

## Article

# Logics for Epistemic Actions: Completeness, Decidability, Expressivity<sup>†</sup>

Alexandru Baltag<sup>1</sup>, Lawrence S. Moss<sup>2,\*</sup> and Sławomir Solecki<sup>3</sup>

<sup>1</sup> Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 1090 GE Amsterdam, The Netherlands; thealexandrubaltag@gmail.com

<sup>2</sup> Mathematics Department, Indiana University, Bloomington, IN 47405-7106, USA

<sup>3</sup> Mathematics Department, Cornell University, Ithaca, NY 14853, USA; ss3777@cornell.edu

\* Correspondence: lmoss@indiana.edu

<sup>†</sup> This paper was intended to be the “journal version” of our 1998 paper “The logic of common knowledge, public announcements, and private suspicions”. It was mainly written in 2004, with a few bibliographic additions coming a few years later. Several pages were rewritten in early 2023.

**Abstract:** We build and study dynamic versions of epistemic logic. We study languages parameterized by an *action signature* that allows one to express epistemic actions such as (truthful) public announcements, completely private announcements to groups of agents, and more. The language  $\mathcal{L}(\Sigma)$  is modeled on dynamic logic. Its sentence-building operations include modalities for the execution of programs, and for knowledge and common knowledge. Its program-building operations include action execution, composition, repetition, and choice. We consider two fragments of  $\mathcal{L}(\Sigma)$ . In  $\mathcal{L}_1(\Sigma)$ , we drop action repetition; in  $\mathcal{L}_0(\Sigma)$ , we also drop common knowledge. We present the syntax and semantics of these languages and sound proof systems for the validities in them. We prove the strong completeness of a logical system for  $\mathcal{L}_0(\Sigma)$  and the weak completeness of one for  $\mathcal{L}_1(\Sigma)$ . We show the finite model property and, hence, decidability of  $\mathcal{L}_1(\Sigma)$ . We translate  $\mathcal{L}_1(\Sigma)$  into PDL, obtaining a second proof of decidability. We prove results on expressive power, comparing  $\mathcal{L}_1(\Sigma)$  with modal logic together with transitive closure operators. We prove that a logical language with operators for private announcements is more expressive than one for public announcements.

**Keywords:** dynamic epistemic logic; update; public announcement; common knowledge; completeness; expressive power



**Citation:** Baltag, A.; Moss, L.S.; Solecki, S. Logics for Epistemic Actions: Completeness, Decidability, Expressivity. *Logics* **2023**, *1*, 97–147. <https://doi.org/10.3390/logics1020006>

Academic Editor: Barteld Kooi

Received: 15 June 2022

Revised: 3 March 2023

Accepted: 18 April 2023

Published: 12 June 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

One of the goals of *dynamic epistemic logic* is to construct logical languages which allow one to represent a variety of possible types of changes affecting the information states of agents in a multi-agent setting. One wants (formal) logical systems with primitive operations corresponding to (informal) notions such as *public announcement*, *completely private announcement*, and *private announcement to one agent with suspicion by another*, etc. And then, after the logics are formulated, one would ideally want technical tools to use in their study and application.

The full formulation of such logics is somewhat of a complicated story. In the first place, one needs a reasonable syntax. On the semantic side, there is an unusual feature in that the truth of a sentence at a point in one model often depends on the truth of a related sentence in a *different* model. In effect, the kinds of actions we are interested in give rise to functions, or relations more generally, on the class of all possible models.

There have been some proposals on logical systems for dynamic epistemic logic beginning with the work of Plaza [1], Gerbrandy [2,3], and Gerbrandy and Groeneveld [4]. These papers formulated logical systems for the informal notions of public announcement and completely private announcement. They left open the matter of axiomatizing the logics in the presence of common-knowledge operators. (Without the common-knowledge

operators, the systems are seen to be variants on standard multi-model logic.) And they also left open the question of the decidability of these systems. Our work began with complete axiomatizations for these systems and for more general systems. Our results were presented in [5]. As it happens, the logics which we constructed in [5] did not have the friendliest syntax. The first two authors pursued this matter for some time and eventually came to the proposals in [6]. The logical systems for the validities of the ultimate languages are certainly related to those in [5]. But due to the different formulation of the overall syntax and semantics, all of the work on soundness and completeness had to be completely reworked. The main purpose of the present paper is to present these results. Secondly, we present some technical results on the systems such as results on expressive power.

### 1.1. Contents of This Paper: A High-Level View

This paper is a long technical development, and for this reason we did not include a full-scale motivation of the logical systems themselves. For a great deal of the motivational material, one should see [5]. Section 2 formulates all of the definitions needed in the paper. The presentation is based closely on the work in [6], so readers familiar with that paper may use Section 2 as a review. Other readers of this paper could take the logical systems here to simply be extensions of propositional dynamic logic which allow one to make “transitions from model to model” in addition to transitions “inside a given model”. In addition, Section 2.8 is new in this paper.

The logical languages studied in this paper are presented in Section 3. Section 3.2 presents some examples of the semantics, chosen to foreshadow work in Section 6 on expressive power. The logical system is presented in Section 4, along with a soundness result for most of the system. The completeness theorem for the logic comes in Section 5. The final section deals with questions of expressive power, and it may be read after Section 4.

Two aspects of our overall machinery are worth pointing out. The first is the use of the *canonical action model*. This is a semantic object built from syntactic objects (sequences of *simple actions*, terms in our language  $\mathcal{L}(\Sigma)$ ). We would like to think that the use of the canonical model makes for a more elegant presentation than one would otherwise have.

The second feature is a *term rewriting system* for dealing with assertions in the language. The semantic equivalences in dynamic epistemic logic are sufficiently complicated that the ‘subsentence’ relation is not the most natural or useful one for many purposes. Instead, one needs to handcraft various ordering relations for use in inductive proofs, or in defining translations. For just one example, in the logic of public announcements one has an *announcement–knowledge axiom* in a form such as

$$[!\varphi]\Box\psi \leftrightarrow (\varphi \rightarrow \Box[!\varphi]\psi). \quad (1)$$

(Here, the notation  $[!\varphi]\psi$  can be read: if  $\varphi$  is true, then after announcing it publicly,  $\psi$  is true.) This is for a logic with just one agent, and in this discussion we are forgetting about common knowledge. To prove that the logic has a translation  $t$  back to ordinary modal logic, one wants to use the equivalence in (1), above, as the key step in the translation, defining  $([\varphi]\Box\psi)^t$  to be  $\varphi^t \rightarrow (\Box[!\varphi]\psi)^t$ . But then one needs to have some reason to say that both  $\varphi$  and (more critically)  $\Box[!\varphi]\psi$  are of *lower complexity* than the original formula  $[\varphi]\Box\psi$ . There are several ways to make this precise. One is to directly assign an element of some well-founded set to each formula and then use this as a measure of complexity. This is carried out in [7], and the well-founded set is the natural numbers. Our treatment is different, mainly because it goes via term rewriting. In effect, one takes the well-founded set to be the sentences themselves, with the order given by substitution using laws such as (1), but oriented in a specified direction (left to right, for example). Then the fact that we have a well-founded relation is a result one proves about oriented sets of laws (term rewriting systems). In our case, we use an interpretation constructed by hand.

Our term rewriting system is presented and studied in Section 5.2. It actually deals not with  $\mathcal{L}(\Sigma)$  but with a different language called  $\mathcal{L}_1(\Sigma)$ . An important by-product of our completeness proof via rewriting is a Normal Form Theorem for  $\mathcal{L}_1(\Sigma)$ . This is an

interesting result in itself, and it does not follow from other completeness proofs such as those in [7]. In order to use the rewriting system, one needs to know that substitution of “equivalent” objects preserves “equivalence” and that the proof system itself is strong enough to reduce sentences to normal form. Unfortunately, this “obvious” point takes a great deal of work. The details are all in this paper, and to our knowledge no other source presents complete proofs on these matters.

### 1.2. Comparison with Some Other Work

The first version of this paper is our 1998 conference publication [5]. We ourselves had several versions of this paper issued as technical reports or posted on web sites, and so in some sense the results here are *public* but not *published*. Since its inception, the subject of dynamic epistemic logic has taken off in a serious way. It sees dozens of papers a year. But it seems fair to say that the overall topics of investigation are not the logical systems presented in any of the original papers but rather are adaptations of the logics to settings involving probability, belief revision, quantum information, and the like. Still, the work reported here has been the subject of several existing publications. And so it makes sense for us to make the case that the results in this paper are still relevant and do not follow from previously published results.

Our 2016 paper [8] is also a follow-up version of [5]. But in [8], we did not have the space to include all of the material in this paper, and so the particular logic in that paper was a fragment of the one here, with a simpler technical treatment. In addition, the first part of [8] is from [6] and goes beyond what we do here.

The book *Dynamic Epistemic Logic* by Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi [7] is a textbook presentation containing some of the content of this paper and [6], with additional material on model checking, belief revision, and other topics. Section 6.4 presents the syntax of what it calls *action model logic* and writes as  $\mathcal{L}_{KC\otimes}^{\text{act}}(A, P)$ . At first glance, the syntax of  $\mathcal{L}_{KC\otimes}^{\text{act}}(A, P)$  seems fairly close to the language  $\mathcal{L}(\Sigma)$  presented in [6] and reviewed in Section 3 of this paper. A relatively small difference concerns the *action models* in  $\mathcal{L}_{KC\otimes}^{\text{act}}(A, P)$ . (In general, action models are like Kripke models together with a “precondition” function mapping worlds to sentences.) These are required to have the property that each agent’s accessibility relation is an equivalence relation; our treatment is more general and, hence, can present logics for epistemic actions such as “cheating” in games. But the main difference is that our language  $\mathcal{L}(\Sigma)$  uses what everyone would take to be a bona-fide syntax: sentences and program expressions are linearly ordered strings of symbols. In contrast, the syntax of  $\mathcal{L}_{KC\otimes}^{\text{act}}(A, P)$  employs action models *directly*. Structured but unordered objects occur inside of sentences. In a different context, it would be like studying formal language and their connection to automata by studying something that was like a regular expression but allowed finite automata to directly occur inside some syntactic object.

We have no objection ourselves to this move. In fact, this was the presentation we chose in our first version [5] of this paper. But over the years we have found quite a lot of resistance to this presentation on the grounds that one is “mixing syntax and semantics”. Again, we do not assert this objection, but we *address* it by employing the large technical machinery that we see in [6] and also Sections 2 and 3. (We are keenly aware of the irony of our being criticized for “mixing syntax and semantics” in [5], and after we “un-mixed” them in [6], we find others making the same natural move.)

We would also like to mention the paper “Logics of Communication and Change”, van Benthem, van Eijck and Kooi [9]. This paper presents completeness theorem for a logical system in the same family as ours. But as it happens, the paper studies a different system. To see the differences, it is worthwhile to note that the source of much of the work in this area concerns assertions of common knowledge following an action of some sort. In our setting, these would be written  $[\alpha]\Box^*\varphi$ . It turns out that there is no equivalence of the form (1) for assertions such as this. This leads us to our action rule, an inference rule allowing the derivation of sentences of the kind under discussion. The idea in [9] is to start with

a more expressive ‘static’ language (than just epistemic logic with common knowledge), namely, an ‘epistemic’ version of propositional dynamic logic, called E-PDL in the paper (‘E’ for ‘epistemic’). Then one adds ‘action’ modalities corresponding to update models  $[U, e]$ ; this modality would be like our  $[\alpha]$ , but in [9] as in [5] and [7], this  $U$  is an action model rather than a bona-fide syntactic object. One can then prove that the action modalities can be translated away into the static base, via reduction axioms. This very interesting and suggestive result is not available in our more restrictive setting, but that seems to be the price for sticking with a language that has a very clear epistemic interpretation. In fact, it is not known whether E-PDL is actually stronger than the logical systems in this paper<sup>1</sup>, and if it is then it is not clear what is the epistemic use of this additional expressive power<sup>2</sup>.

Finally, we would like to mention several articles quite related to our topic which appear in online sources or in handbooks: [10–12]. These sources should be useful to anyone wishing to obtain a wider perspective on the ideas in this paper or a slightly different presentation of them.

## 2. Definitions

This section provides all of our definitions. It is short on motivation, and we have situated the examples in Section 3.2, following all the definitions. See also Sections 1 and 2 of [6] for the motivation from epistemic logic and for a more leisurely presentation.

### 2.1. State Models and Propositions

We fix a set  $\text{AtSen}$  of *atomic sentences* and also a set  $\mathcal{A}$  of *agents*. All of our definitions are relative to these sets.

A *state model* is a triple  $\mathbf{S} = (S, \xrightarrow{\cdot}_{\mathbf{S}}, \|\cdot\|_{\mathbf{S}})$  consisting of a non-empty set  $S$  of “states”; a family  $\xrightarrow{\cdot}_{\mathbf{S}}$  of binary accessibility relations  $\xrightarrow{A}_{\mathbf{S}} \subseteq S \times S$ , one for each agent  $A \in \mathcal{A}$ ; and a “valuation” (or a “truth” map)  $\|\cdot\|_{\mathbf{S}} : \text{AtSen} \rightarrow \mathcal{P}(S)$ , assigning to each atomic sentence  $p$  a set  $\|p\|_{\mathbf{S}}$  of states. These are exactly Kripke models generalized by having one accessibility relation for each agent. We use the terminology of state models because there are other kinds of models, *action models* and *program models*, in our study. When dealing with a single fixed-state model  $\mathbf{S}$ , we often drop the subscript  $\mathbf{S}$  from all the notation.

**Definition 2.1.** Let  $\text{SMod}$  be the collection of all state models. An *epistemic proposition* is an operation  $\varphi$  defined on  $\text{SMod}$  such that for all  $\mathbf{S} \in \text{SMod}$ ,  $\varphi_{\mathbf{S}} \subseteq S$ .

The collection of epistemic propositions is closed in various ways.

1. For each atomic sentence  $p$  we have an atomic proposition  $p$  with  $p_{\mathbf{S}} = \|p\|_{\mathbf{S}}$ .
2. If  $\varphi$  is an epistemic proposition, then so is  $\neg\varphi$ , where  $(\neg\varphi)_{\mathbf{S}} = S \setminus \varphi_{\mathbf{S}}$ .
3. If  $C$  is a set or class of epistemic propositions, then  $\bigwedge C$  is an epistemic proposition, where

$$(\bigwedge C)_{\mathbf{S}} = \bigcap \{\varphi_{\mathbf{S}} : \varphi \in C\}.$$

4. Taking  $C$ , above, to be empty, we have a “*universally true*” epistemic proposition  $\mathbf{tr}$ , with  $\mathbf{tr}_{\mathbf{S}} = S$ .
5. We also may take  $C$  in part 3 to be a two-element set  $\{\varphi, \psi\}$ ; here, we write  $\varphi \wedge \psi$  instead of  $\bigwedge \{\varphi, \psi\}$ . We see that if  $\varphi$  and  $\psi$  are epistemic propositions, then so is  $\varphi \wedge \psi$ , with  $(\varphi \wedge \psi)_{\mathbf{S}} = \varphi_{\mathbf{S}} \cap \psi_{\mathbf{S}}$ .
6. If  $\varphi$  is an epistemic proposition and  $A \in \mathcal{A}$ , then  $\Box_A \varphi$  is an epistemic proposition, with

$$(\Box_A \varphi)_{\mathbf{S}} = \{s \in S : \text{if } s \xrightarrow{A} t, \text{ then } t \in \varphi_{\mathbf{S}}\}. \quad (2)$$

7. If  $\varphi$  is an epistemic proposition and  $\mathcal{B} \subseteq \mathcal{A}$ , then  $\Box_{\mathcal{B}}^* \varphi$  is an epistemic proposition, with

$$(\Box_{\mathcal{B}}^* \varphi)_{\mathbf{S}} = \{s \in S : \text{if } s \xrightarrow{\mathcal{B}^*} t, \text{ then } t \in \varphi_{\mathbf{S}}\}.$$

Here  $s \xrightarrow{B^*} t$  iff there is a sequence

$$s = u_0 \xrightarrow{A_1} u_1 \xrightarrow{A_2} \dots \xrightarrow{A_n} u_{n+1} = t$$

where  $A_1, \dots, A_n \in \mathcal{B}$ . In other words, there is a sequence of arrows labelled with agents from the set  $\mathcal{B}$  taking  $s$  to  $t$ . We allow  $n = 0$  here, so  $\xrightarrow{B^*}$  includes the identity relation on  $S$ .

### Syntactic and Semantic Notions

It will be important for us to make a sharp distinction between syntactic and semantic notions. We have already begun to do this, speaking of atomic *sentences* and atomic *propositions*. The difference for us is that atomic sentences are entirely syntactic objects: we will not treat an atomic sentence  $p$  as anything except an unanalyzed mathematical object. On the other hand, this atomic sentence  $p$  also has associated with it the atomic proposition  $p$ . As defined in point 1,  $p$  will be a function whose domain is the (proper class of) state models, and it is defined by

$$p_S = \{s \in S : s \in \|p\|_S\}. \quad (3)$$

This difference may seem pedantic at first, and surely there are times when it is sensible to blur it. But for various reasons that will hopefully become clear, we need to insist on it.

Up until now, the only syntactic objects have been the atomic sentences  $p \in \text{AtSen}$ . But we can build the collections of *finitary* and *infinitary sentences* by the same definitions that we have seen, and then the work of the past section is the *semantics* of our logical languages. For example, we have sentences  $p \wedge q$ ,  $\Box_A \neg p$ , and  $\Box_B^* q$ . These then have corresponding epistemic propositions as their semantics:  $p \wedge q$ ,  $\Box_A \neg p$ , and  $\Box_B^* q$ , respectively. Note that the latter is a properly infinitary proposition (and so  $\Box_B^* q$  is a properly infinitary sentence); it abbreviates the infinite conjunction

$$p \wedge \Box_B p \wedge \Box_B \Box_B p \wedge \dots$$

### 2.2. Updates

A *transition relation* between state models  $\mathbf{S}$  and  $\mathbf{T}$  is a relation between the sets  $S$  and  $T$ ; i.e., a subset of  $S \times T$ . An *update*  $\mathbf{r}$  is a pair of operations

$$\mathbf{r} = (\mathbf{S} \mapsto \mathbf{S}(\mathbf{r}), \mathbf{S} \mapsto \mathbf{r}_S),$$

the first takes each  $\mathbf{S} \in \text{SMod}$  to a state model  $\mathbf{S}(\mathbf{r})$ , and the second takes each  $\mathbf{S} \in \text{SMod}$  to a relation  $\mathbf{r}_S$  between  $S$  and  $\mathbf{S}(\mathbf{r})$ . For the second map, we write  $\mathbf{r}_S : S \rightarrow \mathbf{S}(\mathbf{r})$ . We call  $\mathbf{S} \mapsto \mathbf{S}(\mathbf{r})$  the *update map*, and  $\mathbf{S} \mapsto \mathbf{r}_S$  the *update relation*.

We continue our general discussion by noting that the collection of updates is closed in various ways.

1. *Skip and Crash*: there exist updates called “Skip”, denoted by  $\mathbf{1}$ , and “Crash”, denoted by  $\mathbf{0}$ . Both keep the original state model unchanged  $\mathbf{S}(\mathbf{1}) = \mathbf{S}(\mathbf{0}) = \mathbf{S}$ , but they differ in the fact that  $\mathbf{1}_S$  is the identity relation on  $S$ , while  $\mathbf{0}_S$  is the empty relation.
2. *Sequential composition*: if  $\mathbf{r}$  and  $\mathbf{s}$  are epistemic updates, then their composition  $\mathbf{r}; \mathbf{s}$  is again an epistemic update, where  $\mathbf{S}(\mathbf{r}; \mathbf{s}) = \mathbf{S}(\mathbf{r})(\mathbf{s})$ , and  $(\mathbf{r}; \mathbf{s})_S = \mathbf{r}_S; \mathbf{s}_{\mathbf{S}(\mathbf{r})}$ . Here, we use on the right side the usual composition  $;$  of relations<sup>3</sup>.
3. *Union (or non-deterministic choice)*: If  $X$  is any non-empty set of epistemic updates, then the union  $\sqcup X$  is an epistemic update, defined as follows. For each  $\mathbf{S}$ , the set of states of the model  $\mathbf{S}(\sqcup X)$  is the *disjoint union* of all the sets of states in each model  $\mathbf{S}(\mathbf{r})$  for  $\mathbf{r} \in X$ :

$$\{(s, \mathbf{r}) : \mathbf{r} \in X \text{ and } s \in \mathbf{S}(\mathbf{r})\}.$$



Similarly, each accessibility relation  $\overset{A}{\rightarrow}$  is defined as the disjoint union of the corresponding accessibility relations in each model:

$$(t, \mathbf{r}) \overset{A}{\rightarrow} (u, \mathbf{s}) \quad \text{iff} \quad \text{if } \mathbf{r} = \mathbf{s} \text{ and } t \overset{A}{\rightarrow} u \text{ in } \mathbf{S}(\mathbf{r}).$$

The valuation  $\|p\|_{\mathbf{S}(\sqcup X)}$  in  $\mathbf{S}(\sqcup X)$  is the disjoint union of the valuations in each state model:

$$\|p\|_{\mathbf{S}(\sqcup X)} = \{(s, \mathbf{r}) : \mathbf{r} \in X \text{ and } s \in \|p\|_{\mathbf{S}(\mathbf{r})}\}.$$

Finally, the update relation  $(\sqcup X)_{\mathbf{S}}$  between  $\mathbf{S}$  and  $\mathbf{S}(\sqcup X)$  is the union of all the update relations  $\mathbf{r}_{\mathbf{S}}$ :

$$t (\sqcup X)_{\mathbf{S}} (u, \mathbf{r}) \quad \text{iff} \quad t \mathbf{r}_{\mathbf{S}} u.$$

For the empty set of updates  $X = \emptyset$ , we conventionally put  $\sqcup \emptyset := \mathbf{0}$  to be the above-defined “crash” update.

4. Special case: *binary union*. The (disjoint) union of two epistemic updates  $\mathbf{r}$  and  $\mathbf{s}$  is an update  $\mathbf{r} \sqcup \mathbf{s}$ , given by  $\mathbf{r} \sqcup \mathbf{s} = \sqcup \{\mathbf{r}, \mathbf{s}\}$ .
5. Another special case: *Kleene star (iteration)*. We have the operation of Kleene star on updates:

$$\mathbf{r}^* = \sqcup \{\mathbf{1}, \mathbf{r}, \mathbf{r} \cdot \mathbf{r}, \dots, \mathbf{r}^n, \dots\}$$

where  $\mathbf{r}^n$  is recursively defined by  $\mathbf{r}^0 = \mathbf{1}$ ,  $\mathbf{r}^{n+1} = \mathbf{r}^n \cdot \mathbf{r}$ .

The operations  $\mathbf{r}; \mathbf{s}$ ,  $\mathbf{r} \sqcup \mathbf{s}$  and  $\mathbf{r}^*$  are the natural analogues of the operations of union of relations, relational composition and iteration, and of the regular operations on programs in PDL. The intended meanings are: for  $\mathbf{r}; \mathbf{s}$ , sequential composition (*do  $\mathbf{r}$ , then do  $\mathbf{s}$* ); for  $\mathbf{r} \sqcup \mathbf{s}$ , non-deterministic choice (*do either  $\mathbf{r}$  or  $\mathbf{s}$* ); for  $\mathbf{r}^*$ , iteration (*repeat  $\mathbf{r}$  some finite number of times*).

### 2.2.1. Standard Updates

An update  $\mathbf{a}$  is *standard* if for all state models  $\mathbf{S}$ ,  $(\mathbf{a}_{\mathbf{S}})^{-1}$  is a partial function. (The relation  $(\mathbf{a}_{\mathbf{S}})^{-1}$  is the inverse of the update relation; it is a subset of  $\mathbf{S}(\mathbf{a}) \times \mathbf{S}$ .)

**Proposition 2.2.** *The update  $\mathbf{1}$  is standard. The composition of standard updates is standard, as is any union of standard updates.*

### 2.2.2. Updates Determine Dynamic Modalities

If  $\varphi$  is an epistemic proposition and  $\mathbf{r}$  an update, then  $[\mathbf{r}]\varphi$  is an epistemic proposition defined by

$$([\mathbf{r}]\varphi)_{\mathbf{S}} = \{s \in S : \text{for all } t \text{ such that } s \mathbf{r}_{\mathbf{S}} t, t \in \varphi_{\mathbf{S}(\mathbf{r})}\}. \quad (4)$$

We should compare (4) and (2). The point is that we may treat updates in a similar manner to other box-like modalities; the structure given by an update allows us to do this.

We also define the dual proposition  $\langle \mathbf{r} \rangle \varphi$  by

$$(\langle \mathbf{r} \rangle \varphi)_{\mathbf{S}} = \{s \in S : \text{for some } t \text{ such that } s \mathbf{r}_{\mathbf{S}} t, t \in \varphi_{\mathbf{S}(\mathbf{r})}\}.$$

### 2.3. Action Models and Program Models

Let  $\Phi$  be the collection of all epistemic propositions. An (*epistemic*) *action model* is a triple  $\Sigma = (\Sigma, \overset{A}{\rightarrow}, \text{pre})$ , where  $\Sigma$  is a non-empty set of *simple actions*,  $\overset{A}{\rightarrow}$  is an  $\mathcal{A}$ -indexed family of binary relations on  $\Sigma$ , and  $\text{pre} : \Sigma \rightarrow \Phi$ .

To model non-deterministic actions and non-simple actions (whose appearances to agents are not uniform on states), we define *epistemic program models*. In effect, this means that we decompose complex actions (“programs”) into “simple” ones: they correspond to sets of simple, deterministic actions from a given action model.

A *program model* is defined as a pair  $\pi = (\Sigma, \Gamma)$  consisting of an action model  $\Sigma$  and a set  $\Gamma \subseteq \Sigma$  of *distinguished simple actions*. Each of the simple actions  $\gamma \in \Gamma$  may be thought of

as a possible “deterministic resolution” of the non-deterministic action  $\pi$ . As announced above, the intuition about the map called  $\text{pre}$  is that an action is executable in a given state only if all its preconditions hold at that state. We often spell out an epistemic program model as  $(\Sigma, \xrightarrow{A}, \text{pre}, \Gamma)$  rather than  $((\Sigma, \xrightarrow{A}, \text{pre}), \Gamma)$ . Also, we usually drop the word “epistemic” and just refer to these as *program models*.

#### 2.4. The Update Product

Given a state model  $\mathbf{S} = (S, \xrightarrow{A}_{\mathbf{S}}, \|\cdot\|_{\mathbf{S}})$  and an action model  $\Sigma = (\Sigma, \xrightarrow{A}, \text{pre})$ , we define their *update product* to be the state model

$$\mathbf{S} \otimes \Sigma = (S \otimes \Sigma, \xrightarrow{A}, \|\cdot\|_{\mathbf{S} \otimes \Sigma}),$$

given by the following: the new states are pairs of old states  $s$  and simple actions  $\sigma$  which are “consistent”, in the sense that all preconditions of the action  $\sigma$  “hold” at the state  $s$

$$S \otimes \Sigma = \{(s, \sigma) \in S \times \Sigma : s \in \text{pre}(\sigma)_{\mathbf{S}}\}. \quad (5)$$

The new accessibility relations are taken to be the “products” of the corresponding accessibility relations in the two frames; i.e., for  $(s, \sigma), (s', \sigma') \in \mathbf{S} \otimes \Sigma$  we put

$$(s, \sigma) \xrightarrow{A} (s', \sigma') \quad \text{iff} \quad s \xrightarrow{A} s' \text{ and } \sigma \xrightarrow{A} \sigma', \quad (6)$$

and the new valuation map  $\|\cdot\|_{\mathbf{S} \otimes \Sigma} : \text{AtSen} \rightarrow \mathcal{P}(S \otimes \Sigma)$  is essentially given by the old valuation:

$$\|p\|_{\mathbf{S} \otimes \Sigma} = \{(s, \sigma) \in S \otimes \Sigma : s \in \|p\|_{\mathbf{S}}\}. \quad (7)$$

#### 2.5. Updates Induced by Program Models

Recall that we defined updates in Section 2.2. And above, in Section 2.3, we defined epistemic *program models*. Note that there is a big difference: the updates are pairs of operations on the class of all state models, and the program models are typically finite structures. We think of *program models* as capturing specific mechanisms, or algorithms, for inducing updates. This connection is made precise in the following definition.

**Definition 2.3.** Let  $(\Sigma, \Gamma)$  be a program model. We define an update which we also denote  $(\Sigma, \Gamma)$  as follows:

1.  $\mathbf{S}(\Sigma, \Gamma) = \mathbf{S} \otimes \Sigma$ .
2.  $s(\Sigma, \Gamma)_{\mathbf{S}}(t, \sigma)$  if  $s = t$  and  $\sigma \in \Gamma$ .

We call this *the update induced by  $(\Sigma, \Gamma)$* .

Note that updates of the form  $(\Sigma, \Gamma)$  have the property that for all state models  $\mathbf{S}$ , the inverse of the update relation  $(\Sigma, \Gamma)_{\mathbf{S}}$  is a partial function. That is, these updates are standard.

#### 2.6. Operations on Program Models

##### 2.6.1. **1** and **0**

We define program models **1** and **0** as follows: for both of them we take our underlying action model to be given by any one-action set  $\{\sigma\}$  with  $\sigma \xrightarrow{A} \sigma$  for all  $A$ ,  $\text{PRE}(\sigma) = \text{tr}$ ; the only difference is that for **1** we take the distinguished set  $\Gamma_{\mathbf{1}} = \{\sigma\}$ , while for **0** we put  $\Gamma_{\mathbf{0}} = \emptyset$ . The point here is that the update induced by the program model for **1** is equivalent to the update **1** from Section 2.2, and, similarly, the update induced by the program model for **0** is what we called **0** in Section 2.2. That explains why we purposely use the same notation for these program models as for the corresponding updates.

### 2.6.2. Sequential Composition

In all settings involving “actions” in some sense or other, sequential composition is a natural operation. In our setting, we would like to define a composition operation on program models, corresponding to the sequential composition of updates. Here is the relevant definition.

Let  $\Sigma = (\Sigma, \xrightarrow{A}, \text{pre}_\Sigma, \Gamma_\Sigma)$  and  $\Delta = (\Delta, \xrightarrow{A}, \text{pre}_\Delta, \Gamma_\Delta)$  be program models. We define the *composition*

$$\Sigma; \Delta = (\Sigma \times \Delta, \xrightarrow{A}, \text{pre}_{\Sigma; \Delta}, \Gamma_{\Sigma; \Delta})$$

to be the following program model:

1.  $\Sigma \times \Delta$  is the cartesian product of the sets  $\Sigma$  and  $\Delta$ .
2.  $\xrightarrow{A}$  in the composition  $\Sigma; \Delta$  is the family of product relations, in the natural way:

$$(\sigma, \delta) \xrightarrow{A} (\sigma', \delta') \quad \text{iff} \quad \sigma \xrightarrow{A} \sigma' \text{ and } \delta \xrightarrow{A} \delta'.$$

3.  $\text{pre}_{\Sigma; \Delta}(\sigma, \delta) = \langle (\Sigma, \sigma) \rangle \text{pre}_\Delta(\delta)$ .
4.  $\Gamma_{\Sigma; \Delta} = \Gamma_\Sigma \times \Gamma_\Delta$ .

In the definition of  $\text{pre}$ ,  $(\Sigma, \sigma)$  is an abbreviation for the induced update  $(\Sigma, \{\sigma\})$ , as defined in Section 2.5.

### 2.6.3. Unions

If  $\Sigma = (\Sigma, \xrightarrow{A}, \text{pre}_\Sigma, \Gamma_\Sigma)$  and  $\Delta = (\Delta, \xrightarrow{A}, \text{pre}_\Delta, \Gamma_\Delta)$ , we take  $\Sigma \sqcup \Delta$  to be the disjoint union of the models, with the union of the distinguished actions. The intended meaning is the *non-deterministic choice* between the programs represented by  $\Sigma$  and  $\Delta$ . Here is the definition in more detail, generalized to arbitrary (possibly infinite) disjoint unions: let  $\{\Sigma_i\}_{i \in I}$  be a family of program models, with  $\Sigma_i = (\Sigma_i, \xrightarrow{A}_i, \text{pre}_i, \Gamma_i)$ ; we define their (*disjoint*) *union*

$$\bigsqcup_{i \in I} \Sigma_i = \left( \bigsqcup_{i \in I} \Sigma_i, \xrightarrow{A}, \text{pre}_i, \Gamma_i \right)$$

to be the model given by:

1.  $\bigsqcup_{i \in I} \Sigma_i$  is  $\bigcup_{i \in I} (\Sigma_i \times \{i\})$ , the disjoint union of the sets  $\Sigma_i$ .
2.  $(\sigma, i) \xrightarrow{A} (\tau, j)$  if  $i = j$  and  $\sigma \xrightarrow{A}_i \tau$ .
3.  $\text{pre}(\sigma, i) = \text{pre}_i(\sigma)$ .
4.  $\Gamma = \bigcup_{i \in I} (\Gamma_i \times \{i\})$ .

### 2.6.4. Iteration

Finally, we define an iteration operation by  $\Sigma^* = \bigsqcup \{\Sigma^n : n \in \mathbb{N}\}$ . Here,  $\Sigma^0 = \mathbf{1}$  and  $\Sigma^{n+1} = \Sigma^n; \Sigma$ .

Our definition of the operations on program models are faithful to the corresponding operations on updates from Section 2.2.

**Proposition 2.4 ([6]).** *The update induced by a composition of program models is isomorphic to the composition of the induced updates. Similarly, for sums and iteration, mutatis mutandis.*

Since we shall not use this result, we omit the proof.

