The Logic of Data Access and Data Exchanges

Alexandru Baltag
University of Amsterdam, ILLC
Amsterdam, Netherlands
thealexandrubaltag@gmail.com

Sonja Smets
University of Amsterdam, ILLC
Amsterdam, Netherlands
s.j.l.smets@uva.nl

We investigate a new logic that extends Dynamic Epistemic Logic (DEL), by combining standard epistemic modalities $K_a\varphi$ and $K_A\varphi$ for (individual and distributed) *propositional knowledge* with operators $K_a^\varphi x$, $K_A^\varphi x$ denoting *(conditional) non-propositional knowledge of a number* (in which an agent a or a group A have knowledge of the value of some variable x, if given additional information φ). We also generalize these operators, by considering formulas $|x|_A^\varphi \leq N$ (for any natural number N), saying that: *conditional on φ, A can narrow down the possible values of variable x to at most N possibilities*. In order to name and compare such hypothetical values, we extend the logic further with *definite descriptions based on minimization operators*: $\mu_N x_A^\varphi$ denotes the *least of the N possible values of x* (according to some fixed order $\leq$) that are considered possible by group A (given condition φ). On this static base, we consider DEL-style extensions with *dynamic modalities for general 'data-exchange events'* (covering private and public propositional announcements, but also secret hacking of a private database, or public sharing of one's data via open-source repositories, etc). In such scenarios, whole 'chunks' of information may be exchanged or modified: once access to a given source is gained, all the 'data' stored at that specific location becomes available. We give complete axiomatizations for the resulting logics, and prove their decidability and co-expressivity.

1 Introduction

Information may come in both propositional form (Boolean variables) and non-propositional form (e.g. numbers, names, addresses, pictures, videos, etc). Such *data* can be encoded as *values of local variables* (e.g. a cryptographic key or a password), that are stored and processed in certain *locations* or 'sites'(e.g., websites, folders, databases, etc), and can be retrieved when these sites are accessed as 'sources'. Each such source can be thought of as an *agent* (either because it actually is the knowledge base of a natural or artificial agent, or because we think of it as an abstract 'agent' possessing exactly the information that is stored at it).[1] Such agents either 'own' their data, or else have gained access to them from other sources: we say that the agents 'know' the values of these variables, and some of them can also modify these values. More complex data may have an *extended location*, the information being *distributed* among a number of agents: one can recover the value of such complex variables only by accessing several sources.

To deal with non-propositional information, multi-agent epistemic logic has been extended in recent years with "knowing-what" operators $K_a x$ or $K_A x$ for (individual or distributed) *knowledge of (the value of) some variable x* (in addition to the traditional modalities $K_a\varphi$ and $K_A\varphi$ for individual or distributed knowledge of a proposition φ). This work traces back to Plaza [22] and subsequent investigations by [18, 25, 26, 27, 20, 16, 21, 3, 12] on formalizing 'knowledge de re'. While Plaza axiomatized the static logic of 'knowing what', the extension with dynamic operators $[!\varphi]\psi$ for (propositional) public announcements $!\varphi$ was only later axiomatized by Wang and Fan [27]. To 'pre-encode' this dynamics using DEL-style reduction axioms [9], these authors needed a *conditional version of 'knowing what'*: $K_a^\varphi x$ means that a knows the value of x given the information that proposition φ was the case.

[1] The same piece of data can be stored/copied at multiple locations, and each location can store multiple pieces of data.

M. Bílková, M. Gattinger, I. van der Giessen, M. Girlando, Y. Wang (Eds.):
Advances in Modal Logic 2026 (AiML 2026)
EPTCS 447, 2026, pp. 93–116, doi:10.4204/EPTCS.447.6

© A. Baltag, S. Smets
This work is licensed under the
Creative Commons Attribution License.

