=0}^{n-1} [T^{\mathsf{U}}_{ij}(\pi)][\mathsf{U},\mathsf{e}_j]\varphi.$$

Proof The result is derived by the following chain of equivalences:

$$
\begin{array}{rl}
& M, w \models [\mathsf{U}, \mathsf{e}_i][\pi]\varphi \\
\text{iff} & M, w \models \mathsf{pre}(\mathsf{e}_i) \text{ implies } M \circ \mathsf{U}, (w, \mathsf{e}_i) \models [\pi]\varphi \\
\text{iff} & \forall v \in W, j \in \{0, \ldots, n-1\} \text{ with } ((w, \mathsf{e}_i), (v, \mathsf{e}_j)) \in [\![\pi]\!]^{M \circ \mathsf{U}}, \\
& \quad M \circ \mathsf{U}, (v, \mathsf{e}_j) \models \varphi \\
\text{iff (Theorem 11)} & \forall v \in W, j \in \{0, \ldots, n-1\} \text{ with } (w, v) \in [\![T^{\mathsf{U}}_{ij}(\pi); ?\mathsf{pre}(\mathsf{e}_j)]\!]^{M}, \\
& \quad M \circ \mathsf{U}, (v, \mathsf{e}_j) \models \varphi \\
\text{iff} & \forall v \in W, j \in \{0, \ldots, n-1\} \text{ with } (w, v) \in [\![T^{\mathsf{U}}_{ij}(\pi)]\!]^{M}, \\
& \quad M, v \models \mathsf{pre}(\mathsf{e}_j) \text{ implies } M \circ \mathsf{U}, (v, \mathsf{e}_j) \models \varphi \\
\text{iff} & \forall v \in W, j \in \{0, \ldots, n-1\} \text{ with } (w, v) \in [\![T^{\mathsf{U}}_{ij}(\pi)]\!]^{M}, \\
& \quad M, v \models [\mathsf{U}, \mathsf{e}_j]\varphi \\
\text{iff} & M, w \models \bigwedge_{j=0}^{n-1} [T^{\mathsf{U}}_{ij}(\pi)][\mathsf{U}, \mathsf{e}_j]\varphi
\end{array}
$$

□

What the Reduction Theorem gives us is that LCC is equivalent to PDL, and hence, that a proof system for LCC can be given in terms of axioms that reduce formulas of the form $[\mathsf{U}, \mathsf{e}]\varphi$ to equivalent formulas ψ with the property that their main operator is not an update modality for U. First, we state the former model-theoretic expressiveness result, which is the first main theorem of this paper:

Theorem 13 ('LCC = E-PDL') The languages of LCC and E-PDL have equal expressive power. □

The earlier-mentioned similarity between finite automata and our current approach is most striking in the definition of the transformation for starred programs. The definition of transformed π^* paths in terms of operators $K_{ijk}(\pi)$ resembles the definition of sets of regular languages L_k generated by moving through a nondeterministic finite automaton without passing through states numbered k or higher, in the well-known proof of Kleene's Theorem. Textbook versions of its proof can be found in many places, e.g., [20, Theorem 2.5.1].

4.5 Reduction Axioms and Completeness for LCC

The results from the previous section point the way to appropriate reduction axioms for LCC.

Definition 27 (Proof system for LCC) The proof system for LCC consists of all axioms and rules of PDL [28, 14, 25], plus the following reduction axioms:

$$
\begin{array}{rcl}
[\mathsf{U}, \mathsf{e}]\top & \leftrightarrow & \top \\
[\mathsf{U}, \mathsf{e}]p & \leftrightarrow & (\mathsf{pre}(\mathsf{e}) \to p^{\mathsf{sub}(\mathsf{e})}) \\
[\mathsf{U}, \mathsf{e}]\neg\varphi & \leftrightarrow & (\mathsf{pre}(\mathsf{e}) \to \neg[\mathsf{U}, \mathsf{e}]\varphi) \\
[\mathsf{U}, \mathsf{e}](\varphi_1 \wedge \varphi_2) & \leftrightarrow & ([\mathsf{U}, \mathsf{e}]\varphi_1 \wedge [\mathsf{U}, \mathsf{e}]\varphi_2) \\
[\mathsf{U}, \mathsf{e}_i][\pi]\varphi & \leftrightarrow & \bigwedge_{j=0}^{n-1} [T^{\mathsf{U}}_{ij}(\pi)][\mathsf{U}, \mathsf{e}_j]\varphi.
\end{array}
$$

and necessitation for update model modalities. □

The last, and most crucial, of the reduction axioms is based on program transformation. If updates with so-called 'multiple pointed update models' (cf. [12]) are also added to the language, we need the following additional reduction axiom:

$$
[\mathsf{U}, \mathsf{W}]\varphi \quad \leftrightarrow \quad \bigwedge_{\mathsf{e} \in \mathsf{W}} [\mathsf{U}, \mathsf{e}]\varphi
$$

