Chapter 1

CONCURRENT DYNAMIC EPISTEMIC LOGIC

Hans van Ditmarsch
Computer Science,
University of Otago, New Zealand
hans@cs.otago.ac.nz

Wiebe van der Hoek
Computer Science,
University of Liverpool, United Kingdom
wiebe@csc.liv.ac.uk

Barteld Kooi
Computing Science
University of Groningen, the Netherlands
barteld@cs.rug.nl

Abstract When giving an analysis of knowledge in multiagent systems, one needs a framework in which higher-order information and its dynamics can both be represented. A recent tradition starting in original work by Plaza treats all of knowledge, higher-order knowledge, and its dynamics on the same foot. Our work is in that tradition. It also fits in approaches that not only dynamize the epistemics, but also epistemize the dynamics: the actions that (groups of) agents perform are epistemic actions. Different agents may have different information about which action is taking place, including higher-order information. We demonstrate that such information changes require subtle descriptions. Our contribution is to provide a complete axiomatization for an action language of van Ditmarsch, where an action is interpreted as a relation between epistemic states (pointed models) and sets of epistemic states. The applicability of the framework is found in every context where multiagent strategic decision making is at stake, and already demonstrated in game-like scenarios such as Cluedo and card games.

Keywords: Dynamic epistemic logic, epistemic group actions, completeness

1. Introduction

Since Hintikka's [Hin62] epistemic logic, the logic of knowledge, has been a subject of research in philosophy [Hin86], computer science [FHMV95], artificial intelligence [MvdH95] and game theory [AB95]. The latter three application areas made it more and more apparent that in multiagent systems *higher-order information*, knowledge about other agents' knowledge, is crucial.

The famous paper [AGM85] by Alchourrón *et al.* put the *change of information*, or belief revision, as a topic on the philosophical and logical agenda: it was followed by a large stream of publications and much research in belief revision, fine-tuning the notion of epistemic entrenchement [MLH00], revising (finite) belief bases [BDPW02], differences between belief revision and belief updates [KM91], and the problem of iterated belief change [DP97]. However, in all these approaches the dynamics is studied on a level above the informational level, making it impossible to reason about change of agents' knowledge and ignorance within the framework, let alone about the change of other agents' information.

Our work takes these the observations on higher-order knowledge and change of information as a starting point: when giving an analysis of knowledge in multiagent systems, one needs a framework in which higher-order information and its dynamics can be represented.

Although the notion of a run in an interpreted system as described in [FHMV95] makes it in principle possible to reason about the dynamics of an agent's knowledge, the interpretation of a run is typically that of a standard program. Further, the pioneering work of Moore [MvdH95] also studies the relation between actions and knowledge: there the emphasis is on epistemic preconditions that are needed to perform certain actions in the world, such as knowing a key-combination in order to open a safe.

From the point of view of expressivity, one can say that the work on interpreted systems enables one to reason about the (change of) knowledge over time, and adding actions to the language, one can also reason about the change of knowledge brought about by performing certain plans. This enables one to express properties like *perfect recall* and *no learning.* Recently, based on work by Alur et al [AHK97], van der Hoek and Wooldridge [vdHW02] added a *social* or *coalitional aspect* to an epistemic framework, giving them the possibility to express that for instance a group can establish that some knowledge is eventually obtained, or that two agents can enforce that they exchange a secret, without a third agent getting to know this.

Our work fits in approaches that not only dynamize the epistemics, but also epistemize the dynamics: the actions that (groups of) agents perform are epistemic actions. Different agents may have different information about which action is taking place, including higher-order information. This rather recent tradition treats all of knowledge, higher-order knowledge, and its dynamics on the same foot. Following an original contribution by Plaza in 1989 [Pla89], a stream of publications appeared around the year 2000 [GG97, Ger99, LR99, Bal99, vD00, vB01, BMS02, vD02b, vD02a, tC02].

The following, possibly simplest example in the setting of multiagent systems (two agents, one atom) attempts to demonstrate that the notions of higher-order information and epistemic actions are indeed important and may be subtle.

> *Anne and Bert are in a bar, sitting at a table. A messenger comes in and delivers a letter that is addressed to Anne. The letter contains either an invitation for a night out in Amsterdam, or an obligation to give a lecture instead. Anne and Bert commonly know that these are the only alternatives.*

This situation can be modelled as follows: There is one atom p, describing 'the letter invites Anne for a night out in Amsterdam', so that $\neg p$ stands for her lecture obligation. There are two agents 1 (Anne) and 2 (Bert). Whatever happens in each of the following action scenarios, is publicly known (to Anne and Bert). Also, assume that in fact p is true.

ACTION SCENARIO 1.1 (tell) *Anne reads the letter aloud.*

ACTION SCENARIO 1.2 (read) *Bert is seeing that Anne reads the letter.*

ACTION SCENARIO 1.3 (mayread) *Bert orders a drink at the bar so that Anne may have read the letter.*

ACTION SCENARIO 1.4 (bothmayread) *Bert orders a drink at the bar while Anne goes to the bathroom. Both may have read the letter.*

After execution of the first scenario it is common knowledge that p: in the resulting epistemic state $C_{12}p$ (i.e. $C_{\{1,2\}}p$) holds. This is not the case in the second scenario, but still, some common knowledge is obtained there: $C_{12}(K_1 p \vee K_1 \neg p)$: it is commonly known that Anne knows the contents of the letter, irrespective of it being p or $\neg p$. Does this higher-order information change in Scenario 3? Yes, in this case Bert does not even know if Anne knows p or knows $\neg p$: $\neg K_2(K_1 p \vee K_1 \neg p)$. In Scenario 4 something similar is happening, that may best be described by saying that the agents concurrently learn that the other may have learnt p or $\neg p$. Note that in this case both agents may have learnt p, so that p is generally known: $E_{12}p$, but they are in that case unaware of each other's knowledge – $\neg C_{12}p$ –, and *that* is commonly known.

Van Ditmarsch has described such actions as *knowledge actions* (with corresponding dynamic modal operators) in a multiagent dynamic epistemic language [vD00, vD02b]. Knowledge actions are interpreted as a relation between epistemic states. The contribution of our paper is that it provides a complete axiomatization for the extension of this language with concurrency as found in [vD01, vD02a]. This builds on work on concurrency in dynamic logic (PDL) [Pel87, HKT00, Gol92] and is partially related to game theoretical semantics for (extensions of) dynamic logic [Par85, Pau00].

The applicability of the framework is found in every context where multiagent strategic decision making is at stake, and already demonstrated in game-like scenarios such as Cluedo and card games [vD00].

Section 1.2 introduces the language and its semantics. Section 1.3 defines the axioms and derivation rules, and prerequisites for their formulation, such as syntactic equivalence of actions, and shows the soundness of this proof system. Section 1.4 shows its completeness. Section 1.5 gives some applications of the language in specifying multiagent system dynamics, and is followed by the conclusions (Section 1.6).

2. Language and semantics

Structures

Given a finite set of *agents* N and a set of *atoms* P, a (Kripke) *model* $M = \langle W, R, V \rangle$ consists of a domain W of *worlds* or *factual states*, for each agent $n \in N$ a binary *accessibility relation* R_n on W, and a *valuation* $V : P \to \mathcal{P}(W)$, or in other words: for each atom $p \in P$, a subset V_p of W. In an *epistemic model* (commonly known as an $S5$ model) all accessibility relations are equivalence relations. We then write $\sim_n$ for the equivalence relation for agent n. If $w \sim_n w'$ we say that *w is the same as w'* for n, or that *w is equivalent to w'* for n. Write $\sim_B$ for $(\bigcup_{n \in B} \sim_n)^*$ (where R^* is the transitive closure of a binary relation R).

Given a epistemic model M and a world $w \in M$, (M, w) is called an *epistemic state*, w the *point* of that epistemic state, and M the model *underlying* that epistemic state.

For a given model M, $\mathcal{D}(M)$ returns its domain. Instead of $w \in \mathcal{D}(M)$ we also write $w \in M$. If $s = (M, w)$ and $v \in M$ we also write $v \in s$ (and $\mathcal{D}(s)$ for $\mathcal{D}(M)$). Write $\mathcal{S}5_N(P)$ for the class of epistemic models for agents N and atoms P, and $\mathcal{S}5_{\subseteq N}(P)$ for $\bigcup_{B \subseteq N} \mathcal{S}5_B(P)$. Write $\bullet\mathcal{S}5_N(P)$ for the class of epistemic states for agents N and atoms P

(i.e., 'pointed' – • – models). We drop the 'P' if it is clear from the context.

Given an epistemic model M or epistemic state s for a set of agents N, the operator gr returns that set: $gr(M) = gr(s) = N$. This is called the *group* of that epistemic model / state. The group of a set is the union of the groups of its members.

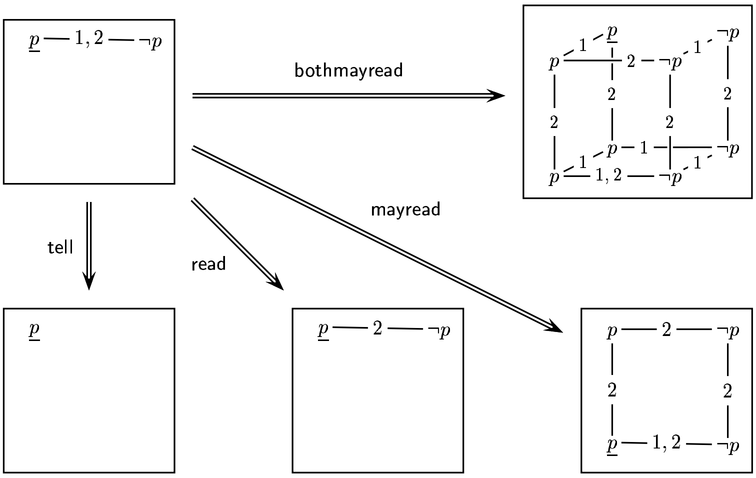

Figure 1.1. Epistemic states resulting from the execution of actions described in the four Action Scenarios. The top left figure represents (Arc, u). Points of epistemic states are underlined. Assume transitivity of access. For mayread and bothmayread only one of more executions is shown.

EXAMPLE 1.5 *The background setting for 'Lecture or Amsterdam' can be represented by an epistemic state. Arc is the model* $\langle \{u, v\}, \sim, V \rangle$ *such that both* $\sim_1$ *and* $\sim_2$ *are the universal relation on* $\{u, v\}$*, and* $V_p = \{u\}$*. The epistemic state* (Arc, u) *corresponds to* p *being actually the case. After Anne has read the letter, an epistemic state is reached that is like* (Arc, u) *but with* $\sim_1 = \{(u, u), (v, v)\}$ *instead. See Figure 1.1.*

Syntax

To a standard multiagent epistemic language with common knowledge for a set N of agents and a set P of atoms [MvdH95, FHMV95], we add dynamic modal operators for programs that are called knowledge actions or just actions. Actions may change the knowledge of the agents involved. The formulas $\mathcal{L}_N(P)$, the actions $\mathcal{L}_N^{\mathsf{act}}(P)$, and the group gr of an action are defined by simultaneous induction:

Definition 1.6 (Formulas and actions)
The formulas $\mathcal{L}_N(P)$ *are defined by*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_n\varphi \mid C_B\varphi \mid [\alpha]\psi$$

where $p \in P$, $n \in N$, $B \subseteq N$, $\alpha \in \mathcal{L}_N^{\mathsf{act}}(P)$, *and* $\psi \in \mathcal{L}_{gr(\alpha)}(P)$. *The* actions $\mathcal{L}_N^{\mathsf{act}}(P)$ *are defined by*

$$\alpha ::= ?\varphi \mid L_B\beta \mid (\alpha \ ! \ \alpha) \mid (\alpha \ ¡ \ \alpha) \mid (\alpha \ ; \ \beta') \mid (\alpha \cup \alpha) \mid (\alpha \cap \alpha)$$

where $\varphi \in \mathcal{L}_N(P)$, $B \subseteq N$, $\beta \in \mathcal{L}_B^{\mathsf{act}}(P)$, *and* $\beta' \in \mathcal{L}_{gr(\alpha)}^{\mathsf{act}}(P)$, *and where the* group $gr(\alpha)$ *of an action* $\alpha \in \mathcal{L}_N^{\mathsf{act}}(P)$ *is defined as:* $gr(?\varphi) := \emptyset$, $gr(L_B\alpha) := B$, *and* $gr(\alpha \bullet \alpha') := gr(\alpha) \cap gr(\alpha')$ *for* $\bullet = \ ! \ , \cap, \cup, \ ; \ .$

Other propositional connectives and modal operators are defined by standard abbreviations, in particular $E_B\varphi := \bigwedge_{n \in B} K_n\varphi$. Outermost parentheses of formulas and actions are deleted whenever convenient. As we may generally assume an arbitrary P, write $\mathcal{L}_N$ instead of $\mathcal{L}_N(P)$, and $\mathcal{L}_N^{\mathsf{act}}$ instead of $\mathcal{L}_N^{\mathsf{act}}(P)$. Instead of, e.g., $C_{\{a,b,c\}}$ we always write C_{abc}. For an arbitrary epistemic ('box'-like) operator K, write $\hat{K}$ for its dual ('diamond'). The dual of $[\alpha]$ is $\langle\alpha\rangle$.

The program constructor L_B is called *learning.* Action $?\varphi$ is a *test*, $(\alpha \ ; \ \alpha')$ is *sequential execution*, $(\alpha \cup \alpha')$ is *nondeterministic choice*, $(\alpha \ ! \ \alpha')$ is called *(left) local choice* and $(\alpha \ ¡ \ \alpha')$ is called *(right) local choice*, and $(\alpha \cap \alpha')$ is *concurrent execution.* The construct $L_B?\varphi$ is pronounced as 'B learn *that* φ'. Local choice $\alpha \ ! \ \alpha'$ may, somewhat inaccurately, be seen as 'from α and α', choose the first.' Local choice $\alpha \ ¡ \ \alpha'$ may be seen as 'from α and α', choose the second.' We will see that the interpretation of local choice '!' and '¡' depends on the context of learning that binds it: in $L_B(\alpha \ ! \ \alpha')$, everybody in B but not in learning operators occurring in α, α', is unaware of the choice for α. That choice is therefore 'local'. Typically, we show properties of local choice for '!' only.

The group *gr* was already used for the agents 'occurring' in epistemic states and models. It serves a similar function on actions, whence the overloading. The constructs $[\alpha]\psi$, $L_B\beta$, and $[\alpha \ ; \ \beta$, wherein *gr* is used (implicitly in $L_B\beta$), guarantee that in an epistemic state for agents N that is the result of action execution, formulas containing modal operators for agents *not* in N are not considered for interpretation.

Example 1.7 *The description in $\mathcal{L}_{12}^{\mathsf{act}}(\{p\})$ of the actions in the introduction are:*

$$\begin{array}{rl}
\mathsf{tell} & L_{12}?p \cup L_{12}?\neg p \\
\mathsf{read} & L_{12}(L_1?p \cup L_1?\neg p) \\
\mathsf{mayread} & L_{12}(L_1?p \cup L_1?\neg p\cup\ ?\top) \\
\mathsf{bothmayread} & L_{12}(\ (L_1?p \cap L_2?p) \cup (L_1?\neg p \cap L_2?\neg p) \\
 & \cup L_1?p \cup L_1?\neg p \cup L_2?p \cup L_2?\neg p\cup\ ?\top)
\end{array}$$

For example, the description of read *(Anne reads the letter) reads as follows: 'Anne and Bert learn that either Anne learns that she is invited for a night out in Amsterdam or that Anne learns that she has to give a lecture instead.' In the last two actions, instead of* $?\top$ *(for 'nothing happens') we may as well write* $?p\cup?\neg p$. *(Associativity of* $\cup$ *is for now assumed, and proved later.)*

A nondeterministic action can have more than one execution in a given epistemic state. The only way to get such an action is to use nondeterministic choice operators $\cup$ in its description. If we use ! operators instead, typically, only some but not all of the agents are aware of the choices made. Constructs $\cup$ and ! are related as follows:

Definition 1.8 (Type and instance of an action) *By replacing all occurrences of '!' and '¡' in an action* α *by '*$\cup$*', except when under the scope of* ?, *we get the* type $t(\alpha)$ *of that action. Slightly informally we can write:*

$$t(\alpha) := \alpha[!/\cup, ¡/\cup]$$

By replacing all occurrences of '$\cup$*' in an action* α *for either '!' or '¡', except when under the scope of* ?, *we get the set of* instances $T(\alpha)$ *of that action. Informally we can write:*

$$T(\alpha) := \{\alpha[\cup/!, ¡]\}$$

If $t(\alpha) = t(\beta)$ *we say that* α *and* β *are the same type of action. Further, if* α *and* β *are identical modulo swapping of occurrences of '!' for '¡' or vice versa, write*

$$\alpha =_T \beta$$

Obviously, $=_T$ *is an equivalence. We say that* α, β *are* comparable *actions.*

Instead of $\alpha\ !\ \alpha'$ we generally write $!\alpha\cup\alpha'$. This expresses more clearly that given choice between α and α', the agents involved in those actions choose α, whereas that choice remains invisible to the agents that learn about these alternatives but are not involved. Similarly, instead of α ¡ α' we generally write $\alpha\cup\ !\alpha'$.