In recent work [12], we extended this setting with operators $K_A^{\varphi} x$ for *conditional distributed knowledge of the value of x by a group A*. This was motivated by the need to deal with a more complex dynamics, going beyond propositional announcements or other traditional DEL-events [9, 7, 5, 17, 6, 8, 23]. In a *data-exchange event*, agents may gain access to 'sources' (i.e., to other agents' knowledge bases), in which case they can be assumed to instantly 'read' (and copy in their own knowledge base) *all* the information stored at those sources. Such events were previously considered in our work [10, 11], and in fact they subsume many other forms of non-propositional informational dynamics that were previously considered, e.g. "tell us all you know" in [2] (and with a different, interrogative interpretation in [24, 14]), in-group sharing [2, 15, 19, 4], and 'resolution' [1]. Other examples of data-exchange events covered in [12] include private changing of the value of a variable (e.g. one's password), parallel data-sharing within different subgroups (e.g., in a poster session), suspected hacking of a private database, private detection of such hacking, etc. As noted in [10], when an agent a gains access to another agent b's knowledge base, her new state of knowledge matches the *distributed knowledge* of the group $\{a,b\}$. So on the side of propositional knowledge, we needed operators $K_A\varphi$ for *distributed knowledge*[2] *within group A*. A similar move was also necessary on the side of "knowing what": operators $K_A x$, expressing that *the group A has distributed knowledge of the value of x*; in fact, to pre-encode public-announcement dynamics we needed the *conditional version* $K_A^{\varphi} x$ of these operators. Note that this type of conditional knowledge may *not* imply knowledge of the 'real' value of x (in the actual world), but only the fact that the agent or group *can uniquely determine a hypothetical value* of x (applicable only to worlds satisfying φ). To axiomatize these notions, in the presence of equality of values $x = y$, we were lead to introduce *definite description terms* x_A^{φ} that can *explicitly refer to such hypothetical values*: x_a^{φ} denotes (in the actual world) the *(unique) value that x would have according to agent a if* φ *were true*. Using this, we were able to provide in [12] a complete axiomatization for the corresponding dynamic-epistemic logic. But for this, we had to *restrict* the dynamics to 'semi-public' events (a class that does *not* include e.g. secret hacking). On the other hand, in [11] we axiomatized a logic with *unrestricted dynamics*, covering 'arbitrary' data-exchange events (including complicated hacking scenarios). But this was based on a *restricted static logic base*, that did *not* include non-propositional knowledge operators, but only the traditional epistemic modalities $K_a\varphi$ and $K_A\varphi$, as well as (a major generalization of) common knowledge $C_A\varphi$ (in the form of polyadic conditional-epistemic group modalities $C_A^{\mathbf{e}}\varphi_1 \dots \varphi_n$, indexed by data-exchange events $\mathbf{e}$). There were *no explicit variables x* in the logic, so no assertions $x = y$ or $Px_1 \dots x_n$ talking about the properties of specific pieces of (non-propositional) data, as well as *no* "knowledge-what" operators $K_a x$, $K_A x$ or $K_A^{\varphi} x$. And, because of these limitations, the class of data-exchange events was still inherently restricted: e.g., there was no event of privately changing the value (such as one's password), and no events involving preconditions of the form $K_a x$ (for instance, no "conditional hacking", in which it is common knowledge that an agent a may be hacking another agent b's database if and only if she came to know b's password).

In this paper, we overcome the limitations present in the frameworks of both [11] and [12], by providing a common generalization, while also greatly increasing the expressivity of the logic. More specifically, we generalize the operators $K_A^{\varphi} x$ to expressions $|x|_A^{\varphi} \leq N$ (for any natural number N), saying that: *if given additional information* φ*, the group A can narrow down the possible values of variable x to at most N possibilities* (for any natural number N). In other words, the group has distributed knowledge that, if φ is the case then the value of x belongs to a given list of N possible values.[3] This is a useful addition when dealing with cryptographic protocols. Indeed, if N is 'small enough' and x is say another

[2]The notation $D_A\varphi$ is typically used for distributed knowledge, but we prefer here $K_A\varphi$ because we think of this as capturing a natural notion of (virtual) group knowledge.

[3]Conditional knowledge $K_A^{\varphi} x$ of the value can now be recovered as a special case (for $N = 1$).