Thus, we see that LCC is no more expressive than PDL; indeed, program transformations can be used to translate LCC to PDL, as follows:

Definition 28 (Translation) The translation function t takes a formula from the language of LCC and yields a formula in the language of PDL.

$$
\begin{array}{lcl@{\qquad\qquad}lcl}
t(\top) & = & \top & r(a) & = & a \\
t(p) & = & p & r(B) & = & B \\
t(\neg\varphi) & = & \neg t(\varphi) & r(?\varphi) & = & ?t(\varphi) \\
t(\varphi_1 \wedge \varphi_2) & = & t(\varphi_1) \wedge t(\varphi_2) & r(\pi_1;\pi_2) & = & r(\pi_1); r(\pi_2) \\
t([\pi]\varphi) & = & [r(\pi)]t(\varphi) & r(\pi_1 \cup \pi_2) & = & r(\pi_1) \cup r(\pi_2) \\
t([\mathsf{U},\mathsf{e}]\top) & = & \top & r(\pi^*) & = & (r(\pi))^*. \\
t([\mathsf{U},\mathsf{e}]p) & = & t(\mathsf{pre}(\mathsf{e})) \to p^{\mathsf{sub}(\mathsf{e})} & & & \\
t([\mathsf{U},\mathsf{e}]\neg\varphi) & = & t(\mathsf{pre}(\mathsf{e})) \to \neg t([\mathsf{U},\mathsf{e}]\varphi) & & & \\
t([\mathsf{U},\mathsf{e}](\varphi_1 \wedge \varphi_2) & = & t([\mathsf{U},\mathsf{e}]\varphi_1) \wedge t([\mathsf{U},\mathsf{e}]\varphi_2) & & & \\
t([\mathsf{U},\mathsf{e}_i][\pi]\varphi & = & \bigwedge_{j=0}^{n-1} [T_{ij}^{\mathsf{U}}(r(\pi))]t([\mathsf{U},\mathsf{e}_j]\varphi) & & & \\
t([\mathsf{U},\mathsf{e}][\mathsf{U}',\mathsf{e}']\varphi & = & t([\mathsf{U},\mathsf{e}]t([\mathsf{U}',\mathsf{e}']\varphi)) & & &
\end{array}
$$

$\Box$

The correctness of this translation follows from direct semantic inspection, using the program transformation corollary for the translation of $[\mathsf{U},\mathsf{e}_i][\pi]\varphi$ formulas.

Theorem 14 (Completeness for LCC) $\models \varphi$ iff $\vdash \varphi$. $\Box$

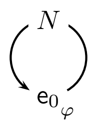
Proof The proof system for PDL is complete, and every formula in the language of LCC is provably equivalent to a PDL formula. $\Box$

5 Analyzing Major Communication Types

The program transformation approach provides a systematic perspective on communicative updates. In the case of public announcement and common knowledge, it was still possible to generate appropriate reduction axioms by hand. Such axioms can also be generated automatically, however, by program transformation, as we will now show. This allows us to deal with much more complicated cases, such as secret group communication and common belief, or subgroup announcement and common knowledge, where axiom generation by hand is not feasible anymore. The following generated axioms may look unwieldy, illustrating the fact that E-PDL functions as an assembler language for detailed analysis of the higher level specifications of communicative updates in terms of update models. But upon closer inspection, they make sense, and indeed, for simple communicative scenarios, they can be seen to liveinside EL-RC.

5.1 Public Announcement and Common Knowledge

The update model for public announcement that φ consists of a single state e_0 with precondition φ and epistemic relation $\{(\mathsf{e}_0, \mathsf{e}_0)\}$ for all agents. Call this model P_φ.