Comparable actions $\alpha =_T \beta$ are 'on the same level of abstraction in the type hierarchy'. This means that they can be 'compared': it can be determined if they are (syntactically) the same for a given agent or not, a notion needed in the proof system to be introduced in Section 1.3.

Example 1.9 *The action* read *where Bert is seeing that Anne reads the letter is different from the instance of that action where* Anne is actually invited for a night out *and Bert is seeing that Anne reads the letter. The last is described as $L_{12}(L_1?p\ !\ L_1?\neg p)$: of the two alternatives $L_1?p$ and $L_1?\neg p$, the first is chosen, but agent 2 is unaware of that choice. A different way of writing that action is $L_{12}(!L_1?p \cup L_1?\neg p)$. The action* read *is its* type. *The other instance of action* read *is $L_{12}(L_1?p$ ¡ $L_1?\neg p)$ $(L_{12}(L_1?p\cup\ !L_1?\neg p))$. Actions $L_{12}(!L_1?p \cup L_1?\neg p)$ and $L_{12}(L_1?p\cup\ !L_1?\neg p)$ are* comparable *to each other.*

Somewhat similarly, the action bothmayread *has four different executions if p is true and another four if p is false: there are eight actions instances (state transformers) of that type.*

Semantics

The semantics of $\mathcal{L}_N(P)$ (on epistemic models) is defined as usual [MvdH95], plus an additional clause for the meaning of dynamic operators. The interpretation of a dynamic operator is a relation between an epistemic state and a set of epistemic states. The *composition* $(R \circ R')$ of two relations $R, R' : W \to \mathcal{P}(W)$ (such as $[\![\cdot]\!]$) is defined as follows: let $v \in W, V \subseteq W$, then: $(R \circ R')(v, V) :\Leftrightarrow \exists X : R(v, X)$ and $\forall x \in X : \exists V_x : R'(x, V_x)$ and $V = \bigcup_{x \in X} V_x$. Further, $R \sqcup R' := \{(v, V) \mid \exists V', V'' : R(v, V'), R'(v, V''),$ and $V = V' \cup V''\}$.

In the semantics, we need a notion of equivalence between sets of epistemic states. We lift equivalence of worlds in an epistemic state to equivalence of epistemic states and to equivalence of sets of epistemic states. Sets of epistemic states will occur as worlds in definition 1.11 of action interpretation, and equivalence of such worlds for an agent will be defined as equivalence of those sets.

Definition 1.10 (Equivalence of sets of epistemic states) *Let $M, M' \in \mathcal{S}5_N$, $v, w \in M$, and $w' \in M'$. Let $S, S' \subseteq \bullet\mathcal{S}5_{\subseteq N}$. Let*

$n \in N$. *Then:*

$$\begin{array}{lll}
(M,w) \sim_n (M,v) & \textit{iff} & w \sim_n v \\
(M,w) \sim_n (M',w') & \textit{iff} & \exists v \in M : (M,v) \underline{\leftrightarrow} (M',w') \ \& \\
 & & (M,w) \sim_n (M,v) \\
S \sim_n S' & \textit{iff} & [\ \forall s \in S : n \in gr(s) \Rightarrow \exists s' \in S' : s \sim_n s'\ ]\ \& \\
 & & [\ \forall s' \in S' : n \in gr(s') \Rightarrow \exists s \in S : s \sim_n s'\ ]
\end{array}$$

In the second clause of the definition, $\underline{\leftrightarrow}$ stands for 'is bisimilar to' [BdRV01]. Bisimilarity is a notion of sameness between epistemic states that implies equivalence of their logical descriptions (theories), though not vice versa. The implicit symmetric closure in the third clause of the definition is needed to keep $\sim_n$ an equivalence relation.

We now continue with the semantics. The interpretation of formulas and actions is defined simultaneously.

Definition 1.11 (Interpretation of formulas and actions) *Let $s = (M,w) \in \bullet\mathcal{S}5_N(P)$, where $M = \langle W, \sim, V\rangle$; let $\varphi \in \mathcal{L}_N(P)$, and let $\alpha \in \mathcal{L}_N^{\mathsf{act}}(P)$. The interpretation $\models$ of φ in (M,w) and the interpretation $[\![\cdot]\!]$ of α in (M,w) are both defined by inductive cases.*

$$\begin{array}{lll}
M,w \models p & :\textit{iff} & w \in V(p) \\
M,w \models \neg\varphi & :\textit{iff} & M,w \not\models \varphi \\
M,w \models \varphi \wedge \psi & :\textit{iff} & M,w \models \varphi \textit{ and } M,w \models \psi \\
M,w \models K_n\varphi & :\textit{iff} & \forall w' : w' \sim_n w \Rightarrow M,w' \models \varphi \\
M,w \models C_B\varphi & :\textit{iff} & \forall w' : w' \sim_B w \Rightarrow M,w' \models \varphi \\
M,w \models [\alpha]\varphi & :\textit{iff} & \forall S \subseteq \bullet\mathcal{S}5_{\subseteq N} : (M,w)[\![\alpha]\!]S \Rightarrow \exists s' \in S : s' \models \varphi
\end{array}$$

$$\begin{array}{rlll}
s[\![?\varphi]\!]S & :\textit{iff} & s \models \varphi \textit{ and } S = \{((\langle W_\varphi, \emptyset, V|W_\varphi\rangle, w))\ \} & \textit{(see below)} \\
s[\![L_B\alpha]\!]S & :\textit{iff} & \exists S' : s[\![\alpha]\!]S' \textit{ and } S = \{((\langle W', \sim', V'\rangle, S'))\ \} & \textit{(see below)} \\
[\![\alpha\ ;\ \alpha']\!] & := & [\![\alpha]\!] \circ [\![\alpha']\!] & \\
[\![\alpha \cup \alpha']\!] & := & [\![\alpha]\!] \cup [\![\alpha']\!] & \\
[\![\alpha\ !\ \alpha']\!] & := & [\![\alpha]\!] & \\
[\![\alpha \cap \alpha']\!] & := & [\![\alpha]\!] \sqcup [\![\alpha']\!] &
\end{array}$$

For arbitrary S: $S \models \varphi$ iff for all $s \in S$: $s \models \varphi$. In the clause of action interpretation for 'test': $W_\varphi = \{v \in \mathcal{D}(M) \mid M,v \models \varphi\}$. In the clause for 'learning': $W' := \{S'' \mid \exists v \in M : (M,v)[\![t(\alpha)]\!]S''\}$; for an arbitrary agent n: $\sim'_n := \sim_n$, where $\sim_n$ is equivalence of sets of epistemic states; and for an arbitrary atom p: $S'' \in V'_p$ iff [for all $((\langle W'', \sim'', V''\rangle, w'') = s'' \in S'' : w'' \in V''_p$].

The notion $\langle\alpha\rangle$ is dual to $[\alpha]$ and can be conveniently defined as

$$s \models \langle\alpha\rangle\varphi \ \text{ iff } \ \exists S : s[\![\alpha]\!]S \text{ and } S \models \varphi$$

This may be intuitively more appealing: from the given epistemic state s, we can reach a set of of epistemic states S where φ holds everywhere ('concurrently'). Our treatment of the dynamic operators is similar to that in dynamic logic [Pel87, Gol92].

A test results in an epistemic state without access for any agent. This is appropriate: how knowledge changes is only expressed in 'learning', so before we encounter a learn operator we cannot say anything at all about the knowledge of the agents in the epistemic state resulting from action execution: no access. One might as well say that, while compositionally interpreting an action, the computation of agents' knowledge is *deferred* until L operators are encountered.

Learning $L_B\alpha$ is defined in terms of $t(\alpha)$, and *this* is how local choice constructions $\alpha \; ! \; \alpha'$ get their meaning from being bound by a learning operator: specifically, $[\![L_B(\beta \; ! \; \beta')]\!]$ is computed from $[\![\beta \cup \beta']\!]$, and therefore from $[\![\beta]\!]$ *and* $[\![\beta']\!]$. To execute an action $L_B\alpha$ in an epistemic state s, we do not just have to execute the *actual* action α in the *actual* epistemic state s, but also any *other* action of the same type $t(\alpha)$ as α in any *other* epistemic state s' with the same underlying model as s.[1] The results are the *worlds* in the epistemic state that results from executing $L_B\alpha'$ in s. Such worlds (that are sets of epistemic states) cannot be distinguished from each other by an agent $n \in B$ if they are indistinguishable as sets. This induces a notion of n-equivalence among action interpretations:

Definition 1.12 (Semantic accessibility of actions) *Let* $\alpha, \alpha' \in \mathcal{L}_N^{\mathsf{act}}$, *and* $n \in N$. *Let* $M \in \mathcal{S}5_N$, $w, w' \in M$, *and* $S \subseteq \bullet\mathcal{S}5_N$. *Then:*

$$\begin{array}{lll} [\![\alpha]\!] \sim_n [\![\alpha']\!] & \textit{iff} & [\; w \sim_n w' \textit{ and } (M,w)[\![\alpha]\!]S \;] \Rightarrow \\ & & [\; \exists S' : (M,w')[\![\alpha']\!]S' \textit{ and } S \sim_n S' \;] \end{array}$$

In other words, modulo nondeterminism and concurrency: two actions are the same for an agent, if always when two worlds are the same for an agent, executing those actions doesn't make them different. An infinite number of actions is similar in this sense. For example, $?p$ is the same for any agent as $?(p \vee p)$. However, given an action, it can be determined which actions that are comparable to it (which is the finite set $[\cdot]_{=_T}$), are the same for a given agent or not. This will be addressed in section 1.3.

[1] An alternative, equivalent, formulation of the semantics of $L_B\alpha$ builds the domain of the resulting epistemic state using all actions that are *comparable* to α:

$$W' := \{S'' \mid \exists v \in M, \exists \beta =_T \alpha : (M,v)[\![\beta]\!]S''\}$$

The semantics may *appear* complex, because worlds in the model resulting from learning are actually sets of epistemic states. It is therefore important to realize that this is *merely a complex naming device* for worlds, but that the semantics is simple where it matters: the accessibility between worlds (simple: use $\sim_n$), and the value of atoms (simple: keep current value).

If the interpretation of α in s is not empty, we say that α is *executable* in s. For all actions except concurrent knowledge actions it is more intuitive to think of their interpretation as a relation between epistemic states than as a relation between an epistemic state and a set of epistemic states: if $s[\![\alpha]\!]\{s'\}$, we like to think of s' as the result of executing α in s. The notational abbreviation $s[\![\alpha]\!]s' :\Leftrightarrow s[\![\alpha]\!]\{s'\}$ allows us to keep using this helpful intuition. Further, if the interpretation is functional as well, write $s[\![\alpha]\!]$ for the unique s' such that $s[\![\alpha]\!]s'$. If this is the case for arbitrary s, we call α a *state transformer.* Note that tests are state transformers.

EXAMPLE 1.13 *The interpretation of* read $= L_{12}(L_1?p \cup L_1?\neg p)$ *(see Action scenario 1.2) on* (Arc, u) *(see Example 1.5) is defined in terms of the interpretation of* $L_1?p \cup L_1?\neg p$ *on* (Arc, u) *and* (Arc, v)*. To interpret* $L_1?p \cup L_1?\neg p$ *on* (Arc, u) *we may either interpret* $L_1?p$ *or* $L_1?\neg p$*. Only the first can be executed. The interpretation of* $L_1?p$ *on* (Arc, u) *is defined in terms of the interpretation of* $?p$ *on any epistemic state* (Arc, x) *where* $?p$ *can be executed, i.e. where* p *holds, that is on* (Arc, u)*;* $(Arc, u)[\![?p]\!]$ *is the singleton epistemic state consisting of world* u *without access. This epistemic state is therefore the single world in the domain of* $(Arc, u)[\![L_1?p]\!]$*. That world has reflexive access for 1, because the epistemic state it stands for lacks access for 1, so that:*

$$(Arc, u)[\![?p]\!] \sim_1 (Arc, u)[\![?p]\!]$$

In the next and final stage of the interpretation, note that (as worlds)

$$(Arc, u)[\![L_1?p]\!] \sim_2 (Arc, u)[\![L_1?\neg p]\!]$$

because agent 2 does not occur in those epistemic states, but that

$$(Arc, u)[\![L_1?p]\!] \not\sim_1 (Arc, u)[\![L_1?\neg p]\!]$$

because $(Arc, u)[\![L_1?p]\!]$ *is not bisimilar to* $(Arc, u)[\![L_1?\neg p]\!]$*. Further details have been omitted. See Figure 1.2.*

EXAMPLE 1.14 *The interpretation of* tell $= L_{12}?p$ *(see Action scenario 1.1) on* (Arc, u) *can be computed along the same lines as that of* read

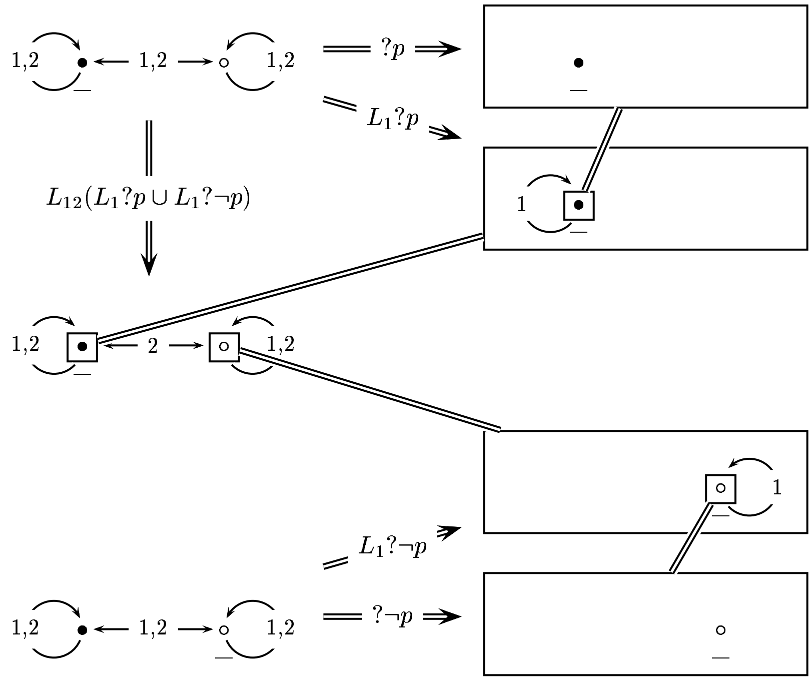

Figure 1.2. Details of the interpretation of action read in (Arc, u). All access is visualized. Atom p holds in • worlds, and does not hold in ∘ worlds. Linked boxes are identical. See also Figure 1.1.

but is much simpler. The interpretation of $\mathsf{bothmayread} = L_{12}((L_1?p \cap L_2?p) \cup (L_1?\neg p \cap L_2?\neg p) \cup L_1?p \cup L_1?\neg p \cup L_2?p \cup L_2?\neg p \cup\ ?p \cup\ ?\neg p)$ *(see Action scenario 1.4) is more complex. The choice action bound by* L_{12} *can be executed in four ways in* (Arc, u)*, and similarly in four ways in* (Arc, v)*. These result in the eight worlds of the cube-shaped model of Figure 1.1. We give a detail of the computation that involves concurrency.*

The world in the left-front-top corner of the cube in Figure 1.1 is the set of epistemic states S *such that* $(Arc, u)[\![L_1?p]\!]S$*, i.e., as the interpretation is clearly functional:* $S = \{(Arc, u)[\![L_1?p]\!]\}$*. The world in the left-back-top corner is the set of epistemic states* S' *such that* $(Arc, u)[\![L_1?p \cap L_2?p]\!]S'$*, i.e.* $S' = S \cup S''$*, with* S'' *such that* $(Arc, u)[\![L_2?p]\!]S''$*. Therefore,* $S' = \{(Arc, u)[\![L_1?p]\!], (Arc, u)[\![L_2?p]\!]\}$*. We have that* $S \sim_1 S'$*, because* $(Arc, u)[\![L_1?p]\!] \in S$ *can be mapped to ditto in* S'*, and because* $(Arc, u)[\![L_1?p]\!] \sim_1 (Arc, u)[\![L_1?p]\!]$*; and because, vice versa,* $(Arc, u)[\![L_1?p]\!]$

$\in S'$ can be mapped to ditto in S, and agent 1 does not occur at all in $(Arc, u)[\![L_2?p]\!] \in S'$.