agent b's communication key or private password, then any intruder a having the capability $|x|_a^\varphi \leq N$ will be able to hack b's communication (by simply trying out all the N possible values), as soon as she may receive the information φ. The same applies to a group A of hackers if they can collectively narrow down the possibilities, i.e., have the capability $|x|_A^\varphi \leq N$. As done in [12] for $K_A^\varphi x$, whenever we have $|x|_a \leq N$ we introduce *definite descriptions denoting each of the N hypothetical values of x*. For this, we assume given a salient *total order* $\leq$ on the set of values. This is useful in numerical applications, but it also allows us to convert cardinality statements $|x|_a \leq N$ into ordinal descriptions $1_N.x_A^\varphi$, $2_N.x_A^\varphi$ etc, denoting "the first (or the second, etc) of the $\leq N$ possible values of x according to A given φ".[4] To capture the properties of the total order, we include in our language *order statements* $x \leq y$ (besides equality of values $x = y$ and possibly other predicates $Px_1 \ldots x_n$). We obtain a very expressive dynamic-epistemic logic, and we are able to completely axiomatize it and prove its decidability. Our proofs make an innovative use of known methods. For the static logic, we first use filtration to obtain a quasi-model: this is not a 'real' model, but just a syntactic construction, in which *variables have no values* (and the group relations are non-standard, as they are *not intersections* of the individual relations). We then follow the usual method for dealing with distributed knowledge, by unraveling the model into an infinite tree and redefining the relations to make them standard. However, new complications ensue due to the variables, and especially due to the combination of equality $x = y$ and non-propositional knowledge $K_A x$; the problems are in fact compounded by the presence of formulas for order $x \leq y$ and narrowing down $|x|_A^\varphi \leq N$. Here, we use our definite descriptions to ensure that the local properties of variables x are preserved and continuously transmitted at far-away nodes of the tree.[5] Defining the total order on values over the tree requires a use of the Order Extension Principle, and hence of the Axiom of Choice. Finally, the completeness for the dynamic logic uses DEL-style reduction axioms; but again, the details are quite tantalizing. Even the soundness of some reduction laws (e.g., the 'Cutoff-Min. Change' axiom) is not at all obvious!

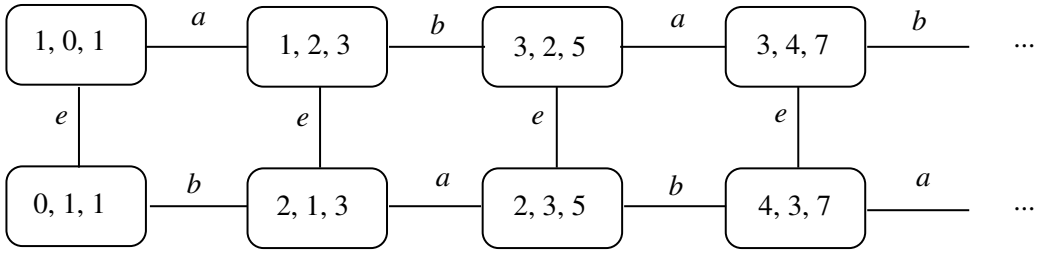
We should mention a self-imposed limitation: due to the page-limit, we chose *not* to include common knowledge operators $C_A\varphi$, whose dynamics introduces another source of complexity, requiring additional technical developments and proofs. We leave this step for the journal-version of this paper.

2 Motivating Examples

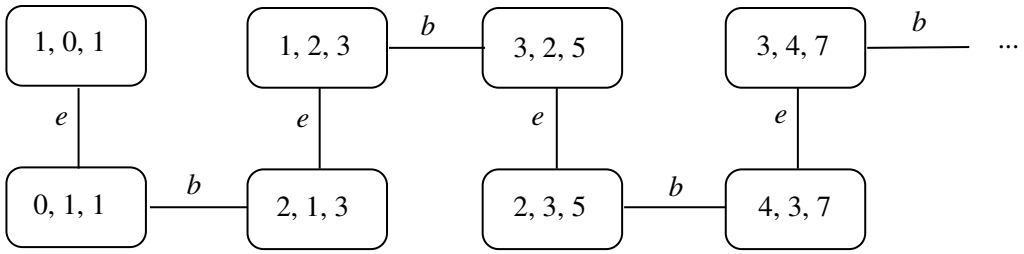
Example 1 Alice and Bob are each given a number $x(a), x(b) \in \mathbb{N} = \{0, 1, 2, \ldots\}$, while the value of the sum $x(e) := x(a) + x(b)$ is stored in a closed envelope e. It is common knowledge that: (1) *each of the two sees his/her own number, but neither of them can see the others' number, nor the sum* $x(e)$; (2) *one of the two numbers* x^a, x^b *is the immediate successor of the other* (i.e. either $x(a) = x(b) + 1$ or $x(b) = x(a) + 1$). We represent this situation using an *infinite Kripke model*, where states are triplets of numbers $(x(a), x(b), x(e))$, and the accessibility relations $\sim_a$ and $\sim_b$ encode the agents' uncertainty. We also have an accessibility relation $\sim_e$ for the 'envelope' (treated as an abstract 'agent'), encoding the information contained in it. The formula $K_a x(a) \wedge K_b x(b)$, saying that *agents know their numbers*, is valid on this model, while $K_e x(e)$ says that the sum $x(e)$ is stored in the envelope. On the other hand, agents a and b can *together* figure out the sum: we have $K_{\{a,b\}} x(e)$, i.e. the *group* $A = \{a, b\}$ has *"distributed" knowledge* of $x(e)$. The formulas $|x(a)|_b \leq 2$, $|x(b)|_a \leq 2$, $|x(e)|_a \leq 2$ and $|x(e)|_b \leq 2$ say that the live agents can *narrow down the possibilities* (for the other's number, as well as for the sum) *to at most two*. To *name* these values, we write e.g. $1_2.x(e)_a$ for the *first* (i.e., the *least*) of the *two* possible values of the sum $x(e)$ according to Alice, $2_2.x(e)_a$ for the *second* least (=largest) of the *two*, etc. The picture is:

[4] We can define all these using a *cutoff-minimization construct* $\mu_N x_A^\varphi$, which is the same as $1_N.x_A^\varphi$: it denotes *the least of the* $\leq N$ *possible values of x (according to A given* φ*)*.

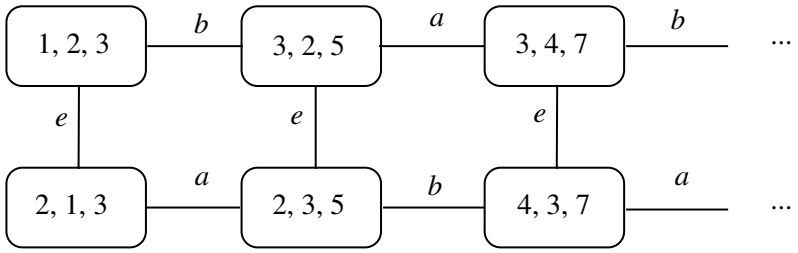
[5] In effect, the 'values' of variables x "hitch a ride" on the back of various group epistemic transitions $\rightarrow_A$ in the tree, and their properties are transmitted by passing them to *other (A-preserved) terms*.



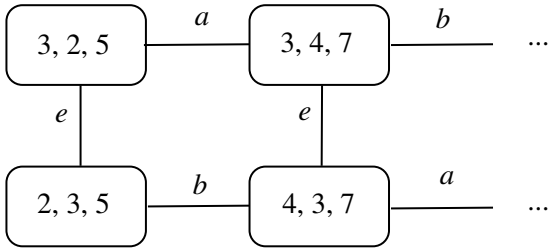
Example 1, continued Next, it is common knowledge that *Alice 'hacks' Bob's database*. This is a 'semi-public' event $!(a:b)$, which *updates the model* by replacing $\sim_a$ with the intersection $\sim_a \cap \sim_b$:



Example 2: an alternative scenario Start in the initial situation from Example 1. This time no hacking is allowed, but Alice and Bob are asked: "*Do you know each other's number*"? They both answer (truthfully, publicly and simultaneously): "*I don't know*". This is a *public announcement* $!(\neg K_a x(b) \wedge \neg K_b x(a))$. The *updated model* is obtained by deleting states $(0,1,1)$ and $(1,0,1)$ from the initial model:



If *asked the same question again*, they again answer "*I don't know*", which deletes $(2,1,3)$ and $(1,2,3)$:



If *asked for the 3rd time*, Alice says " *I know Bob's number*" and Bob says "*I don't know*". The announcement $!(K_a x(b) \wedge \neg K_b x(a))$ *deletes all states but* $(2,3,5)$: so we have $x(a) = 2, x(b) = 3, x(e) = 5$.