We are interested in how public announcement that φ affects common knowledge among group of agents B, i.e., we want to compute $[P_\varphi, \mathsf{e}_0][B^*]\psi$. For this, we need $T_{00}^{P_\varphi}(B^*)$, which is defined as $K_{001}^{P_\varphi}(B)$.

To work out $K^{P_\varphi}_{001}(B)$, we need $K^{P_\varphi}_{000}(B)$, and for $K^{P_\varphi}_{000}(B)$, we need $T^{P_\varphi}_{00}(B)$, which turns out to be $\bigcup_{b\in B}(?\varphi; b)$, or equivalently, $?\varphi; B$. Working upwards from this, we get:

$$K^{P_\varphi}_{000}(B) = ?\top \cup T^{P_\varphi}_{00}(B) = ?\top \cup (?\varphi; B),$$

and therefore:

$$\begin{array}{rcl} K^{P_\varphi}_{001}(B) & = & (K^{P_\varphi}_{000}(B))^* \\ & = & (?\top \cup (?\varphi; B))^* \\ & = & (?\varphi; B)^*. \end{array}$$

Thus, the reduction axiom for the public announcement update P_φ with respect to the program for common knowledge among agents B, works out as follows:

$$\begin{array}{rcl} [P_\varphi, \mathsf{e}_0][B^*]\psi & \leftrightarrow & [T^{P_\varphi}_{00}(B^*)][P_\varphi, \mathsf{e}_0]\psi \\ & \leftrightarrow & [K^{P_\varphi}_{001}(B)][P_\varphi, \mathsf{e}_0]\psi \\ & \leftrightarrow & [(?\varphi; B)^*][P_\varphi, \mathsf{e}_0]\psi. \end{array}$$

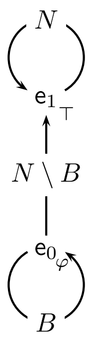
This expresses that every B path consisting of φ worlds ends in a $[P_\varphi, \mathsf{e}_0]\psi$ world, i.e., it expresses exactly what is captured by the special purpose operator $C_B(\varphi, \psi)$ from Section 2.2.

5.2 Secret Group Communication and Common Belief

The logic of secret group communication is the logic of email CCs (assuming that emails arrive immediately and are read immediately). The update model for a secret group message to B that φ consists of two possible events $\mathsf{e}_0, \mathsf{e}_1$, where e_0 has precondition φ and e_1 has precondition $\top$. The group ou, and where the accessibilities T are given by:

$$T = \{\mathsf{e}_0 R(b) \mathsf{e}_0 \mid b \in B\} \cup \{\mathsf{e}_0 R(a) \mathsf{e}_1 \mid a \in N \setminus B\} \cup \{\mathsf{e}_1 R(a) \mathsf{e}_1 \mid a \in N\}.$$

The actual event is e_0. The members of B are aware that φ gets communicated; the others think that nothing happens. In this thought they are mistaken, which is why CC updates generate KD45 models: i.e., CC updates make knowledge degenerate into belief.



We work out the program transformations that this update engenders for common knowledge among group of agents D. Call the update model CC^B_φ.

We will have to work out $K^{\mathrm{CC}^B_\varphi}_{002} D$, $K^{\mathrm{CC}^B_\varphi}_{012} D$, $K^{\mathrm{CC}^B_\varphi}_{112} D$, $K^{\mathrm{CC}^B_\varphi}_{102} D$.

For these, we need $K^{\mathrm{CC}^B_\varphi}_{001} D$, $K^{\mathrm{CC}^B_\varphi}_{011} D$, $K^{\mathrm{CC}^B_\varphi}_{111} D$, $K^{\mathrm{CC}^B_\varphi}_{101} D$.

For these in turn, we need $K^{\mathrm{CC}^B_\varphi}_{000} D$, $K^{\mathrm{CC}^B_\varphi}_{010} D$, $K^{\mathrm{CC}^B_\varphi}_{110} D$, $K^{\mathrm{CC}^B_\varphi}_{100} D$.