EXAMPLE 1.15 *The second clause in definition 1.10, where bisimilarity to an n-equal epistemic state is a sufficient condition for n-equivality of epistemic states,* cannot *be removed from the definition of equivalence between epistemic states.*

E.g., with the stronger condition of identity (or isomorphism, even), agent 1 is unable to distinguish action

$$L_{12}(!L_1L_1?p \cup (L_1?p \cap L_2?p) \cup L_1?\neg p \cup ?\top)$$

from action

$$L_{12}(L_1L_1?p \cup\ !(L_1?p \cap L_2?p) \cup L_1?\neg p \cup ?\top)$$

after execution in epistemic state (Arc, u). Unlike before, we cannot now establish that $(Arc, u)[\![L_1L_1?p]\!] \sim_1 (Arc, u)[\![L_1?p \cap L_2?p]\!]$. This 1-equivalence fails, because $(Arc, u)[\![L_1L_1?p]\!] \not\sim_1 (Arc, u)[\![L_1?p]\!]$, because their underlying models are different*: the domain of the first consists of an epistemic state $(Arc, u)[\![L_1?p]\!]$ (consisting of one world, accessible to 1, p holds), the domain of the second of an epistemic state $(Arc, u)[\![?p]\!]$ (consisting of one world, no access, p holds). However, obviously $(Arc, u)[\![L_1L_1?p]\!] \underline{\leftrightarrow} (Arc, u)[\![L_1?p]\!]$ (both have a domain consisting of one world, that is accessible to 1, and where p holds).*

Without the bisimilarity clause, in the resulting epistemic state agent 1 would know that agent 2 has not learnt p after execution of the first action. So, also from a conceptual point of view, we can easily realise that this cannot be correct.

We close this section with some elementary properties of the semantics. Two actions α, α' are (semantically) *equivalent*, notation $[\![\alpha]\!] = [\![\alpha']\!]$, if they induce the same relation between epistemic states and sets of epistemic states, modulo names of worlds.

PROPOSITION 1.16 (ACTION ALGEBRA) *Let $\alpha, \alpha', \alpha'' \in \mathcal{L}_N^{\mathsf{act}}$. Then:*

$$\begin{array}{rcl}
[\![(\alpha \cup \alpha') \cup \alpha'']\!] & = & [\![\alpha \cup (\alpha' \cup \alpha'')]\!] \\
[\![(\alpha \ ; \ \alpha') \ ; \ \alpha'']\!] & = & [\![\alpha \ ; \ (\alpha' \ ; \ \alpha'')]\!] \\
[\![(\alpha \cup \alpha') \ ; \ \alpha'']\!] & = & [\![(\alpha \ ; \ \alpha'') \cup (\alpha' \ ; \ \alpha'')]\!] \\
[\![(\alpha \ ; \ \alpha') \cup \alpha'']\!] & = & [\![(\alpha \cup \alpha'') \ ; \ (\alpha' \cup \alpha'')]\!]
\end{array}$$

Proof () By using simple relational algebra. We show the third, the rest is similar: $[\![(\alpha \cup \alpha') \ ; \ \alpha'']\!] = [\![\alpha \cup \alpha']\!] \circ [\![\alpha'']\!] = ([\![\alpha]\!] \cup [\![\alpha']\!]) \circ [\![\alpha'']\!] = ([\![\alpha]\!] \circ [\![\alpha']\!]) \cup ([\![\alpha]\!] \circ [\![\alpha'']\!]) = [\![\alpha \ ; \ \alpha']\!] \cup [\![\alpha \ ; \ \alpha'']\!] = [\![(\alpha \ ; \ \alpha') \cup (\alpha \ ; \ \alpha'')]\!]$. ⊣

Associativity of $\cup$ was already assumed in the text before. There are various other algebraic properties of action interpretations, such as $\llbracket L_B L_B \alpha \rrbracket = \llbracket L_B \alpha \rrbracket$. The next proposition relates action instances and action types to other actions.

Proposition 1.17 (Action types and instances)
Let $\alpha \in \mathcal{L}_N^{\mathsf{act}}$. Then:

(a) $\llbracket \alpha \rrbracket \subseteq \llbracket t(\alpha) \rrbracket$
(b) *Action instances have a functional interpretation.*
(c) *if $\beta \in T(\alpha)$ then $\llbracket \beta \rrbracket \subseteq \llbracket \alpha \rrbracket$*
(d) $\llbracket \alpha \rrbracket = \llbracket \bigcup_{\beta \in T(\alpha)} \beta \rrbracket$

Proof ()
(a) Induction on α. The only nontrivial case is $\alpha' \,!\, \alpha''$. We have that: $\llbracket \alpha' \,!\, \alpha'' \rrbracket = \llbracket \alpha' \rrbracket \subseteq \llbracket \alpha' \cup \alpha'' \rrbracket = \llbracket \alpha' \rrbracket \cup \llbracket \alpha'' \rrbracket \subseteq_{IH} \llbracket t(\alpha') \rrbracket \cup \llbracket t(\alpha'') \rrbracket = \llbracket t(\alpha') \cup t(\alpha'') \rrbracket = \llbracket t(\alpha' \,!\, \alpha'') \rrbracket$.

(b) Induction on α. The only nontrivial case is nondeterministic choice. Let $\beta \in T(\alpha' \cup \alpha'')$. Then either $\beta = \beta' \,!\, \beta''$ or $\beta = \beta' \text{ ¡ } \beta''$, with $\beta' \in T(\alpha')$ and $\beta'' \in T(\alpha'')$. In the first case, by induction $\llbracket \beta' \rrbracket$ is functional, and therefore also $\llbracket \beta' \,!\, \beta'' \rrbracket = \llbracket \beta' \rrbracket$. In the second case, this follows from the functionality of $\llbracket \beta'' \rrbracket$.

(c) Induction on α. A typical case: Let $s \in \bullet\mathcal{S}5_N$, $S \subseteq \bullet\mathcal{S}5_{\subseteq N}$, and suppose that $s \llbracket \alpha' \cup \alpha'' \rrbracket S$. Then either $s \llbracket \alpha' \rrbracket S$ or $s \llbracket \alpha'' \rrbracket S$. If $s \llbracket \alpha' \rrbracket S$ then, by induction, there is a $\beta' \in T(\alpha')$ such that $s \llbracket \beta' \rrbracket S$. Let $\beta'' \in T(\alpha'')$ be arbitrary. Then $\beta' \,!\, \beta'' \in T(\alpha' \cup \alpha'')$ and $s \llbracket \beta' \rrbracket S = s \llbracket \beta' \,!\, \beta'' \rrbracket S$.

(d) Induction on α. Some cases. Case $\alpha' \;;\; \alpha''$: use Proposition 1.16. Case $\alpha' \cup \alpha''$: $\llbracket \alpha' \cup \alpha'' \rrbracket =_{IH} \llbracket \bigcup_{\beta' \in T(\alpha')} \beta' \cup \bigcup_{\beta'' \in T(\alpha'')} \beta'' \rrbracket = \llbracket \bigcup_{\beta' \in T(\alpha'), \beta'' \in T(\alpha'')} (\beta' \,!\, \beta'') \cup \bigcup_{\beta' \in T(\alpha'), \beta'' \in T(\alpha'')} (\beta' \text{ ¡ } \beta'') \rrbracket = \llbracket \bigcup_{\beta \in T(\alpha)} \beta \rrbracket$. Case $L_B \alpha'$: use that $s \llbracket L_B \alpha' \rrbracket \{(M', S)\}$ presupposes $s \llbracket \alpha' \rrbracket S$. $\dashv$

Proposition 1.17.a expresses that the interpretation of an action is contained in the interpretation of its type. Proposition 1.17.b entails that the interpretation of state transformers is indeed functional. Proposition 1.17.c expresses that the interpretation of an instance of an action is contained in the interpretation of that action. Proposition 1.17.d expresses that an action is somehow the same (induces the same interpretation) as nondeterministic choice between all its instances.

The two main theorems of interest are the following. The proofs are adapted from [vD02b]. They are by simultaneous induction (referring to each other). For convenience in the proof, we lift the notion of bisimilarity from one between epistemic states to one between sets of epistemic states: $S \underline{\leftrightarrow} S'$ iff for all $s \in S$ there is an $s' \in S'$ such that $s \underline{\leftrightarrow} s'$, and for all $s' \in S'$ there is an $s \in S$ such that $s \underline{\leftrightarrow} s'$.

THEOREM 1.18 (BISIMILARITY IMPLIES MODAL EQUIVALENCE) *Let* $\varphi \in \mathcal{L}_N$. *Let* $s, s' \in \bullet\mathcal{S}5_N$. *If* $s \underline{\leftrightarrow} s'$, *then* $s \models \varphi \Leftrightarrow s' \models \varphi$.

Proof () By induction on the structure of φ. The proof is standard, except for the clause $\varphi = [\alpha]\psi$ that we therefore present in detail. Assume $s \models [\alpha]\psi$. We have to prove $s' \models [\alpha]\psi$. Let S' be arbitrary such that $s'[\![\alpha]\!]S'$. By simultaneous induction hypothesis (Theorem 1.19) it follows from $s'[\![\alpha]\!]S'$ and $s \underline{\leftrightarrow} s'$ that there is an S such that $S \underline{\leftrightarrow} S'$ and $s[\![\alpha]\!]S$. From $s[\![\alpha]\!]S$ and $s \models [\alpha]\psi$ (given) follows that there is an $s'' \in S$ such that $s'' \models \psi$. From $S \underline{\leftrightarrow} S'$, $s'' \in S$, and $s'' \models \psi$, follows that there is an $s''' \in S'$ such that $s''' \models \psi$. From $s''' \models \psi$, $s''' \in S'$, and $s'[\![\alpha]\!]S'$ follows that $s' \models [\alpha]\psi$. $\dashv$

THEOREM 1.19 (ACTION EXECUTION PRESERVES BISIMILARITY) *Let* $\alpha \in \mathcal{L}_N^{\mathsf{act}}$ *and* $s, s' \in \bullet\mathcal{S}5_N$. *If* $s \underline{\leftrightarrow} s'$ *and there is an* $S \subseteq \bullet\mathcal{S}5_{\subseteq N}$ *such that* $s[\![\alpha]\!]S$, *then there is an* $S' \subseteq \bullet\mathcal{S}5_{\subseteq N}$ *such that* $s'[\![\alpha]\!]S'$ *and* $S \underline{\leftrightarrow} S'$.

Proof () By induction on the structure of α, or, to be slightly more accurate: induction on the complexity of α, where $L_B\alpha > t(\alpha)$.

We remind the reader of previously introduced notational conventions used in this proof: If α is a state transformer and executable in s, write $s[\![\alpha]\!]$ for the s' such that $s[\![\alpha]\!]\{s'\}$; if $(M, w) = s$ is an epistemic state and $v \in \mathcal{D}(M)$, we also write: $v \in s$, or $v \in M$.

Case $?\varphi$: Suppose $\mathfrak{R} : s \underline{\leftrightarrow} s'$. By simultaneous induction (Theorem 1.18) it follows from $s \underline{\leftrightarrow} s'$ and $s \models \varphi$ that $s' \models \varphi$. Define, for all $v \in s[\![?\varphi]\!]$, $v' \in s'[\![?\varphi]\!]$: $\mathfrak{R}^{?\varphi}(v, v') :\Leftrightarrow \mathfrak{R}(v, v')$. Then $\mathfrak{R}^{?\varphi} : s[\![?\varphi]\!] \underline{\leftrightarrow} s'[\![?\varphi]\!]$, because (Points:) $\mathfrak{R}^{?\varphi}(w, w)$, (Back and Forth:) both epistemic states have empty access, and (Valuation:) $\mathfrak{R}^{?\varphi}(v, v')$ implies $\mathfrak{R}(v, v')$. In other words: $\{s'[\![?\varphi]\!]\}$ is the required S'.

Case $L_B\alpha'$: Suppose $\mathfrak{R} : s \underline{\leftrightarrow} s'$ and $s[\![L_B\alpha']\!]\{s_+\}$. Let $s = (M, w)$ and $s' = (M', w')$. Let $S_+ \in s_+$ be arbitrary (i.e.: a world S_+ that is a set of epistemic states, occurring in the domain of epistemic state s_+). Then there is a $z \in s$ such that $(M, z)[\![t(\alpha)]\!]S_+$. Because $z \in s$ and $\mathfrak{R} : s \underline{\leftrightarrow} s'$, there is a $z' \in s'$ such that $\mathfrak{R}(z, z')$ and obviously we also have that $\mathfrak{R} : (M, z) \underline{\leftrightarrow} (M', z')$ (the domain of an epistemic state is the domain of its underlying model). By induction, using that the complexity of $t(\alpha)$ is smaller than that of $L_B\alpha$, there is an S'_+ such that $(M', z')[\![t(\alpha)]\!]S'_+$ and $S_+ \underline{\leftrightarrow} S'_+$.

Now define s'_+ as follows: its domain consists of worlds S'_+ constructed according to the procedure just outlined; accessibility between such worlds is accessibility between those worlds as sets of epistemic states, and valuation corresponds to those in the bisimilar worlds of s_+. Finally, the point of s'_+ is a set of epistemic states that is the result of executing α in s' and that is bisimilar to the point of s_+. The accessibility on s'_+ corresponds to that on s_+, because for arbitrary sets of epistemic states (and thus for worlds in s_+, s'_+): if $S_1 \sim_n S_2$, $S_1 \underline{\leftrightarrow} S'_1$, and $S_2 \underline{\leftrightarrow} S'_2$, then $S'_1 \sim_n S'_2$. Therefore $s_+ \underline{\leftrightarrow} s'_+$, $s'[\![L_B\alpha]\!]\{s'_+\}$, and $\{s'_+\}$ is the required S'.

Case $\alpha\ ;\ \beta$: Suppose $s \underline{\leftrightarrow} s'$ and $s[\![\alpha\ ;\ \beta]\!]S$. Note that $[\![\alpha\ ;\ \beta]\!] = [\![\alpha]\!] \circ [\![\beta]\!]$. Let S_1 be such that $s[\![\alpha]\!]S_1$ and for all $s_1 \in S_1$ there is an S_{s_1} such that $s_1[\![\beta]\!]S_{s_1}$, and $S = \cup_{s_1} S_{s_1}$. By induction we have an S'_1 such that $s'[\![\alpha]\!]S'_1$ and $S_1 \underline{\leftrightarrow} S'_1$. Again by induction, for an arbitrary $s_1 \in S_1$ such that $s_1 \underline{\leftrightarrow} s'_1 \in S'_1$ and $s_1[\![\beta]\!]S_{s_1}$, we have an $S'_{s'_1}$ such that $s'_1[\![\beta]\!]S'_{s'_1}$ and $S_{s_1} \underline{\leftrightarrow} S'_{s'_1}$. Let $S' = \cup_{s'_1} S'_{s'_1}$. Then $s'[\![\alpha\ ;\ \beta]\!]S'$ and $S \underline{\leftrightarrow} S'$ (S' may be larger than S, but $S \underline{\leftrightarrow} S'$ then also holds, see case $\cap$.)