### 2.7. Action Signatures

**Definition 2.5.** An *action signature* is a structure

$$\Sigma = (\Sigma, \xrightarrow{A}, (\sigma_1, \sigma_2, \dots, \sigma_n))$$

where  $\Sigma = (\Sigma, \xrightarrow{A})$  is a finite Kripke frame, and  $\sigma_1, \sigma_2, \dots, \sigma_n$  is an enumeration of  $\Sigma$  in a list without repetitions. We call the elements of  $\Sigma$  *action types*. When we deal with an action



signature, our notation for the action types usually includes a subscript (even though this is occasionally redundant); thus, the action types come with a number that indicates their position in the fixed enumeration of the action signature.

An action signature  $\Sigma$  together with an assignment of epistemic propositions to the action types in  $\Sigma$  gives us a full-fledged action model. And this is the exact sense in which an *action signature* is an abstraction of the notion of *action model*. We shall use action signatures in constructing logical languages.

**Example 2.6.** Here is a very simple action signature which we call  $\Sigma_{\text{skip}}$ .  $\Sigma$  is a singleton  $\{\text{skip}\}$ ,  $\text{pre}(\text{skip}) = \text{tr}$ , and  $\text{skip} \xrightarrow{A} \text{skip}$  for all agents  $A$ . In a sense, which we shall make clear later, this is an action in which “nothing happens”, and, moreover, it is common knowledge that this is the case.

The next simplest action signature is the “test” signature  $\Sigma_?$ . We take  $\Sigma_? = \{?, \text{skip}\}$ , with the enumeration  $?, \text{skip}$ . We also take  $? \xrightarrow{A} \text{skip}$ , and  $\text{skip} \xrightarrow{A} \text{skip}$  for all  $A$ . This turns out to be a totally opaque form of test:  $\varphi$  is tested on the real world, but nobody knows this is happening. Its function will be to generate tests  $?\varphi$ , which affect the states precisely in the way dynamic logic tests do.

For each set  $B \subseteq A$  of agents, we define the action signature  $\text{Pri}^B$  of completely private announcements to the group  $B$ . It has  $\Sigma = \{\text{Pri}^B, \text{skip}\}$ ;  $\text{Pri}^B \xrightarrow{B} \text{Pri}^B$  for all  $B \in B$ ,  $\text{Pri}^B \xrightarrow{C} \text{skip}$  for  $C \notin B$ , and  $\text{skip} \xrightarrow{A} \text{skip}$  for all agents  $A$ .

The action signature  $\text{Pri}$  of all completely private announcements (to arbitrary subgroups) can be thought of as the disjoint union of all the action signatures  $\text{Pri}^B$  with  $B \subseteq A$ . However, we can avoid some redundancy by identifying all the  $\text{skip}$  actions regardless of which signature  $\text{Pri}^B$  they come from: in other words, we set  $\Sigma = \{\text{Pri}^B : B \subseteq A\} \cup \{\text{skip}\}$ , with  $\text{Pri}^B \xrightarrow{B} \text{Pri}^B$  for all  $B \in B \subseteq A$ ,  $\text{Pri}^B \xrightarrow{C} \text{skip}$  for  $C \notin B \subseteq A$ , and  $\text{skip} \xrightarrow{A} \text{skip}$  for all agents  $A$ .

Next, we consider the action signature  $\text{Prss}_k^B$  of private announcements to the group  $B$  with secure suspicion of  $k$  possible announcements by the outsiders. It has the following components:  $\Sigma = \{1, 2, \dots, k\} \cup \{1', 2', \dots, k'\} \cup \{\text{skip}\}$  (we assume these sets to be disjoint);  $i \xrightarrow{B} i'$  for all  $B \in B$  and all  $i \leq k$ ;  $i \xrightarrow{C} j$  for all  $i, j \leq k$  and  $C \notin B$ ;  $i' \xrightarrow{B} i'$  for  $i \leq k$  and  $B \in B$ ;  $i' \xrightarrow{C} \text{skip}$  for all  $i \leq k$  and  $C \notin B$ ; and, finally,  $\text{skip} \xrightarrow{A} \text{skip}$  for all agents  $A$ .

The action signature  $\text{Cka}_k^B$  is given by:  $\Sigma = \{1, \dots, k\}$ ;  $i \xrightarrow{B} i$  for  $i \leq k$  and  $B \in B$ ; and, finally,  $i \xrightarrow{C} j$  for  $i, j \leq k$  and  $C \notin B$ . This action signature is called the signature of common knowledge of alternatives for an announcement to the group  $B$ .

### 2.7.1. Signature-Based Program Models

Let  $\Sigma$  be an action signature, let  $n$  be the number of action types in  $\Sigma$ , let  $\Gamma \subseteq \Sigma$ , and let  $\vec{\psi} = \psi_1, \dots, \psi_n$  be a list of epistemic propositions. We obtain a program model  $(\Sigma, \Gamma, \vec{\psi})$  in the following way:

1. The set of simple actions is  $\Sigma$ , and the accessibility relations are those given by the action signature.
2. For  $j = 1, \dots, n$ ,  $\text{pre}(\sigma_j) = \psi_j$ .
3. The set of distinguished actions is  $\Gamma$ .

In the special case that  $\Gamma$  is the singleton set  $\{\sigma_i\}$ , we write the resulting signature-based program model as  $(\Sigma, \sigma_i, \vec{\psi})$ .

Finally, recall from Section 2.5 that every signature-based program model induces an update.

To summarize: every action signature, set of distinguished action types in it, and tuple of epistemic propositions gives a program model in a canonical way. Every program model induces a standard update.

As examples, note that both our program model **1** for “Skip” and our program model **0** for “Crash” (as defined above) are signature-based: indeed, the signature is  $\Sigma_{\text{skip}}$  as defined

above (having only one action type *skip*, with loops for all agents and true precondition), while the designated sets of actions are  $\Gamma_1 = \{\text{skip}\}$  and, respectively,  $\Gamma_0 = \emptyset$ .

### 2.8. Bisimulation-Based Notions of Equivalence

In this section, we discuss natural notions of equivalence for some of the definitions which we have already seen. We begin by recalling the most important notion of equivalence for state models, *bisimulation*.

**Definition 2.7.** Let  $\mathbf{S}$  and  $\mathbf{T}$  be state models. A *bisimulation* between  $\mathbf{S}$  and  $\mathbf{T}$  is a relation  $R \subseteq S \times T$  such that whenever  $s R t$ , the following three properties hold:

1.  $s \in \llbracket p \rrbracket_{\mathbf{S}}$  if  $t \in \llbracket p \rrbracket_{\mathbf{T}}$  for all atomic sentences  $p$ .
2. For  $A \in \mathcal{A}$  and  $s'$  such that  $s \xrightarrow{A} s'$ , there is some  $t'$  such that  $t \xrightarrow{A} t'$  and  $s' R t'$ .
3. For  $A \in \mathcal{A}$  and  $t'$  such that  $t \xrightarrow{A} t'$ , there is some  $s'$  such that  $s \xrightarrow{A} s'$  and  $s' R t'$ .

$R$  is a *total bisimulation* if it is a bisimulation and, in addition: for all  $s \in S$  there is some  $t \in T$  such that  $s R t$ , and vice-versa.

**Proposition 2.8.** If there is a bisimulation  $R$  such that  $s R t$ , then  $s$  and  $t$  agree on all sentences  $\varphi$  in infinitary modal logic:  $s \in \llbracket \varphi \rrbracket_{\mathbf{S}}$  if  $t \in \llbracket \varphi \rrbracket_{\mathbf{T}}$ .

Recall that  $\text{SMod}$  is the class of all state models. We have spoken of states as the elements of state models, but we also use the term *states* to refer to the pointed-model pairs  $(\mathbf{S}, s)$ , with  $\mathbf{S} \in \text{SMod}$  and  $s \in S$ . The class of all states is itself a state model, except that its collection of states is a proper class rather than a set. Still, it makes sense to talk about bisimulation relations on the class of all states. The largest such is given by

$$(\mathbf{S}, s) \equiv (\mathbf{T}, t) \quad \text{iff} \quad \text{there is a bisimulation } R \text{ between } \mathbf{S} \text{ and } \mathbf{T} \text{ such that } s R t.$$

This relation  $\equiv$  is indeed an equivalence relation on ‘states’ (pointed models), called the *bisimilarity* relation. When  $\mathbf{S}$  and  $\mathbf{T}$  are clear from the context, we write  $s \equiv t$  instead of  $(\mathbf{S}, s) \equiv (\mathbf{T}, t)$ .

#### 2.8.1. Equivalence and Preservation by Bisimulation

Since we are discussing notions of equivalence here, it makes sense to also think about epistemic propositions. Two propositions  $\varphi$  and  $\psi$  are equal if they are the same operation on  $\text{SMod}$ . That is, for all  $\mathbf{S}$ ,  $\varphi_{\mathbf{S}} = \psi_{\mathbf{S}}$ . Later, we shall introduce syntactically defined languages and also proof systems to go with them; in due course we shall see other interesting notions of equivalence.

Moving towards a connection of propositions with bisimulation, we say that a proposition  $\varphi$  is *preserved by bisimulations* if whenever  $(\mathbf{S}, s) \equiv (\mathbf{T}, t)$ , then  $s \in \varphi_{\mathbf{S}}$  iff  $t \in \varphi_{\mathbf{T}}$ .

**Proposition 2.9 ([6]).** The propositions which are preserved by bisimulation include  $\mathbf{tr}$  and the atomic propositions  $\mathbf{p}$ , and they are closed under all of the (infinitary) operations on propositions from Section 2.1.

#### 2.8.2. Equivalence of Bisimulation-Preserving Updates

From now we will focus on a special class of well-behaved updates: the ones that preserve bisimulation.

**Definition 2.10.** An update  $\mathbf{r}$  *preserves bisimulations* if the following two conditions hold:

1. If  $s \mathbf{r}_{\mathbf{S}} s'$  and  $(\mathbf{S}, s) \equiv (\mathbf{T}, t)$ , then there is some  $t'$  such that  $t \mathbf{r}_{\mathbf{T}} t'$  and  $(\mathbf{S}(\mathbf{r}), s') \equiv (\mathbf{T}(\mathbf{r}), t')$ .
2. If  $t \mathbf{r}_{\mathbf{T}} t'$  and  $(\mathbf{S}, s) \equiv (\mathbf{T}, t)$ , then there is some  $s'$  such that  $s \mathbf{r}_{\mathbf{S}} s'$  and  $(\mathbf{S}(\mathbf{r}), s') \equiv (\mathbf{T}(\mathbf{r}), t')$ .

An action model  $\Sigma$  *preserves bisimulations* if the epistemic proposition  $\text{pre}(\sigma)$  is preserved under bisimulations for all  $\sigma \in \Sigma$ .

It is easy to see that the updates **1** and **0** are bisimulation-preserving, and, similarly, that the action models for the programs **1** and **0** preserve bisimulation. More examples can be produced using the following two results.

**Proposition 2.11** ([6]). *Concerning bisimulation preservation:*

1. *The bisimulation-preserving updates are closed under composition and (infinitary) unions.*
2. *If  $\varphi$  is preserved by bisimulations and  $\mathbf{r}$  preserves bisimulations, then  $[\mathbf{r}]\varphi$  is preserved by bisimulations.*

**Proposition 2.12** ([6]). *Let  $\Sigma$  be a bisimulation-preserving action model. Let  $\Gamma \subseteq \Sigma$  be arbitrary. Then, the update induced by  $(\Sigma, \Gamma)$  preserves bisimulation.*

We can now define an appropriate notion of equivalence between bisimulation-preserving updates.

**Definition 2.13.** Two bisimulation-preserving updates  $\mathbf{a}$  and  $\mathbf{b}$  are *equivalent* if the following two conditions hold:

1. If  $s \mathbf{a}_{\Sigma} s'$  then there is some  $s''$  such that  $s \mathbf{b}_{\Sigma} s''$  and  $(\mathbf{S}(\mathbf{a}), s') \equiv (\mathbf{S}(\mathbf{b}), s'')$ .
2. If  $s \mathbf{b}_{\Sigma} s''$ , then there is some  $s'$  such that  $s \mathbf{a}_{\Sigma} s'$  and  $(\mathbf{S}(\mathbf{a}), s') \equiv (\mathbf{S}(\mathbf{b}), s'')$ .

We write  $\mathbf{a} \cong \mathbf{b}$  for the update-equivalence relation.

**Proposition 2.14.** *Update equivalence  $\cong$  is an equivalence relation on the class of bisimulation-preserving updates.*

**Proof.** For reflexivity, let  $\mathbf{a}$  be a bisimulation-preserving update. We can check that  $\mathbf{a} \cong \mathbf{a}$  by applying the definition of bisimulation preservation to  $\mathbf{r} := \mathbf{a}$  and to models  $\mathbf{S} = \mathbf{T}$  and states  $s = t$ . Symmetry follows immediately from the definition of  $\cong$  together with the symmetry of the bisimilarity relation  $\equiv$ . For transitivity, assume that  $\mathbf{a} \cong \mathbf{b}$  and  $\mathbf{b} \cong \mathbf{c}$ . To show that  $\mathbf{a} \cong \mathbf{c}$ , we check only condition 1 (since the proof of condition 2 is similar). For this, suppose that we have  $s \mathbf{a}_{\Sigma} s'$ . Since  $\mathbf{a} \cong \mathbf{b}$ , there must exist some  $s''$  such that  $s \mathbf{b}_{\Sigma} s''$  and  $(\mathbf{S}(\mathbf{a}), s') \equiv (\mathbf{S}(\mathbf{b}), s'')$ . The first of these, together with the fact that  $\mathbf{b} \cong \mathbf{c}$ , implies that there must exist some  $s'''$  such that  $s \mathbf{b}_{\Sigma} s'''$  and  $(\mathbf{S}(\mathbf{b}), s'') \equiv (\mathbf{S}(\mathbf{c}), s''')$ . By the transitivity of the bisimilarity relation, we also have  $(\mathbf{S}(\mathbf{a}), s') \equiv (\mathbf{S}(\mathbf{c}), s''')$ , as desired.  $\square$

**Proposition 2.15.** *Let  $\mathbf{S}$  and  $\mathbf{T}$  be models and let  $s, t$  be states. Assume that  $(\mathbf{S}, s) \equiv (\mathbf{T}, t)$ . Let  $\mathbf{a}$  and  $\mathbf{b}$  be bisimulation-preserving updates such that  $\mathbf{a} \cong \mathbf{b}$ . Then we have the following:*

1. *If  $s \mathbf{a}_{\Sigma} s'$ , then there is some  $t'$  such that  $t \mathbf{b}_{\Sigma} t'$  and  $(\mathbf{S}(\mathbf{a}), s') \equiv (\mathbf{T}(\mathbf{b}), t')$ .*
2. *If  $t \mathbf{b}_{\Sigma} t'$ , then there is some  $s'$  such that  $s \mathbf{a}_{\Sigma} s'$  and  $(\mathbf{S}(\mathbf{a}), s') \equiv (\mathbf{T}(\mathbf{b}), t')$ .*

**Proof.** The second clause follows from the first, using the fact that the relations of update equivalence and bisimilarity are symmetric. So it is enough to check the first clause. From  $s \mathbf{a}_{\Sigma} s'$ ,  $(\mathbf{S}, s) \equiv (\mathbf{T}, t)$  and the fact that  $\mathbf{a}$  preserves bisimulation, it follows that there exists some  $t''$  such that  $t \mathbf{a}_{\Sigma} t''$  and  $(\mathbf{S}(\mathbf{a}), s') \equiv (\mathbf{T}(\mathbf{a}), t'')$ . From  $t \mathbf{a}_{\Sigma} t''$  and  $\mathbf{a} \cong \mathbf{b}$ , it follows that there exists some  $t'$  such that  $t \mathbf{b}_{\Sigma} t'$  and  $(\mathbf{T}(\mathbf{a}), t'') \equiv (\mathbf{T}(\mathbf{b}), t')$ . Putting these together and using the transitivity of the bisimilarity relation  $\equiv$ , we conclude that  $(\mathbf{S}(\mathbf{a}), s') \equiv (\mathbf{T}(\mathbf{b}), t')$ , as desired.  $\square$

**Proposition 2.16.** *Let  $\mathbf{a} \cong \mathbf{b}$  and  $\mathbf{c} \cong \mathbf{d}$ . Then,  $\mathbf{a}; \mathbf{c} \cong \mathbf{b}; \mathbf{d}$ , and also  $\mathbf{a} \sqcup \mathbf{c} \cong \mathbf{b} \sqcup \mathbf{d}$ .*

**Proof.** To prove  $\mathbf{a};\mathbf{c} \cong \mathbf{b};\mathbf{d}$ , we only check the first clause in the definition of update equivalence. Suppose that  $s(\mathbf{a};\mathbf{c})_{\mathbf{S}} s'$ , i.e., there exists some  $s''$  such that  $a \mathbf{a}_{\mathbf{S}} s''$  and  $s'' \mathbf{c}_{\mathbf{S}}(\mathbf{a}) s'$ . From  $a \mathbf{a}_{\mathbf{S}} s''$  and  $\mathbf{a} \cong \mathbf{b}$ , we infer that there is some  $t''$  with  $s \mathbf{b}_{\mathbf{S}} t''$  and  $(\mathbf{S}(\mathbf{a}), s'') \equiv (\mathbf{S}(\mathbf{b}), t'')$ . By using these together with the fact that  $\mathbf{c} \cong \mathbf{d}$ , and applying Proposition 2.15, we obtain some  $t'$  such that  $t'' \mathbf{c}_{\mathbf{S}}(\mathbf{b}) t'$  and  $(\mathbf{S}(\mathbf{a})(\mathbf{c}), s') \equiv (\mathbf{S}(\mathbf{b})(\mathbf{d}), t')$ , as desired.

For the second claim, we again only prove the first clause in the definition of  $\mathbf{a} \sqcup \mathbf{c} \cong \mathbf{b} \sqcup \mathbf{d}$ . Suppose that  $s(\mathbf{a} \sqcup \mathbf{c})_{\mathbf{S}} s'$ . By the definition of  $\mathbf{a} \sqcup \mathbf{c}$ , this means that we either have  $s' = (t, \mathbf{a})$  for some  $t$  satisfying  $s \mathbf{a}_{\mathbf{S}} t$ , or else we have  $s' = (t, \mathbf{b})$  for some  $t$  satisfying  $s \mathbf{b}_{\mathbf{S}} t$ . We only treat the first case (the other case is similar), so we have  $s' = (t, \mathbf{a})$  and  $s \mathbf{a}_{\mathbf{S}} t$ . Using the fact that  $\mathbf{a} \sim \mathbf{b}$ , we infer that there exists some  $t'$  such that  $s \mathbf{b}_{\mathbf{S}} t'$  and  $(\mathbf{S}(\mathbf{a}), t) \equiv (\mathbf{S}(\mathbf{b}), t')$ . Using these and taking the state  $s'' := (t', \mathbf{b})$  in the model  $\mathbf{S}(\mathbf{b} \sqcup \mathbf{d})$ , we can easily check that we have  $s(\mathbf{b} \sqcup \mathbf{d})_{\mathbf{S}} s''$  and  $(\mathbf{S}(\mathbf{a} \sqcup \mathbf{c}), s') \equiv (\mathbf{S}(\mathbf{b} \sqcup \mathbf{d}), s'')$ , as desired.  $\square$

**Proposition 2.17.** For all bisimulation-preserving  $\mathbf{a}$ , we have  $\mathbf{a};\mathbf{1} \cong \mathbf{1};\mathbf{a} \cong \mathbf{a}$ , as well as  $\mathbf{a};\mathbf{0} \cong \mathbf{0};\mathbf{a} \cong \mathbf{0}$  and  $\mathbf{a} \sqcup \mathbf{0} \cong \mathbf{0} \sqcup \mathbf{a} \cong \mathbf{a}$ .

**Proof.** Easy verification.  $\square$

**Proposition 2.18.** Let  $\mathbf{a}$  and  $\mathbf{b}$  be standard bisimulation-preserving updates. If  $\varphi$  is preserved by bisimulations and  $\mathbf{a} \cong \mathbf{b}$ , then  $[\mathbf{a}]\varphi = [\mathbf{b}]\varphi$ , and also  $\langle \mathbf{a} \rangle \varphi = \langle \mathbf{b} \rangle \varphi$ .

**Proof.** We check the first assertion only. Fix  $\mathbf{S}$ , and let  $s \in S$  be such that  $s \in ([\mathbf{a}]\varphi)_{\mathbf{S}}$ . Let  $t \in \mathbf{S}(\mathbf{b})$  be such that  $s \mathbf{b}_{\mathbf{S}} t$ . We must show that  $t \in \varphi_{\mathbf{S}(\mathbf{b})}$ .

For this, we use  $s \mathbf{b}_{\mathbf{S}} t$  and  $\mathbf{a} \cong \mathbf{b}$ , and by the definition of update-equivalence we infer that there exists some  $t'$  such that  $s \mathbf{a}_{\mathbf{S}} t'$  and  $(\mathbf{S}(\mathbf{a}), t') \equiv (\mathbf{S}(\mathbf{b}), t)$ . From  $s \in ([\mathbf{a}]\varphi)_{\mathbf{S}}$  and  $s \mathbf{a}_{\mathbf{S}} t'$ , it follows that  $t' \in \varphi_{\mathbf{S}(\mathbf{a})}$ . This, together with  $(\mathbf{S}(\mathbf{a}), t') \equiv (\mathbf{S}(\mathbf{b}), t)$  and the fact that  $\varphi$  is preserved by bisimulations, implies that  $t \in \varphi_{\mathbf{S}(\mathbf{b})}$ , as desired.  $\square$

**Proposition 2.19.** Let  $\mathbf{a}$  and  $\mathbf{b}$  be standard bisimulation-preserving updates. If  $\mathbf{a} \cong \mathbf{b}$ , then  $\mathbf{a}$  and  $\mathbf{b}$  have the same domain. That is, for all state models  $\mathbf{S}$ ,  $\text{dom}(\mathbf{a}_{\mathbf{S}}) = \text{dom}(\mathbf{b}_{\mathbf{S}})$ .

**Proof.** The domains are  $(\langle \mathbf{a} \rangle \mathbf{tr})_{\mathbf{S}}$  and  $(\langle \mathbf{b} \rangle \mathbf{tr})_{\mathbf{S}}$ , where recall that  $\mathbf{tr}$  is the “universally true” proposition. So the result follows from Proposition 2.18 and the fact that  $\mathbf{tr}$  is preserved by bisimulations.  $\square$

**Definition 2.20.** Let  $\Sigma = (\Sigma, \xrightarrow{\mathcal{A}}, \text{pre})$  and  $\Sigma' = (\Sigma', \xrightarrow{\mathcal{A}}, \text{pre})$  be action models. As this notation indicates, we shall not introduce additional notation to differentiate the arrows and the pre functions on these action models. A *bisimulation* between  $\Sigma$  and  $\Sigma'$  is a relation  $R \subseteq \Sigma \times \Sigma'$  such that whenever  $\sigma R \sigma'$ , the following three properties hold:

1.  $\text{pre}(\sigma) = \text{pre}(\sigma')$ .
2. For  $A \in \mathcal{A}$  and  $\rho$  such that  $\sigma \xrightarrow{A} \rho$ , there is some  $\rho'$  such that  $\sigma' \xrightarrow{A} \rho'$  and  $\rho R \rho'$ .
3. For  $A \in \mathcal{A}$  and  $\rho'$  such that  $\sigma' \xrightarrow{A} \rho'$ , there is some  $\rho$  such that  $\sigma \xrightarrow{A} \rho$  and  $\rho R \rho'$ .

We write  $\sigma \sim \sigma'$ , and say that actions  $\sigma$  and  $\sigma'$  are *bisimilar*, if there is a bisimulation  $R$  between  $\Sigma$  and  $\Sigma'$  such that  $\sigma R \sigma'$ .

Let  $\pi = (\Sigma, \Gamma)$  and  $\rho = (\Sigma', \Gamma')$  be program models. A *bisimulation* between  $\pi$  and  $\rho$  is a bisimulation  $R$  between  $\Sigma$  and  $\Sigma'$  such that the following hold:

1. If  $\sigma \in \Gamma$ , then there is some  $\sigma' \in \Gamma'$  such that  $\sigma R \sigma'$ .
2. If  $\sigma' \in \Gamma'$ , then there is some  $\sigma \in \Gamma$  such that  $\sigma R \sigma'$ .

If such a bisimulation exists, then we write  $\pi \sim \rho$ , and say that the program models  $\pi$  and  $\rho$  are *bisimilar*.

**Proposition 2.21.** *If  $\pi = (\Sigma, \Gamma)$  is any program model, **1** is the above-defined program model for “Skip” and **0** is the above-defined program model for “Crash”, then we have  $\pi; \mathbf{1} \sim \mathbf{1}$ ;  $\pi \sim \pi$ , as well as  $\pi; \mathbf{0} \sim \mathbf{0}$ ;  $\pi \sim \mathbf{0}$  and  $\pi \sqcup \mathbf{0} \sim \mathbf{0} \sqcup \pi \sim \pi$ .*

**Proof.** Easy verification.  $\square$

**Proposition 2.22.** *If two bisimulation-preserving program models are bisimilar, then they induce equivalent updates.*

**Proof.** Let  $\pi = (\Sigma, \Gamma)$  and  $\rho = (\Sigma', \Gamma')$  be program models, and let  $R$  be a bisimulation between  $\pi$  and  $\rho$ . To prove our claim, we first construct a bisimulation  $\bar{R}$  between the update products  $\mathbf{S} \otimes \Sigma$  and  $\mathbf{S} \otimes \Sigma'$ , by putting for all states  $(s, \sigma) \in \mathbf{S} \otimes \Sigma$ ,  $(s', \sigma') \in \mathbf{S} \otimes \Sigma'$ :

$$(s, \sigma) \bar{R} (s', \sigma') \text{ iff } s = s' \text{ and } \sigma R \sigma'.$$

It is easy to check that this is a bisimulation relation. The atomic preservation is obvious:  $(s, \sigma) \bar{R} (s', \sigma')$  implies that  $s = s'$ , and so both  $(s, \sigma)$  and  $(s', \sigma')$  satisfy the same atomic sentences as  $s = s'$ . For the forth clause, assume that we have  $(s, \sigma) \xrightarrow{a} (t, \rho)$  and  $(s, \sigma) \bar{R} (s', \sigma')$ . From these, we obtain  $s \xrightarrow{a} t$  and  $\sigma \xrightarrow{a} \rho$ , as well as  $s = s'$  and  $\sigma R \sigma'$ . Since  $R$  is a bisimulation between  $\Sigma$  and  $\Sigma'$ , there must exist some  $\rho' \in \Sigma'$  such that  $\sigma' \xrightarrow{a} \rho'$  and  $\rho R \rho'$ . This gives us that  $\text{pre}(\rho) = \text{pre}(\rho')$ , and since  $(t, \rho) \in \mathbf{S} \otimes \Sigma$ , we have that  $t \in \text{pre}(\rho) = \text{pre}(\rho')$ , and so the state  $(t, \rho') \in \mathbf{S} \otimes \Sigma'$  exists. This state is our witness for the forth clause: it is easy now to check that we have  $(s, \sigma') \xrightarrow{a} (t, \rho')$  and  $(t, \rho) \bar{R} (t, \rho')$ , as desired. The back condition is similar.

We can now prove that the updates  $\pi$  and  $\rho$  are equivalent. Let **a** be the update induced by  $\pi$  (via the update product), and let **b** be the update induced by  $\rho$ . We need to prove that **a**  $\cong$  **b**. For this, we only check the first condition in the definition of update equivalence (the second condition is similar). Assume that  $s \mathbf{a}_{\mathbf{S}} (s, \sigma) \in \Sigma$  with  $\sigma \in \Gamma$ . Since  $R$  is a bisimulation between  $(\Sigma, \Gamma)$  and  $(\Sigma', \Gamma')$ , there exists some  $\sigma' \in \Gamma'$  such that  $\sigma R \sigma'$ . This implies that  $\text{pre}(\sigma) = \text{pre}(\sigma')$ , and so  $(s, \sigma) \in \Sigma$  gives us  $s \in \text{pre}(\sigma) = \text{pre}(\sigma')$ ; hence, the state  $(s, \sigma') \in \mathbf{S} \otimes \Sigma'$  exists. This state is our witness for the first clause in the definition of update equivalence: it is easy now to check that we have  $s \mathbf{b}_{\mathbf{S}} (s, \sigma')$  and  $(s, \sigma) \bar{R} (s, \sigma')$ , and so we also have  $(\mathbf{S} \otimes \Sigma, (s, \sigma)) \equiv (\mathbf{S} \otimes \Sigma', (s, \sigma'))$  (since  $\bar{R}$  is a bisimulation between  $\mathbf{S} \otimes \Sigma$  and  $\mathbf{S} \otimes \Sigma'$ ), as desired.  $\square$

**Note** Bisimilarity is not the weakest (most general) relation on program models that implies update equivalence. Cf. the work in [13], in which a weaker such relation is studied (‘action emulation’). So, in a sense, our notion of program bisimilarity is too strong for the job that it was designed for. However, it has the advantage of simplicity, and it will be sufficient for our purposes. It is also a very natural notion to study in the context of the other concepts of bisimulation equivalence considered in this section.

### 3. Signature-Based Languages $\mathcal{L}(\Sigma)$ and Their Fragments

At this point, we have enough general definitions to present the syntax and semantics of our signature-based languages  $\mathcal{L}(\Sigma)$  and their important sublanguages  $\mathcal{L}_0(\Sigma)$  and  $\mathcal{L}_1(\Sigma)$ . Most of the remaining parts of this paper study these languages.

#### 3.1. Syntax and Semantics

Fix an action signature  $\Sigma$ . See Figure 1 for the syntax of a logical language  $\mathcal{L}(\Sigma)$ , together with its sublanguages  $\mathcal{L}_0(\Sigma)$  and  $\mathcal{L}_1(\Sigma)$ .

| sentences $\varphi$                         | true | $p_i$   | $\neg\varphi$  | $\varphi \wedge \psi$ | $\Box_A \varphi$  | $\Box_B^* \varphi$ | $[\pi]\varphi$ |
|---|------|---|--|-----------------------|---|--------------------|----------------|
| programs $\pi$                              | skip | crash   | $\sigma_i \psi_1 \dots \psi_n$                       | $\pi \sqcup \rho$     | $\pi; \rho$   | $\pi^*$            |                |
| $\llbracket \text{true} \rrbracket$         | =    | <b>tr</b>   | $\llbracket \text{skip} \rrbracket$                  | =                     | <b>1</b>  |                    |                |
| $\llbracket p \rrbracket$                   | =    | <b>p</b>  | $\llbracket \text{crash} \rrbracket$                 | =                     | <b>0</b>  |                    |                |
| $\llbracket \varphi \wedge \psi \rrbracket$ | =    | $\llbracket \varphi \rrbracket \wedge \llbracket \psi \rrbracket$ | $\llbracket \sigma_i \psi_1 \dots \psi_n \rrbracket$ | =                     | $(\Sigma, \sigma_i, \llbracket \psi_1 \rrbracket \dots \llbracket \psi_n \rrbracket)$ |                    |                |
| $\llbracket \neg\varphi \rrbracket$         | =    | $\neg \llbracket \varphi \rrbracket$                              | $\llbracket \pi; \rho \rrbracket$                    | =                     | $\llbracket \pi \rrbracket; \llbracket \rho \rrbracket$                               |                    |                |
| $\llbracket \Box_A \varphi \rrbracket$      | =    | $\Box_A \llbracket \varphi \rrbracket$                            | $\llbracket \pi \sqcup \rho \rrbracket$              | =                     | $\llbracket \pi \rrbracket \sqcup \llbracket \rho \rrbracket$                         |                    |                |
| $\llbracket \Box_B^* \varphi \rrbracket$    | =    | $\Box_B^* \llbracket \varphi \rrbracket$                          | $\llbracket \pi^* \rrbracket$                        | =                     | $\llbracket \pi \rrbracket^*$   |                    |                |
| $\llbracket [\pi]\varphi \rrbracket$        | =    | $\llbracket [\pi] \rrbracket \llbracket \varphi \rrbracket$       |  |                       |   |                    |                |

For  $\mathcal{L}_1(\Sigma)$ , we drop the  $\pi^*$  construct. For  $\mathcal{L}_0(\Sigma)$ , we drop the  $\pi^*$  and  $\Box_B^*$  constructs.

**Figure 1.** The language  $\mathcal{L}(\Sigma)$  and its semantics, and the fragments  $\mathcal{L}_0(\Sigma)$ , and  $\mathcal{L}_1(\Sigma)$ .

As in *PDL*, we have two sorts of syntactic objects: *sentences* and *programs*. The set of sentences and the set of programs are simultaneously defined by mutual recursions. We call programs of the form  $\sigma\psi_1 \dots \psi_n$  *basic actions*, where  $\sigma \in \Sigma$ . Here, the number  $n$  is the number of action types in  $\Sigma$ , and basic actions are formed by postfixing any such action type  $\sigma$  with an  $n$ -long string  $\vec{\psi}$  of already-formed sentences  $\psi_j$  in our language<sup>4</sup>. Arbitrary programs are recursively formed from basic actions, as well as from the programs *skip* and *crash*, by applying the standard regular program operations of *PDL*: non-deterministic choice  $\pi \sqcup \rho$ , sequential composition  $\pi; \rho$  and iteration  $\pi^*$ . In their turn, sentences are recursively built from atomic sentences  $\pi$ , by applying standard Boolean connectives, as well as modalities  $\Box_A \varphi$  (for any agent  $A$ ),  $\Box_B^* \varphi$  (for any set of agents  $B$ ) and  $[\pi]\varphi$  (for any program  $\pi$ ).

The sublanguage  $\mathcal{L}_1(\Sigma)$  is obtained by dropping iteration  $\pi^*$  from the constructs of  $\mathcal{L}(\Sigma)$ , while the sublanguage  $\mathcal{L}_0(\Sigma)$  is obtained by dropping both iteration  $\pi^*$  and the common knowledge modalities  $\Box_B^* \varphi$ .