Example 3 Starting again in the initial situation from Example 1, agents are publicly announced that $x(a) < x(b)$. After that, Alice knows Bob's number, i.e. $[!(x(a) < x(b))]K_a x(b)$. We *can* reason hypothetically about this scenario *in the original model, without updating it*, using *conditional knowledge* $K_a^{x(a)<x(b)} x(b)$ *of the (hypothetical) value of* $x(b)$ *given the condition* $x_a < x_b$. We can also name this hypothetical value using a conditionalized version of the "least-of-N" description introduced above, e.g. writing $1_1.x(b)_a^{x(a)<x(b)}$ for the unique (=least out of only one) value of $x(b)$ that is considered possible by

Alice conditional on $x(a) < x(b)$. But note that, when considering a condition that is *known to be false* (e.g., $x(a) = x(b)$), there are *no such hypothetical values*: so the best we can do is to interpret the 'least' value as the 'infimum' $1_1.x(b)_a^{x(a)=x(b)} = inf\,\emptyset = \infty$. This means that we are working in $\mathbb{N}^\top := \mathbb{N} \cup \{\infty\}$.

Example 4: Back in time Imagine now that our story starts **earlier**, at a time when each 'agent' (Alice a, Bob b, and the envelope e) only 'knows' their own number (but not yet the correlations between numbers). The model is just $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$, with $(x(a),x(b),x(c)) \sim_a (x'(a),x'(b),x'(c))$ iff $x(a) = x'(a)$, and similarly for b and e. Next, the correlations are publicly announced $!((x(a) = x(b)+1 \vee x(b) = x(a)+1) \wedge x(e) = x(a)+x(b))$, and after that the semi-public hacking !(a:b) happens. Can we check that Alice will know $x(e)$ after this scenario *without updating the model* $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$? For this, we will need an operator $K_{a,b}^{(x(a)=x(b)+1 \vee x(b)=x(a)+1) \wedge (x(e)=x(a)+x(b))} x(e)$ for *conditional distributed knowledge of a value*.

To formalize the examples, we need the following ingredients: *equality* $x = y$; *order* $x < y$; *functions* $x = y+1$, $z = x+y$; *(conditional) (distributed) knowledge of a number* $K_A^\varphi x$; a way to express than an agent/group can *(conditionally) narrow down to N possible values* ($|x|_A^\varphi \leq N$), and to *name these values in increasing order*: $1_N.x_A^\varphi$, $2_N.x_A^\varphi$, etc; *public announcements* $!\varphi$, *semi-public hacking* $!(a:b)$, etc.

3 Syntax and Semantics of *LDA* and *LDAE*

Vocabularies: static and dynamic. A *static vocabulary* $\mathscr{V} = (\mathscr{A},V,Prop,Pred,Funct,ar,\varepsilon)$ consists of the following:[6] a finite set $\mathscr{A}$ of *agents* $a,b,c,\ldots$, also called 'locations' (e.g., websites, databases, processors, etc), where data are stored and processed; a set V of *basic variables* v, $v',v'',\ldots$; a set $Prop$ of *atomic propositions* $p,q,\ldots$; a set $Pred$ of *predicate symbols* $P,Q,\ldots$, including *equality* $(=)$ and an *order predicate* $\leq$; a set $Funct$ of *function symbols* $F,G,\ldots$, including two *constants* $\bot,\top$; an *arity* map $ar : Pred \cup Funct \to \mathbb{N}$, sending predicate symbols $P \in Pred$ and function symbols $F \in Funct$ to natural numbers $ar(P), ar(F) \in \mathbb{N}$, with $ar(\top) = ar(\bot) = 0$ and $ar(=) = ar(\leq) = 2$; an auxiliary symbol $\varepsilon \notin \mathscr{A} \cup V \cup Prop \cup Pred \cup Funct$, denoting "the *environment*" (e.g., the 'envelope' in our example).

A *dynamic vocabulary* $(\mathscr{V},\mathscr{E})$ is a pair consisting of a static vocabulary $\mathscr{V}$ and a *countable* set $\mathscr{E}$ of *'event names'* (denoted by $e,e',f,\ldots$), that includes some special symbols $!,?,\tau$.

Groups Given the static vocabulary $\mathscr{V}$, a *group* $A \subseteq \mathscr{A}$ is any non-empty set of agents in $\mathscr{A}$. We use capital letters $A,B,\ldots$ to denote groups.