Case $\alpha \cup \beta$: Suppose $s \underline{\leftrightarrow} s'$ and $s[\![\alpha \cup \beta]\!]S$. Then either $s[\![\alpha]\!]S$ or $s[\![\beta]\!]S$. If $s[\![\alpha]\!]S$, then by induction there is an S' such that $s'[\![\alpha]\!]S'$ and $S \underline{\leftrightarrow} S'$. Therefore also $s'[\![\alpha \cup \beta]\!]S'$. Similarly, if $s[\![\beta]\!]S$.

Cases $\alpha\ !\ \beta$ and α ¡ β are similar to $\alpha \cup \beta$.

Case $\alpha \cap \beta$. Suppose $s \underline{\leftrightarrow} s'$ and $s[\![\alpha \cap \beta]\!]S$. Let S_1 and S_2 be such that $s[\![\alpha]\!]S_1$, $s[\![\beta]\!]S_2$, and $S_1 \cup S_2 = S$. By induction, there are S'_1 and S'_2 such that $s'[\![\alpha]\!]S'_1$, $s'[\![\beta]\!]S'_2$, $S_1 \underline{\leftrightarrow} S'_1$, $S_2 \underline{\leftrightarrow} S'_2$. The required S' such that $S \underline{\leftrightarrow} S'$ is $S' := S'_1 \cup S'_2$. (It doesn't matter if $S_1 \cap S_2$ is empty or not. In the last case, S'_1 and S'_2 may have been chosen so that S' contains more epistemic states than S, but $S \underline{\leftrightarrow} S'$ still holds.) $\dashv$

A corollary of theorem 1.19 is the following:

COROLLARY 1.20 *Let $s, s' \in \bullet\mathcal{S}5_N$, and let $\alpha \in \mathcal{L}_N^{\mathsf{act}}$ be a state transformer that is executable in s. If $s \underline{\leftrightarrow} s'$, then $s[\![\alpha]\!] \underline{\leftrightarrow} s'[\![\alpha]\!]$.*

3. Proof system

In this section we present the proof system for concurrent dynamic epistemic logic. It is based on the dynamic epistemic logics of [Ger99] and [BMS02], and on Concurrent PDL [Pel87]. Before we present the proof system we need a syntactic notion for the executability of an action and a syntactic notion for equivalence of actions, because, among other things, we want to express the following as an axiom (*Action use*): 'after an action α an agent n knows that φ, iff the executability of α implies that n knows that, for each action β that is equivalent to α for n, after β is executed φ holds.' The notion of executibility is captured syntactically by the *precondition*, see Definition 1.29. The notion of equivalence of actions can also be captured syntactically, and this is done in Definition 1.22. This is merely a partition, for each agent, on the set of actions that are comparable to a given action ($[\alpha] =_T$).

EXAMPLE 1.21 *Consider the action $L_{123}(?\top \cup L_1?p \cup (L_1?p \cap L_2?p))$, i.e., everybody learns that either nothing happens, or 1 learns p or both 1 and 2 learn p. Or, in a more 'natural' setting: an outsider tells Anne (1), Bill (2) and Cath (3), who are sitting in a bar, at a table, that he may have told Anne, or both Anne and Bill, that Anne is invited for a lecture. Even if Anne now knows that she is invited, she doesn't know whether Bill knows that too. In other words, Anne cannot distinguish action $L_{123}(?\top \cup\, !L_1?p \cup (L_1?p \cap L_2?p))$ from action $L_{123}(?\top \cup L_1?p \cup\, !(L_1?p \cap L_2?p))$. On the other hand, Bill can distinguish those actions: if he has learnt p, he knows that Anne has learnt p as well. However, both Anne and Bill* can *distinguish nothing happening – $L_{123}(!?\top \cup L_1?p \cup (L_1?p \cap L_2?p))$ – from something happening, whereas Cath cannot do so.*

A notion of accessibility among actions will be essential for the formulation of axioms and rules in the proof system. Semantically, we were already able to make that distinction, by the obvious way of 'lifting' the notion of access within a model, to one between epistemic states, to one between sets of epistemic states, and to one between relations between epistemic states and sets of epistemic states, such as between actions by means of their interpretation $[\![\cdot]\!]$ (see Definition 1.12). We now define a *syntactic* notion of accessibility among actions, such that syntactic access implies semantic access.

DEFINITION 1.22 (SYNTACTIC ACCESSIBILITY) *Let $Gr(\alpha)$ be the set of all agents occurring in learning operators in α, except those under the scope of '?'. Then*

$$\alpha \sim_n \beta \quad \textit{iff} \quad \alpha =_T \beta \textit{ and } \alpha \approx_n \beta$$

where $\approx_n$ is the symmetric closure of the smallest relation satisfying the following conditions:

$$
\begin{array}{lll}
\alpha \approx_n \beta & \textit{if} \;\; n \notin Gr(\alpha) \cup Gr(\beta) & \\
L_B\alpha \approx_n L_B\beta & \textit{if} \;\; \alpha =_T \beta \textit{ and } \alpha \approx_n \beta & \\
\alpha \bullet \alpha' \approx_n \beta \bullet \beta' & \textit{if} \;\; \alpha \approx_n \beta \textit{ and } \alpha' \approx_n \beta' & \textit{for } \bullet = \cap, \; ; \; , \cup \\
\left.\begin{array}{l}
\alpha \approx_n \beta \; ! \; \beta' \\
\alpha \; ! \; \alpha' \approx_n \beta \; ! \; \beta' \\
\alpha' \; \text{¡} \; \alpha \approx_n \beta \; ! \; \beta' \\
\alpha' \; \text{¡} \; \alpha \approx_n \beta' \; \text{¡} \; \beta \\
\alpha \approx_n \beta' \; \text{¡} \; \beta
\end{array}\right\} & \textit{if} \;\; \alpha \approx_n \beta & \\
\left.\begin{array}{l}
\alpha \approx_n \beta \cap \beta' \\
\alpha \approx_n \beta' \cap \beta
\end{array}\right\} & \textit{if} \;\; \alpha \approx_n \beta \textit{ and } n \notin Gr(\beta') &
\end{array}
$$

The ‘large group’ Gr of an action is the union of all groups learning anything anywhere, except when under the scope of ‘?’. Relation $\sim_n$ induces a partition on the set of all actions of the same type (see Definition 1.8 of $=_T$). In the clause for ‘learning’ it is essential that the type requirement is made once more: note that otherwise, e.g., $L_1?p \sim_1 L_1?q$, as tests are indistinguishable from each other for any agent.

Proposition 1.23 *$\sim_n$ is an equivalence.*

Proof () We prove that $\approx_n$ is an equivalence, from which follows that $\sim_n$ is an equivalence. Obviously, $\approx_n$ is reflexive and symmetrical. Transitivity is proven by distinguishing many cases, we merely do the crucial ones.

Suppose $\alpha \; ! \; \alpha' \approx_n \beta \; \text{¡} \; \beta'$ and $\beta \; \text{¡} \; \beta' \approx_n \gamma \; ! \; \gamma'$. Then, using the definition: $\alpha \approx_n \beta'$ and $\beta' \approx_n \gamma$. Using induction, we have $\alpha \approx_n \gamma$. By again using the definition of $\approx_n$: $\alpha \; ! \; \alpha' \approx_n \gamma \; ! \; \gamma'$.

Suppose $\alpha \approx_n \beta \cap \beta'$ and $\gamma \approx_n \beta \cap \beta'$, and that the first holds because $\alpha \approx_n \beta$ and $n \notin Gr(\beta')$, and the second because $\gamma \approx_n \beta'$ and $n \notin Gr(\beta)$. Now observe that, in general, an agent n either learns something in both of two $\approx_n$ equal actions (i.e., n is in Gr of both), or both actions are invisible from n’s viewpoint (i.e., n is in Gr of neither). Therefore, from $\alpha \approx_n \beta$ and $n \notin Gr(\beta)$ follows $n \notin Gr(\alpha)$, and from $\gamma \approx_n \beta'$ and $n \notin Gr(\beta')$ follows $n \notin Gr(\gamma)$. From $n \notin Gr(\alpha)$ and $n \notin Gr(\gamma)$ follows $\alpha \approx_n \gamma$. ⊣

Lemma 1.24 *Given an action α, the set $[\alpha]_{\sim_n}$ is effectively computable.*

Proof () If m is the number of $!$ and ¡ operators in α, there are at most 2^m actions β that are comparable to α (such that $\beta =_T \alpha$). So $[\alpha]_{=_T}$ can easily be determined. The set $[\alpha]_{\sim_n}$ is determined by computing the partition on $[\alpha]_{=_T}$ given $\approx_n$. This can be determined in linear time with respect to the length of the action. ⊣

EXAMPLE 1.25 (SYNTACTIC ACCESSIBILITY) *Consider Example 1.21 at the beginning of this subsection. The type of action discussed was* $L_{123}(?\top \cup L_1?p \cup (L_1?p \cap L_2?p))$. *More precisely, we may choose to read* $L_{123}((?\top \cup L_1?p) \cup (L_1?p \cap L_2?p))$ *(see Proposition 1.16). There are four instances of this type. This is a* $=_T$ *equivalence class. One of those instances is* $L_{123}((?\top \; ! \; L_1?p) \; ! \; (L_1?p \cap L_2?p))$, *which we informally write as* $L_{123}(!?\top \cup L_1?p \cup (L_1?p \cap L_2?p))$. *Note that only three of the four instances are essentially different: once you have chosen 'right', it does not matter whether the subsequent choice on the left is 'right' or 'left'. We now compute the mentioned equivalences on the set of instances: Let* α *be the 'left choice is learned', i.e.,* $\alpha = L_{123}((?\top \text{ ¡ } L_1?p) \; ! \; (L_1?p \cap L_2?p))$, *and let* β *be* $L_{123}((?\top \text{ ¡ } L_1?p) \text{ ¡ } (L_1?p \cap L_2?p))$

'Even if Anne now knows that she is invited, she doesn't know whether Bill is invited too':

$$\begin{array}{lll}
\alpha & \approx_1 & \beta \\
\Leftrightarrow & & \\
(?\top \text{ ¡ } L_1?p) \; ! \; (L_1?p \cap L_2?p) & \approx_1 & (?\top \text{ ¡ } L_1?p) \text{ ¡ } (L_1?p \cap L_2?p) \\
\Leftarrow & & \\
?\top \text{ ¡ } L_1?p & \approx_1 & L_1?p \cap L_2?p \\
\Leftarrow & & \\
L_1?p & \approx_1 & L_1?p \cap L_2?p \\
\Leftarrow & & \\
L_1?p & \approx_1 & L_1?p \qquad \text{and } 1 \notin Gr(L_2?p)
\end{array}$$

'Bill can distinguish those actions'. This proceeds similarly, except for the last step, where $L_1?p \not\approx_2 L_1?p \cap L_2?p$, *because 2 can distinguish an action where he learns something, namely* $L_2?p$, *and that is part of* $L_1?p \cap L_2?p$, *from one that is invisible to him, namely* $L_1?p$ *(on the left).*

Cath cannot distinguish nothing happening from something happening (see above for the meaning of α *and* β*):*

$$\begin{array}{lll}
\alpha & \approx_3 & \beta \\
\Leftrightarrow (...) & & \\
?\top \; ! \; L_1?p & \approx_3 & ?\top \text{ ¡ } L_1?p \\
\Leftarrow & & \\
?\top & \approx_3 & L_1?p \\
\Leftarrow & & \\
3 \notin Gr(?\top) \cup Gr(L_1?p) = \{1\} & &
\end{array}$$

Proposition 1.26 (Syntactic implies semantic equivalence)

$$\beta \sim_n \beta' \Rightarrow \llbracket\beta\rrbracket \sim_n \llbracket\beta'\rrbracket$$

Proof () We prove that $\beta \approx_n \beta' \Rightarrow \llbracket\beta\rrbracket \sim_n \llbracket\beta'\rrbracket$, from which the required follows. The proof is by induction on the structure of the actions β and β' (i.e. prove each inductive case of β by induction on β'). We show the non-trivial cases.

If $n \notin Gr(\beta) \cup Gr(\beta')$, then trivially $\llbracket\beta\rrbracket \sim_n \llbracket\beta'\rrbracket$: given arbitrary s and S (resp. s and S') such that $s\llbracket\beta\rrbracket S$ (resp. $s'\llbracket\beta\rrbracket S'$, the group of S (resp. S') must be $\emptyset$. Therefore $S \sim_n S'$. A fortiori $S \sim_n S'$ when $s \sim_n s'$, so that $\llbracket\beta\rrbracket \sim_n \llbracket\beta'\rrbracket$.

If $L_B\alpha \approx_n L_B\beta$ and $n \in B$, by the definition of $\approx_n$: $\alpha \approx_n \beta$, by induction: $\llbracket\alpha\rrbracket \sim_n \llbracket\beta\rrbracket$, and by the construction of $\sim_n$ in the case 'learning' of the definition of $\llbracket\cdot\rrbracket$: $\llbracket L_B\alpha\rrbracket \sim_n \llbracket L_B\beta\rrbracket$ immediately follows.

If $\alpha \mathbin{!} \beta \approx_n \alpha' \mathbin{¡} \beta'$, then $\alpha \approx_n \beta'$, so by induction $\llbracket\alpha\rrbracket \sim_n \llbracket\beta'\rrbracket$, so, by definition of $\llbracket\cdot\rrbracket$, $\llbracket\alpha \mathbin{!} \beta\rrbracket \sim_n \llbracket\alpha' \mathbin{¡} \beta'\rrbracket$.

Suppose $\alpha \approx_n \beta \cap \beta'$ because $\alpha \approx_n \beta$ and $n \notin Gr(\beta')$. By induction $\llbracket\alpha\rrbracket \sim_n \llbracket\beta\rrbracket$. As for arbitrary sets of epistemic states S, S', S'' it holds that: if $S \sim_n S'$ and $n \notin gr(S'')$, then $S \sim_n S' \cup S''$, the required $\llbracket\alpha\rrbracket \sim_n \llbracket\beta \cap \beta'\rrbracket$ now follows. $\dashv$

Corollary 1.27 (Preservation of accessibility) *Given an epistemic model M, $v, v' \in M$, and actions β, β' such that β is executable in (M, v) and β' is executable in (M, v'). If $v \sim_n v'$ and $\beta \sim_n \beta'$, then for S such that $(M, v)\llbracket\beta\rrbracket S$ there is an S' such that $(M, v')\llbracket\beta\rrbracket S'$ and $S \sim_n S'$.*

Corollary 1.28 *For action instances and state transformers β, β': if $v \sim_n v'$ and $\beta \sim_n \beta'$ (and executability), then $(M, v)\llbracket\beta\rrbracket \sim_n (M, v')\llbracket\beta'\rrbracket$.*

In Proposition 1.26 we have established that if two actions are syntactically the same for an agent, they are also semantically the same: if $\beta \sim_n \beta'$, then $\llbracket\beta\rrbracket \sim_n \llbracket\beta'\rrbracket$. For trivial reasons this is indeed a proper inclusion, because actions of a different type cannot be syntactically the same. For example, we have that $\llbracket ?p\rrbracket \sim_1 \llbracket ?\neg\neg p\rrbracket$ but $?p \not\sim_1 ?\neg\neg p$ (because $?p \neq_T ?\neg\neg p$). Also, $\llbracket L_1?p\rrbracket \sim_1 \llbracket L_1L_1?p\rrbracket$ but $L_1?p \not\sim_1 L_1L_1?p$, and $\llbracket L_1?p \cup L_1?q\rrbracket \not\sim_1 \llbracket L_1?q \cup L_1?p\rrbracket$ but $L_1?p \cup L_1?q \not\sim_1 L_1?q \cup L_1?p$.

What about actions of the same type? All of the above, and more, can easily be incorporated as alternatives into actions of the same type. For example, consider the actions $L_{12}(L_1?p\cup!(L_1?\neg\neg p \cap L_2?q))$ and $L_{12}(!L_1?p \cup (L_1?\neg\neg p \cap L_2?q))$. These are different for 1, because $L_1?p$ is different for 1 from $L_1?\neg\neg p$, because the types of the respective tests are different (this requires syntactic identity!). Therefore, also in non-trivial cases, $\llbracket\beta\rrbracket \sim_n \llbracket\beta'\rrbracket$ does not imply $\beta \sim_n \beta'$.