For these, we need:

$$
\begin{aligned}
T_{00}^{\mathrm{CC}_{\varphi}^{B}} D &= \bigcup_{d \in B \cap D} (?\varphi; d) \;= ?\varphi; (B \cap D) \\
T_{01}^{\mathrm{CC}_{\varphi}^{B}} D &= \bigcup_{d \in D \setminus B} (?\varphi; d) \;= ?\varphi; (D \setminus B) \\
T_{11}^{\mathrm{CC}_{\varphi}^{B}} D &= D \\
T_{10}^{\mathrm{CC}_{\varphi}^{B}} D &= ?\bot
\end{aligned}
$$

It follows that:

$$
\begin{aligned}
K_{000}^{\mathrm{CC}_{\varphi}^{B}} D &= ?\top \cup (?\varphi; (B \cap D)) \\
K_{010}^{\mathrm{CC}_{\varphi}^{B}} D &= ?\varphi; (D \setminus B) \\
K_{110}^{\mathrm{CC}_{\varphi}^{B}} D &= ?\top \cup D, \\
K_{100}^{\mathrm{CC}_{\varphi}^{B}} D &= ?\bot
\end{aligned}
$$

¿From this we can work out the K_{ij1}, as follows:

$$
\begin{aligned}
K_{001}^{\mathrm{CC}_{\varphi}^{B}} D &= (?\varphi; (B \cap D))^{*} \\
K_{011}^{\mathrm{CC}_{\varphi}^{B}} D &= (?\varphi; (B \cap D))^{*}; (D \setminus B) \\
K_{111}^{\mathrm{CC}_{\varphi}^{B}} D &= ?\top \cup D \\
K_{101}^{\mathrm{CC}_{\varphi}^{B}} D &= ?\bot.
\end{aligned}
$$

Finally, we get K_{002} and K_{012} from this:

$$
\begin{aligned}
K_{002}^{\mathrm{CC}_{\varphi}^{B}} D &= K_{001}^{\mathrm{CC}_{\varphi}^{B}} D \cup K_{011}^{\mathrm{CC}_{\varphi}^{B}} D; (K_{111}^{\mathrm{CC}_{\varphi}^{B}} D)^{*}; K_{101}^{\mathrm{CC}_{\varphi}^{B}} D \\
&= K_{001}^{\mathrm{CC}_{\varphi}^{B}} D \;\; \text{(since the righthand expression evaluates to } ?\bot) \\
&= (?\varphi; (B \cap D))^{*} \\
K_{012}^{\mathrm{CC}_{\varphi}^{B}} D &= K_{011}^{\mathrm{CC}_{\varphi}^{B}} D \cup K_{011}^{\mathrm{CC}_{\varphi}^{B}} D; (K_{111}^{\mathrm{CC}_{\varphi}^{B}} D)^{*} \\
&= K_{011}^{\mathrm{CC}_{\varphi}^{B}} D; (K_{111}^{\mathrm{CC}_{\varphi}^{B}} D)^{*} \\
&= (?\varphi; (B \cap D))^{*}; (D \setminus B); D^{*}.
\end{aligned}
$$

Thus, the program transformation for common belief among D works out as follows:

$$
[\mathrm{CC}_{\varphi}^{B}, \mathsf{e}_0][D^{*}]\psi \;\;\leftrightarrow\;\; [(?\varphi; (B \cap D))^{*}][\mathrm{CC}_{\varphi}^{B}, \mathsf{e}_0]\psi \wedge [(?\varphi; (B \cap D))^{*}; (D \setminus B); D^{*}][\mathrm{CC}_{\varphi}^{B}, \mathsf{e}_1]\psi.
$$

This transformation yields a reduction axiom that shows that EL-RC also suffices to provide reduction axioms for secret group communication. Compare [27] for a direct axiomatization of the logic of CCs.

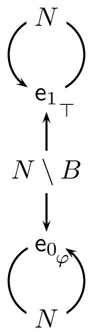
5.3 Group Messages and Common Knowledge

Finally, we consider group messages. This example is one of the simplest cases that shows that program transformations gives us reduction axioms that are no longer feasible to give by hand.

The update model for a group message to B that φ consists of two states $\mathsf{e}_0, \mathsf{e}_1$, where e_0 has precondition φ and e_1 has precondition $\top$, and where the accessibilities T are given by:

$$
T = \{\mathsf{e}_0 R(b) \mathsf{e}_0 \mid b \in B\} \cup \{\mathsf{e}_1 R(b) \mathsf{e}_1 \mid b \in B\} \cup \{\mathsf{e}_0 R(a) \mathsf{e}_1 \mid a \in N \setminus B\} \cup \{\mathsf{e}_1 R(a) \mathsf{e}_0 \mid a \in N \setminus B\}.
$$

This captures the fact that the members of B can distinguish the φ update from the $\top$ update, while the other agents (the members of $N \setminus B$) cannot. The actual event is e_0. Call this model G^B_φ.