We use the following standard abbreviations:  $\text{false} = \neg \text{true}$ ,  $\varphi \vee \psi = \neg(\neg\varphi \wedge \neg\psi)$ ,  $\varphi \rightarrow \psi = \neg(\varphi \wedge \neg\psi)$ ,  $\varphi \leftrightarrow \psi = (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ ,  $\Diamond_A \varphi = \neg \Box_A \neg \varphi$ ,  $\Diamond_B^* = \neg \Box_B^* \neg$ , and  $\langle \pi \rangle \varphi = \neg [\pi] \neg \varphi$ .

The semantics defines two operations by simultaneous recursion on  $\mathcal{L}(\Sigma)$ :

1.  $\varphi \mapsto \llbracket \varphi \rrbracket$ , taking the sentences of  $\mathcal{L}(\Sigma)$  into epistemic propositions.
2.  $\pi \mapsto \llbracket \pi \rrbracket$ , taking the programs of  $\mathcal{L}(\Sigma)$  into program models (and, hence, into induced updates).

The formal definition is given in Figure 1. The first map  $\varphi \mapsto \llbracket \varphi \rrbracket$  might be called the *truth map* for the language. When we began our study of state models, we started with a “valuation” (or a “truth” map)  $\|\cdot\|_S : \text{AtSen} \rightarrow \mathcal{P}(S)$ , assigning to each atomic sentence  $p$  a set  $\|p\|_S$  of states. The truth map here extends this to sentences and actions of  $\mathcal{L}(\Sigma)$ . The overall definition is by simultaneous recursion on  $\mathcal{L}(\Sigma)$ . We employ the standard device of speaking of the definition in terms of a temporal metaphor. That is, we think of the definition of the semantics of a sentence or action as coming “after” the semantics of its subsentences and subactions.

With one key exception, the operations on the right-hand sides are immediate applications of our general definitions of the closure conditions on epistemic propositions from Section 2.1 and the operations on program models from Section 2.6. A good example to explain this is the clause for the semantics of sentences  $[\pi]\varphi$ . Assuming that we have a program model  $\llbracket \pi \rrbracket$ , we obtain an induced update in Section 2.5 which we again denote  $\llbracket \pi \rrbracket$ . We also have an epistemic proposition  $\llbracket \varphi \rrbracket$ . We can, therefore, form the epistemic proposition  $\llbracket [\pi] \rrbracket \llbracket \varphi \rrbracket$  (see Equation (4) in Section 2.2). Note that we have overloaded the square-bracket notation; this is intentional, and we have done the same with other notation as well.

Similarly, the semantics of *skip* and *crash* are the program models **1** and **0** of Section 2.6.



We also discuss the definition of the semantics for basic actions  $\sigma_i \vec{\psi}$ . For this, recall that we have a general definition of a signature-based program model  $(\Sigma, \Gamma, \psi_1, \dots, \psi_n)$ , where  $\Gamma \subseteq \Sigma$  and the  $\psi$ 's are any epistemic propositions. What we have in the semantics of  $\sigma_i \vec{\psi}$  is the special case of this where  $\Gamma$  is the singleton  $\{\sigma_i\}$  and  $\psi_i$  is  $\llbracket \psi_i \rrbracket$ , a proposition which we have already defined when we come to define  $\llbracket \sigma_i \vec{\psi} \rrbracket$ .

### Logics Generated by Families of Signatures

Up until now, we have defined logics of the form  $\mathcal{L}(\Sigma)$  for a single action signature  $\Sigma$ . Recall also that each action signature is finite. It is also natural to think about the logic of *all* finite action signatures. That is, we want to combine all of the logics  $\mathcal{L}(\Sigma)$  even though we cannot literally combine the action signatures.

More generally, given a family  $\mathcal{S}$  of action signatures, we would like to combine all the logics  $\{\mathcal{L}(\Sigma)\}_{\Sigma \in \mathcal{S}}$  into a single logic. Let us assume the signatures  $\Sigma \in \mathcal{S}$  are mutually disjoint (otherwise, just choose mutually disjoint copies of these signatures). We define *the logic  $\mathcal{L}(\mathcal{S})$  generated by the family  $\mathcal{S}$*  in the following way: the syntax is defined by taking the same definition we had in Figure 1 for the syntax of  $\mathcal{L}(\Sigma)$ , but in which on the side of the programs we take instead as *basic actions* all expressions of the form

$$\sigma_i \psi_1 \cdots \psi_n$$

where  $\sigma \in \Sigma$ , for some signature  $\Sigma \in \mathcal{S}$ , and  $n$  is the length of the listing of non-trivial action types of  $\Sigma$ . The semantics is again given by the same definition as in Figure 1, but in which the clause about  $\sigma \psi_1 \cdots \psi_n$  refers to the appropriate signature: for every  $\Sigma \in \mathcal{S}$ , every  $\sigma \in \Sigma$ ; if  $n$  is the length of the listing of  $\Sigma$ , then

$$\llbracket \sigma_i \psi_1 \cdots \psi_n \rrbracket = (\Sigma, \sigma, \llbracket \psi_1 \rrbracket, \dots, \llbracket \psi_n \rrbracket).$$

It is important to note that, in general,  $\mathcal{L}(\mathcal{S})$  is *not* a signature-based language of the form  $\mathcal{L}(\Sigma)$  for any single (finite) signature  $\Sigma$ . However, our definition of signature-based languages has a natural generalization to *locally finite* 'signatures'<sup>5</sup>, and then one can see the languages  $\mathcal{L}(\mathcal{S})$  as being based on such generalized signatures.

### 3.2. Examples

In this section, we provide examples of the concepts from Sections 2.1–3.1. Our examples of state models are chosen with an eye towards one of the inexpressivity results in Sections 6.2 and 6.3 near the end of the paper. So we know that they may appear artificial. Still, we trust that having a suitably detailed example may help the reader.

First, we need to introduce some concrete examples of signature-based languages. The *language of public announcements*  $\mathcal{L}(\Sigma_{\text{pub}})$  is the special case of our general class of languages  $\mathcal{L}(\Sigma)$  that is obtained by taking the action signature  $\Sigma_{\text{pub}}$  of public announcements, as we defined it in Section 2.7:  $\Sigma = \{\text{Pub}\}$ ,  $\text{Pub} \xrightarrow{A} \text{Pub}$ , and  $\text{Pub} \xrightarrow{B} \text{Pub}$ . We can similarly define the language of *completely private announcements*  $\mathcal{L}(\Sigma_{\text{pri}})$  by taking the signature  $\text{Pri}$  of all private announcements, as defined in Section 2.7:  $\Sigma = \{\text{Pri}^B : B \subseteq \mathcal{A}\} \cup \{\text{skip}\}$ , with  $\text{Pri}^B \xrightarrow{B} \text{Pri}^B$  for all  $B \in \mathcal{B} \subseteq \mathcal{A}$ ,  $\text{Pri}^B \xrightarrow{C} \text{skip}$  for  $C \notin \mathcal{B} \subseteq \mathcal{A}$ , and  $\text{skip} \xrightarrow{A} \text{skip}$  for all agents  $A$ . For both public and private announcements, we can consider sublanguages  $\mathcal{L}_0(\Sigma_{\text{pub}})$ ,  $\mathcal{L}_1(\Sigma_{\text{pub}})$ ,  $\mathcal{L}_0(\Sigma_{\text{pri}})$ , and  $\mathcal{L}_1(\Sigma_{\text{pri}})$ , as above, by applying the corresponding syntactic restrictions.

In our following examples, we will more specifically fix our set  $\text{AtSen}$  of atomic sentences to be a two-element set  $\{p, q\}$ , and fix the set  $\mathcal{A}$  of agents to be the two-element set  $\{A, B\}$ .

As a first example, we consider a family of state models  $C_n$ , one for each even positive numbers  $n$ .  $C_n$  is a cycle of  $5n$  points  $a_1, \dots, a_{5n}$  arranged as follows:

$$a_1 \xleftrightarrow{A} a_2 \xleftrightarrow{B} a_3 \quad \cdots \quad a_{5n-1} \xleftrightarrow{A} a_{5n} \xleftrightarrow{B} a_1$$

Since  $n$  is even, for  $1 \leq i \leq 5$ , the connection is  $a_{in-1} \xleftrightarrow{A} a_{in} \xleftrightarrow{B} a_{in+1}$  (We are taking subscripts modulo  $5n$  here.) We also specify that  $p$  is true at all points except  $a_1$  and  $a_{2n+1}$ , and  $q$  is true *only* at  $a_{4n+1}$ .

We will look at the sentence  $\langle Pub \ p \rangle \diamond_{A,B}^* q$ , belonging to the language  $\mathcal{L}_1(\Sigma_{pub})$ , and aim to evaluate in each of the models  $C_n$ . (Without abbreviations, this sentence would be  $\neg[Pub \ p] \Box_{A,B}^* \neg q$ .) The point of this first example is to calculate  $\llbracket \langle Pub \ p \rangle \diamond_{A,B}^* q \rrbracket_{C_n}$ . This takes a few steps.

The definitions tell us that  $\llbracket p \rrbracket = p$ , and so

$$\llbracket p \rrbracket_{C_n} = p_{C_n} = \|p\|_{C_n} = \{a_i : i \neq 1, 2n+1\}.$$

We have a signature-based program model  $(\Sigma_{pub}, Pub, p)$ . (In more detail,  $\Sigma_{pub}$  is an action signature, our distinguished set  $\Gamma \subseteq \Sigma$  is  $\{Pub\}$ , and  $PRE(Pub)$  is the the proposition  $p$ .) We can take the update product of this with  $C_n$  to obtain the model  $C_n \otimes (\Sigma_{pub}, Pub, p)$ . The state set of this state model is

$$\{(a_i, Pub) : 1 \leq i \leq 5n \ \& \ i \neq 1 \ \& \ i \neq 2n+1\}.$$

The accessibilities are given by

$$\begin{aligned} (a_2, Pub) &\xleftrightarrow{B} (a_3, Pub) \xleftrightarrow{A} \dots & (a_{2n-1}, Pub) &\xleftrightarrow{A} (a_{2n}, Pub) \\ (a_{2n+1}, Pub) &\xleftrightarrow{A} (a_{2n+2}, Pub) \xleftrightarrow{B} \dots & (a_{5n-1}, Pub) &\xleftrightarrow{A} (a_{5n}, Pub) \end{aligned}$$

In this model  $p$  is true at all points, and  $q$  is true only at  $(a_{4n+1}, Pub)$ . The update relation  $(\Sigma_{pub}, Pub, p)_{C_n}$  is

$$\{(a_i, (a_i, Pub)) : 1 \leq i \leq 5n \ \& \ i \neq 1 \ \& \ i \neq 2n+1\}.$$

The intuition is that when we relativize the model to  $p$  (that is, we update the model with a public announcement of  $p$ ),  $a_1$  and  $a_{2n+1}$  disappear. The cycle breaks into two disconnected components.

To save on notation, let us write  $D_n$  for the model  $C_n \otimes (\Sigma_{pub}, Pub, p)$ . It follows that

$$\llbracket q \rrbracket_{D_n} = \|q\|_{D_n} = \{(a_{4n+1}, Pub)\},$$

and, therefore, that

$$\llbracket \diamond_{A,B}^* q \rrbracket_{D_n} = (\diamond_{A,B}^* q)_{D_n} = \{(a_i, Pub) : 2n+1 \leq i \leq 5n\}.$$

It now follows that

$$\begin{aligned} \llbracket \langle Pub \ p \rangle \diamond_{A,B}^* q \rrbracket_{C_n} &= \{a_i : (a_i, Pub) \in D_n \ \& \ (a_i, Pub) \in \llbracket \diamond_{A,B}^* q \rrbracket_{D_n}\} \\ &= \{a_i : 2n+2 \leq i \leq 5n\} \end{aligned}$$

Later, we are going to be especially interested in the points  $a_{n+1}$  and  $a_{3n+1}$ . The analysis above shows that  $a_{n+1}$  does not belong to  $\llbracket \langle Pub \ p \rangle \diamond_{A,B}^* q \rrbracket_{C_n}$ , but  $a_{3n+1}$  does belong to it. The intuition is that after we relativize the original cycle to  $p$  by deleting  $a_1$  and  $a_{2n+1}$ , there is no path from  $a_{n+1}$  to  $a_{4n+1}$ . But even after we make the deletion, there is a path from  $a_{3n+1}$  to  $a_{4n+1}$ , and this is why  $a_{3n+1}$  satisfies the sentence  $\langle Pub \ p \rangle \diamond_{A,B}^* q$ .

### A Second Example

We next consider a different example based on the signature  $Pri$  of completely private announcements.  $\mathcal{L}_1(\Sigma_{pri})$  contains the following sentence

$$\chi \equiv \langle Pri^A p, \text{true} \rangle \diamond_A^* \diamond_B \neg p \quad (8)$$

We show in Section 6.3 that  $\chi$  cannot be expressed by any sentence in  $\mathcal{L}_1(\Sigma_{pub})$  and that  $\diamond_A \chi$  is not expressible in  $\mathcal{L}_1(\Sigma_{pub})$ , even by a set of sentences. This means that there is no set  $T$  of sentences of  $\mathcal{L}_1(\Sigma_{pub})$  such that for every state,  $(S, s)$ ,  $(S, s) \models \psi$  for each  $\psi \in T$  iff  $(S, s) \models \langle Pri^A p, \text{true} \rangle \diamond_A^* \diamond_B \neg p$ . We take this inexpressibility result to be a formal confirmation that a logical system with private announcements is more powerful than a system with only public announcements.

**Definition 3.1.** For each non-zero natural number  $n$  we shall construct models  $S_n$  and  $T_n$ . These two models have the same state set:

$$\{a, b\} \cup \{c_i : 1 \leq i \leq n\}.$$

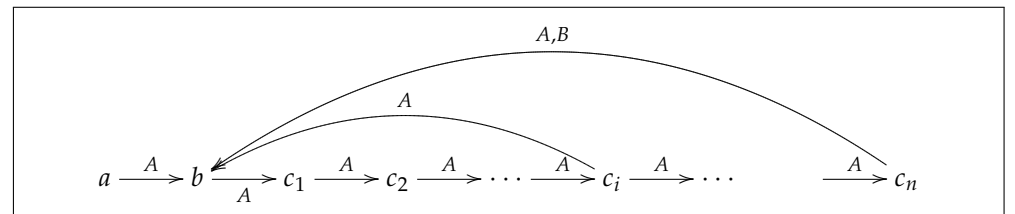
The atomic proposition  $p$  is true at all points except  $b$ , and no other atomic propositions are true anywhere. The arrows in  $S_n$  are given by

1.  $x \xrightarrow{A} x$  for all  $x$ .
2.  $a \xrightarrow{A} b$ .
3.  $b \xrightarrow{A} c_1$ .
4.  $c_i \xrightarrow{A} b$  for all  $1 \leq i \leq n$ .
5.  $c_n \xrightarrow{B} b$ .

$T_n$  has all these arrows and exactly one more:

6.  $a \xrightarrow{A} c_1$ .

The model  $S_n$  is shown in Figure 2. We did not show the reflexive  $\xrightarrow{A}$  arrows.



**Figure 2.** The model  $S_n$  omitting the reflexive  $A$ -arrows. The model  $T_n$  adds the arrow  $a \xrightarrow{A} c_1$ .

For all  $x \neq a$ , the submodels of  $S_n$  and  $T_n$  reachable from  $x$  are the same. Therefore, we have the following fact:

$$\text{for all } x \neq a, (S_n, x) \models \varphi \text{ iff } (T_n, x) \models \varphi \quad (9)$$

This holds for all  $\varphi$  in  $\mathcal{L}(\Sigma_{pub})$ .

Let us calculate  $S_n \otimes (\Sigma_{pri}, Pri^A, p, \text{tr})$  and  $T_n \otimes (\Sigma_{pri}, Pri^A, p, \text{tr})$ . Intuitively, we are making a private announcement of  $p$  to  $A$ . To save on notation, we call the updated models  $\hat{S}_n$  and  $\hat{T}_n$ .

Here are some facts about the structure of  $\hat{S}_n$ :

1.  $(a, Pri^A) \xrightarrow{A} (a, Pri^A)$ .
2. The only  $\xrightarrow{A}$ -successor of  $(a, Pri^A)$  is itself. In particular,  $(a, Pri^A) \xrightarrow{A} (b, Pri^A)$  does not hold, since  $(b, Pri^A)$  does not belong to the model.
3.  $(x, \text{skip}) \xrightarrow{A} (y, \text{skip})$  whenever  $x \xrightarrow{A} y$  in  $S_n$  and similarly for  $(x, \text{skip}) \xrightarrow{B} (y, \text{skip})$ .
4.  $(c_i, Pri^A) \xrightarrow{A} (c_{i+1}, Pri^A)$  for  $1 \leq i < n$ .

The structure of  $\hat{\mathbf{T}}_n$  differs from  $\hat{\mathbf{S}}_n$ . Since  $a \xrightarrow{A} c_1$  in  $\mathbf{T}_n$ , we have a path in  $\hat{\mathbf{T}}_n$ :

$$(a, \text{Pri}^A) \xrightarrow{A} (c_1, \text{Pri}^A) \xrightarrow{A} \dots \xrightarrow{A} (c_n, \text{Pri}^A) \xrightarrow{B} (b, \text{skip}).$$

Once again,  $\hat{\mathbf{S}}_n$  contains no path from  $(a, \text{Pri}^A)$  to any other point, in particular, no path to  $(b, \text{skip})$ .

The update relation between  $\mathbf{S}_n$  and  $\hat{\mathbf{S}}_n$  is  $\{(x, (x, \text{Pri}^A)) : x \neq b\}$ . And from this, we see that with  $\chi$  from (8),

$$\llbracket \chi \rrbracket_{\mathbf{S}_n} = \{x : (x, \text{Pri}^A) \in \llbracket \diamond_A^* \diamond_B \neg p \rrbracket_{\hat{\mathbf{S}}_n}\} = \{c_i : 1 \leq i \leq n\}.$$

In particular,  $a \notin \llbracket \chi \rrbracket_{\mathbf{S}_n}$ .

The update relation between  $\mathbf{T}_n$  and  $\hat{\mathbf{T}}_n$  is the same, and we have

$$\llbracket \chi \rrbracket_{\mathbf{T}_n} = \{x : (x, \text{Pri}^A) \in \llbracket \diamond_A^* \diamond_B \neg p \rrbracket_{\hat{\mathbf{T}}_n}\} = \{x : x \neq b\}.$$

And this time,  $a \in \llbracket \chi \rrbracket_{\mathbf{T}_n}$ .

In Section 6.3, we shall use these models to prove the result announced above: that our sentence  $\diamond_A \chi$  is not expressible by any set of sentences using public announcements only. The intuition is that the models  $\mathbf{S}_n$  and  $\mathbf{T}_n$  differ on  $\chi$  due to the private announcement in that sentence, but as far as public announcements go, the two models are “pretty close”. Of course, “pretty close” needs to be defined and studied, and this is what we do in our later section. The formal proof does not use any of our other results, and the reader may turn to it at this point.

### 3.3. Basic Properties

**Proposition 3.2.** Let  $\varphi$  be a sentence of  $\mathcal{L}(\Sigma)$ , and let  $\alpha$  be an action of  $\mathcal{L}(\Sigma)$ . Then:

1. The epistemic proposition  $\llbracket \varphi \rrbracket$  is preserved by bisimulation.
2. The update  $\llbracket \alpha \rrbracket$  is standard and preserves bisimulation.

**Proof.** By induction on  $\mathcal{L}(\Sigma)$ . For true and the atomic sentences  $p_i$ , we use Proposition 2.9. The same result takes care of the induction steps for all the sentential operators except  $[\pi]\varphi$ . For this, we use Proposition 2.11 part 2, and also the induction hypothesis. Turning to the programs, the standardness comes from Proposition 2.2 and the observation that signature-based program models induce standard updates. The assertion that the interpretation of programs of  $\mathcal{L}(\Sigma)$  preserve bisimulation comes from Proposition 2.11 part 1; for the programs  $\sigma_i \tilde{\psi}$ , we use Proposition 2.12 and the induction hypothesis.  $\square$

In what follows, we shall use Proposition 3.2 without mentioning it.

### 3.4. The Canonical Action Model

We henceforth restrict ourselves to the language  $\mathcal{L}_1(\Sigma)$ , obtained by eliminating iteration  $\pi^*$  from our repertoire. Moreover, in this section, we focus on the *program* expressions of  $\mathcal{L}_1(\Sigma)$ . We introduce some useful meta-syntactic terms and notations, that will allow us to structure (a subset of) our program expressions as a “syntactic program model”, called the *canonical action model*.

**Definition 3.3.** A *simple action* of  $\mathcal{L}_1(\Sigma)$  is a program of  $\mathcal{L}_1(\Sigma)$  that can be obtained from *skip*, *crash* and basic actions of  $\mathcal{L}_1(\Sigma)$  (of the form  $\sigma \tilde{\psi}$ , but without the use of iteration  $\pi^*$  within any precondition) by repeated applications of the sequential composition operation  $\pi; \pi'$  (but no sum operation  $\sqcup$ , and of course no iteration, since iteration is not present in  $\mathcal{L}_1(\Sigma)$ ). So, in particular, simple actions include all the basic actions of  $\mathcal{L}_1(\Sigma)$ . We use letters such as  $\alpha$  and  $\beta$  to denote simple actions. We do so for the rest of this paper, without further notice.

### 3.4.1. The Canonical Action Model $\Omega$

We define a program model  $\Omega$  in several steps. The actions of the model  $\Omega$  (that is, the elements of its carrier set) are the simple actions of the language  $\mathcal{L}_1(\Sigma)$  (as defined just above). For all  $A$ , the accessibility relation  $\overset{A}{\rightarrow}$  is the smallest relation such that

1.  $skip \overset{A}{\rightarrow} skip$ .
2.  $\sigma_i \vec{\varphi} \overset{A}{\rightarrow} \sigma_j \vec{\psi}$  if  $\sigma_i \overset{A}{\rightarrow} \sigma_j$  in  $\Sigma$  and  $\vec{\varphi} = \vec{\psi}$ .
3. If  $\alpha \overset{A}{\rightarrow} \alpha'$  and  $\beta \overset{A}{\rightarrow} \beta'$ , then  $\alpha; \beta \overset{A}{\rightarrow} \alpha'; \beta'$ .

As before, we use the notation  $\overset{C^*}{\rightarrow}$  for the reflexive-transitive closure of the union of all accessibility relations  $\overset{A}{\rightarrow}$  (on simple actions) labelled by agents  $A \in \mathcal{C}$ . In particular, the largest such relation  $\rightarrow_{\mathcal{A}}^*$  is obtained by taking  $\mathcal{C} = \mathcal{A}$  to be the set of all agents.

**Proposition 3.4.** *As a frame,  $\Omega$  is locally finite: for each simple  $\alpha$ , there are only finitely many  $\beta$  such that  $\alpha \rightarrow_{\mathcal{A}}^* \beta$ .*

**Proof.** By induction on  $\alpha$ ; we use heavily the fact that the accessibility relations on  $\Omega$  are the smallest family with their defining property. For the simple action expressions  $\sigma \vec{\psi}$ , we use the assumption that the action model  $\Sigma$  underlying all our definitions is finite. (Taking it to be locally finite would also be sufficient.)  $\square$

Next, we define  $\text{PRE} : \Omega \rightarrow \mathcal{L}_1(\Sigma)$  by recursion so that

$$\begin{aligned} \text{PRE}(skip) &= \text{true} \\ \text{PRE}(crash) &= \text{false} \\ \text{PRE}(\sigma_i \vec{\psi}) &= \psi_i \\ \text{PRE}(\alpha; \beta) &= \langle \alpha \rangle \text{PRE}(\beta) \end{aligned}$$

This function  $\text{PRE}$  is not the function  $\text{pre}$  which is part of the structure of an epistemic action model. However, there is a connection: We are in the midst of defining the epistemic action model  $\Omega$ , and its  $\text{pre}$  function is defined in terms of  $\text{PRE}$ .

Another point: the last clause in the definition of  $\text{PRE}$  could read  $\text{PRE}(\alpha; \beta) = \text{PRE}(\alpha) \wedge [\alpha] \text{PRE}(\beta)$ . This is equivalent to what we have above.

We set

$$\text{pre}(\sigma) = \llbracket \text{PRE}(\sigma) \rrbracket.$$

This simple action model  $\Omega$  is the *canonical epistemic action model*; it plays the same role in our work as the canonical model in modal logic.

**Remark 3.5.** The structure  $(\Omega, \overset{A}{\rightarrow}, \text{PRE})$  is entirely syntactic. It is *not* an action model. This contrasts with the canonical action model  $\Omega = (\Omega, \overset{A}{\rightarrow}, \text{pre})$ ; the last component of this structure is semantic. The syntactic structure  $(\Omega, \overset{A}{\rightarrow}, \text{PRE})$  is the one which is actually used in the statement of the action rule of the logical system. We emphasize this point to allay any suspicion that our logical system is formulated in terms of semantic concepts.

This is also perhaps a good place to remind the reader that neither  $\text{PRE}$  nor  $\text{pre}$  is a first-class symbol in  $\mathcal{L}(\Sigma)$ ; it is only a defined symbol. In Section 5, below, we shall introduce another language called  $\mathcal{L}_1^+(\Sigma)$ . In that language,  $\text{PRE}$  will be a first-class function symbol.

We now proceed to study the main properties of the canonical action model.

**Lemma 3.6.** *For any action type  $\sigma \in \Sigma$  and simple actions  $\alpha, \beta \in \Omega$ , we have the following bisimilarities between program models:*

1.  $(\Sigma, \sigma_i, \llbracket \psi_1 \rrbracket \cdots \llbracket \psi_n \rrbracket) \sim (\Omega, \sigma_i \vec{\psi})$
2.  $(\Omega, \alpha; \beta) \sim (\Omega, \alpha); (\Omega, \beta)$

where in the right-hand side of the second part we use the composition  $;$  of program models.

**Proof.** For the first claim, the bisimulation is the relation  $\{(\sigma_j, \sigma_j \vec{\psi}) : 1 \leq j \leq n\}$ , where  $n$  is the number of elements in  $\Sigma$ .

For the second claim, the bisimulation relates any action of the form  $\alpha'; \beta'$  in  $\Omega$  with the pair  $(\alpha', \beta')$  in  $\Omega \times \Omega$ . It is easy to check that this is a bisimulation between the action models  $\Omega$  and  $\Omega; \Omega$ , which obviously relates  $\alpha; \beta$  with  $(\alpha, \beta)$ , as desired.  $\square$

**Definition 3.7.** For any  $\alpha \in \Omega$ , let  $\hat{\alpha}$  be the program model  $(\Omega, \{\alpha\})$ . As in Section 2.5, we use the same notation  $\hat{\alpha}$  to denote the induced *update*. (However, it will be important to distinguish the two uses, and so we speak of the program model  $\hat{\alpha}$  and also the update  $\hat{\alpha}$ .)

We can now prove the main result on the canonical action model.

**Theorem 3.8.** Let  $\alpha \in \Omega$ . Then, as updates, we have  $\llbracket \alpha \rrbracket \cong \hat{\alpha}$ .

**Proof.** By induction on  $\alpha$ . For *skip* and *crash*, it is easy to check that, as updates, *skip* is equivalent to the identity update **1**, and *crash* is equivalent to **0**.

We next consider an action  $\alpha := \sigma_i \vec{\psi}$ . By the definition of the semantics, we have  $\llbracket \alpha \rrbracket = (\Sigma, \sigma_i, \llbracket \psi_1 \rrbracket \cdots \llbracket \psi_n \rrbracket)$ , while  $\hat{\alpha} = (\Omega, \sigma_i \vec{\psi})$ . By the first claim in the previous Lemma, these program models are bisimilar; hence, they induce equivalent updates.

Finally, consider simple actions of the form  $\alpha; \beta$ . By the semantics,  $\llbracket \alpha; \beta \rrbracket = \llbracket \alpha \rrbracket; \llbracket \beta \rrbracket$ , and by the induction hypothesis, this is the same as  $\hat{\alpha}; \hat{\beta}$ . By the second claim in the previous Lemma, this last program model is bisimilar with  $\hat{\alpha}; \hat{\beta}$ ; hence, the induced updates are equivalent.  $\square$

#### 4. The Logical System for Validity of $\mathcal{L}_1(\Sigma)$ Sentences

We write  $\models \varphi$  to mean that for all state models **S** and all  $s \in S$ ,  $s \in \llbracket \varphi \rrbracket_{\mathbf{S}}$ . In this case, we say that  $\varphi$  is *valid*.

In principle, we are of course interested in the validities of the full languages  $\mathcal{L}(\Sigma)$ . But as has already been mentioned, the satisfiability problems for these languages are, in general, not recursively axiomatizable. (See [14] for details on this.) This is one of the reasons we also consider sublanguages  $\mathcal{L}_0(\Sigma)$  and  $\mathcal{L}_1(\Sigma)$ . It turns out that  $\mathcal{L}_0(\Sigma)$  is the easiest to study: it is of the same expressive power as ordinary multi-modal logic. The main completeness result of the paper is a *sound and complete* proof system for the validities in  $\mathcal{L}_1(\Sigma)$ .

In Figure 3, below, we present the sound and complete logic for  $\mathcal{L}_1(\Sigma)$ . We write  $\vdash \varphi$  if  $\varphi$  can be obtained from the axioms of the system using its inference rules. We often omit the turnstile  $\vdash$  when it is clear from the context.

Most of the system will be quite standard from modal logic. The action axioms and the action rule are new, however. These include the atomic permanence axiom; note that, in this axiom,  $p$  is an atomic sentence. The axiom says that announcements do not change the brute fact of whether or not  $p$  holds. This axiom reflects the fact that our actions do not change any kind of local state.

The Action-Knowledge Axiom gives a criterion for knowledge after an action. It is perhaps easier to appreciate in dual form:

$$\langle \sigma_i \vec{\psi} \rangle \Diamond_A \varphi \leftrightarrow (\psi_i \wedge \bigvee \{ \Diamond_A [\sigma_j \vec{\psi}] \varphi : \sigma_i \xrightarrow{A} \sigma_j \text{ in } \Sigma \}).$$

In words, two sentences are equivalent: first, the assertion that the action  $\sigma_i \vec{\psi}$  may be executed, and as a result of the execution, agent  $A$  will consider  $\varphi$  possible; and second, the precondition  $\psi_i$  of the action holds, and  $A$  considers it possible (before the action) that the action is really  $\sigma_j$  and that there is some possible world in which that action  $\sigma_j$  may be executed and results in a state satisfying  $\varphi$ . This axiom should be compared with the Ramsey axiom in conditional logic.



|  |  |
|--|--|
| <b>Basic Axioms</b>  |  |
| All sentential validities  |  |
| ( $[\pi]$ -normality)  | $\vdash [\pi](\varphi \rightarrow \psi) \rightarrow ([\pi]\varphi \rightarrow [\pi]\psi)$  |
| ( $\Box_A$ -normality)   | $\vdash \Box_A(\varphi \rightarrow \psi) \rightarrow (\Box_A\varphi \rightarrow \Box_A\psi)$   |
| * ( $\Box_C^*$ -normality)   | $\vdash \Box_C^*(\varphi \rightarrow \psi) \rightarrow (\Box_C^*\varphi \rightarrow \Box_C^*\psi)$   |
| * <b>Epistemic Mix Axiom</b>   | $\vdash \Box_C^*\varphi \rightarrow \varphi \wedge \bigwedge \{\Box_A\Box_C^*\varphi : A \in \mathcal{C}\}$  |
| <b>Action Axioms</b>   |  |
| (Atomic Permanence)  | $\vdash [\sigma_i\vec{\psi}]p \leftrightarrow (\psi_i \rightarrow p)$  |
| (Partial Functionality)  | $\vdash [\sigma_i\vec{\psi}]\neg\chi \leftrightarrow (\psi_i \rightarrow \neg[\sigma_i\vec{\psi}]\chi)$  |
| (Action-Knowledge)   | $\vdash [\sigma_i\vec{\psi}]\Box_A\varphi \leftrightarrow (\psi_i \rightarrow \bigwedge \{\Box_A[\sigma_j\vec{\psi}]\varphi : \sigma_i \xrightarrow{A} \sigma_j \text{ in } \Sigma\})$ |
| (Skip Axiom)   | $\vdash [\text{skip}]\varphi \leftrightarrow \varphi$  |
| (Crash Axiom)  | $\vdash [\text{crash}]\text{false}$  |
| (Composition Axiom)  | $\vdash [\pi][\rho]\varphi \leftrightarrow [\pi;\rho]\varphi$  |
| (Choice Axiom)   | $\vdash [\pi \sqcup \rho]\varphi \leftrightarrow [\pi]\varphi \wedge [\rho]\varphi$  |
| <b>Modal Rules</b>   |  |
| (Modus Ponens)   | From $\vdash \varphi$ and $\vdash \varphi \rightarrow \psi$ , infer $\vdash \psi$  |
| ( $[\pi]$ -necessitation)  | From $\vdash \psi$ , infer $\vdash [\pi]\psi$  |
| ( $\Box_A$ -necessitation)   | From $\vdash \varphi$ , infer $\vdash \Box_A\varphi$   |
| * ( $\Box_C^*$ -necessitation)   | From $\vdash \varphi$ , infer $\vdash \Box_C^*\varphi$   |
| * <b>Action Rule</b>   |  |
| Let $\psi$ be a sentence, let $\alpha$ be a simple action, and let $\mathcal{C}$ be a set of agents. Let there be sentences $\chi_\beta$ for all $\beta$ such that $\alpha \xrightarrow{C^*} \beta$ (including $\alpha$ itself), and such that |  |
| 1.   | $\vdash \chi_\beta \rightarrow [\beta]\psi;$   |
| 2.   | If $A \in \mathcal{C}$ and $\beta \xrightarrow{A} \gamma$ , then $\vdash (\chi_\beta \wedge \text{PRE}(\beta)) \rightarrow \Box_A\chi_\gamma.$   |
| From these assumptions, infer $\vdash \chi_\alpha \rightarrow [\alpha]\Box_C^*\psi.$   |  |

**Figure 3.** The logical system for  $\mathcal{L}_1(\Sigma)$ . For  $\mathcal{L}_0(\Sigma)$ , we drop the \* axioms and rules. The definition of  $\text{PRE}$ ,  $\xrightarrow{A}$  and  $\xrightarrow{C^*}$  in the action rule is given in Section 3.4.