This should not necessarily be a problem, as long as we have 'enough' actions that are the same for an agent to a given action to guarantee soudness of the proof system wherein we use syntactic accessibility. But it turns out that we have not enough. The principle of *Action use*, that will indeed occur in the following proof system, says that, if n knows that [after every action that is for n the same as the actual action, φ holds], then [n knows that φ holds] after that action.

Because $L_{12}(L_1?p\cup!(L_1?\neg\neg p \cap L_2?q))$ if different for agent 1 from $L_{12}(!L_1?p \cup (L_1?\neg\neg p \cap L_2?q))$, it is now derivable that after the first action 1 knows that q! In other words: agent 1 can make some epistemic distinction between p and $\neg\neg p$, which does not make sense.

More in general, suppose we have arbitrarily complex and differently described actions $\alpha, \beta \in \mathcal{L}_N^{\mathsf{act}}(P)$ and $n \in N$ such that $\alpha =_T \beta$ and $\llbracket\alpha\rrbracket \sim_n \llbracket\beta\rrbracket$ but $\alpha \not\sim_n \beta$, and let $q \notin P$, $B \supseteq Gr(\alpha) \cup Gr(\beta)$, and $m \notin B$. Consider the action where (everybody learns that) group B, which includes n, learn that either α or β takes place, but that in the second case agent m, not in B, learns that q. Obviously, we do not want n to be able to determine whether q after that action. However, $L_{B+m}(L_B\alpha\cup\,!(L_B\beta \cap L_m?q))$ is different for agent n from $L_{B+m}(!L_B\alpha \cup (L_B\beta \cap L_m?q))$, so after the first action, because of *Action use*, n knows that q, even though, of course, only m is supposed to know that.

Therefore we have to coarsen (as the opposite of 'refine') the partition on $=_T$ for agents n in Definition 1.22, in order to guarantee the soundness of the proof system to be introduced in Definition 1.31: we need more n-equal actions. In all the counterexamples above, the problem was that equivalent actions – actions such that $\llbracket\alpha\rrbracket = \llbracket\beta\rrbracket$, so that they are trivially n-equivalent as well: $\llbracket\alpha\rrbracket \sim_n \llbracket\beta\rrbracket$ – *should* also be syntactically n-equal for any n. We therefore suggest to solve our problem tentatively as follows: introduce a notion of 'general' syntactic equivalence of actions: $\alpha =_\vdash \alpha'$ iff, for arbitrary φ (however, depending on the complexity of α and β): $\vdash [\alpha]\varphi \leftrightarrow [\alpha']\varphi$; now redefine syntactic access $\sim_n$ as

$$\alpha \sim_n \beta \text{ iff } \alpha =_T \beta \text{ and } \exists\gamma : \gamma =_\vdash \alpha \text{ and } \gamma \approx_n \beta$$

This definition is unsatisfactory for two reasons. First, because the auxiliary notion of $\sim_n$ and the proof system yet to come are now defined simultaneously, we must show that the derivation of syntactic action equivalence does not need instances of axioms wherein syntactic action access thus established is used. However, the complexity measure used in $=_\vdash$ may well take care of that. Second, Lemma 1.24 no longer holds. Given an action α, the computation of $[\alpha]_{\sim_n}$ on (the easily determined set) $[\alpha]_{=_T}$ may be undecidable (even though it remains clearly finite), as the notion of action equivalence used in $\sim_n$ is defined in terms of derivability.

Other options out of the dilemma include: redefine $\approx_n$ by relaxing the constraint on $L_B\alpha$ (i), define $\approx_n$ not between arbitrary actions but between normal forms of actions (ii), or we may 'simply' define $\alpha \sim_n \beta$ iff $\alpha =_T \beta$ and $[\![\alpha]\!] \sim_n [\![\beta]\!]$, but incorporating a semantic feature in a syntactic notion seems to 'give away' the computational advantage an axiomatization pretends to have (iii) over the semantics. We hope to improve on the given solution.

In the remainder, assume that if $[\![\alpha]\!] \sim_n [\![\beta]\!]$, then there is a $\gamma =_T \beta$ such that $[\![\alpha]\!] \sim_n [\![\gamma]\!]$. From this follows, that if $\alpha =_T \beta$, then $\alpha \sim_n \beta$ iff $[\![\alpha]\!] \sim_n [\![\beta]\!]$.

We continue by saying when an action can be executed. An action can be executed in an epistemic state if its precondition is true. 'Precondition' is defined as follows.

Definition 1.29 (Preconditions of an action)

1 $\mathsf{pre}(?\varphi) := \varphi$

2 $\mathsf{pre}(\alpha\ ;\ \beta) := \mathsf{pre}(\alpha) \wedge \langle\alpha\rangle\mathsf{pre}(\beta)$

3 $\mathsf{pre}(\alpha \cup \beta) := \mathsf{pre}(\alpha) \vee \mathsf{pre}(\beta)$

4 $\mathsf{pre}(\alpha \cap \beta) := \mathsf{pre}(\alpha) \wedge \mathsf{pre}(\beta)$

5 $\mathsf{pre}(\alpha\ !\ \beta) := \mathsf{pre}(\alpha)$

6 $\mathsf{pre}(L_B\alpha) := \mathsf{pre}(\alpha)$

Lemma 1.30

$$\models \mathsf{pre}(\alpha) \leftrightarrow \langle\alpha\rangle\top$$

Proof () By induction on α. Note that $s \models \langle\alpha\rangle\top$ iff there is a set of epistemic states S such that $s[\![\alpha]\!]S$ (omitting the trivial part: 'and for all $s' \in S : s' \models \top$'). The crucial case 'learning': Suppose $s \models \langle L_B\alpha\rangle\top$. Let S be such that $s[\![L_B\alpha]\!]S$. For a point (world) $S' \in s' \in S$ we have $s[\![\alpha]\!]S'$ (because of the definition of action interpretation). I.e., $s \models \langle\alpha\rangle\top$. By induction $s \models \mathsf{pre}(\alpha)$ and because $\mathsf{pre}(L_B\alpha) = \mathsf{pre}(\alpha)$: $s \models \mathsf{pre}(L_B\alpha)$.

⊣

Now we are ready to provide the proof system. It is based on the proof systems of [BMS02] and [Pel87].

Definition 1.31 (Proof system)

Propositional tautologies	
Knowledge distribution	$K_n(\varphi \rightarrow \psi) \rightarrow (K_n\varphi \rightarrow K_n\psi)$
Truth	$K_n\varphi \rightarrow \varphi$
Positive introspection	$K_n\varphi \rightarrow K_nK_n\varphi$
Negative introspection	$\neg K_n\varphi \rightarrow K_n\neg K_n\varphi$
Common knowledge use	$C_B\varphi \rightarrow (\varphi \wedge E_BC_B\varphi)$
Test	$[?\varphi]\psi \leftrightarrow (\varphi \rightarrow \psi)$
Sequential composition	$[\alpha \ ; \ \alpha']\varphi \leftrightarrow [\alpha][\alpha']\varphi$
Nondeterministic choice	$[\alpha \cup \alpha']\varphi \leftrightarrow ([\alpha]\varphi \wedge [\alpha']\varphi)$
Concurrency	$[\alpha \cap \alpha']\varphi \leftrightarrow ([\alpha]\varphi \vee [\alpha']\varphi)$
Learning	$\langle L_B\alpha\rangle\top \leftrightarrow \mathsf{pre}(L_B\alpha)$
Local choice	$[\alpha \ ! \ \alpha']\varphi \leftrightarrow [\alpha]\varphi$
Actions instances	$[\alpha]\varphi \leftrightarrow \bigwedge_{\beta \in T(\alpha)}[\beta]\varphi$
Atomic permanence	$[\alpha]p \leftrightarrow (\mathsf{pre}(\alpha) \rightarrow p)$
C-knowledge induction	$(\varphi \wedge C_B(\varphi \rightarrow E_B\varphi)) \rightarrow C_B\varphi$
Action use	$[\alpha]K_n\varphi \leftrightarrow (\mathsf{pre}(\alpha) \rightarrow K_n \bigwedge_{\alpha' \sim_n \alpha}[\alpha']\varphi)$
Modus Ponens	*if* φ *and* $\varphi \rightarrow \psi$, *then* ψ
Knowledge necessitation	*if* φ, *then* $K_n\varphi$
Action facilitation	*if* $\varphi \rightarrow \psi$, *then* $[\alpha]\varphi \rightarrow [\alpha]\psi$
Action induction	*if*: *for all* β *such that* $\alpha \sim_B \beta$, *there is a* χ_β*such that* $\chi_\beta \rightarrow [\beta]\varphi$ *and such that* $\beta \sim_n \alpha'$ *implies* $(\chi_\beta \wedge \mathsf{pre}(\beta)) \rightarrow E_B\chi_{\alpha'}$, *then* : $\chi_\alpha \rightarrow [\alpha]C_B\varphi$

A formula φ *is deducable, abbreviated as* $\vdash \varphi$, *iff there exists a finite sequence of formulas such that each formula is either an instantiation of one of the axioms above, or if it is obtained by applying one of the rules above to formulas that appear earlier in the sequence.*

Below we will give a few examples of proofs using this system. In these proofs every formula in the sequence is given a number and written on one line. At the end of the line is given the axiom that the formula is an instantiation of or the rule that was applied to obtain it, including the lines of the formulas that the rule was applied to. If one formula follows by simple propositional reasoning, we write PC, accompanied by the lines of formulas.

Example 1.32 *We show that* $\vdash [L_1?p]K_1p$. *To see that this is the case we first determine the precondition of* $L_1?p$. *It follows from the definition* *it is equal to p. Below we will write $\mathsf{pre}(L_1?p)$ for this to make clear which rules are applied.*

1	$\mathsf{pre}(L_1?p) \rightarrow p$	*PC*
2	$[L_1?p]p \leftrightarrow (\mathsf{pre}(L_1?p) \rightarrow p)$	*Atomic permanence*
3	$[L_1?p]p$	*PC 1,2*
4	$K_1[L_1?p]p$	*Knowledge necessitation 3*
5	$\mathsf{pre}(L_1?p) \rightarrow K_1[L_1?p]p$	*PC 4*
6	$[L_1?p]K_1p \leftrightarrow (\mathsf{pre}(L_1?p) \rightarrow K_1[L_1?p]p)$	*Action use*
7	$[L_1?p]K_1p$	*PC 5,6*

EXAMPLE 1.33 *We now show that $\vdash [L_{12}?p]C_{12}p$. This example shows how the action induction rule can be applied. Again $\mathsf{pre}(L_{12}?p) = p$.*

1	$\mathsf{pre}(L_{12}?p) \rightarrow p$	*PC*
2	$[L_{12}?p]p \leftrightarrow (\mathsf{pre}(L_{12}?p) \rightarrow p)$	*Atomic permanence*
3	$[L_{12}?p]p$	*PC 1,2*
4	$\top \rightarrow [L_{12}?p]p$	*PC 3*
5	$\top$	*PC*
6	$K_1\top$	*Knowledge necessitation 5*
7	$K_2\top$	*Knowledge necessitation 5*
8	$((\top \wedge \mathsf{pre}(L_{12}?p)) \rightarrow K_1\top) \wedge ((\top \wedge \mathsf{pre}(L_{12}?p)) \rightarrow K_2\top)$	*PC 6,7*
9	$\top \rightarrow [L_{12}?p]C_{12}p$	*Action induction 4,8*
10	$[L_{12}?p]C_{12}p$	*PC 9*

One might have expected a distribution axiom for $[\alpha]$, but this is not sound. Such an axiom is also unsound in the logic presented in [Pel87], for the same reason: the interpretation of actions are relations between epistemic states and sets of epistemic states. The modality $[\alpha]$ corresponds to a $\forall\exists$ quantifier and distribution does not hold for that.

We do have a weaker form of distribution in the form of the action facilitation rule. This is all we need in the completeness proof.

The local choice axiom *suggests* that in arbitrary actions, subactions $\alpha\,!\,\alpha$ can be substituted for α. However, this is not the case, because such substitutions can*not* be performed within the scope of an L_B operator.

The following are used in the completeness proof for formulas of the form $[\alpha]C_B\varphi$ and in the soundness proof of the action induction rule.

DEFINITION 1.34 (WITNESS PATH) *A witness path from $w \in M$ for $\langle\alpha\rangle\hat{C}_B\psi$ (see [BMS02]) is a path*

$$w = w_0 \sim_{n_1} w_1 \sim_{n_2} \cdots \sim_{n_k} w_k = v$$

in M, such that $k \geq 0$, $n_i \in N$ and there are actions α_i such that

$$\alpha = \alpha_0 \sim_{n_1} \alpha_1 \sim_{n_2} \cdots \sim_{n_k} \alpha_k = \beta$$

and such that for $0 \leq i \leq k$, $(M, w_i) \models \langle\alpha_i\rangle\top$, and $(M, w_k) \models \langle\alpha_k\rangle\psi$.

LEMMA 1.35
$M, w \models \langle\alpha\rangle\hat{C}_B\psi$ iff [there is a witness path for $\langle\alpha\rangle\hat{C}_B\psi$ from w].

Proof () Note that the right entails that there is a $\beta \sim_B \alpha$ such that $M, w \models \hat{C}_B\langle\beta\rangle\psi$. We reformulate the lemma as one about an arbitrary natural number k and then prove the lemma by induction on k, first the implication to the right, then the one to the left. Write $\sim_B^k$ for a $\sim_B$-path of length k:

Let $\alpha \in \mathcal{L}_N^{\mathsf{act}}$, $(M, w) \in \bullet\mathcal{S}5_N$, $\psi \in \mathcal{L}_N$. For all k: [there is an S_w such that $(M, w)[\![\alpha]\!]S_w$ and for all $(M', w') \in S_w$ there is a $v' \sim_B^k w'$ with $M', v' \models \psi$] iff [there is a witness path of length k for $\langle\alpha\rangle\hat{C}_B\psi$ from w].

First, note that the part '*there is an S_w such that $(M, w)[\![\alpha]\!]S_w$ and for all $(M', w') \in S_w$:*' might as well read '*there is an (M', w') resulting from executing α in (M, w):*.' The more complex formulation is in accordance with the semantics of actions.

'$\Rightarrow$'. ($k = 0$) Let $(M', w') \in S_w$. Given $v' \sim_B^0 w'$ with $M', v' \models \psi$. Choose $\beta = \alpha$ and $v = w$ and $S_v = S_w$. The required holds. ($k = k + 1$) Let $(M', w') \in S_w$. Given $v' \sim_B^{k+1} w'$ with $M', v' \models \psi$. Let n and x' be such that $v' \sim_B^{k+1} w' = v' \sim_n x' \sim_B^k w'$. Choose $\alpha' \sim_n \alpha$ and $x \sim_n w$ such that $(M, x)[\![\alpha']\!]S_x$ and $(M', x') \in S_x$. Now by induction, there is a witness path of length k for $\langle\alpha'\rangle\hat{C}_B\psi$ from x. Let $x \sim_B^k v$ be that path and $\alpha' \sim_B^k \beta$ be the corresponding chain of actions. Then $w \sim_n x \sim_B^k v$ is the required witness path from w, with $\alpha \sim_n \alpha' \sim_B^k \beta$ the required chain of actions.

'$\Leftarrow$'. ($k = 0$) Given $v = w$ and $\alpha = \beta$ (path length 0). We then have $M, w \models \langle\alpha\rangle\psi$. Therefore, trivially, $M, w \models \langle\alpha\rangle\hat{C}\psi$. Therefore, there is (by the definition of $\models$) an S_w where ψ holds throughout, as required for 'zero length'. ($k = k + 1$) Let $(M', w') \in S_w$. Given v and β such that $w \sim_B^{k+1} v$ and $\alpha \sim_B^{k+1} \beta$ as required. Let the first steps in those chains be $w \sim_n x$ and $\alpha \sim_n \alpha'$, respectively. By induction, there is an S_x such that for all $(M', x') \in S_x$, there is a $v' \sim_B^k x'$ with $M', v' \models \psi$. We now complete this to a path of length $k + 1$ as follows. As $w \sim_n x$ and $\alpha \sim_n \alpha'$, by Corollary 1.27 we may choose S_w such that $(M, w)[\![\alpha]\!]S_w$ and $S_w \sim_n S_x$. We can now complete our required path by choosing $(M', w') \in S_w$ such that $w' \sim_n x'$ (and note that - as required - *for each* $s \in S_w$ there must be such a corresponding epistemic state $s' \in S_x$, because of the definition of $\sim_n$ between sets of epistemic states; therefore, (M', w') may be considered arbitrary). From $(M, w)[\![\alpha]\!]S_w$, $(M', w') \in S_w$ arbitrary, $w' \sim_n x' \sim_B^k v'$, and $M', v' \models \psi$, follows: $M, w \models \langle\alpha\rangle\hat{C}_B\psi$. $\dashv$

We are now ready to establish soundness of the proof system, with which we close the section. The following section established the completeness of the proof system.