Ordered Value Domains A *value domain* (for a given vocabulary $\mathscr{V}$) is a first-order model $\mathbf{D} = (D,I)$ for $\mathscr{V}$, consisting of: a *domain of 'values'* D, of cardinality $|D| > 1$; and an *interpretation function* I, mapping each functional symbol F of arity n into a function $I(F) = F_\mathbf{D} : D^n \to D$ (so that the constants $\bot$ and $\top$ are interpreted as values $\bot_\mathbf{D}, \top_\mathbf{D} \in D$), and each relational symbol P of arity n into a set $I(P) = P_\mathbf{D} \subseteq D^n$ of n-tuples of values in D, subject to the following conditions: the equality symbol $=$ is interpreted as the *identity relation* $=_\mathbf{D}$ on D; and the order symbol $\leq$ is interpreted as *some total order* $\leq_\mathbf{D} \subseteq D \times D$ on the set D having $\bot_\mathbf{D}$ *as its bottom element and* $\top_\mathbf{D}$ *as its top element* (i.e., $\bot_\mathbf{D} \leq_\mathbf{D} d \leq_\mathbf{D} \top_\mathbf{D}$ for all $d \in D$).

Minimization Given an ordered value domain $\mathbf{D} = (D,I)$, it is easy to see that *every finite non-empty subset* $D' \subseteq D$ *has a unique minimum* $min\,D' \in D'$, s.t. $min\,D' \leq_\mathbf{D} d'$ for all $d' \in D'$.

Notation: 'Cutoff Minimization' We can also consider, for any natural number $N \geq 1$, an "N-cutoff" version of minimization min_N, defined on *all* sets $D' \subseteq D$ (including the infinite ones), by putting:

$$\begin{array}{lcll} min_N\,D' & := & \top_\mathbf{D}\,(\text{'trivial'}), & \text{when } D' = \emptyset, \\ min_N\,D' & := & min\,D', & \text{when } 1 \leq |D'| \leq N, \text{ and} \\ min_N\,D' & := & \bot_\mathbf{D}\,(\text{'undefined'}), & \text{otherwise (for } |D'| > N). \end{array}$$

[6] All the sets $(\mathscr{A},V,Prop,Pred,Funct,\mathscr{E})$ in a vocabulary are assumed to be mutually disjoint.

Notation: 'Cutoff' n^{th} element For all sets $D' \subseteq D$ and all $1 \leq n \leq N$, we can now introduce a notation $n_N(D')$ for *"the n^{th} element of the $\leq N$ elements" of D*, by putting:

$$1_N(D') := min_N D', \quad (n+1)_N(D') := min_{N-1}(D' - \{n_N(D')\}) \text{ for } 1 \leq n < N.$$

Epistemic State Models An *epistemic state model over a value domain* $\mathbf{D} = (D, I)$ is a tuple $\mathbf{M} = (S, \sim, \underline{\bullet}(\bullet))$, where: (i) S is a set of *states* (or 'possible worlds'), typically denoted by $s, w, \ldots$; (ii) $\sim: \mathscr{A} \to \mathscr{P}(S \times S)$ maps agents $a \in \mathscr{A}$ to *equivalence relations* $\sim_a \subseteq S \times S$, called *'indistinguishability' relations*; (iii) $\underline{\bullet}(\bullet) : S \times (V \cup Prop) \to D$ is an *assignment map* ('valuation'), mapping pairs $(s, v) \in S \times V$ into arbitrary values $\underline{s}(v) \in D$, and mapping pairs $(s, p) \in S \times Prop$ into values $\underline{s}(p) \in \{\bot_{\mathbf{D}}, \top_{\mathbf{D}}\}$.

Group indistinguishability Given a state model $\mathbf{M} = (S, \sim, \underline{\bullet}(\bullet))$, we define *group indistinguishability relations* $\sim_A := \bigcap_{a \in A} \sim_a$ on S (for all groups $A \subseteq \mathscr{A}$) by *taking intersections*.

Syntax of *LDAE* For a fixed dynamic vocabulary $(\mathscr{V}, \mathscr{E})$, the *(dynamic) Logic of (group) Data Access & Exchange* (*LDAE*) has a syntax consisting of: (1) a set $Var := Var(\mathscr{V})$ of compound *variables* (or 'terms') x; (2) a set $Fml := Fml(\mathscr{V})$ of *formulas* φ; (3) a set of *'syntactic' event models*; (4) a set $Events$ of *data-exchange events*. These components are simultaneously defined by mutual recursion, as follows:

(1) & (2) **Variables and Formulas** The sets Var and Fml are given by the clauses:

$$\begin{array}{lllllllll} x & ::= & v & | & F(\bar{x}) & | & \varphi \to x | x & | & \mu_N x_A^{\varphi} \quad | \quad \mathbf{e}(x) \\ \varphi & ::= & p & | & P\bar{x} & | & \varphi \to \varphi & | & K_A \varphi \quad | \quad [\mathbf{e}]\varphi \end{array}$$

where: $v \in V$ are basic variables; $p \in Prop$ are atoms; $\bar{x} = (x_1, \ldots, x_n)$ are tuples of terms; $P \in Pred$ are n-ary predicate symbols; F are n-ary function symbols; $a \in \mathscr{A}$ are agents; $A \subseteq \mathscr{A}$ are groups; $N \geq 1$ are integers; and the *events* $\mathbf{e}$ are technically "pointed event models" $(\mathbf{E}, e)$, as defined below.

(3) & (4) **Event Models and Events** An *event model* $\mathbf{E} = (E, \sim, \underline{\bullet}(\bullet))$ consists of: (i) a finite set $E \subseteq \mathscr{E}$ of event names; (ii) an equivalence relation $\sim_a \subseteq E \times E$ (*agent a's indistinguishability* over events) for each $a \in \mathscr{A}$; (iii) a *change map* $\underline{e}(\bullet) : Prop \cup V \cup \mathscr{A} \cup \{\varepsilon\} \to Fml \cup Var \cup \mathscr{P}(\mathscr{A})$ for each event $e \in E$, mapping ε to a formula $\underline{e}(\varepsilon) \in Fml$, atoms $p \in Prop$ to formulas $\underline{e}(p) \in Fml$, basic variables $v \in V$ to terms $\underline{e}(v) \in Var$, and agents $a \in \mathscr{A}$ to groups $\underline{e}(a) \subseteq \mathscr{A}$. We require these items to satisfy *two conditions*:

1. *Self-Access* (agents access their own database): $a \in \underline{e}(a)$;
2. *Known Access* (agents know their sources): $e \sim_a f$ implies $\underline{e}(a) = \underline{f}(a)$;

As mentioned, a *data-exchange event* is just a *'pointed' event model* $\mathbf{e} = (\mathbf{E}, e)$, i.e. a *pair* of an event model $\mathbf{E} = (E, \sim, \underline{\bullet}(\bullet))$ and an event name $e \in E$. We denote by $Events$ the set of data-exchange events.

The Static Logic *LDA* The 'static' fragment of our logic, called the *(static) Logic of (group) Data Access* (*LDA*), consists of all formulas of *LDAE* that are built without the use of dynamic operators $[\mathbf{e}]\varphi$ or $\mathbf{e}(x)$.

Precondition and Postconditions The *precondition* of any event $\mathbf{e} = (\mathbf{E}, e)$ is the formula $pre_{\mathbf{e}} := \underline{e}(\varepsilon)$, which intuitively gives *the event's condition of possibility*: $\mathbf{e}$ can only happen in states satisfying its precondition $pre_{\mathbf{e}}$. The *event* $\mathbf{e}$*'s postcondition for* $p \in Prop$ is the formula $post_{\mathbf{e}}(p) := \underline{e}(p)$; similarly, the *event's postcondition for* $v \in V$ is the term $post_{\mathbf{e}}(v) := \underline{e}(v)$. Intuitively, the postconditions determine the way an event changes the values of atoms and variables: the new value of variable v (or atom p) after event $\mathbf{e}$ coincides with the 'old' value of the formula $post_{\mathbf{e}}(p)$ (or the term $post_{\mathbf{e}}(v)$) before the event.

(Extended) Access Map *Event* $\mathbf{e}$*'s access map* is the restriction of $\underline{e}$ to $\mathscr{A}$, specifying for each agent $a \in \mathscr{A}$ the group $\underline{e}(a) \subseteq \mathscr{A}$ of *all sources/agents (whose locations/databases are) accessed by agent a during the event*. We can also *extend* the access map to *groups* $A \subseteq \mathscr{A}$, by putting $\underline{e}(A) := \bigcup\{\underline{e}(a) : a \in A\}$, for the set of *sources that are distributedly accessible to the group A during event e*.