The statement of our action rule uses the meta-syntactic terminology introduced in Section 3.4; in particular, the notion of simple actions, the syntactic accessibility relations  $\xrightarrow{A}$  and  $\xrightarrow{C^*}$ , and the syntactic precondition map  $\text{PRE}$  defined on simple actions in the (syntactic version of the) canonical action model. As for its meaning, the action rule gives a necessary criterion for *common knowledge* after a simple action. Since common knowledge is formalized by the  $\Box_B^*$  construct, this rule is a kind of induction rule. (The sentences  $\chi_\beta$  play the role of strong induction hypotheses.)

**Remark 4.1.** Recall that  $\xrightarrow{C^*}$  is an abbreviation for the reflexive and transitive closure of the relation  $\bigcup_{A \in \mathcal{C}} \xrightarrow{A}$ . Recall also from Proposition 3.4 that there are only finitely many  $\beta$  such that  $\alpha \xrightarrow{C^*} \beta$ . So even though the action rule might look like it takes infinitely many premises, it really only takes finitely many.

**Remark 4.2.** It is possible to drop the composition axiom in favor of a more involved version of the action rule. The point is we shall later introduce normal forms for sentences, and using the composition axiom will greatly simplify these normal forms. Moreover, adding the composition axiom leads to shorter proofs. The action rule here is geared towards those normal forms. So if we were to drop the composition axiom, we would need

a stronger, more complicated, formulation of the action rule, one which involved *sequences* of actions. It is not terribly difficult to formulate such a rule, and completeness can be obtained by an elaboration of the work, which we shall do.

#### 4.1. Soundness of the Axioms

In this section, we check the soundness of the axioms of the system (but not yet the soundness of the action rule). The basic axioms are all routine, and so we omit the details on them.

**Proposition 4.3.** *The atomic permanence axiom  $[\sigma_i \vec{\psi}]p \leftrightarrow (\psi_i \rightarrow p)$  is sound.*

**Proof.** Recall from Section 2.7.1 how  $\llbracket \sigma_i \vec{\psi} \rrbracket$  works; call this update  $\mathbf{r}$ . Fix a state model  $\mathbf{S}$ . The following are equivalent:

1.  $s \in \llbracket [\sigma_i \vec{\psi}]p \rrbracket_{\mathbf{S}}$ .
2.  $s \in ([\mathbf{r}]p)_{\mathbf{S}}$ .
3. If  $s \mathbf{r}_{\mathbf{S}} t$ , then  $t \in p_{\mathbf{S}(\mathbf{r})}$ .
4. If  $(s, \sigma_i) \in \mathbf{S}(\mathbf{r})$ , then  $(s, \sigma_i) \in p_{\mathbf{S}(\mathbf{r})}$ .
5. If  $s \in \llbracket \psi_i \rrbracket_{\mathbf{S}}$ , then  $s \in \llbracket p \rrbracket_{\mathbf{S}} = p_{\mathbf{S}}$ .
6.  $s \in \llbracket \psi_i \rightarrow p \rrbracket_{\mathbf{S}}$ .

Most of the individual equivalences are easy, and we only comment on (3)  $\iff$  (4) and (4)  $\iff$  (5). The definition of the update  $\mathbf{r} = \llbracket \sigma_i \vec{\psi} \rrbracket$  implies that  $\mathbf{S}(\mathbf{r}) = \mathbf{S} \otimes \Sigma$ . And  $\mathbf{r}_{\mathbf{S}}$  has the property that everything which  $s \in S$  is related to by  $\mathbf{r}_{\mathbf{S}}$  is of the form  $(s, \sigma_i)$ , where  $\sigma_i$  is from the statement of this axiom and  $(s, \sigma_i) \in \mathbf{S}(\mathbf{r})$ . These points imply the equivalence (3)  $\iff$  (4). For (4)  $\iff$  (5), note that  $(s, \sigma_i) \in \mathbf{S}(\mathbf{r})$  if  $s \in \text{pre}(\sigma_i)_{\mathbf{S}} = \llbracket \psi_i \rrbracket_{\mathbf{S}}$ ; also the definition of the update product in Equation (7) implies that  $(s, \sigma_i) \in p_{\mathbf{S}(\mathbf{r})}$  iff  $s \in \llbracket p \rrbracket_{\mathbf{S}}$ .  $\square$

**Proposition 4.4.** *The partial functionality axiom  $[\sigma_i \vec{\psi}] \neg \chi \leftrightarrow (\psi_i \rightarrow \neg [\sigma_i \vec{\psi}] \chi)$  is sound.*

**Proof.** Again, let  $\mathbf{r} = \llbracket \sigma_i \vec{\psi} \rrbracket$ . Also, let  $\chi = \llbracket \chi \rrbracket$ . Fix a state model  $\mathbf{S}$ . The following are equivalent:

1.  $s \in \llbracket [\sigma_i \vec{\psi}] \neg \chi \rrbracket_{\mathbf{S}}$ .
2.  $s \in ([\mathbf{r}] \neg \chi)_{\mathbf{S}}$ .
3. If  $s \mathbf{r}_{\mathbf{S}} t$ , then  $t \in \neg \chi_{\mathbf{S}(\mathbf{r})}$ .
4. If  $s \mathbf{r}_{\mathbf{S}} t$ , then  $t \notin \chi_{\mathbf{S}(\mathbf{r})}$ .
5. If  $s \in \llbracket \psi_i \rrbracket_{\mathbf{S}}$ , then  $(s, \sigma_i) \notin ([\mathbf{r}] \chi)_{\mathbf{S}}$ .
6. If  $s \in \llbracket \psi_i \rrbracket_{\mathbf{S}}$ , then  $s \notin \neg \chi_{\mathbf{S}(\mathbf{r})}$ .
7. If  $s \in \llbracket \psi_i \rrbracket_{\mathbf{S}}$ , then  $s \in ([\mathbf{r}] \chi)_{\mathbf{S}}$ .
8.  $s \in \llbracket \psi_i \rightarrow \neg [\sigma_i \vec{\psi}] \chi \rrbracket_{\mathbf{S}}$ .

The crucial equivalence here is (6)  $\implies$  (5). The reason this holds is that in the action model for  $\sigma_i \vec{\psi}$ , there is just one distinguished world. So for each fixed  $s \in S$ , there is at most one  $t$  such that  $s \mathbf{r}_{\mathbf{S}} t$ ; i.e.,  $(s, \sigma_i)$ .  $\square$

**Proposition 4.5.** *The Action-Knowledge Axiom*

$$[\sigma_i \vec{\psi}] \Box_A \varphi \leftrightarrow (\psi_i \rightarrow \bigwedge \{ \Box_A [\sigma_j \vec{\psi}] \varphi : \sigma_i \xrightarrow{A} \sigma_j \text{ in } \Sigma \})$$

*is sound.*

**Proof.** Again, let  $\mathbf{r} = \llbracket \sigma_i \vec{\psi} \rrbracket$ . Fix a state model  $\mathbf{S}$ . The following are equivalent:

1.  $s \in \llbracket [\sigma_i \vec{\psi}] \Box_A \varphi \rrbracket_{\mathbf{S}}$ .
2.  $s \in [\mathbf{r}] \llbracket \Box_A \varphi \rrbracket_{\mathbf{S}}$ .
3. If  $s \mathbf{r}_{\mathbf{S}} u$ , then  $u \in \llbracket \Box_A \varphi \rrbracket_{\mathbf{S}(\mathbf{r})}$ .
4. If  $s \in \llbracket \psi_i \rrbracket_{\mathbf{S}}$ , then  $(s, \sigma_i) \in \llbracket \Box_A \varphi \rrbracket_{\mathbf{S}(\mathbf{r})}$ .

5. If  $(s, \sigma_i) \in \mathbf{S}(\mathbf{r})$ , then for all  $(t, \sigma_j)$  such that  $(t, \sigma_j) \in \mathbf{S}(\mathbf{r})$ ,  $s \xrightarrow{A} t$  and  $\sigma_i \xrightarrow{A} \sigma_j$ , we have  $(t, \sigma_j) \in \llbracket \varphi \rrbracket_{\mathbf{S}(\mathbf{r})}$ .
6. If  $(s, \sigma_i) \in \mathbf{S}(\mathbf{r})$ , then for all  $t$  and all  $\sigma_j$  such that  $s \xrightarrow{A} t$  and  $\sigma_i \xrightarrow{A} \sigma_j$ : if  $(t, \sigma_j) \in \mathbf{S}(\mathbf{r})$ , then  $(t, \sigma_j) \in \llbracket \varphi \rrbracket_{\mathbf{S}(\mathbf{r})}$ .
7. If  $(s, \sigma_i) \in \mathbf{S}(\mathbf{r})$ , then for all  $t$  and  $\sigma_j$  such that  $s \xrightarrow{A} t$  and  $\sigma_i \xrightarrow{A} \sigma_j$ :  $t \in \llbracket [\sigma_j \vec{\psi}] \varphi \rrbracket_{\mathbf{S}}$ .
8. If  $s \in \llbracket \psi_i \rrbracket_{\mathbf{S}}$ , then for all  $\sigma_j$  such that  $\sigma_i \xrightarrow{A} \sigma_j$ ,  $s \in \llbracket \Box_A [\sigma_j \vec{\psi}] \varphi \rrbracket_{\mathbf{S}}$ .
9.  $s \in \llbracket \psi_i \rightarrow \bigwedge \{ \Box_A [\sigma_j \vec{\psi}] \varphi : \sigma_i \xrightarrow{A} \sigma_j \} \rrbracket_{\mathbf{S}}$ .

This completes the proof.  $\square$

**Proposition 4.6.** *The action-mix axiom  $[\pi^*] \varphi \rightarrow \varphi \wedge [\pi][\pi^*] \varphi$  is sound.*

**Proof.** This is standard.  $\square$

**Proposition 4.7.** *The skip axiom  $[\text{skip}] \varphi \leftrightarrow \varphi$  is sound.*

**Proof.** Fix a state model  $\mathbf{S}$ . Recall that the semantics of *skip* is the update 1. So the following are equivalent:

1.  $s \in \llbracket [\text{skip}] \varphi \rrbracket_{\mathbf{S}}$ .
2. If  $s \mathbf{1}_{\mathbf{S}} t$ , then  $t \in \llbracket \varphi \rrbracket_{\mathbf{S}(\mathbf{1})}$ .
3.  $s \in \llbracket \varphi \rrbracket_{\mathbf{S}}$ .

This completes the proof.  $\square$

**Proposition 4.8.** *The crash axiom  $[\text{crash}] \text{false}$  is sound.*

**Proof.** Recall that  $\llbracket [\text{crash}] \rrbracket = \mathbf{0}$ , the update having an empty transition relation  $\mathbf{0}_{\mathbf{S}}$ . Working through the definitions,  $[\text{crash}] \text{false}$  is easily seen to hold vacuously.  $\square$

**Proposition 4.9.** *The composition axiom  $[\pi; \rho] \varphi \leftrightarrow [\pi][\rho] \varphi$  is sound.*

**Proof.** Write  $\mathbf{r}$  for  $\llbracket \pi \rrbracket$  and  $\mathbf{b}$  for  $\llbracket \rho \rrbracket$ . Fix a state model  $\mathbf{S}$ . The following are equivalent:

1.  $s \in \llbracket [\pi; \rho] \varphi \rrbracket_{\mathbf{S}}$ .
2. If  $s \llbracket \pi; \rho \rrbracket_{\mathbf{S}} u$ , then  $u \in \llbracket \varphi \rrbracket_{\mathbf{S}(\mathbf{r}; \mathbf{b})}$ .
3. If  $s \llbracket \mathbf{r}; \mathbf{b} \rrbracket_{\mathbf{S}} u$ , then  $u \in \llbracket \varphi \rrbracket_{\mathbf{S}(\mathbf{r}; \mathbf{b})}$ .
4. If  $s \mathbf{r}_{\mathbf{S}} t$  and  $t \mathbf{b}_{\mathbf{S}(\mathbf{r})} u$ , then  $u \in \llbracket \varphi \rrbracket_{\mathbf{S}(\mathbf{r}; \mathbf{b})}$ .
5. If  $s \mathbf{r}_{\mathbf{S}} t$ , then  $t \in \llbracket [\rho] \varphi \rrbracket_{\mathbf{S}(\mathbf{r})}$ .
6.  $s \in \llbracket [\pi][\rho] \varphi \rrbracket_{\mathbf{S}}$ .

The equivalence of (3) and (4) is by the definition of relational composition. The remaining equivalences are from the semantic definitions. The equivalence of (4) and (5) is by the fact that  $\mathbf{S}(\mathbf{r}; \mathbf{b}) = \mathbf{S}(\mathbf{r})(\mathbf{b})$ .  $\square$

**Proposition 4.10.** *The choice axiom  $[\pi \sqcup \rho] \varphi \leftrightarrow [\pi] \varphi \wedge [\rho] \varphi$  is sound.*

**Proof.** Fix a state model  $\mathbf{S}$ ; we drop  $\mathbf{S}$  from the notation in the rest of this proof. Write  $\gamma$  for  $\pi \sqcup \rho$ ,  $\mathbf{r}$  for  $\llbracket \pi \rrbracket$ ,  $\mathbf{b}$  for  $\llbracket \rho \rrbracket$ , and  $\mathbf{c}$  for  $\llbracket \gamma \rrbracket$ . Then, the following are equivalent:

1.  $s \in \llbracket [\pi \sqcup \rho] \varphi \rrbracket_{\mathbf{S}}$ .
2. If  $s \mathbf{c} u$ , then  $u \in \llbracket \varphi \rrbracket_{\mathbf{S}(\mathbf{c})}$ .
3. If  $s \mathbf{r} t$ , then  $t \in \llbracket \varphi \rrbracket_{\mathbf{S}(\mathbf{c})}$ ; and if  $s \mathbf{b} t'$ , then  $t' \in \llbracket \varphi \rrbracket_{\mathbf{S}(\mathbf{c})}$ .
4. If  $s \mathbf{r} t$ , then  $t \in \llbracket \varphi \rrbracket_{\mathbf{S}(\mathbf{r})}$ ; and if  $s \mathbf{b} t'$ , then  $t' \in \llbracket \varphi \rrbracket_{\mathbf{S}(\mathbf{b})}$ .
5.  $s \in \llbracket [\pi] \varphi \rrbracket_{\mathbf{S}}$  and  $s \in \llbracket [\rho] \varphi \rrbracket_{\mathbf{S}}$ .
6.  $s \in \llbracket [\pi] \varphi \wedge [\rho] \varphi \rrbracket_{\mathbf{S}}$ .

The equivalence (2)  $\iff$  (3) comes from the fact that each of the  $u$  such that  $s \mathbf{c} u$  is either (a) an element  $t$  of  $\mathbf{S}(\mathbf{r})$  related to  $s$  by  $\mathbf{r}$ , or else (b) an element  $t'$  of  $\mathbf{S}(\mathbf{b})$  related to  $s$  by  $\mathbf{b}$ . This is by the definition of  $\mathbf{r} \sqcup \mathbf{b}$ .

We show the equivalence (3)  $\iff$  (4) by considering the cases (a) and (b) noted just above. We use the fact that the natural injections of  $\mathbf{S}(\mathbf{r})$  and  $\mathbf{S}(\mathbf{b})$  in  $\mathbf{S}(\mathbf{c})$  are bisimulations, and also the fact that  $\llbracket \varphi \rrbracket$  is preserved by bisimulations (see Proposition 3.2).  $\square$

#### 4.2. Soundness of the Action Rule

To show that the action rule is sound, we need the following preliminary result.

**Lemma 4.11.**  $s \in \llbracket \langle \alpha \rangle \diamond_{\mathcal{C}}^* \varphi \rrbracket_{\mathbf{S}}$  if there is a sequence of states from  $S$

$$s = s_0 \rightarrow_{A_1} s_1 \rightarrow_{A_2} \cdots \rightarrow_{A_{k-1}} s_{k-1} \rightarrow_{A_k} s_k$$

where  $k \geq 0$ , and also a sequence of actions of the same length  $k$ ,

$$\alpha = \alpha_0 \rightarrow_{A_1} \alpha_1 \rightarrow_{A_2} \cdots \rightarrow_{A_{k-1}} \alpha_{k-1} \rightarrow_{A_k} \alpha_k$$

such that  $A_i \in \mathcal{C}$  and  $s_i \in \text{pre}(\alpha_i)_{\mathbf{S}}$  for all  $0 \leq i < k$ , and  $s_k \in \llbracket \langle \alpha_k \rangle \varphi \rrbracket_{\mathbf{S}}$ .

**Remark 4.12.** The case  $k = 0$  just says that  $s \in \llbracket \langle \alpha \rangle \diamond_{\mathcal{C}}^* \varphi \rrbracket_{\mathbf{S}}$  is implied by  $s \in \llbracket \langle \alpha \rangle \varphi \rrbracket_{\mathbf{S}}$ .

**Proof.** The following are equivalent:

1.  $s \in \llbracket \langle \alpha \rangle \diamond_{\mathcal{C}}^* \varphi \rrbracket_{\mathbf{S}}$ .
2.  $s \in \langle \llbracket \alpha \rrbracket \rrbracket \diamond_{\mathcal{C}}^* \varphi \rrbracket_{\mathbf{S}}$ .
3.  $s \in \langle \hat{\alpha} \rrbracket \diamond_{\mathcal{C}}^* \varphi \rrbracket_{\mathbf{S}}$ .
4.  $s \in \text{pre}(\alpha)_{\mathbf{S}}$ , and  $(s, \alpha) \in \llbracket \diamond_{\mathcal{C}}^* \varphi \rrbracket_{\mathbf{S} \otimes \Omega}$ .
5.  $s \in \text{pre}(\alpha)_{\mathbf{S}}$ , and for some  $k \geq 0$  there is a sequence in  $\mathbf{S} \otimes \Omega$ ,

$$(s, \alpha) = t_0 \rightarrow_{A_1} t_1 \rightarrow_{A_2} \cdots \rightarrow_{A_{k-1}} t_{k-1} \rightarrow_{A_k} t_k$$

such that  $A_i \in \mathcal{C}$  and  $t_k \in \llbracket \varphi \rrbracket_{\mathbf{S} \otimes \Omega}$ .

6. There are sequences of states  $s = s_0, \dots, s_k$  and actions  $\alpha_0, \dots, \alpha_k$  as in the statement of this lemma.

The first equivalence is by the semantics of  $\mathcal{L}(\Sigma)$ . The equivalence (2)  $\iff$  (3) uses the equality  $\llbracket \alpha \rrbracket = \hat{\alpha}$  which we saw in the concluding statement of Theorem 3.8. in Section 3.4.1, and also Proposition 2.18. Equivalence (3)  $\iff$  (4) uses our overall definitions and the structure of  $\Omega$  as an action model. (4)  $\iff$  (5) is just the semantics of  $\diamond_{\mathcal{C}}^*$ . Finally, the equivalence (5)  $\iff$  (6) again uses the conclusion of Theorem 3.8. That is, for all  $i$ ,  $(s_i, \alpha) \in \mathbf{S} \otimes \Omega$ , if  $s_i \in \text{pre}(\alpha_i)_{\mathbf{S}}$ .  $\square$

**Proposition 4.13.** *The action rule is sound.*

**Proof.** Assume that  $s \in \llbracket \chi_{\alpha} \rrbracket_{\mathbf{S}}$  but also  $s \in \llbracket \langle \alpha \rangle \diamond_{\mathcal{C}}^* \neg \psi \rrbracket_{\mathbf{S}}$ . According to Lemma 4.11, there is a labeled sequence of states from  $S$

$$s = s_0 \rightarrow_{A_1} s_1 \rightarrow_{A_2} \cdots \rightarrow_{A_{k-1}} s_{k-1} \rightarrow_{A_k} s_k$$

where  $k \geq 0$  and each  $A_i \in \mathcal{C}$ , and also a sequence of actions of length  $k$ , with the same labels,

$$\alpha = \alpha_0 \rightarrow_{A_1} \alpha_1 \rightarrow_{A_2} \cdots \rightarrow_{A_{k-1}} \alpha_{k-1} \rightarrow_{A_k} \alpha_k$$

such that  $s_i \in \text{pre}(\alpha_i)_{\mathbf{S}}$  for all  $0 \leq i < k$ , and  $s_k \in \llbracket \langle \alpha_k \rangle \neg \psi \rrbracket_{\mathbf{S}}$ . If  $k = 0$ , we would have  $s \in \llbracket \langle \alpha \rangle \neg \psi \rrbracket_{\mathbf{S}}$ . But one of the assumptions in the statement of the action rule is that  $\vdash \chi_{\alpha} \rightarrow [\alpha] \psi$ . So we would have  $s \in \llbracket [\alpha] \psi \rrbracket_{\mathbf{S}}$ . This would be a contradiction.

Now we argue the case  $k > 0$ . We show by induction on  $1 \leq i \leq k$  that  $s_i \in \llbracket \chi_{\alpha_i} \rrbracket \mathbf{s}$ . The case  $i = 0$  is the opening assumption of this proof. Assume that  $s_i \in \llbracket \chi_{\alpha_i} \rrbracket \mathbf{s}$ . By hypothesis,  $s_i \in \text{pre}(\alpha_i) \mathbf{s}$ . In view of the second assumption in the action rule,  $s_i \in \llbracket \Box_{A_{i+1}} \chi_{\alpha_{i+1}} \rrbracket \mathbf{s}$ . Hence,  $s_{i+1} \in \llbracket \chi_{\alpha_{i+1}} \rrbracket \mathbf{s}$ . This completes our induction.

In particular,  $s_k \in \llbracket \chi_{\alpha_k} \rrbracket \mathbf{s}$ . Using again the first assumption in the action rule, we have  $s_k \in \llbracket [\alpha_k] \psi \rrbracket \mathbf{s}$ . This is a contradiction.  $\square$

#### 4.3. Syntactic Facts

At this point, we have presented the semantic facts which we need concerning  $\mathcal{L}_1(\Sigma)$ . These include the soundness of the logical system for validity. To prove completeness, we also need some syntactic facts. We could have presented this section earlier, but since it leans on the action rule, we have delayed it until establishing the soundness of that rule.

In this section,  $\alpha, \beta$ , etc., denote simple actions in  $\mathcal{L}_1(\Sigma)$ .

##### 4.3.1. A Stronger form of the Action-Knowledge Axiom

At this point, we establish a stronger form of the action-knowledge axiom. In Figure 3 in Section 4, this axiom was stated only for basic actions. The strengthenings here is to *simple* actions, that is, to compositions of basic actions, *skip* and *crash*.

**Lemma 4.14.** *The Action-Knowledge Axiom is provable for all simple actions  $\alpha$ :*

$$\vdash [\alpha] \Box_A \varphi \leftrightarrow (\text{PRE}(\alpha) \rightarrow \bigwedge \{ \Box_A [\beta] \varphi : \alpha \xrightarrow{A} \beta \text{ in } \Omega \}) \quad (10)$$

**Proof.** By induction on  $\alpha$ . If  $\alpha$  is *skip*,  $\text{PRE}(\alpha) = \text{true}$ , the only  $\beta$  with  $\alpha \xrightarrow{A} \beta$  is *skip*, and equivalence (10) reads:

$$\vdash [\text{skip}] \Box_A \varphi \leftrightarrow (\text{true} \rightarrow \Box_A [\text{skip}] \varphi)$$

And this is an easy consequence of propositional reasoning, the skip axiom, and  $\Box_A$  necessitation. If  $\alpha$  is *crash*,  $\text{PRE}(\alpha) = \text{false}$ , and there are no  $\beta$  such that  $\alpha \xrightarrow{A} \beta$ . Equivalence (10) then reads:

$$\vdash [\text{crash}] \Box_A \varphi \leftrightarrow (\text{false} \rightarrow \text{true}).$$

By the crash axiom and modal reasoning,  $\vdash [\text{crash}] \Box_A \varphi$ ; and so the assertion just above holds.

If  $\alpha$  is of the form  $\sigma_i \vec{\psi}$ , then we simply have the Action-Knowledge Axiom in the form we know it.

So assume our lemma for  $\alpha'$  and  $\alpha$ ; we prove it for  $\alpha'; \alpha$ . We show that

$$\vdash [\alpha'; \alpha] \Box_A \varphi \leftrightarrow (\text{PRE}(\alpha'; \alpha) \rightarrow \bigwedge \{ \Box_A [\beta'; \beta] \varphi : \alpha' \xrightarrow{A} \beta'; \beta \text{ in } \Omega \}) \quad (11)$$

We start with the equivalence in (10), use  $[\alpha']$  necessitation and normality, and obtain

$$\vdash [\alpha'] [\alpha] \Box_A \varphi \leftrightarrow ([\alpha'] \text{PRE}(\alpha) \rightarrow \bigwedge \{ [\alpha'] \Box_A [\beta] \varphi : \alpha \xrightarrow{A} \beta \text{ in } \Omega \}) \quad (12)$$

By the induction hypothesis on  $\alpha'$  and several uses of the composition axiom, we have that for all  $\beta$ ,

$$\vdash [\alpha'] \Box_A [\beta] \varphi \leftrightarrow (\text{PRE}(\alpha') \rightarrow \bigwedge \{ \Box_A [\beta'; \beta] \varphi : \alpha' \xrightarrow{A} \beta'; \beta \text{ in } \Omega \}).$$

The composition axiom and (12) lead to the provable equivalence of  $[\alpha'; \alpha] \Box_A \varphi$  and

$$[\alpha'] \text{PRE}(\alpha) \rightarrow (\text{PRE}(\alpha') \rightarrow \bigwedge \{ \Box_A [\beta'; \beta] \varphi : \alpha \xrightarrow{A} \beta \text{ and } \alpha' \xrightarrow{A} \beta' \text{ in } \Omega \}).$$

But  $\vdash \text{PRE}(\alpha'; \alpha) \leftrightarrow \text{PRE}(\alpha') \wedge [\alpha'] \text{PRE}(\alpha)$ . In addition, we have a general fact  $\alpha'; \alpha \xrightarrow{A} \beta'; \beta$  if  $\alpha' \xrightarrow{A} \beta'$  and  $\alpha \xrightarrow{A} \beta$ . Using these observations and some propositional reasoning, we obtain (11), as desired.  $\square$

**Remark 4.15.** There are no sound versions of the Action-Knowledge Axiom for actions containing  $\sqcup$ . For example, let  $\mathcal{A}$  be a singleton, hence omitted from the notation. Consider  $\Sigma = \{\sigma, \tau\}$  as an action signature with the enumeration  $\sigma, \tau$ , and with (say) the arrows  $\sigma \rightarrow \tau$  and  $\tau \rightarrow \sigma$ . Note that by atomic permanence,  $[\sigma pq]r$  is equivalent to  $p \rightarrow r$ ;  $[\tau pq]r$  is equivalent to  $q \rightarrow r$ ; by action-knowledge and atomic permanence,  $[\sigma pq]\Box r$  is equivalent to  $p \rightarrow \Box(q \rightarrow r)$ ; and  $[\tau pq]\Box r$  is equivalent to  $q \rightarrow \Box(p \rightarrow r)$ . Consider also the sentence  $[(\sigma pq) \sqcup (\tau pq)]\Box r$ . By the choice axiom, it is equivalent to

$$(p \rightarrow \Box(q \rightarrow r)) \wedge (q \rightarrow \Box(p \rightarrow r)). \quad (13)$$

Now, we did not define  $\text{PRE}((\sigma pq) \sqcup (\tau pq))$ , but the most reasonable choice is  $p \vee q$ . And we did not define the accessibility structure of actions containing  $\sqcup$ , but in this case the most likely choice is to have  $(\sigma pq) \sqcup (\tau pq)$  relate to itself and nothing else. But then when we write out the right-hand side of the equivalence (10), we obtain

$$(p \vee q) \rightarrow (\Box(p \rightarrow r) \wedge \Box(q \rightarrow r)).$$

Clearly, this is not equivalent to (13). Even if one were to change  $\text{PRE}((\sigma pq) \sqcup (\tau pq))$  to, say, true, or to  $p \wedge q$ , we still would not have an equivalence: (13) is stronger.

#### 4.3.2. A Stronger form of the Partial Functionality Axiom

We shall also need the following result, a version of the partial functionality axiom:

**Lemma 4.16.**  $\vdash \langle \alpha \rangle \varphi \leftrightarrow \text{PRE}(\alpha) \wedge [\alpha] \varphi$ .

**Proof.** This is an induction, much like the proof of Lemma 4.14, above.  $\square$

#### 4.3.3. Syntactic Bisimulation

Our next set of results pertains to a syntactic notion of action equivalence, one with a bisimulation-like flavor.

**Definition 4.17.** A *syntactic bisimulation* is a relation  $R$  on the set  $\Omega$  of simple actions of  $\mathcal{L}(\Sigma)$  with the following property: if  $\alpha R \beta$ , then

1.  $\vdash \text{PRE}(\alpha) \leftrightarrow \text{PRE}(\beta)$ .
2. For all  $\alpha'$  and  $A$  such that  $\alpha \xrightarrow{A} \alpha'$ , there is some  $\beta'$  such that  $\beta \xrightarrow{A} \beta'$  and  $\alpha' R \beta'$ .
3. For all  $\beta'$  and  $A$  such that  $\beta \xrightarrow{A} \beta'$ , there is some  $\alpha'$  such that  $\alpha \xrightarrow{A} \alpha'$  and  $\alpha' R \beta'$ .

We say that  $\alpha$  and  $\beta$  are *provably equivalent* if there is some syntactic bisimulation relating them. We write  $\alpha \equiv \beta$  in this case.

For sentences, we write  $\varphi \equiv \psi$  and say that  $\varphi$  and  $\psi$  are *equivalent* if  $\vdash \varphi \leftrightarrow \psi$ .

**Lemma 4.18.**  $\vdash \langle \alpha \rangle \text{true} \leftrightarrow \text{PRE}(\alpha)$ .

**Proof.** By induction on  $\alpha$ . For  $\alpha = \text{skip}$  and  $\alpha = \text{crash}$ , we use the skip axiom and the crash axiom, respectively. Here is the argument for  $\alpha$  of the form  $\sigma_i \vec{\psi}$ . First, by necessitation we have  $\vdash [\sigma_i \vec{\psi}] \text{true}$ . And by this and partial functionality,  $\vdash [\sigma_i \vec{\psi}] \neg \text{true} \leftrightarrow \neg \psi_i$ . So  $\vdash \neg [\sigma_i \vec{\psi}] \neg \text{true} \leftrightarrow \psi_i$ . That is,  $\vdash \langle \sigma_i \vec{\psi} \rangle \text{true} \leftrightarrow \psi_i$ .