THEOREM 1.36 (SOUNDNESS) *For all* $\varphi \in \mathcal{L}_N(P)$

$$\vdash \varphi \;\Rightarrow\; \models \varphi$$

Proof () By induction on the length of the proof. We omit the familiar purely epistemic cases, for that see, e.g., [MvdH95]. Throughout the proof, let $(M, w) \in \mathcal{S}5_N(P)$ be arbitrary. Further, assume all introduced sets of epistemic states to be subsets of some proper $\bullet\mathcal{S}5_B$. We remind the reader that '$S \models \varphi$' means '$\forall s \in S : s \models \varphi$.'

Case 'Test'.
Note that in $[?\varphi]\psi \leftrightarrow (\varphi \rightarrow \psi)$, ψ must be a purely propositional formula ($\psi \in \mathcal{L}_\emptyset(P)$), because $gr((M, w)[\![?\varphi]\!]) = \emptyset$. The truth of propositional formulas is unaffected by action execution.

Suppose $M, w \models [?\varphi]\psi$ and $M, w \models \varphi$. Then $(M, w)[\![?\varphi]\!] \models \psi$. Because $\psi \in \mathcal{L}_\emptyset(P)$, also $M, w \models \psi$. Therefore $M, w \models \varphi \rightarrow \psi$.

Suppose $M, w \models \varphi \rightarrow \psi$. If $M, w \not\models \varphi$, then $(M, w) \models [?\varphi]\psi$ trivially holds. Otherwise, because $M, w \models \varphi$, $(M, w)[\![?\varphi]\!]$ exists; and from $M, w \models \varphi$ and $M, w \models \varphi \rightarrow \psi$ follows $M, w \models \psi$. Because $\psi \in \mathcal{L}_\emptyset(P)$, also $(M, w)[\![?\varphi]\!] \models \psi$. Therefore, as well, $(M, w) \models [?\varphi]\psi$.

The axioms 'Sequential composition', 'Nondeterministic choice', and 'Concurrency' are intuitively more appealing in their dual form: $\langle\alpha\ ;\ \alpha'\rangle\varphi \leftrightarrow$

$\langle\alpha\rangle\langle\alpha'\rangle\varphi$, $\langle\alpha\cup\alpha'\rangle\varphi\leftrightarrow(\langle\alpha\rangle\varphi\vee\langle\alpha'\rangle\varphi)$, and $\langle\alpha\cap\alpha'\rangle\varphi\leftrightarrow(\langle\alpha\rangle\varphi\wedge\langle\alpha'\rangle\varphi)$. We therefore show the validity of those.

Case 'Sequential composition'.
$M,w\models\langle\alpha\ ;\ \alpha'\rangle\varphi$
$\Leftrightarrow$
$\exists S:(M,w)[\![\alpha\ ;\ \alpha']\!]S$ and $\forall s\in S:s\models\varphi$
$\Leftrightarrow$
$\exists S,S':(M,w)[\![\alpha]\!]S'$ and $\forall s'\in S',\exists S_{s'}:s'[\![\alpha]\!]S_{s'}$ and $S=\bigcup_{s'\in S'}S_{s'}$ and $\forall s\in S:s\models\varphi$
$\Leftrightarrow$
$\exists S,S':(M,w)[\![\alpha]\!]S'$ and $\forall s'\in S',\exists S_{s'}\subseteq S:s'[\![\alpha]\!]S_{s'}$ and $\forall s''\in S_{s'}:s''\models\varphi$ and $S=\bigcup_{s'\in S'}S_{s'}$
$\Leftrightarrow$ take $S=\bigcup_{s'\in S'}S_{s'}$
$\exists S':(M,w)[\![\alpha]\!]S'$ and $\forall s'\in S':s'\models\langle\alpha'\rangle\varphi$
$\Leftrightarrow$
$M,w\models\langle\alpha\rangle\langle\alpha'\rangle\varphi$

Case 'Nondeterministic choice'.
$M,w\models\langle\alpha\cup\alpha'\rangle\varphi$
$\Leftrightarrow$
$\exists S:(M,w)[\![\alpha\cup\alpha']\!]S$ and $\forall s\in S:s\models\varphi$
$\Leftrightarrow$
$\exists S:[\ (M,w)[\![\alpha]\!]S$ or $(M,w)[\![\alpha']\!]S\ ]$ and $\forall s\in S:s\models\varphi$
$\Leftrightarrow$
$\exists S:[\ (M,w)[\![\alpha]\!]S$ and $\forall s\in S:s\models\varphi\ ]$ or $[\ (M,w)[\![\alpha']\!]S$ and $\forall s\in S:s\models\varphi\ ]$
$\Leftrightarrow$
$[\ \exists S:(M,w)[\![\alpha]\!]S$ and $\forall s\in S:s\models\varphi\ ]$ or $[\ \exists S:(M,w)[\![\alpha']\!]S$ and $\forall s\in S:s\models\varphi\ ]$
$\Leftrightarrow$
$M,w\models\langle\alpha\rangle\varphi$ or $M,w\models\langle\alpha'\rangle\varphi$
$\Leftrightarrow$
$M,w\models\langle\alpha\rangle\varphi\vee\langle\alpha'\rangle\varphi$

Case 'Concurrency'.
$M,w\models\langle\alpha\cap\alpha'\rangle\varphi$
$\Leftrightarrow$
$\exists S:(M,w)[\![\alpha\cap\alpha']\!]S$ and $\forall s\in S:s\models\varphi$
$\Leftrightarrow$
$\exists S_1,S_2:(M,w)[\![\alpha]\!]S_1$ and $(M,w)[\![\alpha']\!]S_2$ and $\forall s\in S_1,S_2:s\models\varphi$
$\Leftrightarrow$

$M, w \models \langle\alpha\rangle\varphi$ and $M, w \models \langle\alpha'\rangle\varphi$
$\Leftrightarrow$
$M, w \models \langle\alpha\rangle\varphi \wedge \langle\alpha'\rangle\varphi$

Case 'Learning'.
This follows immediately from Lemma 1.30, for case $L_B\alpha$. Incidentally, all other cases of Lemma 1.30 are derivable and therefore not listed as axioms, see Lemma 1.48, in the continuation.

Case 'Local choice'.
Trivial, as $[\![\alpha \ ! \ \alpha']\!] = [\![\alpha]\!]$.

Case 'Action instances'.
This follows directly from Proposition 1.17 (an action is equivalent to nondeterministic choice between its instances), and repeated application of axiom 'Nondeterministic choice', that has already been proved sound.

Case 'Atomic permanence'.
Trivial.

Case 'Action use'.
We prove the validity of the dual form of action use which is $\langle\alpha\rangle\hat{K}_n\varphi \leftrightarrow (\mathsf{pre}(\alpha) \wedge \hat{K}_n \bigvee_{\alpha' \sim_n \alpha}\langle\alpha'\rangle\varphi)$. Note that $M, w \models \langle\alpha\rangle\hat{K}_n\varphi$ can be rewritten as

$$\exists S : (M, w)[\![\alpha]\!]S \text{ and } S \models \hat{K}_n\varphi \qquad (i)$$

Next, $M, w \models (\mathsf{pre}(\alpha) \wedge \hat{K}_n \bigvee_{\alpha' \sim_n \alpha}\langle\alpha'\rangle\varphi)$ can be rewritten as

$$M, w \models \mathsf{pre}(\alpha) \ \& \ \exists v \sim_n w, \exists\alpha' \sim_n \alpha, \exists S' : (M, v)[\![\alpha']\!]S' \ \& \ S' \models \varphi \qquad (ii)$$

Using Corollary 1.27, the executability of α in (ii) guarantees the existence of an S such that: $(M, w)[\![\alpha]\!]S$ and $S \sim_n S'$. From this follows by definition of K_n that $S \models \hat{K}_n\varphi$: i.e., (i). Vice versa, given (i), an assumed S' such that $S' \sim_n S$ and $S' \models \varphi$, can only be established by executing some alternative α' to α in a v with $v \sim_n w$ (an agent cannot 'forget' distinctions). We now have $[\![\alpha']\!] \sim_n [\![\alpha]\!]$ by definition, and choosing α' from $[\alpha]_{=_T}$ we get $\alpha \sim_n \alpha'$ as required: i.e., (ii).

Case 'Action facilitation'.
As in other cases, we prove a dual version, namely: '*from* $\varphi \rightarrow \psi$ *follows* $\langle\alpha\rangle\varphi \rightarrow \langle\alpha\rangle\psi$'. Assume $\models \varphi \rightarrow \psi$. Suppose $M, w \models \langle\alpha\rangle\varphi$. Then there

is an S such that $(M,w)[\![\alpha]\!]S$ and $S \models \varphi$. As we may assume $\varphi \to \psi$ to be valid, we have $S \models \varphi \to \psi$. From $S \models \varphi$ and $S \models \varphi \to \psi$ follows $S \models \psi$. From $(M,w)[\![\alpha]\!]S$ and $S \models \psi$ follows $M,w \models \langle\alpha\rangle\psi$. Winding up: $M,w \models \langle\alpha\rangle\varphi \to \langle\alpha\rangle\psi$.

Case 'Action induction'.
The soundness of this rule is proved along the lines of [BMS02], using Corollary 1.35. ⊣

4. Completeness

The completeness proof is based on [BMS02], [FHMV95], and [Pel87]. We quite closely follow the structure of the proof in [BMS02]. The main difficulty in the proof is the truth lemma, which is proven by induction on formulas. We show that every formula is provably equivalent to a formula in a sublanguage of the full language. The induction follows the structure of the formulas in the sublanguage. We first define the translation to the sublanguage.

Definition 1.37 (Translation)
Define the following function $f : \mathcal{L}_N(P) \cup \mathcal{L}_N^{\mathsf{act}}(P) \to \mathcal{L}_N(P) \cup \mathcal{L}_N^{\mathsf{act}}(P)$*:*

$$\begin{array}{lcl}
f(p) & = & p \\
f(\neg\varphi) & = & \neg f(\varphi) \\
f(\varphi \wedge \psi) & = & f(\varphi) \wedge f(\psi) \\
f(K_n\varphi) & = & K_n f(\varphi) \\
f(C_B\varphi) & = & C_B f(\varphi)
\end{array}$$

$$\begin{array}{lcl}
f([?\varphi]\psi) & = & f(\varphi) \to f(\psi) \\
f([L_B\alpha]p) & = & f(\mathsf{pre}(\alpha)) \to p \\
f([L_B\alpha]\neg\varphi) & = & \bigvee_{\beta \in T(L_B\alpha)} \neg f([\beta]\varphi) \\
f([L_B\alpha](\varphi \wedge \psi)) & = & f([L_B\alpha]\varphi) \wedge f([L_B\alpha]\psi) \\
f([L_B\alpha]K_n\varphi) & = & f(\mathsf{pre}(\alpha)) \to K_n \bigwedge_{\beta \sim_n L_B\alpha} f([\beta]\varphi) \\
f([L_B\alpha]C_B\varphi) & = & [L_B f(\alpha)]C_B f(\varphi) \\
f([L_B\alpha][\beta]\varphi) & = & f([L_B\alpha]f([\beta]\varphi)) \\
f([\alpha \;!\; \beta]\psi) & = & f([\alpha]\psi) \\
f([\alpha \;;\; \beta]\psi) & = & f([\alpha]f([\beta]\psi)) \\
f([\alpha \cup \beta]\psi) & = & f([\alpha]\psi) \wedge f([\beta]\psi) \\
f([\alpha \cap \beta]\psi) & = & f([\alpha]\psi) \vee f([\beta]\psi)
\end{array}$$

$$\begin{array}{lcl} f(?\varphi) & = & ?f(\varphi) \\ f(\alpha\ ;\ \beta) & = & f(\alpha)\ ;\ f(\beta) \\ f(\alpha \cup \beta) & = & f(\alpha) \cup f(\beta) \\ f(\alpha \cap \beta) & = & f(\alpha) \cap f(\beta) \\ f(\alpha\ !\ \beta) & = & f(\alpha)\ !\ f(\beta) \\ f(L_B\alpha) & = & L_B f(\alpha) \end{array}$$

LEMMA 1.38 *Given a formula $\varphi \in \mathcal{L}_N$, we have that $f(\varphi) \in \mathcal{L}_N^f$, where $\mathcal{L}_N^f$ the following BNF:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_n\varphi \mid C_B\varphi \mid [L_B\alpha]C_B\varphi$$

where all ψ in $L_B\alpha$ are also in $\mathcal{L}_N^f$.

Proof () By induction on $\varphi \in \mathcal{L}_N(P)$. ⊣

The next lemma shows, that if we can prove by induction that something holds for every formula $\varphi \in \mathcal{L}_N^f$, then it also applies to every $\varphi \in \mathcal{L}_N$.

LEMMA 1.39 (WELL-FOUNDED ORDER) There exists a well-founded order $<$ on the language $\mathcal{L}_N$ with the following properties:

1 $\psi < \varphi$, for any subformula ψ of φ

2 $<$ is transitive

3 $f(\varphi) \leq \varphi$

Proof () The proof is very much like a similar proof given in the appendix of [BMS02], where a theorem about lexicographic path orders is used. We apply the same technique but there is a key difference. A formula of the form $[\alpha]\varphi$ can be viewed as a function $[\cdot]\cdot$ on α and φ. The order of the arguments is important for a lexicographic path order. In [BMS02] the order of the arguments is switched in the definition of the lexicographic path order. Here we do not switch the arguments.

Let Σ be the signature of $\mathcal{L}_N$, where atomic propositions are viewed as 0-ary functions, where negations, individual epistemic operators, common knowledge operators, tests, and learn operators are viewed as unary functions, and where conjunctions, sequential composition, non-deterministic choice, concurrent knowledge action, and the $[\cdot]\cdot$ operators are viewed as binary operators.

Now consider the lexicographical path order $<$ that is induced by fixing an order on these such that $[\cdot]\cdot$ is the greatest and the rest of the signature is unordered. It follows directly from the theorem cited in [BMS02] that:

1 $<$ is transitive.

2 $<$ has the subterm property.

3 $<$ is monotonic.

4 $<$ is well-founded.

We now prove that $f(\varphi) \leq \varphi$ by induction on φ. The cases for atoms, negation, conjunction, individual epistemic operators, and common knowledge operators is straightforward.

Suppose φ is of the form $[?\psi]\chi$. Now we can show that $f(\psi) \rightarrow f(\chi) < [?\psi]\chi$, because $[\cdot]\cdot$ is greater than $\rightarrow$ and because $f(\psi) \leq \psi$ and $f(\chi) \leq \chi$ by the induction hypothesis. Now ψ and χ are both subterms of $[?\psi]\chi$. Therefore $f(\psi) \rightarrow f(\chi) < [?\psi]\chi$.

Suppose φ is of the form $[\alpha\ ;\ \beta]\psi$. Now we can show that $f([\alpha]f([\beta]\psi))$ $< [\alpha\ ;\ \beta]\psi$, because $[\alpha][\beta]\psi < [\alpha\ ;\ \beta]\psi$. Note that $\alpha < \alpha\ ;\ \beta$, because of the subterm property. For the same reason we have $\alpha < [\alpha\ ;\ \beta]\psi$. Now we need to show that $[\beta]\psi < [\alpha\ ;\ \beta]\psi$. And again this is induced by the subterm property. The cases for nondeterministic choice, concurrent execution, and local choice are analogous.