A difference with the CC case is that group messages are S5 models. Since updates of S5 models with S5 models are S5, group messages engender common knowledge (as opposed to mere common belief). Let us work out the program transformation that this update engenders for common knowledge among group of agents D.

We will have to work out $K^{G^B_\varphi}_{002} D$, $K^{G^B_\varphi}_{012} D$, $K^{G^B_\varphi}_{112} D$, $K^{G^B_\varphi}_{102} D$.

For these, we need $K^{G^B_\varphi}_{001} D$, $K^{G^B_\varphi}_{011} D$, $K^{G^B_\varphi}_{111} D$, $K^{G^B_\varphi}_{101} D$.

For these in turn, we need $K^{G^B_\varphi}_{000} D$, $K^{G^B_\varphi}_{010} D$, $K^{G^B_\varphi}_{110} D$, $K^{G^B_\varphi}_{100} D$.

For these, we need:

$$
\begin{array}{rcl}
T^{G^B_\varphi}_{00} D & = & \bigcup_{d \in D} (?\varphi; d) = ?\varphi; D, \\
T^{G^B_\varphi}_{01} D & = & \bigcup_{d \in D \setminus B} (?\varphi; d) = ?\varphi; (D \setminus B), \\
T^{G^B_\varphi}_{11} D & = & D, \\
T^{G^B_\varphi}_{10} D & = & D \setminus B.
\end{array}
$$

It follows that:

$$
\begin{array}{rcl}
K^{G^B_\varphi}_{000} D & = & ?\top \cup (?\varphi; D), \\
K^{G^B_\varphi}_{010} D & = & ?\varphi; (D \setminus B), \\
K^{G^B_\varphi}_{110} D & = & ?\top \cup D, \\
K^{G^B_\varphi}_{100} D & = & D \setminus B.
\end{array}
$$

¿From this we can work out the K_{ij1}, as follows:

$$
\begin{array}{rcl}
K^{G^B_\varphi}_{001} D & = & (?\varphi; D)^*, \\
K^{G^B_\varphi}_{011} D & = & (?\varphi; D)^*; ?\varphi; D \setminus B, \\
K^{G^B_\varphi}_{111} D & = & ?\top \cup D \cup (D \setminus B; (?\varphi; D)^*; ?\varphi; D \setminus B), \\
K^{G^B_\varphi}_{101} D & = & D \setminus B; (?\varphi; D)^*.
\end{array}
$$

Finally, we get K_{002} and K_{012} from this:

$$\begin{array}{rcl}
K_{002}^{G_\varphi^B} D & = & K_{001}^{G_\varphi^B} D \cup K_{011}^{G_\varphi^B} D; (K_{111}^{G_\varphi^B} D)^*; K_{101}^{G_\varphi^B} D \\
& = & (?\varphi; D)^* \cup \\
& & (?\varphi; D)^*; ?\varphi; D \setminus B; (D \cup (D \setminus B; (?\varphi; D)^*; ?\varphi; D \setminus B))^* ; D \setminus B; (?\varphi; D)^*, \\
K_{012}^{G_\varphi^B} D & = & K_{011}^{G_\varphi^B} D; (K_{111}^{G_\varphi^B} D)^* \\
& = & (?\varphi; D)^*; ?\varphi; D \setminus B; (D \cup (D \setminus B; (?\varphi; D)^*; ?\varphi; D \setminus B))^*.
\end{array}$$

Abbreviating $D \cup (D \setminus B; (?\varphi; D)^*; ?\varphi; D \setminus B)$ as π, we get the following transformation for common knowledge among D after a group message to B that φ:

$$\begin{array}{rcl}
[G_\varphi^B, \mathsf{e}_0][D^*]\psi & \leftrightarrow & [(?\varphi; D)^* \cup ((?\varphi; D)^*; ?\varphi; D \setminus B; \pi^*; D \setminus B; (?\varphi; D)^*)][G_\varphi^B, \mathsf{e}_0]\psi \\
& & \wedge \\
& & [(?\varphi; D)^*; ?\varphi; D \setminus B; \pi^*][G_\varphi^B, \mathsf{e}_1]\psi.
\end{array}$$

This formula makes clear that although we can translate every formula of LCC to PDL, the higher order descriptions of updates using update models are more convenient for reasoning about information change.