Finally, assume the result for  $\beta$ . Then, by normality and necessitation,  $\vdash \langle \alpha \rangle \langle \beta \rangle \text{true} \leftrightarrow \langle \alpha \rangle \text{PRE}(\beta)$ . So  $\vdash \langle \alpha \rangle \langle \beta \rangle \text{true} \leftrightarrow \text{PRE}(\alpha; \beta)$ . We conclude by showing as a general fact that  $\vdash \langle \alpha \rangle \langle \beta \rangle \varphi \leftrightarrow \langle \alpha; \beta \rangle \varphi$ . For this, the composition axiom tells us that  $\vdash [\alpha][\beta] \neg \varphi \leftrightarrow [\alpha; \beta] \neg \varphi$ . So  $\vdash \neg [\alpha] \neg [\beta] \neg \varphi \leftrightarrow \neg [\alpha; \beta] \neg \varphi$ . Thus,  $\vdash \langle \alpha \rangle \langle \beta \rangle \varphi \leftrightarrow \langle \alpha; \beta \rangle \varphi$ , as desired.  $\square$



**Lemma 4.19.** *The following monoid-type laws hold:*

1.  $\vdash \text{PRE}(\alpha; \text{skip}) \leftrightarrow \text{PRE}(\alpha) \leftrightarrow \text{PRE}(\text{skip}; \alpha).$
2.  $\vdash \text{PRE}(\alpha; (\beta; \gamma)) \leftrightarrow \text{PRE}((\alpha; \beta); \gamma).$

**Proof.** We use the definitions to calculate

$$\begin{aligned} \text{PRE}(\alpha; \text{skip}) &= \langle \alpha \rangle \text{true} \\ \text{PRE}(\text{skip}; \alpha) &= \langle \text{skip} \rangle \text{PRE}(\alpha). \end{aligned}$$

We obtain a formal proof using Lemma 4.18 and the skip axiom. Turning to the second law,

$$\begin{aligned} \text{PRE}(\alpha; (\beta; \gamma)) &= \langle \alpha \rangle \text{PRE}(\beta; \gamma) \\ &= \langle \alpha \rangle \langle \beta \rangle \text{PRE}(\gamma) \\ &\equiv \langle \alpha; \beta \rangle \text{PRE}(\gamma) \\ &= \text{PRE}((\alpha; \beta); \gamma). \end{aligned}$$

The equivalence above which mentions  $\equiv$  uses the composition axiom.  $\square$

**Lemma 4.20.** *Concerning the syntactic equivalence  $\equiv$  on  $\Omega$ :*

1. *The relation  $\equiv$  is an equivalence relation.*
2. *If  $\varphi_1 \equiv \psi_1, \dots, \varphi_n \equiv \psi_n$ , then  $\sigma_i \vec{\varphi} \equiv \sigma_i \vec{\psi}$ .*
3.  *$\alpha; \text{skip} \equiv \alpha \equiv \text{skip}; \alpha$ .*
4.  *$\alpha; (\beta; \gamma) \equiv (\alpha; \beta); \gamma$ .*
5. *If  $\alpha \equiv \alpha'$  and  $\beta \equiv \beta'$ , then  $\alpha; \beta \equiv \alpha'; \beta'$ .*

**Proof.** The first part is routine. Part (2) uses the bisimulation consisting of all pairs  $(\sigma_j \vec{\varphi}, \sigma_j \vec{\psi})$  such that  $1 \leq j \leq n$  and for all  $i$ ,  $\varphi_i \equiv \psi_i$ . Lemma 4.19 is used in parts (3) and (4). The rest of the argument for part (3) is easy and we omit it. For part (4), we take  $R$  to be the set of pairs

$$(\alpha'; (\beta'; \gamma'), (\alpha'; \beta'); \gamma')$$

such that for some sequence  $w$  of  $\rightarrow_A$  with  $A \in \mathcal{A}^*$ ,  $\alpha'$  is reachable from  $\alpha$  via  $w$ , and similarly for  $\beta'$  and  $\gamma'$  (via the same  $w$ ). Part (5) is similar.  $\square$

**Lemma 4.21.** *For all  $A \in \mathcal{C}$  and all  $\beta$  such that  $\alpha \rightarrow_A \beta$ ,*

1.  $\vdash [\alpha] \Box_{\mathcal{C}}^* \psi \rightarrow [\alpha] \psi.$
2.  $\vdash [\alpha] \Box_{\mathcal{C}}^* \psi \wedge \text{PRE}(\alpha) \rightarrow \Box_A [\beta] \Box_{\mathcal{C}}^* \psi.$

**Proof.** Part (1) follows from the epistemic-mix axiom and modal reasoning. For part (2), we start with a consequence of the epistemic-mix axiom:  $\vdash \Box_{\mathcal{C}}^* \psi \rightarrow \Box_A \Box_{\mathcal{C}}^* \psi$ . Then, by modal reasoning,  $\vdash [\alpha] \Box_{\mathcal{C}}^* \psi \rightarrow [\alpha] \Box_A \Box_{\mathcal{C}}^* \psi$ . By the Action-Knowledge Axiom in the generalized form of Lemma 4.14, we have  $\vdash [\alpha] \Box_{\mathcal{C}}^* \psi \wedge \text{PRE}(\alpha) \rightarrow \Box_A [\beta] \Box_{\mathcal{C}}^* \psi$ .  $\square$

We present next our main result on the syntactic equivalence of actions. It is a syntactic version of Proposition 2.18.

**Lemma 4.22.** *Let  $\alpha$  and  $\beta$  be simple actions. If  $\alpha \equiv \beta$ , then for all  $\varphi$ ,  $\vdash [\alpha] \varphi \leftrightarrow [\beta] \varphi$ .*

**Proof.** By induction on  $\varphi$ . Fix a syntactic bisimulation  $R$  relating  $\alpha$  and  $\beta$ . For  $\varphi = \text{true}$  or an atomic sentence  $p_i$ , our result is easy. The induction steps for  $\neg$  and  $\wedge$  are trivial. The step for  $\Box_A$  is not hard, and so we omit it.

We next check the result for sentences  $[\gamma] \varphi$ . We need to see that

$$\vdash [\alpha][\gamma] \varphi \leftrightarrow [\alpha'][\gamma] \varphi.$$

For this, it is sufficient by the composition axiom to show that  $\vdash [\alpha; \gamma]\varphi \leftrightarrow [\alpha'; \gamma]\varphi$ . By Lemma 4.20, parts (1) and (5),  $\alpha; \gamma \equiv \alpha'; \gamma$ . So we have completed the induction hypothesis.

This leaves the step for sentences of the form  $\Box_B^* \varphi$ , assuming the result for  $\varphi$ . We use the action rule to show that  $\vdash [\alpha]\Box_C^* \varphi \rightarrow [\beta]\Box_C^* \varphi$ . We need a functional witness to  $R$ ; that is, a map  $\beta' \mapsto \alpha'$  for all  $\beta'$  such that  $\beta \xrightarrow{A^*} \beta'$ , such that  $\alpha' R \beta'$  for all these actions. Further, let  $\chi_{\beta'}$  be  $[\alpha']\Box_C^* \varphi$ . We need to show that for all  $A \in C$  and all  $\beta' \xrightarrow{A} \beta''$ ,

- a.  $\vdash [\alpha']\Box_C^* \varphi \rightarrow [\beta']\varphi$ ;
- b. If  $\beta' \xrightarrow{A} \beta''$ , then  $\vdash [\alpha']\Box_C^* \varphi \wedge \text{PRE}(\beta') \rightarrow \Box_A[\alpha'']\Box_C^* \varphi$ .

For (a), we know from Lemma 4.21 that  $\vdash [\alpha']\Box_C^* \varphi \rightarrow [\alpha']\varphi$ . By induction hypothesis on  $\varphi$ ,  $\vdash [\alpha']\varphi \leftrightarrow [\beta']\varphi$ . And this implies (a). For (b), Lemma 4.21 tells us that under the assumptions,

$$\vdash [\alpha']\Box_C^* \varphi \wedge \text{PRE}(\alpha') \rightarrow \Box_A[\alpha'']\Box_C^* \varphi.$$

The fact that  $R$  is a syntactic bisimulation tells us that  $\vdash \text{PRE}(\alpha') \leftrightarrow \text{PRE}(\beta')$ . This implies (b).

This completes the induction on  $\varphi$ .  $\square$

Lemma 4.22 will be used in several places to come.

## 5. Completeness Theorems

In this section, we prove the completeness of our logical systems for  $\mathcal{L}_0(\Sigma)$  and  $\mathcal{L}_1(\Sigma)$ . (See Figure 1 for these.) Recall that the difference between the two languages is that the second has the common-knowledge propositional operators  $\Box_B^*$  while the first does not. This difference makes the bigger system much more expressive, and more difficult to study. As it happens, the extension of  $\mathcal{L}_1(\Sigma)$  to the full logic  $\mathcal{L}(\Sigma)$ , the extension via the action iteration operation  $\pi^*$ , leads to a logical language whose validity problem is complete  $\Pi_1^1$ . So there cannot be a recursively axiomatized logical system for the validities of  $\mathcal{L}(\Sigma)$ . Returning to the smaller  $\mathcal{L}_1(\Sigma)$ , even here the common-knowledge operators  $\Box_B^*$  give a logical system which is not compact. So we cannot have a *strongly* complete logic for it; that is, we cannot axiomatize the notion of validity under hypotheses  $T \models \varphi$ . The best one can hope for is weak completeness:  $\vdash \varphi$  if and only if  $\models \varphi$ . We prove this in Theorem 5.31. As a result of some preliminary results aimed toward that result, we establish the strong completeness of our axiomatization of the weaker logic  $\mathcal{L}_0(\Sigma)$ . The work there is easier because it relies on a translation into modal logic.

In a later section, we show that in contrast to our translation results for  $\mathcal{L}_0(\Sigma)$ , the larger language  $\mathcal{L}_1(\Sigma)$  cannot be translated into  $\mathcal{L}$  or even to  $\mathcal{L}(\Box^*)$  (modal logic with extra modalities  $\Box_B^*$ ). So completeness results for  $\mathcal{L}_1(\Sigma)$  cannot simply be based on translation.

### 5.1. The Ideas

Our completeness proofs are somewhat involved, and it might help the reader to have a preview of some of the ideas before we get started. For  $\mathcal{L}_0(\Sigma)$ , the leading idea is that we can translate the logic back to ordinary modal logic. Then we obtain completeness by taking any logical system for modal logic and adding whatever principles are needed in order to make the translation. This idea is simple enough, and after looking at a few examples one can see how the translation of  $\mathcal{L}_0(\Sigma)$  to modal logic should go. But the formal definition of the translation is complicated. Our definition goes via a *term rewriting system* related to the axioms of the logic. Part of our work in this section will be to prove the *termination* of our system. It would have been nice to use some off-the-shelf results of term rewriting theory to get the termination of our system, but this does not seem to be possible. In any case, we prove termination by establishing a *decreasing interpretation* of the system; that is, rewriting a sentence leads to a decrease in an order  $<$  that we study at length. Our interpretation is exponential rather than polynomial, and it was found by hand.

When we turn to  $\mathcal{L}_1(\Sigma)$ , our model is the filtration proof of the completeness of PDL due to Kozen and Parikh [15]. We need to use the action rule rather than an induction rule,

but modulo this difference, the work is similar. We also need to use some of the details on the ordering  $<$  mentioned above. That is, even if one were interested in the completeness theorem of  $\mathcal{L}_0(\Sigma)$  alone, our proof would still involve the general rewriting apparatus.

**Definition 5.1.** Let  $\mathcal{NF}$  be the smallest set of ground sentences and actions in  $\mathcal{L}_1(\Sigma)$  with the following properties:

1. Each atomic  $p$  belongs to  $\mathcal{NF}$ .
2. If  $\varphi, \psi \in \mathcal{NF}$ , then also  $\neg\varphi$ ,  $\varphi \wedge \psi$ ,  $\Box_A\varphi$ , and  $\Box_C^*\varphi$  belong to  $\mathcal{NF}$ .
3. If  $\alpha \in \mathcal{NF}$ ,  $\varphi \in \mathcal{NF}$ , and  $\mathcal{C} \subseteq \mathcal{A}$ , then  $[\alpha]\Box_C^*\varphi$  belongs to  $\mathcal{NF}$ .
4. If  $k \geq 0$ , if  $\vec{\psi}^1, \dots, \vec{\psi}^k$  is a sequence of sequences of length  $n(\Sigma)$  of elements of  $\mathcal{NF}$ , and if  $\sigma_1, \dots, \sigma_k \in \Sigma$ ,

$$(\dots((\sigma_1\vec{\psi}^1; \sigma_2\vec{\psi}^2); \dots; \sigma_k\vec{\psi}^k) \quad (14)$$

is an action term in  $\mathcal{NF}$ .

**Lemma 5.2.** There is a function  $\text{nf} : \mathcal{L}_1(\Sigma) \rightarrow \mathcal{NF}$  such that for all  $\varphi$ ,  $\vdash \varphi \leftrightarrow \text{nf}(\varphi)$ . Moreover, if  $\varphi \in \mathcal{L}_0(\Sigma)$ , then  $\text{nf}(\varphi)$  is a purely modal sentence (it contains no actions).

**Lemma 5.3.** There is a well-order  $<$  on  $\mathcal{L}_1(\Sigma)$  such that for all  $\varphi$  and  $\alpha$ :

1.  $\text{nf}(\varphi) \leq \varphi$ .
2.  $\text{PRE}(\alpha) < [\alpha]\varphi$ .
3. If  $\alpha \xrightarrow{A} \beta$ , then  $[\beta]\varphi < \neg[\beta]\varphi < [\alpha]\Box_C^*\varphi$ .

**Lemma 5.4.** For every  $\varphi$  there is a finite set  $f(\varphi) \subseteq \mathcal{NF}$  such that

1.  $f(\varphi)$  is closed under subsentences.
2. If  $[\gamma]\Box_C^*\chi \in f(\varphi)$ ,  $\gamma \xrightarrow{C} \delta$ , and  $A \in \mathcal{C}$ , then  $f(\varphi)$  also contains  $\Box_A[\delta]\Box_C^*\chi$ ,  $[\delta]\Box_C^*\chi$ ,  $\text{nf}(\text{PRE}(\delta))$ , and  $\text{nf}([\delta]\chi)$ .

The proofs of these will appear in Section 5.2 just below. The proofs are technical, and so the reader not interested in those details might wish to omit the next section on a first reading of this paper. We shall use Lemmas 5.2–5.4 in the work on completeness below, but neither the details of the proofs nor the other results of Section 5.2 will be used in the rest of this paper.

## 5.2. Proofs of the Main Facts on Normal Forms and the Well-Order $<$

We turn to the proofs of Lemmas 5.2–5.4, just above. Our proofs are complicated and circuitous, so there might be shorter arguments. For example, we do not know of any proofs that avoid term rewriting theory. For the record, here are some of the reasons why we feel that the study of our system is complicated:

1. The original statement of axioms such as the action-knowledge axiom is in terms of actions of the form  $[\sigma_i\vec{\psi}]$ .
2. On the other hand, the action rule is best stated in terms of actions which are compositions of the actions  $[\sigma_i\vec{\psi}]$ . In a term-rewriting setting, this point and the previous one work against each other. Our work will be to work with the versions of the axioms that are generalized to the case of all simple actions.
3. Again mentioning term rewriting, we will need to pick an orientation for the composition axiom. This will either be  $[\alpha][\beta]\varphi \rightsquigarrow [\alpha;\beta]\varphi$ , or  $[\alpha;\beta]\varphi \rightsquigarrow [\alpha][\beta]\varphi$ <sup>6</sup>. Both alternatives lead to difficulties at various points. We chose the first alternative, and for this reason, we will need a formulation of the Action-Knowledge Axiom as an infinite scheme.
4. Our language has the program union operator  $\sqcup$ , but because the axioms are not, in general, sound for sums, we need to reformulate things to avoid  $\sqcup$ .

It is convenient to replace  $\mathcal{L}_1(\Sigma)$  by a slightly different language which we call  $\mathcal{L}_1^+(\Sigma)$ . This new language is shown in Figure 4. For the purposes of this section, we stress the formulation of this as an algebra for a two-sorted signature. Starting with some fixed action signature  $\Sigma$ , we construct a two-sorted signature  $\Delta = \Delta(\Sigma)$  of terms, obtained in the following way. Let  $n$  be the number of simple actions in  $\Sigma$ .

1.  $\Delta$  has two sorts:  $s$  (for sentences) and  $a$  (for actions).
2. Each  $p \in \text{AtSen}$  is a constant symbol of sort  $s$ .
3.  $\neg$ ,  $\Box_A$ , and  $\Box_B^*$  are function symbols of type  $s \rightarrow s$ .
4.  $\wedge$  and  $\rightarrow$  are binary function symbols of type  $s \times s \rightarrow s$ .
5. Each  $\sigma \in \Sigma$  is a function symbol of sort  $s^n \rightarrow a$ .
6.  $;$  is function symbols of sort  $a \times a \rightarrow a$ .
7.  $app$  is a binary symbol of type  $a \times s \rightarrow s$ .
8.  $\text{PRE}$  is a function symbol of sort  $a \rightarrow s$ .

|                     |                               |                 |                       |                            |                  |                    |                   |                      |
|---------------------|-------------------------------|-----------------|-----------------------|----------------------------|------------------|--------------------|-------------------|----------------------|
| sentences $\varphi$ | $p_i$                         | $\neg\varphi$   | $\varphi \wedge \psi$ | $\varphi \rightarrow \psi$ | $\Box_A \varphi$ | $\Box_B^* \varphi$ | $[\alpha]\varphi$ | $\text{PRE}(\alpha)$ |
| actions $\alpha$    | $\sigma\psi_1, \dots, \psi_n$ | $\alpha; \beta$ |                       |                            |                  |                    |                   |                      |

**Figure 4.** The language  $\mathcal{L}_1^+(\Sigma)$ .

The most important addition here is that we have  $\text{PRE}$  as a first-class part of the syntax; previously, it had been an abbreviation. We also add the implication symbol  $\rightarrow$ , but this is only for convenience. Obviously,  $\rightarrow$  may be dropped from the system. On the other hand, we dropped  $\Box$  (as we mentioned, some axioms are not sound as equations in general if we have  $\Box$ ). We might as well drop *skip* as well since it, too, can be translated away.

Incidentally, the fact that  $\text{PRE}$  is not a symbol of  $\mathcal{L}_1(\Sigma)$  makes the issue of translating between  $\mathcal{L}_1(\Sigma)$  and  $\mathcal{L}_1^+(\Sigma)$  delicate. The reader might wish to formulate a careful translation in order to appreciate some of the features of our ordering  $<$  which we shall introduce in due course.

We adopt the usual notational conventions that  $\wedge$ ,  $\leftrightarrow$ , and  $;$  are used as infix symbols. We continue our practice of writing  $\sigma_i \vec{\psi}$  for what technically would be  $\sigma_1(\psi_1, \dots, \psi_n)$ . Also, we write  $[\alpha]\psi$  instead of  $app(\alpha, \psi)$ .

When dealing with  $\mathcal{L}_1^+(\Sigma)$ , we let  $\alpha, \beta$ , etc., range over terms of sort  $a$ , and  $\varphi, \psi, \chi$ , etc., range over terms of sort  $s$ . Finally, we use letters such as  $t$  and  $u$  for terms of either sort (so as to shorten many of our statements). We also will need to adjoin new variables to our signature in order to formulate the notion of *substitution* that leads to a *term rewriting system*. For this, we let  $X_a$  and  $X_s$  be sets of new symbols. In order to simplify our notation, we will use letters such as  $x, y$ , and  $z$  to range over *both* of these sets. That is, we will not notationally distinguish the two sorts of variables. The context will always make it clear what the sort of any given variable is.

Let  $\mathcal{L}_1^+(\Sigma, X)$  be the terms built from our signature which now may contain the variables from  $X$ . Examples of such terms may be found in Figure 5.

|                                      |                                       |                    |   |
|--------------------------------------|---------------------------------------|--------------------|---|
| (r1)                                 | $x \rightarrow y$                     | $\rightsquigarrow$ | $\neg(x \wedge \neg y)$   |
| (r2)                                 | $\text{PRE}(\sigma_i y_1 \cdots y_n)$ | $\rightsquigarrow$ | $y_i$   |
| (r3)                                 | $\text{PRE}(x; y)$                    | $\rightsquigarrow$ | $\text{PRE}(x) \wedge [x]\text{PRE}(y)$   |
| (r4)                                 | $[x]p$                                | $\rightsquigarrow$ | $\text{PRE}(x) \rightarrow p$   |
| (r5)                                 | $[x]\neg y$                           | $\rightsquigarrow$ | $\text{PRE}(x) \rightarrow \neg[x]y$  |
| (r6)                                 | $[x](y \wedge z)$                     | $\rightsquigarrow$ | $[x]y \wedge [x]z$  |
| (r7 <sub><math>\alpha</math></sub> ) | $[\alpha]\Box_A x$                    | $\rightsquigarrow$ | $\text{PRE}(\alpha) \rightarrow \bigwedge \{ \Box_A [\beta]x : \alpha \xrightarrow{A} \beta \}$ |
| (r8)                                 | $[x][y]z$                             | $\rightsquigarrow$ | $[x; y]z$   |
| (r9)                                 | $x; (y; z)$                           | $\rightsquigarrow$ | $(x; y); z$   |

**Figure 5.** The rewrite system  $\mathcal{R}$

**Lemma 5.5.** Let  $\varphi$  be a sentence of  $\mathcal{L}_1(\Sigma)$ . Then there is some  $\varphi^\dagger \in \mathcal{L}_1(\Sigma)$  which does not contain  $\sqcup$ , crash or skip such that  $\vdash \varphi \leftrightarrow \varphi^\dagger$ .

**Proof.** (J. Sack, personal communication) We define  $\varphi \mapsto \varphi^\dagger$  as follows:

$$\begin{array}{llll}
 \text{true}^\dagger & = & \text{true} & ([\text{skip}]\varphi)^\dagger = \varphi^\dagger \\
 p_i^\dagger & = & p_i & ([\text{crash}]\varphi)^\dagger = \text{true} \\
 (\neg\varphi)^\dagger & = & \neg\varphi^\dagger & ([\sigma_i\psi_1 \cdots \psi_n]\varphi)^\dagger = [\sigma_i\psi_1^\dagger \cdots \psi_n^\dagger]\varphi^\dagger \\
 (\varphi \wedge \psi)^\dagger & = & \varphi^\dagger \wedge \psi^\dagger & ([\pi \sqcup \rho]\varphi)^\dagger = ([\pi]\varphi)^\dagger \wedge ([\rho]\varphi)^\dagger \\
 (\Box_A \varphi)^\dagger & = & \Box_A \varphi^\dagger & ([\pi; \rho]\varphi)^\dagger = ([\pi][\rho]\varphi)^\dagger \\
 (\Box_B^* \varphi)^\dagger & = & \Box_B^* \varphi^\dagger & 
 \end{array}$$

The definition is by recursion on the number  $k$  of composition symbols ( $;$ ) in  $\varphi$ . For a fixed  $k$ , we then use recursion on the total number of symbols. The overall recursion allows us to define  $([\pi; \rho]\varphi)^\dagger$  to be  $([\pi][\rho]\varphi)^\dagger$ . And the ‘inside’ recursion on the number of symbols allows us to define  $([\pi \sqcup \rho]\varphi)^\dagger$  in terms of  $([\pi]\varphi)^\dagger$  and  $([\rho]\varphi)^\dagger$ .

We indicate two of the verifications that our definition works. Here are the details for the line involving  $([\sigma_i\psi_1 \cdots \psi_n]\varphi)^\dagger$ . Assuming the relevant induction hypotheses, we see that  $\sigma_i\psi_1 \cdots \psi_n \equiv \sigma_i\psi_1^\dagger \cdots \psi_n^\dagger$  (see Lemma 4.20, part 2). And then by Lemma 4.22,

$$[\sigma_i\psi_1 \cdots \psi_n]\varphi \equiv [\sigma_i\psi_1^\dagger \cdots \psi_n^\dagger]\varphi.$$

Using necessitation, we also have equivalence to  $[\sigma_i\psi_1^\dagger \cdots \psi_n^\dagger]\varphi^\dagger$ .

Some of the other cases use necessitation in this way also. In all cases, the verifications are similar.  $\square$

**Remark 5.6.** In the remainder of this paper, we shall assume that sentences and actions of  $\mathcal{L}_1(\Sigma)$  do not contain  $\sqcup$ , crash or skip. We also regard them as *ground terms* of  $\mathcal{L}_1^+(\Sigma)$ : these are terms without variables.

Incidentally, the proof of Lemma 5.5 shows that  $;$  is also eliminable. However, we shall *not* assume that  $;$  is not found in our sentences. In fact, the rewriting system  $\mathcal{R}$  that we present shortly will introduce compositions.

### 5.2.1. The Rewriting System $\mathcal{R}$ and Its Interpretation

We recall here the general notion of term rewriting as it applies to  $\mathcal{L}_1^+(\Sigma, X)$ . For more on this topic, especially on the notion of termination, see, e.g., Dershowitz [16]. Consider a *rewrite rule* (r) of the form  $l \rightsquigarrow r$ , where  $l$  and  $r$  are elements of  $\mathcal{L}_1^+(\Sigma, X)$ . This (r) generates a relation of *immediate rewriting* on  $\mathcal{L}_1^+(\Sigma, X)$ : we say that  $t_1$  *rewrites to*  $t_2$  *via* (r) if there is a term  $u$  with exactly one occurrence of a variable  $x$  (of either sort) and a substitution  $\sigma$  such that  $u(x \leftarrow l^\sigma) = t_1$  and  $u(x \leftarrow r^\sigma) = t_2$ . That is,  $t_1$  and  $t_2$  result from  $u$  by substituting  $l$  and  $r$  in for  $x$ ; however, we need not use  $l$  and  $r$  literally, but we could as well take a substitution instance of them. We would write  $t_1 \xrightarrow{r} t_2$  for this.

Given a set  $\mathcal{R}$  of rewrite rules, we write  $t_1 \xrightarrow{\mathcal{R}} t_2$  if for some  $r \in \mathcal{R}$ ,  $t_1 \xrightarrow{r} t_2$ . We naturally consider the transitive closure  $\xrightarrow{\mathcal{R}^*}$  of  $\xrightarrow{\mathcal{R}}$ . We say that  $t_1$  *rewrites to*  $t_2$  *via*  $\mathcal{R}$  if they stand in this relation  $\xrightarrow{\mathcal{R}^*}$ .

We are mostly interested in the rewriting of ground terms. But we need the notion of variables to formulate this, and this is the only reason for introducing variables.

Now that we have made this preliminary discussion, we consider the term rewriting system  $\mathcal{R}$  shown in Figure 5. We have numbered the rules, and we use this numbering in the sequel.

**Remark 5.7.**  $(r7_\alpha)$  is an infinite scheme, and, in it,  $\alpha$  is a term of sort  $a$ . But  $\alpha$  need not be an action variable. (Indeed, the scheme is only interesting when  $\alpha$  is a composition of actions of the form  $\sigma_i\vec{\psi}$ .) The right side of  $(r7_\alpha)$  depends on  $\alpha$ , and this is why we use a scheme rather than a single rule. We need to explain what it means. First, the action terms

of  $\mathcal{L}_1^+(\Sigma, X)$  carry the structure of a frame by taking for each  $A$  the smallest relation  $\xrightarrow{A}$  such that the following two conditions hold:

1. If  $i \xrightarrow{A} j$  in  $\Sigma$ , then  $\sigma_i \vec{\psi} \xrightarrow{A} \sigma_j \vec{\psi}$ .
2. If  $\alpha \xrightarrow{A} \alpha'$  and  $\beta \xrightarrow{A} \beta'$ , then  $\alpha; \alpha' \xrightarrow{A} \beta; \beta'$ .

Moreover, we use the notation  $\bigwedge \{ \Box_A [\beta] \varphi : \alpha \xrightarrow{A} \beta \}$  to stand for some fixed, but arbitrary, rendering of the conjunction as a nested list of binary conjuncts. The order will not matter, but we shall need an estimate of how many conjuncts there are.

We have a length function  $\ell$  on action terms:  $\ell(x) = 0$  for variables  $x$ ,  $\ell(\sigma_i \vec{\psi}) = 1$ , and  $\ell(\alpha; \beta) = \ell(\alpha) + \ell(\beta)$ . For each  $\alpha$ , the number of  $\beta$  such that  $\alpha \xrightarrow{A} \beta$  is at most  $n^{\ell(\alpha)}$ , where  $n$  is the size of  $\Sigma$ . This is easy to check by induction on  $\ell(\alpha)$ .

Also on (r7<sub>α</sub>): one is tempted to replace this infinite scheme by the finite one

$$(r10) \quad [\sigma_i \vec{\psi}] \Box_A \varphi \rightsquigarrow \psi_i \rightarrow \bigwedge \{ \Box_A [\sigma_j \vec{\psi}] \varphi : i \xrightarrow{A} j \}$$

If one did this, one would have to reverse (r8) to read

$$(r11) \quad [\alpha; \beta] \varphi \rightsquigarrow [\alpha][\beta] \varphi$$

The reason is that (r10), above, only allows us to reduce expressions  $[\alpha] \Box_A \varphi$  when  $\alpha$  is of the form  $\sigma_i \vec{\psi}$ . The problem is that the normal forms of (r1)–(r6) + (r9) + (r10) + (r11) would include terms such as

$$[\sigma_1 \vec{\psi}^1][\sigma_2 \vec{\psi}^2] \cdots [\sigma_k \vec{\psi}^k] \Box_C^* \varphi,$$

and these are not suitable for use in connection with the action rule. We will show about our system (r1)–(r9) that its normal forms includes terms such as

$$[(\cdots (\sigma_1 \vec{\psi}^1; \sigma_2 \vec{\psi}^2) \cdots \sigma_k \vec{\psi}^k)] \Box_C^* \varphi.$$

Again, this is mainly due to our choice in the direction of (r8).

### 5.2.2. Our Interpretation

An *interpretation* of a signature  $\Delta$  is a  $\Delta$  *algebra*. This is a *carrier set* for the sentences, a carrier set for the actions, and for each  $n$ -ary function symbol  $f$  of  $\Delta$  a function of the appropriate sort. In our setting, both carrier sets will be  $N_{\geq 3}$ , the set of natural numbers which are at least 3. We shall use  $a, b$ , etc., to range over  $N_{\geq 3}$  in this section. Our interpretation is shown in Figure 6. By recursion on terms  $t(x_1, \dots, x_k)$ , we build an interpretation<sup>7</sup>.

$$\llbracket t \rrbracket(a_1, \dots, a_k) : (N_{\geq 3})^k \rightarrow N_{\geq 3}.$$

The interpretation of each function symbol is strictly monotone in each argument. (For example, we check this for  $\llbracket ; \rrbracket$ . For fixed  $n$  and  $m$ , we consider the functions  $\lambda a. n^{a+2}$  and  $\lambda a. a^{m+2}$ . Then, if  $a < b$ ,  $n^{a+2} < n^{b+2}$  and  $a^{m+2} < b^{m+2}$ .) So, by induction, each  $\llbracket t \rrbracket(a_1, \dots, a_k)$  is strictly monotone as a function in each argument.

|   |               |   |                            |
|---|---------------|---|----------------------------|
| $\llbracket p_i \rrbracket$               | $= 3$         | $\llbracket \Box_B^* \rrbracket(a)$               | $= a + 1$                  |
| $\llbracket \neg \rrbracket(a)$           | $= a + 1$     | $\llbracket app \rrbracket(a, b)$                 | $= a^b$                    |
| $\llbracket \wedge \rrbracket(a, b)$      | $= a + b$     | $\llbracket PRE \rrbracket(a)$                    | $= a$                      |
| $\llbracket \rightarrow \rrbracket(a, b)$ | $= a + b + 3$ | $\llbracket \sigma_i \rrbracket(a_1, \dots, a_n)$ | $= a_1 + \cdots + a_n + 1$ |
| $\llbracket \Box_A \rrbracket(a)$         | $= a + 2$     | $\llbracket ; \rrbracket(a, b)$                   | $= a^{b+1}$                |

**Figure 6.** The interpretation.

**Lemma 5.8.** Let  $n = |\Sigma|$ . For all action terms  $\alpha$ , and all maps  $\iota$  of variables to  $N_{\geq 3}$ ,

$$\llbracket \alpha \rrbracket(\iota(x_1), \dots, \iota(x_n)) > n^{\ell(\alpha)}.$$



**Proof.** By induction on  $\alpha$ . If  $\alpha$  is a variable  $x$ , then  $\ell(\alpha) = 0$ . So  $\llbracket \alpha \rrbracket(\iota(x)) = \iota(x) \geq 3 > 1 = n^{\ell(\alpha)}$ . If  $\alpha$  is a term of the form  $\sigma_i \vec{\psi}$ , then  $\ell(\alpha) = 1$ , and

$$\begin{aligned} \llbracket \sigma_i \psi_1 \cdots \psi_n \rrbracket(\iota(x_1), \dots, \iota(x_n)) &= \llbracket \psi_1 \rrbracket(\iota(x_1), \dots, \iota(x_n)) + \cdots + \llbracket \psi_n \rrbracket(\iota(x_1), \dots, \iota(x_n)) \\ &> 3n \\ &> n^{\ell(\alpha)} \end{aligned}$$

Now assume our lemma for  $\alpha$  and  $\beta$ .

$$\begin{aligned} \llbracket \alpha; \beta \rrbracket(\iota(x_1), \dots, \iota(x_n)) &= (\llbracket \alpha \rrbracket(\iota(x_1), \dots, \iota(x_n))) \llbracket \beta \rrbracket(\iota(x_1), \dots, \iota(x_n)) + 1 \\ &> n^{\ell(\alpha); (n^{\ell(\beta)+1})} \\ &> n^{\ell(\alpha) + \ell(\beta)} \quad \text{see below} \\ &= n^{\ell(\alpha; \beta)} \end{aligned}$$

In asserting that  $\ell(\alpha); n^{\ell(\beta)} > \ell(\alpha) + \ell(\beta)$ , we assume that  $n > 1$ . If  $n = 1$ , then our lemma is trivial.  $\square$

**Proposition 5.9.** If  $\alpha$  is an action term of  $\mathcal{L}_1^+(\Sigma)$  and  $\alpha \xrightarrow{A} \beta$ , then  $\llbracket \alpha \rrbracket$  and  $\llbracket \beta \rrbracket$  are the same function. In particular, if  $\alpha$  and  $\beta$  are ground action terms, then  $\llbracket \alpha \rrbracket = \llbracket \beta \rrbracket$ .

**Proof.** By induction on  $\alpha$ . The point is that the interpretation of the actions  $\sigma_i \vec{\psi}$  does not use the number  $i$  in any way. The rest follows by an easy induction.  $\square$

The following proposition is a technical but elementary result on exponentiation. It will be used in Lemma 5.11, below. One might notice that if we replace 3 by 2 in the statement, then some of the parts are no longer true. This is the reason why our overall interpretation uses  $N_{\geq 3}$  rather than  $N$  or  $N_{\geq 2}$ .

**Proposition 5.10.** Let  $a, b, c \geq 3$ .

1.  $2a^b > a + 4$ .
2.  $a^{b+c} > a^b + a^c$ .
3.  $a^2 > a + 5$ .
4.  $a^{b+1} > 3a + 3$ .
5.  $a^b > (a+1)(b+1)$ .

**Lemma 5.11.** For each of (r1)–(r9), the interpretation of the left side of the rule is strictly larger than the right side on all tuples of arguments in  $N_{\geq 3}$ .

**Proof.** We remind the reader that letters  $a, b, c$ , etc., denote elements of  $N_{\geq 3}$ . We use Proposition 5.10 without explicit mention.

- (r1)  $\llbracket x \rightarrow y \rrbracket(a, b) = a + b + 3 > a + b + 2 = \llbracket \neg(x \wedge \neg y) \rrbracket(a, b)$ .
- (r2)  $\llbracket \text{PRE}(\sigma_i \vec{y}) \rrbracket(a_1, \dots, a_n) = \llbracket \sigma_i \vec{y} \rrbracket(a_1, \dots, a_n) \geq a_i + 1 > \llbracket y_i \rrbracket(a_1, \dots, a_n)$ .
- (r3)

$$\begin{aligned} \llbracket \text{PRE}(x; y) \rrbracket(a, b) &= a^{b+1} \\ &> a + a^b \\ &= \llbracket \text{PRE}(x) \rrbracket(a, b) + \llbracket [x] \text{PRE}(y) \rrbracket(a, b) \\ &= \llbracket \text{PRE}(x) \wedge [x] \text{PRE}(y) \rrbracket(a, b) \end{aligned}$$

- (r4)  $\llbracket [x] p \rrbracket(a) = a^2 > a + 5 = \llbracket \text{PRE}(x) \rightarrow p \rrbracket(a)$ .