Now we turn to formulas of the form $[L_B\alpha]\psi$. If ψ is an atom p, note that $\mathsf{pre}(\alpha) < [\alpha]\psi$ for all actions α, and p is obviously a subterm of $[L_B\alpha]p$.

If ψ is a negation, we again use the observation that $[\cdot]\cdot$ is greater than disjunctions, and negations. Then we can simply apply the induction hypothesis. The case for conjunctions and individual epistemic operators is analogous.

In case ψ is of the form $C_B\chi$, we observe that the induction hypothesis immediately implies that $f(\alpha) \leq \alpha$. This together with applying the induction hypothesis to χ show that $[f(L_B\alpha)]C_Bf(\chi) \leq [L_B\alpha]C_B\chi$.

The case where ψ is of the form $[\beta]\chi$, follows immediately from the induction hypothesis. ⊣

Lemma 1.40 *We have both (a) and (b):*
(a) $\vdash f(\varphi) \leftrightarrow \varphi$
(b) $\models f(\varphi) \leftrightarrow \varphi$

Proof () (a) follows immediately from the axioms given in 1.31 whereas (b) follows from (a) and the soundness theorem 1.36. ⊣

We can now start constructing the canonical model. Because logics with reflexive transitive closure operators are generally not compact we need to construct a finite canonical model for every formula. That means we only look at maximally consistent sets with respect to some finite set of sentences. This set of sentences is called the closure.

DEFINITION 1.41 (CLOSURE) *Let $\varphi \in \mathcal{L}^f_N$. The* closure *of φ is the minimal set $\mathcal{C}l(\varphi) \subseteq \mathcal{L}^f_N$ such that*

1 $\varphi \in \mathcal{C}l(\varphi)$.

2 If $\psi \in \mathcal{C}l(\varphi)$ and χ is a subformula of ψ, then $\chi \in \mathcal{C}l(\varphi)$.

3 If $\psi \in \mathcal{C}l(\varphi)$ and ψ itself is not a negation, then $\neg\psi \in \mathcal{C}l(\varphi)$

4 If $C_B\psi \in \mathcal{C}l(\varphi)$, then $K_n C_B\psi \in \mathcal{C}l(\varphi)$, for all $a \in B$

5 If $[\alpha]C_B\psi \in \mathcal{C}l(\varphi)$, then for all β and all $n \in B$ such that $\alpha \sim_n \beta$: $K_n[\beta]C_B\psi, [\beta]\psi \in \mathcal{C}l(\varphi)$.

LEMMA 1.42 *For any formula $\varphi \in \mathcal{L}^f_A$, the closure of φ is finite.*

Proof () One can first apply the clauses 1 and 2. This yields a finite set. To this set, the clauses 4 and 5 can be applied, which preserves finiteness (by Lemma 1.24 we know that there are only finitely many β such that $\alpha \sim_n \beta$). Finally, applying clause 3 to this set at most doubles its size. ⊣

From now on, we will often write Φ for $\mathcal{C}l(\varphi)$.

DEFINITION 1.43 (MAXIMALLY CONSISTENT IN Φ) *A finite set of sentences Γ such that $\Gamma \subseteq \Phi$ is maximally consistent in Φ iff:*

1 Γ is consistent, i.e. $\not\vdash \neg(\bigwedge_{\psi\in\Gamma}\psi)$.

2 There is no $\Gamma' \subset \Phi$, such that $\Gamma' \supset \Gamma$ and Γ' is consistent.

LEMMA 1.44 (LINDENBÄUMCHEN) *Let Φ be the closure of a consistent $\varphi \in \mathcal{L}^f_N$. If $\Gamma \subseteq \Phi$ is consistent in Φ, then there is a set $\Gamma' \supseteq \Gamma$ which is maximally consistent in Φ.*

Proof () As Φ is finite, the members of Φ can be enumerated. Let us suppose $\#(\Phi) = k$ and that φ_i $(1 \leq i \leq k)$is the i-th formula sentence of this enumeration. Now define Γ_i $(0 \leq i \leq k)$ as follows:

$$\begin{array}{lcl} \Gamma_0 & = & \Gamma \\ \Gamma_{i+1} & = & \begin{cases} \Gamma_i & \text{if } \Gamma_i \cup \{\varphi_{i+1}\} \text{ is inconsistent} \\ \Gamma_i \cup \{\varphi_{i+1}\} & \text{otherwise} \end{cases} \end{array}$$

It is easily seen that Γ_k is maximally consistent in Φ. We can think of $\Gamma \in W_\Phi$ both as a set of formulas of Φ and as its conjunction: if we mean the latter we write $\underline{\Gamma} = \bigwedge_{\varphi\in\Gamma}\gamma$.

$\dashv$

DEFINITION 1.45 (Φ-CANONICAL MODEL) $M^\Phi = (W^\Phi, R^\Phi, V^\Phi)$

- $W^\Phi = \{\Gamma \subseteq \Phi : \Gamma$ *is maximally* Φ*-consistent*$\}$
- $\Gamma R^\Phi_n \Delta$ *iff* $\{\psi \in \Phi \mid K_n\psi \in \Gamma\} = \{\psi \in \Phi \mid K_n\psi \in \Gamma\}$ *for all* $n \in N$
- $V^\Phi(p) = \{\Gamma : p \in \Gamma\}$

Note that M^Φ is finite: it contains at most $2^{|\Phi|}$ elements. Moreover, note that it is a model in S_B, where B is the group of agents that occur in φ, because the accessibility relations R^Φ_n are all equivalence relations.

DEFINITION 1.46 (GOOD PATH) *A good path from* $\Gamma \in M^\Phi$ *for* $\langle\alpha\rangle\hat{C}_B\psi$ *is a path in* M^Φ

$$\Gamma = \Gamma_0 \sim_{n_1} \Gamma_1 \sim_{n_2} \cdots \sim_{n_k} \Gamma_k$$

such that $k \geq 0$*,* $n_i \in N$ *and there are actions* α_i *such that*

$$\alpha = \alpha_0 \sim_{n_1} \alpha_1 \sim_{n_2} \cdots \sim_{n_k} \alpha_k$$

such that $\langle\alpha_i\rangle\top \in \Gamma_i$ *(*$0 \leq i \leq k$*) and* $\langle\alpha_k\rangle\psi \in \Gamma_k$*.*

The relationship of a 'good path' to the semantically motivated 'witness path' (Definition 1.34) will be obvious.

LEMMA 1.47 *Suppose* $[\alpha]C_B\chi \in \Phi$*. Then: if there is a good path from* Γ_0 *for* $\langle\alpha\rangle\hat{C}_B\neg\psi$*, then* $\langle\alpha\rangle\hat{C}_B\neg\psi \in \Gamma_0$*.*

Proof () By induction on the length k of the path. If $k = 0$, then, since $\alpha_0 = \alpha$, we have $\langle\alpha\rangle\neg\chi \in \Gamma_0$. If $\langle\alpha\rangle\hat{C}_B\neg\psi \notin \Gamma_0$ then, we have $[\alpha]C_B\chi \in \Gamma_0$, and hence, by clause 5 of definition 1.41, and the fact that $\vdash [\alpha]C_B\chi \to [\alpha]\chi$, we have that $[\alpha]\chi \in \Gamma_0$; a contradiction.

Now suppose we have proven the result for k, and suppose that there is a good path from Γ_0 for $\langle\alpha\rangle\hat{C}_B\neg\psi$ of length $k+1$. This gives us a good path of length k from Γ_1 for $\langle\alpha_1\rangle\hat{C}_B\neg\chi$. We also have $[\alpha_1]C_B\chi \in \Phi$, and hence, by using the induction hypothesis, we have $\langle\alpha_1\rangle\hat{C}_B\neg\chi \in \Gamma_1$. Now suppose $\langle\alpha\rangle\hat{C}_B\neg\psi \notin \Gamma_0$, then $[\alpha]C_B\chi \in \Gamma_0$. By lemma the action use axiom we know that $\Gamma_0 \vdash [\alpha]C_B\chi \wedge \mathsf{pre}(\alpha) \to K_n[\alpha_1]C_B\chi$ and hence $K_n[\alpha_1]C_B\chi \in \Gamma_0$. $\dashv$

Lemma 1.48 *The precondition of an action α is provably equivalent with $\langle\alpha\rangle\top$.*

Proof () By induction on α. The case for test follows from the test axiom. Using the induction hypothesis the cases for sequential composition, non-deterministic choice, concurrent action, local choice follow directly using the appropriate axioms. The case for the learn operator follows from the Learning axiom. $\dashv$

Lemma 1.49 *If $\langle\alpha\rangle\hat{C}_B\neg\psi \in \Gamma$ then there is a good path from Γ for $\langle\alpha\rangle\hat{C}_B\neg\psi$*

Proof () Suppose $\langle\alpha\rangle\hat{C}_B\neg\psi \in \Gamma$. For each β such that $\alpha \sim_B \beta$, let S_β be the (finite) set of all $\Gamma \in M_\Phi$ such that there is *no* good path from Γ for $\langle\beta\rangle\hat{C}_B\psi$. We need to see that $\Gamma \notin S_\alpha$. Suppose toward a contradiction that $\Gamma \in S_\alpha$. Let

$$\chi_\beta = \bigvee_{\Delta \in S_\beta} \underline{\Delta}$$

Note that $\neg\chi_\beta$ is logically equivalent to $\bigvee_{\Delta' \notin S_\beta} \underline{\Delta'}$. Since we assumed $\Gamma \in S_\alpha$, we have $\vdash \underline{\Gamma} \to \chi_\alpha$.

We first claim that $\chi_\beta \wedge \langle\beta\rangle\chi$ is inconsistent. Otherwise, there would be a $\Delta \in S_\beta$ such that $\vdash \underline{\Delta} \to \chi_\beta \wedge \langle\beta\rangle\chi$. Note that $\vdash \langle\beta\rangle\chi \to \langle\beta\rangle\top$, and, by Lemma 1.48, $\vdash \langle\beta\rangle\top \leftrightarrow \mathsf{pre}(\beta)$. But then the one point path Δ is a good path from Δ for $\langle\beta\rangle\hat{C}_B\psi$. Thus $\Delta \notin S_\beta$, and this is a contradiction. So indeed, $\chi_\beta \wedge \langle\beta\rangle\chi$ is inconsistent. Therefore, $\vdash \chi_\beta \to [\beta]\neg\chi$.

We will need the following standard claim: If $\underline{\Delta} \wedge \hat{K}_n\underline{\Delta'}$ is consistent, then $\Delta R^\Phi_n \Delta'$. To see this, suppose that not $\Delta R^\Phi_n \Delta'$. Then there must be a formula $K_n\chi$, such that $K_n\chi \in \Delta$ and $K_n\chi \notin \Delta'$. Therefore, $\Delta \vdash K_nK_n\chi$ and $\hat{K}_n\underline{\Delta'} \vdash \hat{K}_n\neg K_n\chi$. This implies that $\underline{\Delta} \wedge \hat{K}_n\underline{\Delta'}$ is inconsistent.

We next show that for all $n \in B$, and all β and μ such that $\beta \sim_B \mu$, the formula $\chi_\beta \wedge \mathsf{pre}(\beta) \wedge \hat{K}_n\neg\chi_\mu$ is inconsistent. Suppose that it is consistent. Because χ_β is a disjunction, we can pick a disjunct $\underline{\Theta}$ for which $\underline{\Theta} \wedge \mathsf{pre}(\beta) \wedge \hat{K}_n\neg\chi_\mu$ is consistent. Note that $\Theta \in S_\beta$. Since Θ is maximally consistent in Φ, we have $\mathsf{pre}(\beta) \in \Theta$. Thus, we now have that $\underline{\Theta} \wedge \hat{K}_n\neg\chi_\mu$ is consistent. Note that $\neg\chi_\mu$ is logically equivalent to $\bigvee_{\Theta' \notin S_\mu} \underline{\Theta'}$. As $\hat{K}_n$ distributes over disjunction, we can now pick a disjunct $\hat{K}_n\underline{\Theta'}$ such that $\underline{\Theta} \wedge \hat{K}_n\underline{\Theta'}$ is consistent. Therefore, by the claim above, $\Theta \sim_n \Theta'$. Since $\mathsf{pre}(\beta) \in \Theta$ and $\beta \sim_n \mu$ and there is a good path for $\langle\mu\rangle\hat{C}_B\psi$ from Θ', there is a good path for $\langle\alpha\rangle\hat{C}_B\neg\psi$ from Θ. This contradicts that $\Theta \in S_\beta$. Therefore, $\vdash \chi_\beta \wedge \mathsf{pre}(\beta) \to K_n\chi_\mu$.

Now we can apply the Action induction rule to show that $\Gamma \vdash \chi_\alpha \rightarrow [\alpha]C_B\neg\chi$ and $\Gamma \vdash \chi_\alpha$. Therefore, $[\alpha]C_B\neg\chi \in \Gamma$. This contradicts our initial assumption, therefore, there *is* a good path for $\langle\alpha\rangle\hat{C}_B\neg\psi$ from Γ. ⊣

Lemma 1.50 (Truth Lemma) *If $\Gamma \in W_\Phi$, then for all $\psi \in \Phi$ it holds that $(M_\Phi, \Gamma) \models \psi$ iff $\psi \in \Gamma$.*

Proof () By induction on ψ. Suppose $\psi \in \Phi$. For atoms p, negations, conjunctions, individual epistemic operators and common knowledge operators we refer to [FHMV95].

Let ψ be of the form $[\alpha]C_B\chi$. We have the following equivalences:

$$\begin{array}{lr}
(M_\Phi, \Gamma) \not\models [\alpha]C_B\chi & \\
\Leftrightarrow & \{\text{Definition of } \langle\alpha\rangle\} \\
(M^\Phi, \Gamma) \models \langle\alpha\rangle\hat{C}_B\neg\chi & \\
\Leftrightarrow & \{\text{Lemma 1.35}\} \\
\text{There exists a witness path for } \langle\alpha\rangle\hat{C}_B\neg\chi \text{ from } \Gamma & \\
\Leftrightarrow & \{\text{Induction Hypothesis}\} \\
\text{There exists a good path for } \langle\alpha\rangle\hat{C}_B\neg\chi \text{ from } \Gamma & \\
\Leftrightarrow & \{\text{Lemma's 1.49 and 1.47}\} \\
\langle\alpha\rangle\hat{C}_B\neg\chi \in \Gamma & \\
\Leftrightarrow & \{\Gamma \text{ is maximally consistent in } \Phi\} \\
[\alpha]C_B\chi \notin \Gamma &
\end{array}$$

⊣

Theorem 1.51 (Completeness) *If $\not\vdash \varphi$, then there is a model (M, w) such that $(M, w) \not\models \varphi$*

Proof () Suppose $\not\vdash \varphi$. Then, $\neg\varphi$ is consistent. Take $f(\neg\varphi)$. Note that $\neg\varphi$ and $f(\neg\varphi)$ are provably equivalent (Lemma 1.40). Now, there is a maximally consistent set Γ in the closure Φ of $f(\neg\varphi)$ such that $f(\neg\varphi) \in \Gamma$. Because of the truth lemma we may conclude that $(M^\Phi, \Gamma) \models f(\neg\varphi)$, and therefore (Lemma 1.40), $(M_\Phi, \Gamma) \models \neg\varphi$ and thus $(M_\Phi, \Gamma) \not\models \varphi$ ⊣

Corollary 1.52 (Decidability) *The validity problem for concurrent dynamic epistemic logic is decidable.*

Proof () As we noted earlier the canonical model for a formula φ has at most $2^{|\Phi|}$ worlds. There are finitely many models with $2^{|\Phi|}$ worlds. As we can check whether φ holds in a model in finite time, we can check for all these models whether they satisfy φ. If they all satisfy φ, then we can conclude that $\vdash \varphi$. ⊣

5. Applications

In various publications this language has been applied to describe the dynamics of concrete multiagent systems [vD00, vD02b, vD02a, vD02c]. We give an overview of application areas by examples.