One interesting side-effect of this bunch of illustrations is that it demonstrates the computational character of our analysis. Cf. [9] on the use of computational tools in exploring the universe of iterated epistemic updates.

6 Conclusion and Further Research

Dynamic-epistemic logics provide systematic means for studying exchange of factual and higher-order information. In this many-agent setting, common knowledge is an essential concept. We have presented two extended languages for dynamic-epistemic logic that admit explicit reduction axioms for common knowledge resulting from an update: one (PAL-RC) for public announcement only, and one (LCC) for general update actions. These systems make proof and complexity analysis for informative actions more perspicuous than earlier attempts in the literature. Still, PAL and epistemic LCC are just two extremes on a spectrum, and many further natural update logics may lie in between.

We conclude by pointing out some further research topics that arise on our analysis.

- *Downward in expressive power from* E-PDL. Which weaker language fragments are in 'dynamic-static harmony', in the sense of having reduction axioms and the corresponding meaning-preserving translation? Our program transformer approach does work also with certain restrictions on tests in our full logic LCC, but we have not yet been able to identify natural intermediate levels,

- *Upward in expressive power from* E-PDL. Which *richer* languages are in dynamic-static harmony? A typical candidate is the epistemic *μ-calculus*, which allows arbitrary operators for defining smallest and greatest fixed-points. Indeed, we have a proof that the results of Section 2 extend to the calculus PAL-μ for public announcements, which takes the complete epistemic *μ-calculus* for its static language (allowing no binding into announcement positions). Our conjecture is that our expressivity and axiomatization results of Section 4 also extend to the full μ-calculus version of LCC. Cf. [7] for the current state of affairs.

- *Other notions of group knowledge.* Another test of our methodology via reduction axioms are further notions of group knowledge. For instance, instead of common knowledge, consider *distributed group knowledge* $D_B\varphi$ consisting of those statements which are available implicitly to the group, in the sense of φ being true at every world reachable from the current one by the intersection of all epistemic accessibility relations. The following simple reduction holds for public announcements: $[A]D_B\varphi \leftrightarrow (A \rightarrow [A]D_B\varphi)$. We have not yet investigated our full system LCC extended with distributed knowledge.

- *Program constructions over update models.* One can add the usual regular operations of composition, choice, and iteration over update models, to obtain a calculus describing effects of more complex information-bearing events. It is known that this extension makes PAL undecidable, but what about partial axiomatizations in our style?,

- *Alternative questions about update reasoning.* With one exception, our reduction axioms are all *schematically valid* in the sense that substituting arbitrary formulas for proposition letters again yields a valid formula. The exception is the base clause, which really only holds for atomic proposition letters p. As dicussed in [5], this means that certain schematically valid laws of update need not be derivable from our axioms in an explicit schematic manner, even though all their concrete instances will be by our completeness theorem. An example is the schematic law stating the associativity of successive announcements. It is not known whether schematic validity is decidable, even for PAL, and no complete axiomatization is known either. This is just one instance of open problems concerning PAL and its ilk for public announcement (cf. the survey in [6]), which all return for LCC, with our program transformations as a vehicle for generalizing the issues.

- *General logical perspectives.* Languages with relativizations are very common in logic. Indeed, closure under relativization is sometimes stated as a defining condition on logics in abstract model theory. Basic modal or first-order logic as they stand are closed under relativizations $[A]\varphi$, often written $(\varphi)^A$. The same is true for logics with fixed-point constructions, like PDL (cf. [4]) or the modal μ-calculus. E.g., computing a relativized least fixed-point $[A]\mu p.\varphi(p)$ works much as evaluation of $\mu p.\varphi(p) \wedge A$ – which actually suggests a corresponding dynamic epistemic reduction axiom (cf. [7]). In the setting of Section 4, lifts relativization to some sort of 'update closure' for general logical languages, referring to *relative interpretations* in definable submodels of products. Languages with this property include again first-order logic and its fixed-point extensions, as well as fragments of the μ-calculus, and temporal **UNTIL** logics.