(r5)

$$\begin{aligned}
\llbracket [x] \neg y \rrbracket(a, b) &= a^{b+1} \\
&\geq a^b + 2a^b \\
&> a + a^b + 4 \\
&= a + \llbracket [x] y \rrbracket(a, b) + 4 \\
&= \llbracket \text{PRE}(x) \rightarrow \neg[x] y \rrbracket(a, b)
\end{aligned}$$

$$(r6) \llbracket [x](y \wedge z) \rrbracket(a, b, c) = a^{b+c} > a^b + a^c = \llbracket [x] y \wedge [x] z \rrbracket(a, b, c).$$

(r7<sub>α</sub>) Let  $y_1, \dots, y_n$  be the free variables of  $\alpha$ . Fix  $c_1, \dots, c_n \in N_{\geq 3}$ . Let  $a = \llbracket \alpha \rrbracket(c_1, \dots, c_n)$ , and let  $b \in N_{\geq 3}$  be arbitrary. Then

$$\begin{aligned}
\llbracket [\alpha] \Box_A x \rrbracket(c_1, \dots, c_n, b) &= a^{b+2} \\
&> a^{b+1} + a^{b+1} \\
&> a^{b+1} + 3a + 3 \\
&= a + a(2 + a^b) + 3 \\
&> a + n^{\ell(\alpha)}(2 + a^b) + 3 \\
&\geq \llbracket \text{PRE}(\alpha) \rightarrow \bigwedge \{ \Box_A [\beta] x : \alpha \xrightarrow{A} \beta \} \rrbracket(c_1, \dots, c_n, b)
\end{aligned}$$

Notice that we used Proposition 5.9 and our observation that the size of the conjunction on the right side of (r7<sub>α</sub>) is at most  $n^{\ell(\alpha)}$ .

$$(r8) \llbracket [x][y]z \rrbracket(a, b, c) = a^{b^c} > a^{(b+1)^c} = (a^{b+1})^c = \llbracket [x; y]z \rrbracket(a, b, c).$$

$$(r9) \llbracket x; (y; z) \rrbracket(a, b, c) = a^{b^{c+1}+1} > a^{(b+1)(c+1)} = (a^{b+1})^{c+1} = \llbracket (x; y); z \rrbracket(a, b, c). \quad \square$$

**Definition 5.12.** Let  $\varphi$  and  $\psi$  be sentences in  $\mathcal{L}_1(\Sigma)$ . We regard these as ground terms in  $\mathcal{L}_1^+(\Sigma)$ . We write  $\varphi > \psi$  if  $\llbracket \varphi \rrbracket > \llbracket \psi \rrbracket$ , where our interpretation is the one in Figure 6. We write  $\varphi < \psi$  if  $\varphi \leq \psi$  to mean the obvious things.

**Theorem 5.13.**  $\mathcal{R}$  is terminating: there are no infinite sequences

$$t_1 \xrightarrow{\mathcal{R}} t_2 \xrightarrow{\mathcal{R}} \dots \xrightarrow{\mathcal{R}} t_n \xrightarrow{\mathcal{R}} t_{n+1} \dots$$

**Proof.** We recall a standard argument. Assume we had a counter example, an infinite sequence of rewrites. We may assume without loss of generality that each  $t_i$  is a ground term. Recall Lemma 5.11 and our observation that each  $\llbracket t \rrbracket(x_1, \dots, x_k)$  is strictly monotone as a function in each argument. From this, it follows that  $\llbracket t_1 \rrbracket > \llbracket t_2 \rrbracket > \dots$ , and these are all numbers. So we have a contradiction.  $\square$

### 5.2.3. Normal Forms

We remind the reader that we formulated the notions of equivalences of simple actions and also equivalence of sentences of  $\mathcal{L}_1(\Sigma)$  in Section 4.3. We consider now a translation  $\text{trans} : \mathcal{L}_1^+(\Sigma) \rightarrow \mathcal{L}_1(\Sigma)$ . The definition is by recursion on the well-order  $<$ . We take  $\text{trans}$  to be a homomorphism for all symbols except PRE. For this, we require the following:

$$\begin{aligned}
\text{trans}(\text{PRE}(\sigma; \psi_1, \dots, \psi_n)) &= \text{trans}(\psi_i) \\
\text{trans}(\text{PRE}(\alpha; \beta)) &= \langle \text{trans}(\alpha) \rangle \text{trans}(\text{PRE}(\beta)) \\
&= \neg[\text{trans}(\alpha)] \neg \text{trans}(\text{PRE}(\beta))
\end{aligned}$$

The reason we need recursion on  $<$  is that  $\text{trans}(\text{PRE}(\beta))$  figures in  $\text{trans}(\text{PRE}(\alpha; \beta))$ . So we need to know that  $\text{PRE}(\beta) < \text{PRE}(\alpha; \beta)$ . Of course, we do not really need the specific  $<$  that we constructed for this; for this minor point, we might as well define  $\varphi < \psi$  if  $\varphi$  has fewer symbols than  $\psi$ .

**Lemma 5.14.** For all action terms  $\alpha$ ,  $\text{trans}(\text{PRE}(\alpha)) = \text{PRE}(\text{trans}(\alpha))$ .

**Proof.** By induction on  $<$ . The important thing again is that when we write  $\text{PRE}(\text{trans}(\alpha))$  we are using  $\text{PRE}$  here as a defined symbol. Its recursion equations match the definition of  $\text{trans}$ .  $\square$

**Lemma 5.15.** Let  $\alpha$  be a ground action term of  $\mathcal{L}_1^+(\Sigma)$ . Then,

$$\{\text{trans}(\beta) : \alpha \xrightarrow{A} \beta \text{ in } \mathcal{L}_1^+(\Sigma)\} = \{\beta : \text{trans}(\alpha) \xrightarrow{A} \beta \text{ in } \mathcal{L}_1(\Sigma)\}.$$

**Proof.** By induction on  $\alpha$ .  $\square$

**Lemma 5.16.** If  $t_1$  and  $t_2$  are ground terms of  $\mathcal{L}_1^+(\Sigma)$  such that  $t_1 \xrightarrow{R} t_2$ , then  $\text{trans}(t_1) \equiv \text{trans}(t_2)$ .

**Proof.** Let  $u$  be a term with exactly one free variable  $x$ , and let  $\sigma$  be such that for some  $i$ ,  $t_1 = u(x \leftarrow l^\sigma)$  and  $t_2 = u(x \leftarrow r^\sigma)$ . We argue by induction on  $u$ . If  $u$  is just  $x$ , then we examine the rules of the system. All of them pertain to sentences except for (r9), and in the case of (r9) we use Lemma 4.20, part (4). The arguments for the other rules use Lemma 5.14, and we will give the details for two of them, (r3) and (r7).

For (r3),

$$\begin{aligned} \text{trans}(\text{PRE}(\alpha; \beta)) &\equiv \langle \text{trans}(\alpha) \rangle \text{trans}(\text{PRE}(\beta)) && \text{definition of trans} \\ &\equiv \text{PRE}(\text{trans}(\alpha)) \wedge [\text{trans}(\alpha) \text{trans}(\text{PRE}(\beta))] && \text{by Lemma 4.16} \\ &\equiv \text{trans}(\text{PRE}(\alpha)) \wedge [\text{trans}(\alpha)] \text{trans}(\text{PRE}(\beta)) && \text{using Lemma 5.14} \\ &\equiv \text{trans}(\text{PRE}(\alpha) \wedge [\alpha] \text{PRE}(\beta)) \end{aligned}$$

The infinite scheme (r7 <sub>$\alpha$</sub> ) is repeated below:

$$(r7_\alpha) \quad [\alpha] \Box_A x \rightsquigarrow \text{PRE}(\alpha) \rightarrow \bigwedge \{ \Box_A [\beta] x : \alpha \xrightarrow{A} \beta \}.$$

Recall that  $\alpha$  might have variables. What we need to check here is that ground instances of (r7 <sub>$\alpha$</sub> ) have equivalent translations. So take a ground sentence of the form  $[\alpha] \Box_A \varphi$ . Its translation is  $[\text{trans}(\alpha)] \Box_A (\text{trans}(\varphi))$ . Using the Action-Knowledge Axiom and Lemmas 5.15 and 5.14:

$$\begin{aligned} [\text{trans}(\alpha)] \Box_A (\text{trans}(\varphi)) &\equiv \text{PRE}(\text{trans}(\alpha)) \rightarrow \Box_A \bigwedge \{ [\beta] \text{trans}(\varphi) : \text{trans}(\alpha) \xrightarrow{A} \beta \} \\ &\equiv \text{trans}(\text{PRE}(\alpha)) \rightarrow \bigwedge \{ \Box_A [\text{trans}(\beta)] \text{trans}(\varphi) : \alpha \xrightarrow{A} \beta \} \\ &\equiv \text{trans} \left( \text{PRE}(\alpha) \rightarrow \bigwedge \{ \Box_A [\beta] \varphi : \alpha \xrightarrow{A} \beta \} \right) \end{aligned}$$

This settles all of the cases in this lemma when  $u$  is a variable by itself. In the general case, we need a fact about substitutions. Let  $t$  be any term of  $\mathcal{L}^+(\Sigma, X)$ . Let  $\sigma$  and  $\tau$  be two ground substitutions (these are substitutions all of whose values contain no variables) such that  $\text{trans}(\sigma(x))$  and  $\text{trans}(\tau(x))$  are equivalent (actions or sentences) for all variables  $x$ . Then,  $\text{trans}(t^\sigma)$  and  $\text{trans}(t^\tau)$  are again equivalent. The argument here is an easy induction on terms of  $\mathcal{L}^+(\Sigma, X)$ . It amounts to several facts which we have seen before:  $\equiv$  is a congruence for all of the syntactic operations on sentences, and also for those on programs (see Lemma 4.20); equivalent actions have equivalent preconditions (by definition); and if  $\alpha \equiv \beta$ , then  $[\alpha] \varphi \equiv [\beta] \varphi$  (Lemma 4.22).  $\square$

A *normal form* in a rewriting system is a term which cannot be rewritten in the system. In a terminating system such as our  $\mathcal{R}$ , one may define a normal form for a term by rewriting as much as possible in some fixed way (for example, by always rewriting in the leftmost possible way). We shall need some information on the normal forms of  $\mathcal{R}$ , and we shall turn to this shortly.

Near the beginning of Section 5, we defined a set  $\mathcal{NF} \subseteq \mathcal{L}_1(\Sigma)$ .  $\mathcal{NF}$  is the smallest set containing the atomic propositions and closed under all the boolean and modal operators;

with the property that if  $\alpha$  and  $\varphi$  belong to  $\mathcal{NF}$ , then so does  $[\alpha]\Box_C^*\varphi$  for all  $C \subseteq \mathcal{A}$ ; and, finally, that the programs built from sentences in  $\mathcal{NF}$  are themselves in  $\mathcal{NF}$  (see also (14) for a more complete description of this last closure condition.)

**Lemma 5.17.** *A ground term  $t \in \mathcal{L}_1^+(\Sigma)$  is a normal form of  $\mathcal{R}$  if  $t \in \mathcal{NF}$ . In particular,  $\text{PRE}$  and  $\rightarrow$  do not occur in normal forms.*

**Proof.** An induction on  $\mathcal{NF} \subseteq \mathcal{L}_1(\Sigma)$  shows that all sentences in  $\mathcal{NF}$  are ground terms of  $\mathcal{L}_1^+(\Sigma)$  and are, moreover, normal forms of the rewriting system  $\mathcal{R}$ : no rules of  $\mathcal{R}$  can apply at any point. Going the other way, we show by induction on  $\mathcal{L}_1^+(\Sigma)$  that if  $t$  is a (ground) normal form, then (regarding  $t$  as a sentence or action in  $\mathcal{L}_1(\Sigma)$ )  $\varphi \in \mathcal{NF}$ . The base case and the induction steps for  $\neg$ ,  $\wedge$ ,  $\Box_A$ , and  $\Box_B^*$  are all easy. Suppose our result holds for  $\varphi$ , and consider a normal form  $[\alpha]\varphi$ . Then,  $\alpha$  must be an action of the form  $[\sigma_i\vec{\psi}]$  with all  $\psi_j$  in normal form; if not, some rule would apply to  $\alpha$  and, hence, to  $[\alpha]\varphi$ . We are left to consider an action  $\alpha$ . By induction hypothesis, the subsentences of  $\alpha$ , too, are normal forms and, hence, belong to  $\mathcal{NF}$ . We claim that  $\alpha$  must be of the right-branching form in Equation (14). This is because all of the other possibilities are reducible in the system using (r9).  $\square$

**Lemma 5.18.** *A sentence  $\varphi \in \mathcal{L}_0(\Sigma)$  is a normal form of  $\mathcal{R}$  if  $\varphi$  is a modal sentence (that is, if  $\varphi$  contains no actions).*

**Proof.** By the easy part of Lemma 5.17, the modal sentences are normal forms. In the more significant direction, one first checks by induction on rewrite sequences that if  $\varphi \in \mathcal{L}_0(\Sigma)$  and  $\varphi'$  is reachable from  $\varphi$  by a finite number of rewrites in  $\mathcal{R}$ , then  $\Box^*$  does not appear in  $\varphi'$ . So every normal form of  $\varphi$  is a purely modal sentence. It follows that if  $\varphi$  were a normal form to begin with, then  $\varphi$  would be a modal sentence.  $\square$

**Corollary 5.19.** *For every  $\varphi \in \mathcal{L}_1(\Sigma)$ , there is a normal form  $\text{nf}(\varphi)$  such that  $\vdash \varphi \leftrightarrow \text{nf}(\varphi)$ . Moreover,  $\text{nf}(\varphi) \leq \varphi$ .*

**Proof.** Regard  $\varphi$  as a term in  $\mathcal{L}_1^+(\Sigma)$ . Let

$$\varphi = \varphi_1 \rightsquigarrow \varphi_2 \rightsquigarrow \cdots \rightsquigarrow \varphi_n = \text{nf}(\varphi)$$

be some sequence of rewrites which leads to a normal form of  $\varphi$ <sup>8</sup>. By Lemma 5.16,  $\vdash \text{trans}(\varphi) \leftrightarrow \text{trans}(\text{nf}(\varphi))$ . But neither  $\varphi$  nor  $\text{nf}(\varphi)$  contain  $\text{PRE}$  or  $\rightarrow$ , so they are literally equal to their translations. Thus,  $\vdash \varphi \leftrightarrow \text{nf}(\varphi)$ , just as desired.  $\square$

**Lemma 5.20.** *For all  $\varphi$  and  $\alpha$  of  $\mathcal{L}_1^+(\Sigma)$ :*

1.  $\text{PRE}(\alpha) < [\alpha]\varphi$ .
2. If  $\alpha \xrightarrow{A} \beta$ , then  $[\beta]\varphi < \neg[\beta]\varphi < [\alpha]\Box_C^*\varphi$ .
3. If  $[\alpha]\Box_C^*\psi$  is a normal form sentence and  $\alpha \xrightarrow{A} \beta$ , then  $\Box_A[\beta]\Box_C^*\psi$  and  $[\beta]\Box_C^*\psi$  are again normal forms.
4. Every subterm of a normal form is a normal form.
5. If  $\alpha$  is a normal-form action and  $\alpha \xrightarrow{A} \beta$ , then  $\beta$  is a normal-form action as well.

**Proof.** Part (1) is an easy calculation. In part (2), we use the fact that  $\llbracket \alpha \rrbracket$  and  $\llbracket \beta \rrbracket$  are the same number, by Proposition 5.9. We use the fact here (and here only) that  $\llbracket \Box^* \rrbracket(a) = a + 1$  rather than  $\llbracket \Box^* \rrbracket(a) = a$ . The remaining parts are also easy using the characterization of normal forms of Lemma 5.17.  $\square$

#### 5.2.4. The Function $f(\varphi)$

The last piece of business in this section is to define the function  $f(\varphi)$  from Lemma 5.4 and then to prove the required properties.

**Definition 5.21.** For  $t$ , an action or a sentence, let  $s(t)$  be the set of subsentences of  $t$ , including  $t$  itself (if  $t$  is a sentence). This includes all sentences occurring in actions which occur in  $\varphi$  and their subsentences. We define a function  $f : \mathcal{NF} \rightarrow \mathcal{P}(\mathcal{NF})$  by recursion on the well-founded relation  $<$  as follows:

$$\begin{aligned}
 f(p) &= \{p\} \\
 f(\neg\varphi) &= f(\varphi) \cup \{\neg\varphi\} \\
 f(\varphi \wedge \psi) &= f(\varphi) \cup f(\psi) \cup \{\varphi \wedge \psi\} \\
 f(\Box_A \varphi) &= f(\varphi) \cup \{\Box_A \varphi\} \\
 f(\Box_B^* \varphi) &= f(\varphi) \cup \{\Box_B^* \varphi\} \cup \{\Box_A \Box_B^* \varphi : A \in \mathcal{B}\} \\
 f([\alpha] \Box_C^* \varphi) &= \bigcup \{s(\Box_A [\beta] \Box_C^* \varphi) : \alpha \xrightarrow{C^*} \beta \text{ \& } A \in \mathcal{C}\} \\
 &\quad \bigcup \{f(\psi) : (\exists \beta) \alpha \xrightarrow{C^*} \beta \text{ \& } \psi \in s(\beta)\} \\
 &\quad \bigcup \{f(nf(\text{PRE}(\beta))) : \alpha \xrightarrow{C^*} \beta\} \\
 &\quad \bigcup f(\Box_C^* \varphi) \\
 &\quad \bigcup \{f(nf([\beta] \varphi)) : \alpha \xrightarrow{C^*} \beta\}
 \end{aligned}$$

The definition makes sense because the calls to  $f$  on the right-hand sides are all smaller than the arguments on the left-hand sides; see Lemma 5.20.)

**Lemma 5.22.** For all  $\varphi \in \mathcal{NF}$ :

1.  $\varphi \in f(\varphi)$ .
2.  $f(\varphi)$  is a finite set of normal form sentences.
3. If  $\psi \in f(\varphi)$ , then  $s(\psi) \subseteq f(\varphi)$ .
4. If  $\psi \in f(\varphi)$ , then  $f(\psi) \subseteq f(\varphi)$ .
5. If  $[\gamma] \Box_C^* \chi \in f(\varphi)$ ,  $\gamma \xrightarrow{C^*} \delta$ , and  $A \in \mathcal{C}$ , then  $f(\varphi)$  also contains  $\Box_A [\delta] \Box_C^* \chi$ ,  $[\delta] \Box_C^* \chi$ ,  $nf(\text{PRE}(\delta))$ , and  $nf([\delta] \chi)$ .

**Proof.** Part (1) is by cases on  $\varphi$ . The only interesting case is for  $[\alpha] \Box_C^* \varphi$ , and this is a subsentence of  $\Box_A [\alpha] \Box_C^* \varphi$  for any  $A \in \mathcal{C}$ .

All of other the parts are by induction on  $\varphi$  in the well-ordered  $<$ . For part (2), we use Lemma 5.20.

In part (3), we argue by induction on normal forms.

The result is immediate when  $\varphi$  is an atomic sentence  $p$ , and the induction steps for  $\neg$ ,  $\wedge$ , and  $\Box_A$  are easy. For  $\Box_B^* \varphi$ , note that since  $\varphi < \Box_B^* \varphi$ , our induction hypothesis implies the result for  $\varphi$ ; we verify it for  $\Box_B^* \varphi$ . When  $\psi$  is  $\Box_B^* \varphi$ , then

$$s(\psi) = s(\varphi) \cup \{\Box_B^* \varphi\} \subseteq f(\varphi) \cup \{\Box_B^* \varphi\} \subseteq f(\Box_B^* \varphi).$$

And then, when  $\psi$  is  $\Box_A \Box_B^* \varphi$  for some  $A \in \mathcal{B}$ , we have

$$s(\psi) = s(\Box_B^* \varphi) \cup \{\Box_A \Box_B^* \varphi\} \subseteq f(\Box_B^* \varphi).$$

To complete part (3), we consider  $[\alpha] \Box_C^* \varphi$ . If there is some  $\chi < [\alpha] \Box_C^* \varphi$  such that  $\psi \in f(\chi)$  and  $f(\chi) \subseteq f([\alpha] \Box_C^* \varphi)$ , then we argue as follows: by induction hypothesis,  $s(\psi) \subseteq f(\chi)$ ; and by hypothesis  $f(\chi) \subseteq f([\alpha] \Box_C^* \varphi)$ . This covers all of the cases except for  $\psi$  a subsentence of  $\Box_A [\beta] \Box_C^* \varphi$ . And here,  $s(\psi) \subseteq s(\Box_A [\beta] \Box_C^* \varphi) \subseteq f([\alpha] \Box_C^* \varphi)$ .

In Part (4), we again argue by induction on normal forms. The result is immediate when  $\varphi$  is an atomic sentence  $p$ . The induction steps for  $\neg$ ,  $\wedge$ , and  $\Box_A$  are easy. For  $\Box_B^* \varphi$ , note that since  $\varphi < \Box_B^* \varphi$ , our induction hypothesis implies the result for  $\varphi$ ; we verify it for  $\Box_B^* \varphi$ . The only interesting case is when  $\psi$  is  $\Box_A \Box_B^* \varphi$  for some  $A \in \mathcal{B}$ . And in this case

$$f(\psi) = f(\Box_B^* \varphi) \cup \{\Box_A \Box_B^* \varphi\} \subseteq f(\Box_B^* \varphi).$$

To complete part (4), we consider  $[\alpha]\Box_C^*\varphi$ . If there is some  $\chi < [\alpha]\Box_C^*\varphi$  such that  $\psi \in f(\chi)$  and  $f(\chi) \subseteq f([\alpha]\Box_C^*\varphi)$ , then we are easily done by the induction hypothesis. This covers all of the cases except for  $\psi$  of the form  $[\beta]\Box_C^*\varphi$  or of the form  $\Box_A[\beta]\Box_C^*\varphi$ . (If  $\psi$  is a subsentence of some  $\beta$ , then  $f(\psi) \subseteq f([\alpha]\Box_C^*\varphi)$  directly. If  $\psi$  is a subsentence of  $\Box_C^*\varphi$ , then by induction hypothesis,  $f(\psi) \subseteq f(\Box_C^*\varphi)$ ; and directly,  $f(\Box_C^*\varphi) \subseteq f([\alpha]\Box_C^*\varphi)$ .) For  $[\beta]\Box_C^*\varphi$ , we use the transitivity of  $\xrightarrow{C^*}$  to check that  $f([\beta]\Box_C^*\varphi) \subseteq f([\alpha]\Box_C^*\varphi)$ . And now the case of  $\Box_A[\beta]\Box_C^*\varphi$  follows:

$$f(\Box_A[\beta]\Box_C^*\varphi) = f([\beta]\Box_C^*\varphi) \cup \{\Box_A[\beta]\Box_C^*\varphi\} \subseteq f([\alpha]\Box_C^*\varphi).$$

For part (5), assume that  $[\gamma]\Box_C^*\chi \in f(\varphi)$ . By part (2),  $[\gamma]\Box_C^*\chi$  is a normal form. The definition of  $f$  implies that  $\Box_A[\delta]\Box_C^*\chi$ ,  $[\delta]\Box_C^*\chi$ ,  $\text{nf}(\text{PRE}(\delta))$ , and  $\text{nf}([\delta]\chi)$  all belong to  $f([\gamma]\Box_C^*\chi)$ , and then part (4) tells us that  $f([\gamma]\Box_C^*\chi) \subseteq f(\varphi)$ .  $\square$

### 5.2.5. Summary

The purpose of this section was to prove Lemmas 5.2–5.4 in Section 5. Lemma 5.2 is Corollary 5.19. Lemma 5.3 comes from Corollary 5.19 and Lemma 5.20. Lemma 5.4 is contained in Lemma 5.22.

### 5.3. Strong Completeness for $\mathcal{L}_0(\Sigma)$

Recall that in the languages  $\mathcal{L}_0(\Sigma)$ , we do not have the common-knowledge operators  $\Box_B^*$  or the action iterations  $\pi^*$ . At this point, we can put together several results from our previous work to obtain a completeness theorem for languages of the form  $\mathcal{L}_0(\Sigma)$ . The overall ideas are: (1) we need a logical system which is strong enough to translate each sentence of  $\mathcal{L}_0(\Sigma)$  to a normal form; (2) this normal form will be a purely modal sentence; and (3) the system should be at least as strong as multimodal  $K$ .

**Proposition 5.23.** *Every sentence  $\varphi$  of  $\mathcal{L}_0(\Sigma)$  is provably equivalent to a sentence  $\varphi^*$  in which there are no occurrences of  $\Box$ , crash,  $;$ , or skip.*

**Theorem 5.24.** *The logical system for  $\mathcal{L}_0(\Sigma)$  is strongly complete: for all sets  $T \subseteq \mathcal{L}_0(\Sigma)$ ,  $T \vdash \varphi$  iff  $T \models \varphi$ .*

**Proof.** The soundness half being easy, we only need to show that if  $T \models \varphi$ , then  $T \vdash \varphi$ .

First, we may assume that the symbols  $\Box$ , crash,  $;$ , and skip do not occur in  $T$  or  $\varphi$ . Thus, we may work with  $\mathcal{L}_0(\Sigma) \cap \mathcal{L}_1^+(\Sigma)$ . In particular, we have normal forms.

Next, for each  $\chi$  of  $\mathcal{L}_0(\Sigma)$ ,  $\vdash \chi \leftrightarrow \text{nf}(\chi)$ . As a result,  $T \vdash \text{nf}(\chi)$  for all  $\chi \in T$ .

Finally, write  $\text{nf}(T)$  for  $\{\text{nf}(\chi) : \chi \in T\}$ . By soundness,  $\text{nf}(T) \models \text{nf}(\varphi)$ . Since our system extends the standard complete proof system of modal logic,  $\text{nf}(T) \vdash \text{nf}(\varphi)$ . So  $T \vdash \text{nf}(\varphi)$ . As we know,  $\vdash \varphi \leftrightarrow \text{nf}(\varphi)$ . So we have our desired conclusion:  $T \vdash \varphi$ .  $\square$

### 5.4. Weak Completeness for $\mathcal{L}_1(\Sigma)$

The proof of completeness and decidability of  $\mathcal{L}_1(\Sigma)$  is based on the filtration argument for completeness of PDL due to Kozen and Parikh [15]. We show that every consistent  $\varphi$  has a finite model, and that the size of the model is recursive in  $\varphi$ . As in the last section, we depend on the results of Lemmas 5.2–5.4.

The Set  $\Delta = \Delta(\varphi)$

Fix a sentence  $\varphi$ . We set  $\Delta = f(\varphi)$  (i.e., we drop  $\varphi$  from the notation). This set  $\Delta$  is the version for our logic of the Fischer–Ladner closure of  $\varphi$ , originating in [17]. Let  $\Delta = \{\psi_1, \dots, \psi_n\}$ . Given a maximal consistent set  $U$  of  $\mathcal{L}_1(\Sigma)$ , let

$$U^* = \pm \psi_1 \wedge \dots \wedge \pm \psi_n,$$



where the signs are taken in accordance with membership in  $U$ . That is, if  $\psi_i \in U$ , then  $\psi$  is a conjunct of  $U^*$ ; but if  $\psi_i \notin U$ , then  $\neg\psi_i$  is a conjunct.

Two (standard) observations are in order. Notice that if  $U^* \neq V^*$ , then  $U^* \wedge V^*$  is inconsistent. Also, for all  $\psi \in \Delta$ ,

$$\vdash \psi \leftrightarrow \bigvee \{W^* : W \text{ is maximal consistent and } \psi \in W\}. \quad (15)$$

and

$$\vdash \neg\psi \leftrightarrow \bigvee \{W^* : W \text{ is maximal consistent and } \neg\psi \in W\}. \quad (16)$$

(The reason is that  $\psi$  is equivalent to the disjunction of *all* complete conjunctions which contain it. However, some of those complete conjunctions are inconsistent and these can be dropped from the big disjunction. The others are consistent and, hence, can be extended to maximal consistent sets.)

**Definition 5.25.** We consider maximal consistent sets  $U$  in the logic for  $\mathcal{L}_1(\Sigma)$ . Let  $U \equiv V$  if  $U^* = V^*$  (if  $U \cap \Delta = V \cap \Delta$ ). The *filtration*  $\mathcal{F}$  is the model whose worlds are the equivalence classes  $[U]$  in this relation. Furthermore, we set

$$[U] \xrightarrow{\Delta} [V] \text{ in } \mathcal{F} \quad \text{iff} \quad \text{whenever } \Box_A \psi \in U \cap \Delta, \text{ then also } \psi \in V. \quad (17)$$

We complete the specification of a state model with the valuation:

$$\|p\|_{\mathcal{F}} = \{[U] : p \in U \cap \Delta\}. \quad (18)$$

The definitions in Equations (18) and (17) are independent of the choice of representatives: we use part (1) of Lemma 5.4 to see that if  $\Box_A \chi \in \Delta$ , then also  $\chi \in \Delta$ .

**Proposition 5.26.** If  $U^* \wedge \Diamond_A V^*$  is consistent, then  $[U] \rightarrow_A [V]$ .

**Proof.** Assume  $\Box_A \psi \in U \cap \Delta$  and toward a contradiction that  $\psi \notin V$ . Since  $\psi \in \Delta$  and  $\neg\psi \in V$ , we have  $\vdash V^* \rightarrow \neg\psi$ . Thus,  $\vdash \Diamond_A V^* \rightarrow \Diamond_A \neg\psi$ , and so  $\vdash U^* \wedge \Diamond_A V^* \rightarrow \Box_A \psi \wedge \Diamond_A \neg\psi$ . Hence  $U^* \wedge \Diamond_A V^*$  is inconsistent.  $\square$

**Definition 5.27.** Let  $\langle \alpha \rangle \Diamond_{\mathcal{C}}^* \psi$  be a normal form. A *good path* from  $[V_0]$  for  $\langle \alpha \rangle \Diamond_{\mathcal{C}}^* \psi$  is a path in  $\mathcal{F}$

$$[V_0] \rightarrow_{A_1} [V_1] \rightarrow_{A_2} \cdots \rightarrow_{A_{k-1}} [V_{k-1}] \rightarrow_{A_k} [V_k] \quad (19)$$

such that  $k \geq 0$ , each  $A_i \in \mathcal{C}$ , and such that there exist actions

$$\alpha = \alpha_0 \rightarrow_{A_1} \alpha_1 \rightarrow_{A_2} \cdots \rightarrow_{A_{k-1}} \alpha_{k-1} \rightarrow_{A_k} \alpha_k$$

such that  $\text{PRE}(\alpha_i) \in V_i$  for all  $0 \leq i \leq k$ , and  $\langle \alpha_k \rangle \psi \in V_k$ .

The idea behind a good path comes from considering Lemma 4.11 in  $\mathcal{F}$ . Of course, the special case of that result would require that  $\langle \mathcal{F}, [V_i] \rangle \models \text{PRE}(\alpha_i)$  rather than  $\text{PRE}(\alpha_i) \in V_i$ , and similarly for  $\langle \alpha_k \rangle \psi$  and  $V_k$ . The exact formulation above was made in order that the Truth Lemma will go through for sentences of the form  $\langle \alpha \rangle \Diamond_{\mathcal{C}}^* \psi$  (see the final paragraphs of the proof of Lemma 5.30).

**Lemma 5.28.** Let  $[\alpha] \Box_{\mathcal{C}}^* \psi \in \Delta$ . If there is a good path from  $[V_0]$  for  $\langle \alpha \rangle \Diamond_{\mathcal{C}}^* \neg\psi$ , then  $\langle \alpha \rangle \Diamond_{\mathcal{C}}^* \neg\psi \in V_0$ .

**Proof.** By induction on the length  $k$  of the path. If  $k = 0$ , then  $\langle \alpha \rangle \neg\psi \in V_0$ . If  $\langle \alpha \rangle \Diamond_{\mathcal{C}}^* \neg\psi \notin V_0$ , then  $[\alpha] \Box_{\mathcal{C}}^* \psi \in V_0$ . By Lemma 5.4, part (2), we have  $\text{nf}([\alpha] \psi) \in V_0$ . This is a contradiction.

Assume the result for  $k$ , and suppose that there is a good path from  $[V_0]$  for  $\langle \alpha \rangle \Diamond_{\mathcal{C}}^* \neg\psi$  of length  $k + 1$ . We adopt the notation from (19) for this good path. Then there is a good

path of length  $k$  from  $[V_1]$  for  $\langle \alpha_1 \rangle \diamond_C^* \neg \psi$ . Also,  $[\alpha_1] \Box_C^* \psi \in \Delta$ , by Lemma 5.4, part (2). By induction hypothesis,  $\langle \alpha_1 \rangle \diamond_C^* \neg \psi \in V_1$ .