Example 1.53 (Card game actions)
Assume three players $1, 2, 3$ *and three cards* a, b, c. *Each player is dealt one card. Atom* a_1 *represents the fact where card a is held by player 1, etc. The action where player 1 picks up his card, so that the others cannot see which card it is, is described by the action*

$$\mathsf{pickup} = L_{123}(L_1?a_1 \cup L_1?b_1 \cup L_1?b_1)$$

In some epistemic state s where each player is dealt one card and all players have picked up their cards (for details, see [vD02b]), player 1 puts his card face up on the table. This is described by the action

$$\mathsf{table} = L_{123}?a_1 \cup L_{123}?b_1 \cup L_{123}?c_1$$

Note that in a given epistemic state only one of these alternatives can be executed. Now in that same epistemic state s we can also execute two rather different actions: firstly, player 1 can show his card to player 2 without player 3 seeing which card is shown. This action is described by

$$\mathsf{show} = L_{123}(L_{12}?a_1 \cup L_{12}?b_1 \cup L_{12}?c_1)$$

Next, player 2 can ask player 1 "please whisper in my ear the name of a card that you do not have," after which player 1 responds to 2's request. That action is described by

$$\mathsf{whisper} = L_{123}(L_{12}?\neg a_1 \cup L_{12}?\neg b_1 \cup L_{12}?\neg c_1)$$

In this case, whatever the actual epistemic state, 1 can choose one of two cards to whisper (and indeed, the complexity of the resulting epistemic state has now increased).

Example 1.54 (Cluedo) *The 'murder game' Cluedo is a card game where actions as in the previous example can take place. Other typical actions in Cluedo are 'ending your move' and 'winning the game'. For a perfect logician, ending a move in Cluedo is publicly announcing that you cannot win the game yet. This is the action* $L_N?\neg\mathsf{win}_n$, *where* win_n *is an epistemic formula describing knowledge of the 'murder cards', the cards 'held by the table' (agent 0) so to speak, i.e.* $\mathsf{win}_n = K_n(\mathsf{scarlett}_0 \wedge \mathsf{knife}_0 \wedge \mathsf{kitchen}_0) \vee K_n ...$ *(all murder cards combinations).*

EXAMPLE 1.55 (DIFFERENT CARDS)
Two players $1, 2$ *face three cards* a, b, c *lying face-down in two stacks on the table. Let a be the atom describing 'card a is in the stack with two cards', etc. Consider the following two actions:*

- *Player 1 draws a card from the two-cards stack, looks at it, returns it, and then player 2 draws a card from the two-cards stack and looks at it.*
- *Player 1 draws a card from the two-cards stack, and then player 2 takes the remaining card from that stack. They both look at their card.*

The first action is described by the sequence

$$L_{12}(L_1?p \cup L_1?q \cup L_1?r) \;;\; L_{12}(L_2?p \cup L_2?q \cup L_2?r)$$

Alternatively, the first action could have been described by

$$L_{12}(\bigcup_{x,y=a,b,c} (L_1?x \cap L_2?y))$$

The second action is described by

$$L_{12}(\bigcup_{x \neq y=a,b,c} (L_1?x \cap L_2?y))$$

The first action has nine different executions, the second one six only. The second action, where the alternatives chosen by 1 and 2 depend on each other, has also a different description as a sequence of two actions, namely:

$$\begin{array}{l} L_{12}(L_1?p \cup L_1?q \cup L_1?r) \;; \\ L_{12}(L_2?(p \wedge \neg K_1 p) \cup L_2?(q \wedge \neg K_1 q) \cup L_2?(r \wedge \neg K_1 r)) \end{array}$$

For example, $L_2?(p \wedge \neg K_1 p)$ *expresses that player 2 only learns p when player 1 has not learnt p already (after which 1 knows that p). That dependence could also have been put explicitly in local choices, which would have resulted in:*

$$\begin{array}{l} L_{12}(\\ \quad \begin{array}{lcll} L_{12}(!L_1?p \cup L_1?q \cup L_1?r) & ; & L_{12}(L_2?p\cup\, !L_2?q \cup L_2?r) & \cup \\ L_{12}(!L_1?p \cup L_1?q \cup L_1?r) & ; & L_{12}(L_2?p \cup L_2?q\cup\, !L_2?r) & \cup \\ L_{12}(L_1?p\cup\, !L_1?q \cup L_1?r) & ; & L_{12}(!L_2?p \cup L_2?q \cup L_2?r) & \cup \\ L_{12}(L_1?p\cup\, !L_1?q \cup L_1?r) & ; & L_{12}(L_2?p \cup L_2?q\cup\, !L_2?r) & \cup \\ L_{12}(L_1?p \cup L_1?q\cup\, !L_1?r) & ; & L_{12}(!L_2?p \cup L_2?q \cup L_2?r) & \cup \\ L_{12}(L_1?p \cup L_1?q\cup\, !L_1?r) & ; & L_{12}(L_2?p\cup\, !L_2?q \cup L_2?r) & \end{array} \\ \quad) \end{array}$$

EXAMPLE 1.56 (SUSPICION) *Suppose that an action* α may have *taken place. In what sense? Some agents are involved in that action, others aren't, and the agents not involved can imagine one of two things to have taken place: either that action, or nothing. They cannot distinguish one from the other. One might say that they only* suspect *but do not know that the action has taken place, or that they* consider *the action to have taken place (which is less biased towards one of the alternatives). The result can be described as*

$$\mathsf{consider}(\alpha) := L_N(\alpha \cup\ ?\top)$$

In words: group N learn that either α *happens or that nothing happens (that the 'test on verity succeeds': always).*

It is a generalization of Action Scenario 1.3, where Bert can imagine Anne to have read the letter. This action was described as $L_{12}(L_1?p \cup L_1?\neg p \cup\ ?\top)$.

EXAMPLE 1.57 (MUDDY CHILDREN) *We assume familiarity with the 'muddy children problem' [FHMV95]. All actions taking place in the 'muddy children problem' are public announcements. Public announcement of* φ *corresponds in* $\mathcal{L}_N$ *to a knowledge action* $L_N?\varphi$. *Suppose there are three children 1, 2, and 3. First 'father' tells them that at least one of them is muddy. This is described by* $L_{123}?(m_1 \vee m_2 \vee m_3)$ *(where* m_i *stand for 'child i is muddy'). And then father tells them, that who knows whether (s)he is muddy may step forward. When nobody steps forward, that action* noforward *is generally [Pla89, GG97, Bal02] analysed as the public announcement of a conjunction describing that none of the children knows whether he/she is muddy:* $L_{123}?((\neg K_1 m_1 \wedge \neg K_1 \neg m_1) \wedge (\neg K_2 m_2 \wedge \neg K_2 \neg m_2) \wedge (\neg K_3 m_3 \wedge \neg K_3 \neg m_3))$. *Although correct, this description is more from an external observer's point of view than from the point of view of a child having decided not to step forward. We prefer an analysis where this is clear, in other words, where the action 'nobody steps forward' is composed of subactions '1 does not step forward', '2 does not step forward' and '3 does not step forward':*

$$\begin{array}{lll} \mathsf{noforward} & = L_{123}(& L_{123}?(\neg K_1 m_1 \wedge \neg K_1 \neg m_1) \cap \\ & & L_{123}?(\neg K_2 m_2 \wedge \neg K_2 \neg m_2) \cap \\ & & L_{123}?(\neg K_3 m_3 \wedge \neg K_3 \neg m_3) \quad) \end{array}$$

Suppose that child 1 and 2 are actually muddy, and let this epistemic state be described by $(Cube, 110)$. *Then after the execution of* $L_{123}?(m_1 \vee m_2 \vee m_3)$ *and* noforward *the epistemic state of Figure 1.3 results, where it is the case that child 1 and 2 know that they are muddy (so* will *step forward at the next stage).*

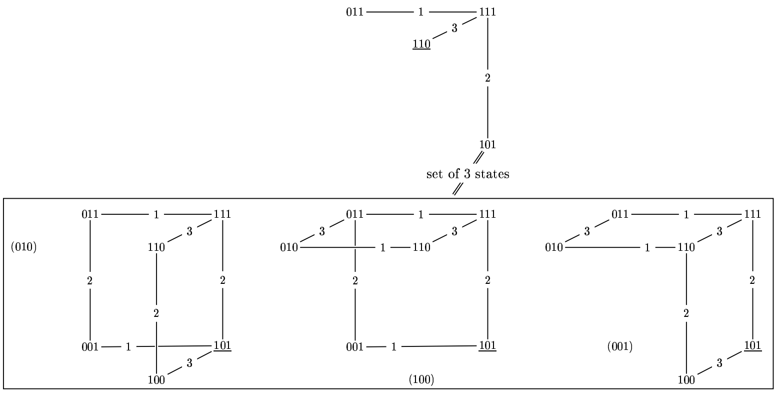

Figure 1.3. None of the children has stepped forward after father has told that at least one of them is muddy. The actual epistemic state is 110: 1 and 2 muddy. Each world of the resulting epistemic state is actually a set of three epistemic states. This is visualized for world 101.

In Figure 1.3 we have visualized the precise structure of one of the worlds of the resulting epistemic state, namely 101*. The world named* 101 *is actually a set of three epistemic states, corresponding to, from left to right, 2, 1, and 3 making public that they do not know whether they are muddy:*

$$\begin{array}{c}(Cube, 101)[\![L_{123}?(\neg K_2 m_2 \wedge \neg K_2 \neg m_2)]\!]\\(Cube, 101)[\![L_{123}?(\neg K_1 m_1 \wedge \neg K_1 \neg m_1)]\!]\\(Cube, 101)[\![L_{123}?(\neg K_3 m_3 \wedge \neg K_3 \neg m_3)]\!]\end{array}$$

Note that this set of three epistemic states is indeed merely a complex name for that world in the resulting model: the only thing that counts is 'its valuation', namely that child 1 and 3 are muddy in that world and that child 2 is not muddy (and of course, all three epistemic states in the set already corresponded on that!).

Example 1.58 (Security protocols) *From a pack of seven known cards (0, 1, 2, 3, 4, 5, 6) two players Anne (a) and Bill (b) each draw three cards and a third player Crow (c) gets the remaining card. How can Anne and Bill openly (publicly) inform each other about their cards, without Crow learning from any of their cards who holds it? There are* *many solutions to this problem [vD02c]. Suppose Anne actually holds {0,1,2}* (012_a)*, Bill {3,4,5}, and Crow card 6. One of the solutions consists of Anne saying "My hand is one of* $012, 034, 056, 135, 246$*" after which Bill says "Crow has card 6". This is described by the sequence of two public announcements*

$$L_{abc}?K_a(012_a \vee 034_a \vee 056_a \vee 135_a \vee 246_a) \ ; \ L_{abc}?K_b6_c$$

Hereafter, it is common knowledge that Anne knows Bill's cards, Bill knows Anne's cards, and Crow doesn't know any of Anne's or Bill's cards.

6. Conclusions

We have presented a proof system, and proved it to be sound and complete, for a dynamic epistemic logic in which higher-order information and belief change, and even higher-order belief change, can all be elegantly expressed. The crucial technical features of the language are, (1) that the notion of epistemic accessibility is lifted from one between worlds of an epistemic state to one between more complex semantic objects, such as sets of epistemic states, (2) the notion of the group of models, of epistemic states, and of actions, and (3) that actions are interpreted as a relation between epistemic states and sets of epistemic states. In view of proving completeness, we introduced a useful notion of syntactic access between actions. We gave an overview of the wide range of applications of this language for concrete multiagent system specification. We intend to continue this research by generalizing the semantics to include (not just knowledge but also) belief.

Acknowledgments

Wiebe van der Hoek carried out part of this research while visiting the University of Otago as a William Evans fellow. Barteld Kooi carried out this research with the assistance of the Netherlands Organization for Scientific Research (NWO) while visiting the University of Otago.

References

[AB95] R.J. Aumann and A. Brandeburger. Epistemic conditions for Nash equilibrium. *Econometrica*, (63):1161–1180, 1995.

[AGM85] C.E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[AHK97] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 100–109, Florida, October 1997.

[Bal99] A. Baltag. A logic of epistemic actions. In W. van der Hoek, J.-J. Meyer, and C. Witteveen, editors, *(Electronic) Proceedings of the ESSLLI 1999 workshop on Foundations and Applications of Collective Agent-Based Systems.* Utrecht University, 1999.

[Bal02] A. Baltag. A logic for suspicious players: Epistemic actions and belief updates in games. *Bulletin of Economic Research*, 54(1):1–45, 2002.

[BDPW02] S. Benferhat, D. Dubois, H. Prade, and M.A. Williams. A practical approach to revising prioritized knowledge bases. *Studia Logica*, 70(1), 2002.

[BdRV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, Cambridge, 2001. Cambridge Tracts in Theoretical Computer Science 53.

[BMS02] A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements, common knowledge and private suspicions. Originally presented at TARK 98, accepted for publication in Annals of Pure and Applied Logic, 2002.

[DP97] A. Darwiche and J. Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89(1-2):1–29, 1997.

[FHMV95] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge.* MIT Press, Cambridge MA, 1995.

[Ger99] J.D. Gerbrandy. *Bisimulations on Planet Kripke.* PhD thesis, University of Amsterdam, 1999. ILLC Dissertation Series DS-1999-01.

[GG97] J.D. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language, and Information*, 6:147–169, 1997.

[Gol92] R. Goldblatt. *Logics of Time and Computation.* CSLI Publications, Stanford, 2 edition, 1992. CSLI Lecture Notes No. 7.

[Hin62] J. Hintikka. *Knowledge and Belief.* Cornell University Press: Ithaca, NY, 1962.

[Hin86] J. Hintikka. Reasoning about knowledge in philosophy. In J. Y. Halpern, editor, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 63–80. Morgan Kaufmann Publishers: San Mateo, CA, 1986.

[HKT00] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic.* MIT Press, Cambridge MA, 2000. Foundations of Computing Series.

[KM91] H. Katsuno and A. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, pages 387–394, 1991.

[LR99] A.R. Lomuscio and M. D. Ryan. An algorithmic approach to knowledge evolution. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 13(2), 1999. Special issue on Temporal Logic in Engineering.

[MLH00] T.A. Meyer, W.A. Labuschagne, and J. Heidema. Refined epistemic entrenchment. *Journal of Logic, Language and Information*, 9:237–259, 2000.

[MvdH95] J.-J.Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science.* Cambridge Tracts in Theoretical Computer Science 41. Cambridge University Press, Cambridge, 1995.

[Par85] R. Parikh. The logic of games and its applications. In M. Karpinski and J. van Leeuwen, editors, *Topics in the theory of computation* Annals of Discrete Mathematics 24, pages 111–139, Amsterdam, 1985. Elsevier Science.

[Pau00] M. Pauly. Game logic for game theorists. Technical report, CWI, Amsterdam, 2000. CWI Technical Report INS-R0017.

[Pel87] D. Peleg. Concurrent dynamic logic. *Journal of the ACM*, 34(2):450–479, 1987.

[Pla89] J.A. Plaza. Logics of public communications. In M.L. Emrich, M.S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.

[tC02] B. ten Cate. Internalizing epistemic actions. In Maricarmen Martinez, editor, *Proceedings of the NASSLLI-2002 student session*, Stanford University, 2002.

[vB01] J.F.A.K. van Benthem. Logics for information update. In J.F.A.K. van Benthem, editor, *Proceedings of TARK VIII*, pages 51–88, Los Altos, 2001. Morgan Kaufmann.

[vD00] H.P. van Ditmarsch. *Knowledge games.* PhD thesis, University of Groningen, 2000. ILLC Dissertation Series DS-2000-06.

[vD01] H.P. van Ditmarsch. The semantics of concurrent knowledge actions. In M. Pauly and G. Sandu, editors, *ESSLLI 2001 workshop on Logic and Games*, 2001.

[vD02a] H.P. van Ditmarsch. The description of game actions in cluedo. In L.A. Petrosian and V.V. Mazalov, editors, *Game Theory and Applications*, volume 8, pages 1–28, Commack, NY, USA, 2002. Nova Science Publishers.

[vD02b] H.P. van Ditmarsch. Descriptions of game actions. *Journal of Logic, Language and Information*, 11:349–365, 2002.

[vD02c] H.P. van Ditmarsch. The russian cards problem: a case study in cryptography with public announcements. Technical report, Department of Computer Science, University of Otago, 2002. OUCS-2002-08.

[vdHW02] W. van der Hoek and M.J. Wooldridge. Tractable multi agent planning for epistemic goals. In C. Castelfranchi and W.L. Johnson, editors, *Proceedings of the First Internation Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, pages 1167–1174, New York, USA, 2002. ACM Press.