Acknowledgements We thanks Tomasz Sadzik, as well as other participants of the 'Stanford-ILLC Seminar' in December 2004 for their helpful comments and questions.

References

[1] A. Baltag, L. S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. Technical Report SEN-R9922, CWI, Amsterdam, 1999.

[2] A. Baltag and L.S. Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004.

[3] A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. Technical report, Dept of Cognitive Science, Indiana University and Dept of Computing, Oxford University, Bloomington and Oxford, 2003. Updated version of [1].

[4] J. van Benthem. Update as relativization. ILLC, University of Amsterdam, 1999. Manuscript.

[5] J. van Benthem. One is a lonely number: on the logic of communication. Technical Report PP-2002-27, ILLC, Amsterdam, 2002. To appear in P. Köpke, ed. *Colloquium Logicum, Münster, 2001*, AMS Publications, Providence.

[6] J. van Benthem. Open problems in logical dynamics. ILLC Amsterdam, 2005. Manuscript.

[7] J. van Benthem, J. van Eijck, and B. Kooi. A fixed-point look at update logic, 2005. Manuscript, ILLC, Amsterdam.

[8] H. van Ditmarsch. *Knowledge Games*. PhD thesis, ILLC, Amsterdam, 2000.

[9] J. van Eijck. Dynamic epistemic modelling. Technical Report SEN-E0424, CWI, Amsterdam, December 2004. Available from `http://db.cwi.nl/rapporten/`.

[10] J. van Eijck. Guarded actions. Technical Report SEN-E0425, CWI, Amsterdam, December 2004. Available from `http://db.cwi.nl/rapporten/`.

[11] J. van Eijck. Reducing dynamic epistemic logic to PDL by program transformation. Technical Report SEN-E0423, CWI, Amsterdam, December 2004. Available from `http://db.cwi.nl/rapporten/`.

[12] J. van Eijck, J. Ruan, and Th. Sadzik. Action emulation, 2005. Manuscript, CWI Amsterdam.

[13] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning About Knowledge*. MIT Press, Cambridge, Massachusetts, 1995.

[14] M.J. Fischer and R.E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[15] J. Gerbrandy. *Bisimulations on Planet Kripke*. ILLC Dissertation Series, Amsterdam, 1999.

[16] D. Harel. Dynamic logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic Volume II: extensions of classical logic*, volume 165 of *Synthese Library Studies in epistemology, logic, methodology, and philosophy of science*, chapter 10, pages 497–604. D. Reidel Publishing Company, Dordrecht Boston Lancaster, 1984.

[17] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. Foundations of Computing. MIT Press, Cambridge, Massachusetts, 2000.

[18] S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3–41. Princeton University Press, Princeton, N.J, 1956.

[19] B. Kooi and J. van Benthem. Reduction axioms for epistemic actions. In R. Schmidt, I Pratt-Hartmann, M. Reynolds, and H. Wansing, editors, *AiML-2004: Advances in Modal Logic*, number UMCS-04-9-1 in Technical Report Series, pages 197–211. University of Manchester, 2004.

[20] H.R. Lewis and C.H. Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall, 1981.

[21] C. Lutz. Expressiveness and complexity of the logic of public announcements, 2004. Manuscript, Technical University of Dresden.

[22] R. van der Meyden. Constructing finite state implementations of knowledge based programs with perfect recall. In *Intelligent Agent Systems: Theoretical and Practical Issues*, volume 1209 of *Lecture Notes in Artificial Intelligence*, pages 135–152. Springer, 1997.

[23] J.-J.Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. Cambridge University Press, Cambridge, 1995.

[24] J.S. Miller and L.S Moss. The undecidability of iterated modal relativization. Technical report, Dept of Computer Science, Indiana University, 2003.

[25] R. Parikh. The completeness of propositional dynamic logic. In *Mathematical Foundations of Computer Science 1978*, pages 403–415. Springer, 1978.

[26] J. A. Plaza. Logics of public communications. In M. L. Emrich, M. S. Pfeifer, M. Hadzikadic, and Z. W. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.

[27] J. Ruan. Exploring the update universe. Master's thesis, ILLC, Amsterdam, 2004.

[28] K. Segerberg. A completeness theorem in the modal logic of programs. In T. Traczyck, editor, *Universal Algebra and Applications*, pages 36–46. Polish Science Publications, 1982.