If  $\langle \alpha \rangle \diamond_C^* \neg \psi \notin V_0$ , then  $[\alpha] \Box_C^* \psi \in V_0$ .  $V_0$  contains  $[\alpha] \Box_C^* \psi \wedge \text{nf}(\text{PRE}(\alpha)) \rightarrow \Box_A[\alpha_1] \Box_C^* \psi$ . (That is, this sentence is valid by Lemma 4.21, part (2). Hence, it belongs to every maximal consistent set.) So  $V_0$  contains  $\Box_A[\alpha_1] \Box_C^* \psi$ . This sentence belongs to  $\Delta$  by Lemma 5.4, part (2). Now by definition of  $\xrightarrow{\Delta}$  in  $\mathcal{F}$ , we see that  $[\alpha_1] \Box_C^* \psi \in V_1$ . This is a contradiction to our observation at the end of the previous paragraph.  $\square$

**Lemma 5.29.** *If  $V_0^* \wedge \langle \alpha \rangle \diamond_C^* \psi$  is consistent, then there is a good path from  $[V_0]$  for  $\langle \alpha \rangle \diamond_C^* \psi$ .*

**Proof.** For each  $\beta$  such that  $\alpha \xrightarrow{C^*} \beta$ , let  $S_\beta$  be the (finite) set of all  $[W] \in \mathcal{F}$  such that there is *no* good path from  $[W]$  for  $\langle \beta \rangle \diamond_C^* \psi$ . We need to see that  $[V_0] \notin S_\alpha$ ; suppose toward a contradiction that  $[V_0] \in S_\alpha$ . Let

$$\chi_\beta = \bigvee \{W^* : W \in S_\beta\}.$$

Note that  $\neg \chi_\beta$  is logically equivalent to  $\bigvee \{X^* : [X] \in \mathcal{F} \text{ and } X \notin S_\beta\}$ . Since we assumed  $[V_0] \in S_\alpha$ , we have  $\vdash V_0^* \rightarrow \chi_\alpha$ .

We first claim for  $\beta$  such that  $\alpha \xrightarrow{C^*} \beta$ ,  $\chi_\beta \wedge \langle \beta \rangle \psi$  is inconsistent. Otherwise, there would be  $[W] \in S_\beta$  such that  $\chi_\beta \wedge \langle \beta \rangle \psi \in W$ . Note that, by the partial-functionality axiom,  $\vdash \langle \beta \rangle \psi \rightarrow \text{PRE}(\beta)$ . But then the one-point path  $[W]$  is a good path from  $[W]$  for  $\langle \beta \rangle \diamond_C^* \psi$ . Thus,  $[W] \notin S_\beta$ , and this is a contradiction. So indeed,  $\chi_\beta \wedge \langle \beta \rangle \psi$  is inconsistent. Therefore,  $\vdash \chi_\beta \rightarrow [\beta] \neg \psi$ .

We next show that for all  $A \in \mathcal{C}$  and all  $\beta$  such that  $\beta \rightarrow_A \gamma$ ,  $\chi_\beta \wedge \text{PRE}(\beta) \wedge \diamond_A \neg \chi_\gamma$  is inconsistent. Otherwise, there would be  $[W] \in S_\beta$  with  $\chi_\beta$ ,  $\text{PRE}(\beta)$ , and  $\diamond_A \neg \chi_\gamma$  in it. Then,  $\bigvee \{\diamond_A X^* : X \notin S_\gamma\}$ , being equivalent to  $\diamond_A \neg \chi_\gamma$ , would belong to  $W$ . It follows that  $\diamond_A X^* \in W$  for some  $[X] \notin S_\gamma$ . By Proposition 5.26,  $[W] \rightarrow_A [X]$ . Since  $[X] \notin S_\gamma$ , there is a good path from  $[X]$  for  $\langle \gamma \rangle \diamond_C^* \psi$ . But since  $\beta \rightarrow_A \gamma$  and  $W$  contains  $\text{PRE}(\beta)$ , we also have a good path from  $[W]$  for  $\langle \beta \rangle \diamond_C^* \psi$ . This again contradicts  $[W] \in S_\beta$ . As a result, for all relevant  $A$ ,  $\beta$ , and  $\gamma$ ,  $\vdash \chi_\beta \wedge \text{PRE}(\beta) \rightarrow \Box_A \chi_\gamma$ .

By the Action Rule,  $\vdash \chi_\alpha \rightarrow [\alpha] \Box_C^* \neg \psi$ . Now  $\vdash V_0^* \rightarrow \chi_\alpha$ . So  $\vdash V_0^* \rightarrow [\alpha] \Box_C^* \neg \psi$ . This contradicts the assumption with which we began this proof.  $\square$

**Lemma 5.30 (Truth Lemma).** *Consider a sentence  $\varphi$ , and also the set  $\Delta = f(\varphi)$ . For all  $\chi \in \Delta$  and  $[U] \in \mathcal{F}$ :  $\chi \in U$  if  $\langle \mathcal{F}, [U] \rangle \models \chi$ .*

**Proof.** We argue by induction on the well-founded  $<$  that if  $\chi \in \Delta$ , then:  $\chi \in U$  if  $\langle \mathcal{F}, [U] \rangle \models \chi$ . The case of  $\chi$  atomic is trivial. Now assume this Truth Lemma for sentences  $< \chi$ . Recall that  $\Delta \subseteq \mathcal{NF}$ , our set of normal forms (see Section 5.2.3). We argue by cases on  $\chi$ .

The cases that  $\chi$  is either a negation or conjunction are trivial.

Suppose next that  $\chi \equiv \Box_A \psi$ . Suppose  $\Box_A \psi \in U$ ; we show  $\langle \mathcal{F}, [U] \rangle \models \Box_A \psi$ . Let  $[V]$  be such that  $[U] \xrightarrow{\Delta} [V]$ . Then by definition of  $\xrightarrow{\Delta}$ ,  $\psi \in V$ . The induction hypothesis applies to  $\psi$ , since  $\psi < \Box_A \psi$ , and since  $\psi \in \Delta$  by Lemma 5.4, part (1). So by induction hypothesis,  $\langle \mathcal{F}, [V] \rangle \models \psi$ . This gives half of our equivalence. Conversely, suppose that  $\langle \mathcal{F}, [U] \rangle \models \Box_A \psi$ . Suppose towards a contradiction that  $\diamond_A \neg \psi \in U$ . So  $U^* \wedge \diamond_A \neg \psi$  is consistent. We use Equation (16) and the fact that  $\diamond_A$  distributes over disjunctions to see that  $U^* \wedge \diamond_A \neg \psi$  is logically equivalent to  $\bigvee (U^* \wedge \diamond_A V^*)$ , where the disjunction is taken over all  $V$  which contain  $\neg \psi$ . Since  $U^* \wedge \diamond_A \neg \psi$  is consistent, one of the disjuncts  $U^* \wedge \diamond_A V^*$  must be consistent. The induction hypothesis again applies, and we use it to see that  $\langle \mathcal{F}, [V] \rangle \models \neg \psi$ . By Proposition 5.26,  $[U] \xrightarrow{\Delta} [V]$ . We conclude that  $\langle \mathcal{F}, [U] \rangle \models \diamond_A \neg \psi$ , and this is a contradiction.

For  $\chi$  of the form  $\Box_C^* \psi$ , we use the standard argument for PDL (see Kozen and Parikh [15]). This is based on lemmas that parallel Lemmas 5.28 and 5.29. The work

is somewhat easier than what we do below for sentences of the form  $[\alpha]\Box_C^*\psi$ , and so we omit these details.

We conclude with the case when  $\chi$  is a normal form sentence of the form  $[\alpha]\Box_C^*\psi \in \Delta$ . Assume that  $[\alpha]\Box_C^*\psi \in \Delta$ . First, suppose that  $[\alpha]\Box_C^*\psi \notin U$ . Then, by Lemma 5.29, there is a good path from  $[U]$  for  $\langle\alpha\rangle\Diamond_C^*\neg\psi$ . We want to apply Lemma 4.11 in  $\mathcal{F}$  to assert that  $\langle\mathcal{F}, [U]\rangle \models \langle\alpha\rangle\Diamond_C^*\neg\psi$ . Let  $k$  be the length of the good path. For  $i \leq k$ ,  $\text{PRE}(\alpha_i) \in U_i$ . Now each  $\text{nf}(\text{PRE}(\alpha_i))$  belongs to  $\Delta$  by Lemma 5.4, part (2), and is  $< [\alpha]\Box_C^*\psi$ . So by induction hypothesis,  $\langle\mathcal{F}, [U_i]\rangle \models \text{nf}(\text{PRE}(\alpha_i))$ . By soundness,  $\langle\mathcal{F}, [U_i]\rangle \models \text{PRE}(\alpha_i)$ . We also need to check that  $\langle\mathcal{F}, [U_k]\rangle \models \langle\alpha_k\rangle\neg\psi$ . For this, recall from Lemma 5.3 that  $\Delta$  contains  $\text{nf}(\neg[\alpha_k]\psi) \leq \neg[\alpha_k]\psi < [\alpha]\Box_C^*\psi$ . Since the path is good,  $U_k$  contains  $\langle\alpha_k\rangle\neg\psi$ ; thus, it contains  $\neg[\alpha_k]\psi$ ; and, finally, it contains  $\text{nf}(\neg[\alpha_k]\psi)$ . By induction hypothesis,  $\langle\mathcal{F}, [U_k]\rangle \models \text{nf}(\neg[\alpha_k]\psi)$ . By soundness,  $\langle\mathcal{F}, [U_k]\rangle \models \neg[\alpha_k]\psi$ . Thus,  $\langle\mathcal{F}, [U_k]\rangle \models \langle\alpha_k\rangle\neg\psi$ . Now it does follow from Lemma 4.11 that  $\langle\mathcal{F}, [U]\rangle \models \langle\alpha\rangle\Diamond_C^*\neg\psi$ .

Going the other way, suppose that  $\langle\mathcal{F}, [U]\rangle \models \langle\alpha\rangle\Diamond_C^*\neg\psi$ . By Lemma 4.11, we obtain a path in  $\mathcal{F}$  witnessing this. The argument of the previous paragraph shows that this path is a good path from  $[U]$  for  $\langle\alpha\rangle\Diamond_C^*\neg\psi$ . By Lemma 5.28,  $U$  contains  $\langle\alpha\rangle\Diamond_C^*\neg\psi$ . This completes the proof.  $\square$

**Theorem 5.31.** (Completeness). *For all  $\varphi$ ,  $\vdash \varphi$  if  $\models \varphi$ . Moreover, this relation is decidable.*

**Proof.** By Lemma 5.2,  $\vdash \varphi \leftrightarrow \text{nf}(\varphi)$ . Let  $\varphi$  be consistent. By the Truth Lemma,  $\text{nf}(\varphi)$  holds at some world in the filtration  $\mathcal{F}$ . So  $\text{nf}(\varphi)$  has a model; thus,  $\varphi$  has one, too. This establishes completeness. For decidability, note that the size of the filtration is computable in the size of the original  $\varphi$ . (Another proof of decidability: we show in Section 6.1 that  $\mathcal{L}_1(\Sigma)$  can be translated into propositional dynamic logic (PDL) fairly directly, and that logic is decidable. Neither argument gives a good estimate of the complexity.)  $\square$

### 5.5. Extensions to the Completeness Theorem

We briefly mention extensions of the Completeness Theorem 5.31.

First, consider the case of S5 (or K45) actions. We change our logical system by restricting to these S5 actions, and we add the S5 axioms to our logical system. We interpret this new system on S5 models. It is easy to check that applying an S5 action to an S5 model gives another S5 model. Further, the S5 actions are closed under composition. Finally, if  $\alpha$  is an S5 action and  $\alpha \rightarrow_A \beta$ , then  $\beta$  also is an S5 action. These easily imply the soundness of the new axioms. For completeness, we need only check that if we assume the S5 axioms, then the filtration  $\mathcal{F}$  from the previous section has the property that each  $\overset{A}{\rightarrow}$  is an equivalence relation. This is a standard exercise in modal logic (see, e.g., Fagin et al. [18], Theorem 3.3.1).

Second, we also have completeness not only for the languages  $\mathcal{L}_1(\Sigma)$ , but also for the languages in [6] that are constructed from *families*  $\mathcal{S}$  of action signatures. This construction is most significant in the case when  $\mathcal{S}$  is an infinite set of signatures; for example,  $\mathcal{S}$  might contain a copy of every finite signature. In that setting, the language  $\mathcal{L}(\mathcal{S})$  would not be the language of any finite signature. But for the (weak) completeness result of this section, the advantage of the extended definition is lost. The point is that for single sentences, we may restrict attention to a finite subset of  $\mathcal{S}$ . And for finite sets  $\mathcal{S}$ ,  $\mathcal{L}(\mathcal{S})$  is literally the language of the coproduct signature

$$\oplus\{\Sigma : \Sigma \in \mathcal{S}\}.$$

We already have completeness for such languages.

Our final extension concerns the move from actions as we have been working with them to actions which change the truth values of atomic sentences. If we make this move, then the axiom of Atomic Permanence is no longer sound. However, it is easy to formulate the relevant axioms. For example, if we have an action  $\alpha$  which effects the change  $p := p \wedge \neg q$ , then we would take an axiom  $[\alpha]p \leftrightarrow (\text{PRE}(\alpha) \rightarrow p \wedge \neg q)$ . Having made

these changes, all of the rest of the work we have done goes through. In this way, we get a completeness theorem for this logic.

#### Endnotes

Special cases of Theorem 5.24 for some of the target logics are due to Plaza [1], Gerbrandy [2,3], and Gerbrandy and Groeneveld [4]. The proofs in these sources also go via translation to modal logic.

## 6. Results on Expressive Power

In this section, we study a number of expressive power issues related to our logics. The four subsections are for the most part independent.

### 6.1. Translation of $\mathcal{L}_1(\Sigma)$ into PDL

In this section,  $\Sigma$  is an arbitrary action signature. In Section 5.2.3, we saw normal forms for  $\mathcal{L}_0(\Sigma)$  and  $\mathcal{L}_1(\Sigma)$ . We showed in that section that every sentence in  $\mathcal{L}_1(\Sigma)$  is provably equivalent to its normal form, and the normal forms of sentences of  $\mathcal{L}_0(\Sigma)$  are exactly the purely modal sentences. This proves that in terms of expressive power,  $\mathcal{L}_0(\Sigma)$  is equivalent to  $\mathcal{L}_0$ , ordinary modal logic.

Further, we can show that  $\mathcal{L}_1(\Sigma)$  and, indeed, the full language  $\mathcal{L}(\Sigma)$  is a sublogic of  $\mathcal{L}_0^\omega$ , the extension of modal logic with countable Boolean conjunction and disjunction. The idea is contained in the following clauses:

$$\begin{aligned} (\Box_B^* \varphi)^t &= \bigwedge_{\langle A_1, \dots, A_n \rangle \in B^*} (\Box_{A_1} \cdots \Box_{A_n} \varphi)^t \\ ([\alpha] \Box_B^* \psi)^t &= \bigwedge_{\langle A_1, \dots, A_n \rangle \in B^*} ([\alpha] \Box_{A_1} \cdots \Box_{A_n} \psi)^t \\ ([\pi]^* \varphi)^t &= \bigwedge_n ([\pi]^n \varphi)^t \end{aligned}$$

It is natural to ask whether there are any finite logics which have been previously studied and into which our logics can be embedded. One possibility is the Modal Iteration Calculus (MIC) introduced in Dawar, Grädel, and Kreutzer [19]. The full language  $\mathcal{L}(\Sigma_{\text{pub}})$  is a sublanguage of MIC; see [14] for details. It is likely that this result extends to all other finite action signatures. In another direction, we ask whether the fragments  $\mathcal{L}_1(\Sigma)$  are sublogics of previously studied systems.

**Theorem 6.1.** *Every sentence of  $\mathcal{L}_1(\Sigma)$  is equivalent to a sentence of PDL.*

**Proof.** (Sketch) We argue by induction on the well-order  $<$  introduced and studied in Section 5. It is sufficient to show that each sentence in the set  $\mathcal{NF}$  of normal forms of  $\mathcal{L}_1(\Sigma)$  is equivalent to a sentence of PDL. (The normal forms were introduced in Section 5.2.3, and the reader may wish to look back at Lemma 5.17.) We just give the main induction step.

Suppose that  $[\alpha] \Box_C^* \varphi$  is a normal-form sentence. Our induction hypothesis implies that each  $\psi < [\alpha] \Box_C^* \varphi$  is equivalent to some PDL sentence  $\psi'$ . We shall show that  $\langle \alpha \rangle \Diamond_C^* \varphi$  is itself equivalent to some PDL sentence; hence,  $[\alpha] \Box_C^* \varphi$  also has this property. For this, we use the semantic equivalent given in Lemma 4.11. Recall first that there are only finitely many  $\beta$  such that  $\alpha \xrightarrow{C^*} \beta$ . Let  $X$  be the (finite) set of all such  $\beta$  such that  $\alpha \xrightarrow{C^*} \beta$ . That is, the actions  $\beta$  reachable from  $\alpha$  by a path labeled by agents in the set  $\mathcal{C}$ . We consider  $X$  as a sub-action structure of  $\Omega$ .

We consider  $\mathcal{C} \cup X$ , and we assume that this union is disjoint. We consider the set  $(\mathcal{C} \cup X)^*$  of all finite words on this set. We are interested in finite  $\mathcal{C}$ -labeled paths through  $X$  beginning at  $\alpha$  and ending at an arbitrary element of  $X$ . At this point, we shall develop our proof only by example.

Suppose that  $\mathcal{C} = \{A, B\}$  and that  $X = \{\alpha, \beta\}$ , with  $\alpha \xrightarrow{A} \alpha$ ,  $\alpha \xrightarrow{B} \beta$ , and  $\beta \xrightarrow{A} \alpha$ . Then one of the paths of interest would be

$$\alpha \quad A \quad \alpha \quad B \quad \beta \quad A \quad \alpha$$

(Note that this corresponds to  $\alpha \xrightarrow{A} \alpha \xrightarrow{B} \beta \xrightarrow{A} \alpha$ .) We are only interested in paths that respect the structure of  $\Omega$ . By Kleene's Theorem, the set  $P$  of finite paths of this type beginning at our fixed action  $\alpha$  is a regular language on  $\mathcal{C} \cup X$ . In our example,  $P$  is given by the regular expression

$$((\alpha(A\alpha)^*(B\beta A))^*(\epsilon + B\beta)).$$

With each such regular expression we associate a PDL program that represents it. In our example, we would have

$$(?PRE'(\alpha); ((\langle A \rangle; ?PRE'(\alpha))^*; (\langle B \rangle; ?PRE'(\beta); \langle A \rangle))^*; (?true + \langle B \rangle; ?PRE'(\beta))).$$

(The notation  $PRE'(\gamma)$  means the PDL translation of  $PRE(\gamma)$ ; such a PDL sentence exists since  $PRE(\gamma) < [\alpha]\Box_{\mathcal{C}}^*\varphi$ .) Again, this is a PDL program  $\pi = \pi(X, \alpha)$  whose denotation  $\llbracket \pi \rrbracket_{\mathbf{S}}$  in a state model  $\mathbf{S}$  is the set of pairs  $(s, t)$  of states such that there is a path

$$s = s_0 \rightarrow_{A_1} s_1 \rightarrow_{A_2} \cdots \rightarrow_{A_{k-1}} s_{k-1} \rightarrow_{A_k} s_k$$

and also a sequence of actions of the same length  $k$ ,

$$\alpha = \alpha_0 \rightarrow_{A_1} \alpha_1 \rightarrow_{A_2} \cdots \rightarrow_{A_{k-1}} \alpha_{k-1} \rightarrow_{A_k} \alpha_k$$

such that each  $A_i \in \mathcal{C}$ , each  $s_i \in \llbracket PRE(\alpha_i) \rrbracket_{\mathbf{S}}$  for all  $0 \leq i \leq k$ , and, finally, such that  $s_k = t$ . Consider now the PDL sentence  $\langle \pi \rangle (\langle \alpha_k \rangle \varphi)'$ . (As above,  $(\langle \alpha_k \rangle \varphi)'$  means the PDL translation of  $\langle \alpha_k \rangle \varphi$ . This exists because  $[\alpha_k]\varphi < [\alpha]\Box_{\mathcal{C}}^*\varphi$ ; see Lemma 5.17.) A state  $s$  satisfies  $\langle \pi \rangle (\langle \alpha_k \rangle \varphi)'$  in  $\mathbf{S}$  if there is some  $(s, t) \in \llbracket \pi \rrbracket_{\mathbf{S}}$  such that  $t \in \llbracket \langle \alpha_k \rangle \varphi \rrbracket_{\mathbf{S}}$ . Putting together our description of  $\llbracket \pi \rrbracket_{\mathbf{S}}$  with this, we see that  $(s, t) \in \llbracket \pi \rrbracket_{\mathbf{S}}$  iff  $s \in \llbracket \langle \alpha \rangle \Diamond_{\mathcal{C}}^* \rrbracket_{\mathbf{S}}$ ; see Lemma 4.11.  $\square$

**Remark 6.2.** Theorem 6.1 also follows from the closure of a different system, epistemic PDL (not discussed in this paper), under action modalities. This was proved by Jan van Eijck [20], and it also appears in [9]. The method of proof is very similar to what we have presented.

At this point, we know that  $\mathcal{L}_1(\Sigma)$  is a sublogic of PDL. So it is interesting to ask whether this extends to the full logic  $\mathcal{L}(\Sigma)$ ; recall that this last language has the operation of *program* iteration  $\pi^*$ . It turns out that  $\mathcal{L}(\Sigma)$  lacks the finite-model property even when  $\Sigma$  is as simple as the signature of public announcements and, indeed, when the only sentence announced publically and repeatedly is  $\Diamond true$  (see Section 6.4, below). Since PDL has the finite model property, we see that  $\mathcal{L}(\Sigma)$  is not a sublogic of PDL. In fact, it also shows that  $\mathcal{L}(\Sigma)$  is not even a sublogic of the modal mu-calculus.

## 6.2. $\mathcal{L}_1(\Sigma_{pub})$ is More Expressive than $\mathcal{L}_1$

Recall that  $\mathcal{L}_1$  in this paper is multi-agent modal logic together with the common-knowledge operators  $\Box_B^*$  for sets of agents. Our main result here is that  $\mathcal{L}_1$  is strictly weaker than  $\mathcal{L}_1(\Sigma_{pub})$ , the logic obtained by adding public announcements to  $\mathcal{L}_1$ .

We define a *rank*  $|\varphi|$  on sentences from  $\mathcal{L}_1(\Sigma_{pri})$ . Let  $|p| = 0$  for  $p$  atomic,  $|\neg\varphi| = |\varphi|$ ,  $|\varphi \wedge \psi| = \max(|\varphi|, |\psi|)$ ,  $|\Diamond_A \varphi| = 1 + |\varphi|$ , for all  $A \in \mathcal{A}$ , and  $|\Diamond_B^* \varphi| = 1 + |\varphi|$  for all  $B \subseteq \mathcal{A}$ .

### 6.2.1. Games for $\mathcal{L}_1$

The main technique in the proof is an adaptation of Fraisse–Ehrenfeucht games to the setting of modal logic. Let  $(\mathbf{S}, s)$  and  $(\mathbf{T}, t)$  be states; i.e., model-world pairs. By recursion on the natural number  $n$ , we define a game  $G_n((\mathbf{S}, s), (\mathbf{T}, t))$ . For  $n = 0$ ,  $I$  immediately wins if the following holds: for all  $p \in \text{AtSen}$ ,  $(\mathbf{S}, s) \models p$  if  $(\mathbf{T}, t) \models p$ . And if  $s$  and  $t$  differ on some atomic sentence,  $I$  immediately wins. Continuing, here is how we define  $G_{n+1}((\mathbf{S}, s), (\mathbf{T}, t))$ . As in the case of the  $G_0$  games, we first check if  $s$  and  $t$  differ on some

atomic sentence. If they do, then  $I$  immediately wins. Otherwise, the play continues. Now  $I$  can make two types of moves.

1. A  $\diamond_A$ -move:  $I$  has a choice of playing from  $\mathbf{S}$  or from  $\mathbf{T}$ , and also some agent  $A$ . If  $I$  chooses  $\mathbf{S}$ , then  $I$  continues by choosing some  $s'$  such that  $s \xrightarrow{A} s'$  in  $\mathbf{S}$ . Then  $II$  replies with some  $t' \in T$  such that  $t \xrightarrow{A} t'$ . Of course, if  $I$  had chosen in  $\mathbf{T}$ , then  $II$  would have chosen in  $S$ . Either way, points  $s'$  and  $t'$  are determined, and the two players then play  $G_n((\mathbf{S}, s'), (\mathbf{T}, t'))$ .
2. A  $\diamond_B^*$ -move:  $I$  plays by selecting  $\mathbf{S}$  (or  $\mathbf{T}$ , but we ignore this symmetric case below), and some set  $B$  of agents, and then  $I$  continues by playing some  $s'$  (say) reachable from  $s$  in the reflexive-transitive closure  $\xrightarrow{B^*}$  of  $\xrightarrow{B}$ ;  $II$  responds with a point  $t'$  in the other model,  $\mathbf{T}$ , which is similarly related to  $t$ .

We write  $(\mathbf{S}, s) \sim_n (\mathbf{T}, t)$  if  $II$  has a winning strategy in the game  $G_n((\mathbf{S}, s), (\mathbf{T}, t))$ . It is easy to check that by induction on  $m$  that if  $(\mathbf{S}, s) \sim_n (\mathbf{T}, t)$  and  $m < n$ , then  $(\mathbf{S}, s) \sim_m (\mathbf{T}, t)$ .

**Proposition 6.3.** *If  $(\mathbf{S}, s) \sim_n (\mathbf{T}, t)$ , then for all  $\varphi$  with  $|\varphi| \leq n$ ,  $(\mathbf{S}, s) \models \varphi$  if  $(\mathbf{T}, t) \models \varphi$ .*

The proof is standard, except perhaps for the easy extra step for  $\diamond^*$  moves.

We have two results that show that public announcements add expressive power to  $\mathcal{L}_1$ . The first is for one agent on arbitrary models, and the second is for two agents on equivalence relations.

In the first result, let  $\mathcal{A}$  be a singleton  $\{A\}$ . We drop the  $A$  from the notation.

**Theorem 6.4.** *The  $\mathcal{L}_1(\Sigma_{pub})$  sentence  $\langle Pub \ p \rangle \diamond^* q$  is not expressible in  $\mathcal{L}_1$ , even by a set of sentences.*

**Proof.** Fix a number  $n$ . We first show that  $\langle Pub \ p \rangle \diamond^* q$  is not expressible by any single sentence of  $\mathcal{L}_1$  of rank  $n$ . Let  $A_n$  be the cycle

$$a_0 \rightarrow a_1 \rightarrow \cdots \rightarrow a_{n+2} \rightarrow a_{n+3} \rightarrow \cdots a_{2n+4} = a_0.$$

We set  $p$  to true everywhere except  $a_{n+2}$  and  $q$  true only at  $a_0$ .

Announcing  $p$  means that we delete  $a_{n+2}$ . So,  $A_n(Pub \ p)$  splits into two disjoint pieces. This means that in  $A_n(Pub \ p)$ ,  $a_1$  does not satisfy  $\diamond^* q$ . But  $a_{n+3}$  does satisfy it.

We show that  $a_1$  and  $a_{n+3}$  agree on all sentences of  $\mathcal{L}_1$  of rank  $\leq n$ . For this, we show that  $II$  has a winning strategy in the  $n$ -round game on between  $(A_n, a_{n+1})$  and  $(A_n, a_{n+3})$ . Then we appeal to Proposition 6.3.  $II$ 's strategy is as follows: if  $I$  ever makes a  $\diamond^*$  move,  $II$  should make a move on the other side to the exact same point. (Recall that  $A_n$  is a cycle.) Thereafter,  $II$  should mimic  $I$ 's moves exactly. Since the play will end with the same point in the two structures,  $II$  wins. But if  $I$  never makes a  $\diamond^*$  move, the play will consist of  $n$   $\diamond$ -moves.  $II$  should simply make the same moves in the appropriate structures. Since  $a_{n+2}$  is  $n+1$  steps from  $a_1$ , and  $a_0 = a_{2n+4}$  is  $n+1$  steps from  $a_{n+3}$ ,  $II$  will win the play in this case.

So at this point we conclude that for all  $n$ ,  $\langle Pub \ p \rangle \diamond^* q$  is not expressible by any single sentence of  $\mathcal{L}_1$  of rank  $n$ . We conclude by extending this to show that  $\langle Pub \ p \rangle \diamond^* q$  is not expressible by any set of sentences of  $\mathcal{L}_1$ . Suppose towards a contradiction that  $\langle Pub \ p \rangle \diamond^* q$  were equivalent to the set  $T \subseteq \mathcal{L}_1$ . Consider the following models  $\mathcal{A}$  and  $\mathcal{B}$ :  $\mathcal{A} = \oplus_{n \geq 0} (A_n, a_1^n)$ ; i.e., take the disjoint union of the models  $\oplus A_n$ , and then identify all points  $a_1^n$ . We also rename the common  $a_1^n$  point to be  $a$ , and we take this as the distinguished point. So we consider  $(\mathcal{A}, a)$ . Similarly, let  $\mathcal{B} = \oplus_{n \geq 0} (A_n, a_{n+3}^n)$ . For clarity we will rename each point  $a_j^m$  to be  $b_j^m$ . And we write  $b$  for the distinguished point of  $\mathcal{B}$ . The construction insures that  $(\mathcal{A}, a) \models \neg \langle Pub \ p \rangle \diamond^* q$  and  $(\mathcal{B}, b) \models \langle Pub \ p \rangle \diamond^* q$ .

By definition of  $T$ ,  $(\mathcal{B}, b) \models \varphi$  for all  $\varphi \in T$ . Let  $\varphi \in T$  be such that  $(\mathcal{A}, a) \models \neg \varphi$ . Let  $m = |\varphi|$ . Let

$$\mathcal{C} = \oplus (\{(A_n, a_1^n) : n \neq m\} \cup \{(A_m, a_{m+3}^m)\}).$$



Notice that we switched exactly one of the identified points. Just as before, we will rename the points of  $\mathcal{C}$  and call the overall distinguished point  $c$ . It is easy to check that  $(\mathcal{C}, c) \models \langle \text{Pub } p \rangle \Diamond^* q$ . And since this sentence is equivalent to  $T$ , we also have  $(\mathcal{C}, c) \models \varphi$ . But we now show that  $(\mathcal{A}, a)$  and  $(\mathcal{C}, c)$  agree on all sentences of rank  $\leq m$ . This implies that  $(\mathcal{A}, a) \models \varphi$ , giving the needed contradiction.

Here is a winning strategy for  $II$  in the  $m$ -round game between  $(\mathcal{A}, a)$  and  $(\mathcal{C}, c)$ . If  $I$  opens with anything besides  $c_{m+3}^m$  or  $a_1^m$ ,  $II$  should play the corresponding point on the other side and, thereafter, play in the obvious way. If  $I$  opens with  $c_{m+3}^m$ ,  $II$  should play with  $a_1^m$  and, thereafter, play with basically the same strategy as in the first part of this theorem; similarly, if  $I$  opens with  $a_1^m$ ,  $II$  should reply  $c_{m+3}^m$  and, thereafter, play via the strategy in the first part of this proof, where we dealt with single sentences.  $\square$

### 6.2.2. Partition Models

Often in epistemic logic one is concerned with models in which every accessibility relation  $\xrightarrow{A}$  is an equivalence relation. We can obtain a version of Theorem 6.4 which shows that even on this smaller class of models, public announcements add expressive power to  $\mathcal{L}_1$ . However, we must use two agents: with one agent and an equivalence relation  $\Box^* \varphi$  is equivalent to  $\Box \varphi$ . Thus, every sentence in  $\mathcal{L}_1(\Sigma_{\text{pub}})$  on one-agent equivalence relations is equivalent to a purely modal sentence.

**Theorem 6.5.** *The  $\mathcal{L}_1(\Sigma_{\text{pub}})$  sentence  $\langle \text{Pub } p \rangle \Diamond_{A,B}^* q$  is not expressible by any set of sentences of  $\mathcal{L}_1$ , even on the class of models in which  $\xrightarrow{A}$  and  $\xrightarrow{B}$  are equivalence relations.*

**Proof.** We fix a number  $N$  and first show that  $\langle \text{Pub } p \rangle \Diamond^* q$  is not expressible by any single sentence of  $\mathcal{L}_1$  of rank  $N$ . Let  $n$  be the smallest even number strictly larger than  $N$ . Let  $C_n$  be as defined in Section 3.2 above. Note that  $C_n$  does have the property that  $\xrightarrow{A}$  and  $\xrightarrow{B}$  are equivalence relations. In addition,  $(\xrightarrow{A})^*$  is the same relation as  $\xrightarrow{A}$ . So,  $\Box_A^* \varphi$  is equivalent to  $\Box_A \varphi$  (and similarly for  $B$ ). Finally,  $(\xrightarrow{A} \cup \xrightarrow{B})^*$  is the universal relation. So if any point whatsoever satisfies a sentence  $\Box_{A,B}^* \varphi$ , then all points satisfy it.

The analysis of Section 3.2 shows that the points  $a_{n+1}$  and  $a_{3n+1}$  differ on our sentence  $\langle \text{Pub } p \rangle \Diamond_{A,B}^* q$ .

We continue by showing that  $II$  has a winning strategy in the  $n$ -round game on  $C_n$  from  $a_{n+1}$  and  $a_{3n+1}$ . If  $I$  ever makes a  $\Diamond^*$  move, then  $II$  should move to the same point and, thereafter, mimic  $I$  perfectly. Thus, we may assume that  $I$  never makes any  $\Diamond^*$  moves. In this case,  $I$  will never move either point to  $a_1$ ,  $a_{2n+1}$ , or  $a_{4n+1}$ . In other words, all points in the play will satisfy  $p \wedge \neg q$ . So, as long as  $I$  plays  $\Diamond$ -moves,  $II$  can follow arbitrarily. Once again, since the game goes for  $n$  rounds, this will be a winning strategy.

At this point we know that  $\langle \text{Pub } p \rangle \Diamond^* q$  is not expressible by any single sentence of  $\mathcal{L}_1$  of rank  $n > N$ . We use the same idea as in Theorem 6.4 to show that  $\langle \text{Pub } p \rangle \Diamond^* q$  is not expressible by any set of sentences of  $\mathcal{L}_1$ . In fact, virtually the same proof goes through.  $\square$

### 6.3. $\mathcal{L}_1(\Sigma_{\text{pri}})$ is More Expressive than $\mathcal{L}_1(\Sigma_{\text{pub}})$

It is natural to assume that *privacy* adds expressive power to logics of communication. The result of this section is the first result we know that establishes this. In it, we assume that our set of agents is the doubleton  $\{A, B\}$ .

**Theorem 6.6.** *The sentence  $\chi$  of  $\mathcal{L}_1(\Sigma_{\text{pri}})$  from (8) and repeated below*

$$\chi \equiv \langle \text{Pri}^A p, \text{true} \rangle \Diamond_A^* \Diamond_B \neg p$$

*cannot be expressed by any sentence of  $\mathcal{L}_1(\Sigma_{\text{pub}})$ , and  $\Diamond_A \chi$  cannot be expressed by any set of sentences of  $\mathcal{L}_1(\Sigma_{\text{pub}})$ .*

We define a function  $|\varphi|$  on  $\mathcal{L}(\Sigma_{\text{pub}})$ :  $|p| = 0$  for  $p$  atomic,  $|\neg\varphi| = |\varphi|$ ,  $|\varphi \wedge \psi| = \max(|\varphi|, |\psi|)$ ,  $|\diamond_A\varphi| = 1 + |\varphi|$ ,  $|\diamond_B\varphi| = 1$ ,  $|\diamond_B^*\varphi| = |\varphi|$  for all  $\mathcal{B} \subseteq \mathcal{A}$ , and  $|\llbracket \text{Pub } \varphi \rrbracket \psi| = \max(|\varphi|, |\psi|)$ .

**Lemma 6.7.** *Let  $\varphi \in \mathcal{L}(\Sigma_{\text{pub}})$ , and let  $1 \leq k \leq n$ . If  $n - k \geq |\varphi|$ , then*

$$(\mathbf{S}_n, a) \models \varphi \quad \text{iff} \quad (\mathbf{S}_n, c_k) \models \varphi \quad \text{iff} \quad (\mathbf{T}_n, a) \models \varphi.$$

**Proof.** By induction on  $\varphi$ . The base step for atomic  $p$  and the induction steps for the Boolean connectives are all trivial.

Here is the induction step for  $\diamond_A\varphi$ . Fix  $k$  with  $n - k \geq |\diamond_A\varphi| = 1 + |\varphi|$ . Thus,  $n - k > n - (k + 1) \geq |\varphi|$ . We prove the needed equivalences in a cycle.

Assume first that  $(\mathbf{S}_n, a) \models \diamond_A\varphi$ . The only arrows from  $a$  in  $\mathbf{S}_i$  are to  $a \xrightarrow{A} a$  and  $a \xrightarrow{A} b$ . If  $(\mathbf{S}_n, a) \models \varphi$ , then the reflexive arrow on  $c_k$  and the induction hypothesis show that  $(\mathbf{S}_n, c_k) \models \diamond_A\varphi$ . If  $(\mathbf{S}_n, b) \models \varphi$ , then clearly  $(\mathbf{S}_n, c_k) \models \diamond_A\varphi$  via  $c_k \xrightarrow{A} b$ .

Second, assume that  $(\mathbf{S}_n, c_k) \models \diamond_A\varphi$ . We show that  $(\mathbf{T}_n, a) \models \diamond_A\varphi$ . Notice that one of the following holds:  $(\mathbf{S}_n, c_k) \models \varphi$ ,  $(\mathbf{S}_n, b) \models \varphi$ , or  $(\mathbf{S}_n, c_{k+1}) \models \varphi$ . The first case is handled easily by the induction hypothesis. In the second, we have by (9) that  $(\mathbf{T}_n, b) \models \varphi$ ; hence  $(\mathbf{T}_n, a) \models \diamond_A\varphi$ . We are left with the case  $(\mathbf{S}_n, c_{k+1}) \models \varphi$ . Since  $n - (k + 1) \geq |\varphi|$ , the induction hypothesis implies that  $(\mathbf{T}_n, a) \models \varphi$ . Since  $a \xrightarrow{A} a$ ,  $(\mathbf{T}_n, a) \models \diamond_A\varphi$ .

Finally, assume that  $(\mathbf{T}_n, a) \models \diamond_A\varphi$ . We verify that  $(\mathbf{S}_n, a) \models \diamond_A\varphi$ . In view of the fact that  $a \xrightarrow{A} c_1$  in  $\mathbf{T}$ , one of the following holds:  $(\mathbf{T}_n, a) \models \varphi$ ,  $(\mathbf{T}_n, b) \models \varphi$ , or  $(\mathbf{T}_n, c_1) \models \varphi$ . The only important case is the last. Since  $n - 1 \geq n - k \geq |\varphi|$ , our induction hypothesis implies that in this case,  $(\mathbf{S}_n, a) \models \diamond_A\varphi$ , as desired.

We continue with the induction step for  $\diamond_B\varphi$ . Recall that  $|\diamond_B\varphi| = 1$ . Since we assume  $n - k \geq 1$ , we have  $k < n$ . There are no  $B$ -arrows from  $c_k$  or from  $a$ . All three needed statements in our lemma are false, so all three are equivalent.

Next, we have the induction step for  $\diamond_B^*\varphi$ , where  $\mathcal{B}$  is either  $\{A\}$ ,  $\{B\}$ , or  $\{A, B\}$ . With two agents,  $\mathcal{B}$  can be  $\{A\}$ ,  $\{B\}$  or  $\{A, B\}$  here. Note that  $\xrightarrow{B}$  is already transitive, so all the work for  $\diamond_B^*\varphi$  has been done above already. Also,  $\xrightarrow{B}$  is a subrelation of  $\xrightarrow{A}$  in both  $\mathbf{S}_n$  and  $\mathbf{T}_n$ . Thus, we only need to work with  $\mathcal{B} = \{A\}$  in this induction step.

Assume first that  $(\mathbf{S}_n, a) \models \diamond_B^*\varphi$ . There are a number of cases. If  $(\mathbf{S}_n, a) \models \varphi$ , then by induction hypothesis,  $(\mathbf{S}_n, c_k) \models \varphi$ . So,  $(\mathbf{S}_n, c_k) \models \diamond_B^*\varphi$  as well. The other case is where  $a \xrightarrow{B^*} x$ ,  $x \models \varphi$ , and  $x$  is among the points  $b, c_1, \dots, c_n$ . Note in this case that  $c_k \xrightarrow{B^*} x$  as well: all non-null paths from  $a$  go through  $b$ , and  $c_k \xrightarrow{A} b$ . Thus, in this case, we have  $(\mathbf{S}_n, c_k) \models \diamond_B^*\varphi$ .

Next, assume that  $(\mathbf{S}_n, c_k) \models \diamond_B^*\varphi$ . Let  $x$  be such that  $c_k \xrightarrow{B^*} x$  and  $(\mathbf{S}_n, x) \models \varphi$ . Clearly,  $x \neq a$ . By (9),  $(\mathbf{T}_n, x) \models \varphi$ . Also,  $a \xrightarrow{A} b \xrightarrow{B^*} c_k \xrightarrow{B^*} x$ . So  $(\mathbf{T}_n, a) \models \diamond_B^*\varphi$ .

Suppose, finally, that  $(\mathbf{T}_n, a) \models \diamond_B^*\varphi$ . If  $(\mathbf{T}_n, a) \models \varphi$ , then by induction hypothesis we have  $(\mathbf{S}_n, a) \models \varphi$ . So, in this case,  $(\mathbf{S}_n, a) \models \diamond_B^*\varphi$  as well. The other case is where  $(\mathbf{T}_n, x) \models \varphi$  for some  $x \neq a$ . By (9) again,  $(\mathbf{S}_n, x) \models \varphi$ . And in  $\mathbf{S}_n$ ,  $a \xrightarrow{B^*} x$ . Thus  $(\mathbf{S}_n, a) \models \diamond_B^*\varphi$ , as desired.

The most interesting step is the induction step for  $\llbracket \text{Pub } \varphi \rrbracket \psi$ . Fix  $k$  with  $n - k > |\llbracket \text{Pub } \varphi \rrbracket \psi| = \max(|\varphi|, |\psi|)$ . Thus,  $n - k \geq |\varphi|, |\psi|$ , and our induction hypothesis applies to both  $\varphi$  and  $\psi$ . In particular, the following are equivalent:

$$(\mathbf{S}_n, a) \models \varphi \quad (\mathbf{S}_n, c_k) \models \varphi \quad (\mathbf{T}_n, a) \models \varphi \tag{20}$$

Let us save on some notation by defining

$$\mathbf{S}_n^* = \mathbf{S} \otimes (\Sigma_{\text{pub}}, \text{Pub}, \varphi)$$

and, similarly, for  $\mathbf{T}_n^*$ . Moreover

$$\llbracket [Pub \ \varphi] \psi \rrbracket_{\mathbf{S}_n} = \{x \in S_n : \text{if } x \in \llbracket \varphi \rrbracket_{\mathbf{S}_n}, \text{ then } (x, Pub) \in \llbracket \psi \rrbracket_{\mathbf{S}_n^*}\} \quad (21)$$

A similar equation holds for  $\mathbf{T}_n$ .

Case I:  $(\mathbf{S}_n, a) \not\models \varphi$ . Since  $n - k \geq |\varphi|$ , we have by (20) that  $(\mathbf{S}_n, c_k) \not\models \varphi$ , and  $(\mathbf{T}_n, a) \not\models \varphi$ . So, automatically, all three of the relevant model-world pairs satisfy our sentence  $[Pub \ \varphi] \psi$ .

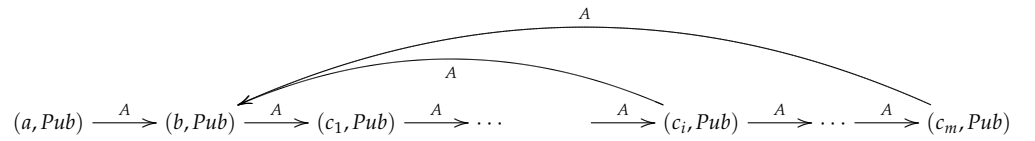
Case II:  $(\mathbf{S}_n, a) \models \varphi$  and  $(\mathbf{S}_n, b) \not\models \varphi$ . Then by induction hypothesis,  $(\mathbf{S}_n, c_k) \models \varphi$ , and  $(\mathbf{T}_n, a) \models \varphi$ , and by (9),  $(\mathbf{T}_n, b) \not\models \varphi$ . In  $\mathbf{S}_n^*$  and  $\mathbf{T}_n^*$ , each point satisfies  $p$  and has no  $\xrightarrow{B}$ -successors, and every point has an  $\xrightarrow{A}$ -successor (itself). Thus, the following are equivalent:

$$(\mathbf{S}_n^*, (a, Pub)) \models \psi \quad (\mathbf{S}_n^*, (c_k, Pub)) \models \psi \quad (\mathbf{T}_n^*, (a, Pub)) \models \psi \quad (22)$$

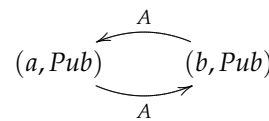
In more detail, the three model-world pairs above are all bisimilar to a one-point model which is a point satisfying  $p$  with a loop for  $\xrightarrow{A}$ . Hence, we have (22). This implies that the following are indeed equivalent:

$$(\mathbf{S}_n, a) \models [Pub \ \varphi] \psi \quad (\mathbf{S}_n, c_k) \models [Pub \ \varphi] \psi \quad (\mathbf{T}_n, a) \models [Pub \ \varphi] \psi. \quad (23)$$

Case III:  $(\mathbf{S}_n, a) \models \varphi$ ,  $(\mathbf{S}_n, b) \models \varphi$ , and for some  $\ell$  such that  $1 \leq \ell \leq n$ ,  $(\mathbf{S}_n, c_\ell) \not\models \varphi$ . We take the smallest such  $\ell$ . Since  $1 \leq k$ ,  $n - 1 \geq n - k \geq |\varphi|$ . So  $(\mathbf{S}_n, c_1) \models \varphi$  by induction hypothesis. Thus,  $\ell > 1$ . Let  $m = \ell - 1$ . Then  $m \geq 1$ . The point  $(c_m, Pri^A)$  in the updated model  $\mathbf{S}_i^*$  has  $A$  arrows only to itself and to  $(b, Pub)$ . Indeed, the model is



What we have drawn is the part of the model accessible from the points shown. We also have not drawn the reflexive  $A$  arrows. Notice that there are no  $B$  arrows whatsoever. As before,  $p$  is true at all points except  $(b, Pub)$ . This model is bisimilar to



Again, we have omitted the  $A$  loops on both points. The same happens with  $\hat{\mathbf{T}}_n$ ; the edge  $a \xrightarrow{A} c_1$  in  $\mathbf{T}_n$  does not change things. Note that the bisimulations relate  $(c_k, Pub)$  to  $(a, Pub)$ . The bisimulation shows the same equivalences (22) which we saw above, and this leads to the desired equivalences in (23).

Case IV:  $(\mathbf{S}_n, a) \models \varphi$ ,  $(\mathbf{S}_n, b) \models \varphi$ , and for all  $\ell$  such that  $1 \leq \ell \leq n$ ,  $(\mathbf{S}_n, c_\ell) \models \varphi$ . That is,  $\varphi$  holds at all worlds of  $\mathbf{S}_n$ . Recall that  $n - k \geq |\varphi|$ . So our induction hypothesis tells us that the following are equivalent

$$(\mathbf{S}_n, a) \models \psi \quad (\mathbf{S}_n, c_k) \models \psi \quad (\mathbf{T}_n, a) \models \psi \quad (24)$$

In this case, the models  $\mathbf{S}_n$  and  $\mathbf{S}_n^*$  are isomorphic, via  $x \mapsto (Pub, x)$ . That is, we are making a public announcement of a sentence  $\varphi$  that holds at all worlds of our model  $\mathbf{S}_n$ . But in this case we claim that  $\varphi$  is also true at all worlds  $x$  of  $\mathbf{T}_n$ . This follows from (9) for  $x \neq a$ , and for  $x = a$  it follows from (20) and the first assumption in this case. So  $\mathbf{T}_n$  and  $\mathbf{T}_n^*$  are isomorphic, again via  $x \mapsto (Pub, x)$ . Thus, we have the equivalences in (22). These, together with those in (20), imply those in (23), once again.

This completes the induction, and, hence, the proof of our lemma.  $\square$

It follows from what we have seen that our sentence  $\chi \equiv \langle \text{Pri}^A p, \text{true} \rangle \diamond_A^* \diamond_B \neg p$  is not equivalent to any single sentence of  $\mathcal{L}_1(\Sigma_{\text{pub}})$ . We would like to see that some sentence of  $\mathcal{L}_1(\Sigma_{\text{pri}})$  is not equivalent to any set of sentences of  $\mathcal{L}_1(\Sigma_{\text{pub}})$ . For this, we add a  $\diamond_A$  and consider  $\diamond_A \langle \text{Pri}^A p, \text{true} \rangle \diamond_A^* \diamond_B \neg p$ . We begin with a general model-theoretic construction which will be used in the remainder of this section.

**Definition 6.8.** Let  $I$  be any set; let  $A \in \mathcal{A}$ ; and for each  $i \in I$ , let  $(H_i, h_i)$  be a model-world pair. Let  $h$  be a new world. Take disjoint copies of the  $H_i$ 's and add an  $A$  arrow from  $h$  to each  $h_i$ . All other arrows are within the  $H_i$ 's and stay the same as in  $H_i$ . No atomic sentences are true at  $h$ . Atomic sentences true in the worlds belonging to the copy of  $H_i$  in  $\oplus_{i \in I}(H_i, h_i)$  are precisely those true at the corresponding worlds of  $H_i$ .

**Lemma 6.9.** Let  $\varphi$  be a sentence in  $\mathcal{L}_1(\Sigma_{\text{pub}})$ . Let  $I$  be any set, and for  $i \in I$ , let  $(H_i, h_i)$  and  $(K_i, k_i)$  be model-world pairs. Assume that for all  $i \in I$ ,  $(H_i, h_i)$  and  $(K_i, k_i)$  agree on all  $\psi \in \mathcal{L}_1(\Sigma_{\text{pub}})$  with  $|\psi| \leq |\varphi|$ . Then,  $(\oplus_{i \in I}(H_i, h_i), h)$  and  $(\oplus_{i \in I}(K_i, k_i), k)$  agree on  $\varphi$ .

**Proof.** By induction on  $\varphi$ . It is clear for atomic sentences. The induction steps for Boolean connectives are trivial. We omit the easy arguments for the induction steps for  $\diamond$  and  $\diamond^*$  with various subscripts. It remains to consider the case when  $\varphi = [\text{Pub } \varphi_1] \varphi_2$ . (The cases where we announce to  $\{A\}$  or to  $\{B\}$  are similar.) Fix  $I$ , and  $(H_i, h_i)$  and  $(K_i, k_i)$  for  $i \in I$ , and assume that these agree on sentences  $\psi$  with  $|\psi| \leq |[\text{Pub } \varphi_1] \varphi_2| = \max(|\varphi_1|, |\varphi_2|)$ . In particular, for each  $i \in I$ ,  $(H_i, h_i) \models \varphi_1$  if and only if  $(K_i, k_i) \models \varphi_1$ . Let

$$J = \{i \in I : (H_i, h_i) \models \varphi_1\} = \{i \in I : (K_i, k_i) \models \varphi_1\}.$$

For  $i \in J$ , let  $(H_i, h'_i)$  and  $(K'_i, k'_i)$  be obtained by “updating  $(H_i, h_i)$  and  $(K_i, k_i)$  by  $[\text{Pub } \varphi_1]$ ”. That is,

$$H'_i = H_i \otimes (\Sigma_{\text{pub}}, \text{Pub}, \varphi_1)$$

and similarly for  $K'_i$ . By our semantics of public announcements for all  $i \in J$ ,  $(H_i, h'_i)$  and  $(K'_i, k'_i)$  agree on  $\varphi_2$ . Therefore, by our inductive hypothesis

$$(\oplus_{i \in J} H'_i, h) \models \varphi_2 \quad \text{iff} \quad (\oplus_{i \in J} K'_i, k) \models \varphi_2.$$

However,

$$(\oplus_{i \in I} H_i, h) \models \varphi \quad \text{iff} \quad (\oplus_{i \in I} H_i, h) \models \varphi_1 \text{ implies } (\oplus_{i \in J} H'_i, h') \models \varphi_2,$$

and

$$(\oplus_{i \in I} K_i, k) \models \varphi \quad \text{iff} \quad (\oplus_{i \in I} K_i, k) \models \varphi_1 \text{ implies } (\oplus_{i \in J} K'_i, k') \models \varphi_2,$$

and we are done.  $\square$

At long last, we complete the proof of Theorem 6.6.

Recall that  $\chi$  is  $\langle \text{Pri}^A p, \text{true} \rangle \diamond_A^* \diamond_B \neg p$ . Assume towards a contradiction that the sentence at the end of our theorem,  $\diamond_A \chi$ , was equivalent to a set  $\Phi$  of sentences in  $\mathcal{L}_1(\Sigma_{\text{pub}})$ . Let  $N$  be the set of natural numbers  $\geq 1$ . For each  $i \in N$ , consider  $\mathbf{S}_i$  and  $\mathbf{T}_i$  from Definition 3.1. Let  $s_i = a$ . In this notation,  $(\mathbf{S}_i, s_i) \not\models \chi$ . Also, writing  $t_i$  for  $a$ , we also have  $(\mathbf{T}_i, t_i) \models \chi$ . Moreover, for all  $i$ ,  $(\mathbf{S}_i, s_i)$  and  $(\mathbf{T}_i, t_i)$  agree on sentences of  $\mathcal{L}_1(\Sigma_{\text{pub}})$  of rank  $\leq i - 1$ . (We are using Lemma 6.7 with  $k = 1$ .)

We consider  $(\oplus_{i \in N} \mathbf{S}_i, s)$  from above, with the new point being called  $s$ . Notice that  $(\oplus_{i \in N} \mathbf{S}_i, s) \not\models \diamond_A \chi$ . So, we have some  $\varphi^* \in \Phi$  such that  $(\oplus_{i \in N} \mathbf{S}_i, s) \not\models \varphi^*$ . Let  $n^* = |\varphi^*| + 1$ . Let  $(\mathbf{S}'_i, s'_i)$  be  $(\mathbf{S}_i, s_i)$  for  $i \neq n^*$ , and let  $(\mathbf{S}_{n^*}, s'_{n^*})$  be  $(\mathbf{T}_{n^*}, t_{n^*})$ . Then, for all  $i$ ,  $(\mathbf{S}_i, s_i)$  and  $(\mathbf{S}'_i, s'_i)$  agree on all sentences  $\psi$  of  $\mathcal{L}_1(\Sigma_{\text{pub}})$  with  $|\psi| \leq n^* - 1 = |\varphi^*|$ . By Lemma 6.9,

$(\oplus_{i \in N} S_i, s)$  and  $(\oplus_{i \in N} S'_i, s')$  agree on  $\varphi^*$ . (In this,  $s'$  is the new point in  $(\oplus_{i \in N} S'_i, s')$ .) This tells us that  $(\oplus_{i \in N} S'_i, s') \not\models \varphi^*$ . A fortiori,  $(\oplus_{i \in N} S'_i, s') \not\models \Diamond_A \chi$ . Since  $s' \xrightarrow{A} s_{n^*}$  in this model, we see that  $(\mathbf{T}_{n^*}, t_{n^*}) \not\models \chi$ . But this contradicts a fact which we recalled above.

#### 6.4. $\mathcal{L}(\Sigma_{pub})$ Lacks the Finite-Model Property

We conclude with a result on *iterating* epistemic actions. We gave a semantics of sentences of the form  $[\alpha^*]\varphi$  earlier in the paper. The ability to iterate actions is useful in algorithms and even in informal presentations of scenarios. As it happens, the iteration operation on actions makes our systems quite expressive, so much so that the finite-model property fails for  $\mathcal{L}_1(\Sigma)$  as soon as  $\Sigma$  contains public announcements.

**Proposition 6.10.**  $[\text{Pub } \Diamond \text{true}]^* \Diamond \Box \text{false}$  is satisfiable, but not in any finite model.

**Proof.** A state  $s$  in a model  $\mathbf{S}$  is called an *end state* if  $s$  has no successors. For each model  $\mathbf{S}$ , let  $\mathbf{S}'$  be the same model, except with the end states removed.  $\mathbf{S}'$  is isomorphic to what we have written earlier as  $\mathbf{S} \otimes (\text{Pub } \Diamond \text{true})$ , that is, the result of publically announcing that some world is possible. So we see that our sentence  $[\text{Pub } \Diamond \text{true}]^* \Diamond \Box \text{false}$  holds of  $s$  just in case the following holds for all  $n$ :

1.  $s \in \mathbf{S}^{(n)}$ , the  $n$ -fold application of the derivative operation to  $\mathbf{S}$ .
2.  $s$  has some child which is an end node (and, hence,  $t$  would not belong to  $\mathbf{S}^{(n+1)}$ ).

It is clear that any model of  $[\text{Pub } \Diamond \text{true}]^* \Diamond \Box \text{false}$  must be infinite, since the sets  $\mathbf{S}^{(n+1)} \setminus \mathbf{S}^{(n)}$  are pairwise disjoint and non-empty.

There are well-known models of  $[\text{Pub } \Diamond \text{true}]^* \Diamond \Box \text{false}$ . One would be the set of decreasing sequences of natural numbers, with  $s \rightarrow t$  if  $t$  is a one-point extension of  $s$ . The end nodes are the sequences that end in 0. For each  $n$ ,  $\mathbf{S}^{(n)}$  is the submodel consisting of the sequences which end in  $n$ .  $\square$

This has several dramatic consequences for this work. First, it means that the logics cannot be translated into the modal mu-calculus, since that logic is known to have the finite-model property. More importantly, we have the following extension of this result:

**Theorem 6.11** (Miller and Moss [14]). *Concerning  $\mathcal{L}(\Sigma_{pub})$ :*

1.  $\{\varphi : \varphi \text{ is satisfiable}\}$  is  $\Sigma_1^1$ -complete.
2.  $\{\varphi : \varphi \text{ is satisfiable on a finite (tree) model}\}$  is  $\Sigma_1^0$ -complete.

Indeed, these even hold when  $\mathcal{L}(\Sigma_{pub})$  is replaced by small fragments, such as the fragment built from  $[\text{Pub } \Diamond \text{true}]$ ,  $\Diamond^*$ ,  $\Diamond$ , and Boolean connectives (yet without atomic sentences); or, the fragment built from arbitrary iterated relativizations and modal logic (without  $\Diamond^*$ ). These negative results go via reduction from domino problems.

The upshot is that logics which allow for arbitrary finite iterations of epistemic actions are not going to be axiomatizable.

We feel that our results on expressive power are just a sample of what could be done in this area. We did not investigate the next natural questions. These questions have to do with notions of “suspicion” and “common knowledge of suspicion”. It would take us too far afield to mention them here, but one could see, for example, Refs. [6,21] for precise definitions. Do announcements with suspicious outsiders extend the expressive power of modal logic with all secure private announcements and common knowledge operators? And, then, do announcements with common knowledge of suspicion add further expressive power?

#### 6.5. Conclusions

This paper advanced the subject of dynamic epistemic logic (DEL) by introducing logical systems for the logical languages which it studied and also by obtaining a number

of results about those logical systems. Prior to this paper, there had been logical systems: see, for example, [1–4]. These were mainly interested in what has become known as *public-announcement logic* (PAL), and while these papers point out forcefully that common knowledge was an important topic for the area, they did not obtain completeness theorems for PAL with common knowledge. Our advance on these was (a) to add to PAL the ability to express a much wider array of epistemic actions; (b) to use action signatures in the syntax, and updates, program models, and the update product in the semantics; and (c) to obtain the completeness of the all the logical systems proposed, including common-knowledge operators. This paper did not go into much detail on *why* one would want to add epistemic operators beyond public announcement; this was done in [6,21] and many other papers. It did formulate logical systems as desired; a look at the first three sections of this paper shows that this formulation takes a lot of work. It also obtained the completeness results for these logical systems, and it did so by calling on results on the canonical action model and also results in term rewriting theory. Those particular results have not been superseded: while others (for example, [7]) have obtained completeness theorems, the logics involved were different, and the proofs also were different. As a final contribution, this paper was perhaps the first to obtain expressive-power results. In addition to being mathematically interesting, they suggest that the large amount of work needed to formulate the syntax and semantics of the logics in this paper is worthwhile. For example, since the logic of private announcements goes beyond what could be expressed in PAL, one really needs a bigger logical system if one wants a general account of private announcements.

An early version of this paper, and several other papers, played an important part in advancing the DEL area. At the present time (2023), there are over 1000 references to [5], the first version of this paper. It is beyond anyone’s ability to look at all of them, regardless of how (or even whether) they used the ideas here or modified them. We know that many of the important contributions to the area of dynamic epistemic logic (broadly conceived) do go beyond what we did here. They modify the overall framework, and they aim at applications which go beyond what one could treat using what we did. Still, we maintain that the published (and unpublished) preliminary versions of this paper made an important contribution to the area. We also think that the current, more polished version is a smaller but still significant further contribution, adding both more technical results and a better, clearer formulation of the main concepts and setting of contemporary dynamic epistemic logic.

**Author Contributions:** Writing—original draft, A.B., L.S.M. and S.S.; Writing—review & editing, A.B., L.S.M. and S.S.; Supervision, A.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by grant #586136 from the Simons Foundation.

**Acknowledgments:** We are grateful to three anonymous reviewers for comments and criticism. Certainly, their work improved this paper. All remaining shortcomings are our own. We thank Valentin Goranko for encouraging us to publish our paper in the first place and for his great patience with us.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Notes

- <sup>1</sup> If it is, then this would show that the completeness result here is definitely not a special case of that in [9]. But again, the matter is open at this time (2023): one cannot derive our result from [9] since the systems are *prima-facie* different
- <sup>2</sup> As the authors of [9] admit: “We have found no practical use for these [complex combinations of agent accessibility relations] at present”. It is true that they go on to argue that, as a result of our more restrictive syntax, “it’s completeness theorem [from [5]] is correspondingly messy”. However, the additional complications of our proof are mainly due to our choice of a strictly “syntactic syntax”, and partly to the explicit rewriting work (taken for granted in [9]), and only in small part due to our restricted static base
- <sup>3</sup> We are writing relational composition in left-to-right order in this paper
- <sup>4</sup> Note that basic actions are just syntactic expressions, in contrast to the signature-based *program models* in Section 2.2, which came with a list  $\vec{\psi}$  of *epistemic propositions* (which we wrote using boldface letters  $\psi_1, \dots, \psi_n$ ). Also note that basic actions might not be “atomic” in the sense that the sentences  $\psi_j$  might themselves contain programs



- 5 These are Kripke frames similar to our signatures, with the difference that one requires that only finitely many action types are reachable from a given action type via any concatenations of arrows. The definition of the languages  $\mathcal{L}(\Sigma)$  is significantly more complex for locally finite signatures, so in this paper we restrict our signatures to finite ones only
- 6 Another possibility which we did not explore is to consider rewriting modulo the identity  $[\alpha; \beta]\varphi = [\alpha][\beta]\varphi$
- 7 We use the same symbol  $\llbracket \cdot \rrbracket$  for our interpretation as we used for the valuation in a state model in Section 2.1. Since our use of interpretations is confined to this section of the paper, we feel that the confusion due to overloading the symbol  $\llbracket \cdot \rrbracket$  should be minimal
- 8 It does not matter which sequence or which normal form is chosen

## References

1. Plaza, J. Logics of public communications. In Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems, Charlotte, NC, USA, 12–14 October 1989.
2. Gerbrandy, J. Dynamic epistemic logic. In *Logic, Language, and Information*; Moss, L.S., Ed.; CSLI Publications, Stanford University: Stanford, CA, USA, 1999; Volume 2.
3. Gerbrandy, J. Bisimulations on Planet Kripke. Ph.D. Thesis, University of Amsterdam, Amsterdam, The Netherlands, 1999.
4. Gerbrandy, J.; Groeneveld, W. Reasoning about information change. *J. Logic Lang. Inf.* **1997**, *6*, 147–169. [\[CrossRef\]](#)
5. Baltag, A.; Moss, L.S.; Solecki, S. The logic of common knowledge, public announcements, and private suspicions. In Proceedings of TARK-VII (Theoretical Aspects of the Rationality and Knowledge) Conference, Evanston, IL, USA, 22–24 July 1998.
6. Baltag, A.; Moss, L.S. Logics for epistemic programs. *Synthese* **2004**, *139*, 165–224. [\[CrossRef\]](#)
7. van Ditmarsch, H.; van der Hoek, W.; Kooi, B. Synthese Library. In *Dynamic Epistemic Logic*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 337.
8. Baltag, A.; Moss, L.S.; Solecki, S. The logic of common knowledge, public announcements, and private suspicions. In *Readings in Formal Epistemology*; Costa, H.A., Ed.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 773–812.
9. van Benthem, J.; van Eijck, J.; Kooi, B. Logics of Communication and Change. *Inf. Comput.* **2006**, *204*, 1620–1662. [\[CrossRef\]](#)
10. Alexandru, B.; Renne, B. Dynamic Epistemic Logic. In *The Stanford Encyclopedia of Philosophy (Winter 2016 Edition)*; Zalta, E.N., Ed.; Stanford University: Stanford, CA, USA, 2016. Available online: <https://plato.stanford.edu/archives/win2016/entries/dynamic-epistemic> (accessed on 2 March 2023).
11. Van Ditmarsch, H.; van der Hoek, W.; Kooi, B. Internet Encyclopedia of Philosophy. In *Dynamic Epistemic Logic*; Springer: Berlin/Heidelberg, Germany, 2016.
12. Moss, L.S.; Logic, D.E. *Handbook of Epistemic Logic*; van Ditmarsch, H., Halpern, J.Y., van der Hoek, W., Kooi, B., Eds.; College Publications: London, UK, 2015; pp. 261–312.
13. van Eijck, J.; Sadzik, T. Action emulation. *Synthese* **2012**, *185*, 131–151. [\[CrossRef\]](#)
14. Miller, J.S.; Moss, L.S. The undecidability of iterated modal relativization. *Stud. Log.* **2005**, *79*, 373–407. [\[CrossRef\]](#)
15. Kozen, D.; Parikh, R. An elementary proof of the completeness of PDL. *Theor. Comput. Sci.* **1981**, *14*, 113–118. [\[CrossRef\]](#)
16. Dershowitz, N. Termination of rewriting. *J. Symbolic Comput.* **1987**, *3*, 69–115. [\[CrossRef\]](#)
17. Fischer, M.J.; Ladner, R.E. Propositional modal logic of programs. *J. Comput. System Sci.* **1979**, *18*, 194–211. [\[CrossRef\]](#)
18. Fagin, R.; Halpern, J.Y.; Moses, Y.; Vardi, M.Y. *Reasoning About Knowledge*; MIT Press: Cambridge, MA, USA, 1996.
19. Dawar, A.; Grädel, E.; Kreutzer, S. Inflationary fixed points in modal logic. *Acm Trans. Comput. Log.* **2004**, *5*, 282–315. [\[CrossRef\]](#)
20. Van Eijck, J. *Reducing Dynamic Epistemic Logic to PDL by Program Transformation*; University of Amsterdam: Amsterdam, The Netherlands, 2004. Available online: <https://staff.fnwi.uva.nl/d.j.n.vaneijck2/papers/04/pdfs/delpdl.pdf> (accessed on 2 March 2023).
21. Baltag, A.; van Ditmarsch, H.; Lawrence, S. Moss, Epistemic Logic and Information Update. In *Handbook of the Philosophy of Information*; Adriaans, P., van Benthem, J., Eds.; Elsevier: Amsterdam, The Netherlands, 2008; pp. 369–463.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.