Subexpression-complexity An *expression* α is any formula φ, term x or event $\mathbf{e}$ of *LDAE*. The *subexpression complexity order* $<$ is the least transitive relation s.t.: (1) every formula is $>$ its subformulas, and also $>$ all terms and events occurring in it; (2) every term is $>$ its subterms, and also $>$ all formulas and events in it; (3) every event $\mathbf{e} = (\mathbf{E}, e)$ is $>$ all preconditions and postconditions of the form pre_f, $\underline{f}(p)$ and $\underline{f}(v)$ with $f \in E$, $p \in Prop$ and $v \in V$. It is easy to see that $<$ *is a well-founded partial order*.

Semantics We simultaneously define three semantic notions on state models $\mathbf{M} = (S, \sim, \underline{\bullet}(\bullet))$:

1. a *satisfaction relation* $s \models_{\mathbf{M}} \varphi$ between states $s \in S$ (in any state model $\mathbf{M}$) and formulas $\varphi \in Fml$;
2. an *extended assignment/valuation function* from $S \times (Var \cup Fml)$ to D, mapping $(state, variable)$-pairs $(s,x) \in S \times Var$ into arbitrary values $s(x)_{\mathbf{M}} \in D$, and mapping $(state, formula)$-pairs $(s, \varphi) \in S \times Fml$ into extreme values $s(\varphi)_{\mathbf{M}} \in \{\bot_{\mathbf{D}}, \top_{\mathbf{D}}\}$.
3. a *product update operation*, mapping state models $\mathbf{M} = (S, \sim, \underline{\bullet}(\bullet))$ and event models $\mathbf{E} = (E, \sim, \underline{\bullet}(\bullet))$ into *updated state models* $\mathbf{M} \bigotimes \mathbf{E} = (S \otimes E, \sim, \underline{\bullet}(\bullet))$ (over the same value domain $\mathbf{D}$).

For this definition, we need an auxiliary notation: for variables $x \in Var$, groups A, formulas φ and states $s \in S$, the *set of possible values of x at state s according to group A conditional on* φ is

$$(x_A^{\varphi})_{s,\mathbf{M}} := \{w(x)_{\mathbf{M}} : w \sim_A s, w \models_{\mathbf{M}} \varphi\}.$$

With this notation, we define our three semantic notions by mutual recursion:

$s \models_{\mathbf{M}} p$	iff	$\underline{s}(p) = \top_{\mathbf{D}}$
$s \models_{\mathbf{M}} Px_1 \dots x_n$	iff	$(s(x_1)_{\mathbf{M}}, \dots, s(x_n)_{\mathbf{M}}) \in I(P)$
$s \models_{\mathbf{M}} \varphi \to \psi$	iff	$s \models_{\mathbf{M}} \varphi$ implies $s \models_{\mathbf{M}} \psi$
$s \models_{\mathbf{M}} K_A \varphi$	iff	$w \models_{\mathbf{M}} \varphi$ for all $w \sim_A s$
$s \models_{\mathbf{M}} [\mathbf{e}]\varphi$	iff	$(s,e) \in S \otimes E$ implies $(s,e) \models_{\mathbf{M} \bigotimes \mathbf{E}} \varphi$, where $\mathbf{e} = (\mathbf{E}, e)$.
$s(v)_{\mathbf{M}}$	$=$	$\underline{s}(v)$ as given in the model $\mathbf{M}$
$s(\varphi \to x \vert y)_{\mathbf{M}}$	$=$	$s(x)_{\mathbf{M}}$ if $s \models_{\mathbf{M}} \varphi$, and
$s(\varphi \to x \vert y)_{\mathbf{M}}$	$=$	$s(y)_{\mathbf{M}}$ if $s \not\models_{\mathbf{M}} \varphi$
$s(F(x_1, \dots, x_n))_{\mathbf{M}}$	$=$	$(I(F))(s(x_1)_{\mathbf{M}}, \dots, s(x_n)_{\mathbf{M}})$
$s(\mu_N x_A^{\varphi})_{\mathbf{M}}$	$=$	$min_N Val(x_A^{\varphi})_{s,\mathbf{M}}$
$s(\mathbf{e}(x))_{\mathbf{M}}$	$=$	$(s,e)(x)_{\mathbf{M} \bigotimes \mathbf{E}}$ if $\mathbf{e} = (\mathbf{E}, e)$ is s.t. $(s,e) \in S \otimes E$, and
$s(\mathbf{e}(x))_{\mathbf{M}}$	$=$	$\top_{\mathbf{D}}$ otherwise.
$s(\varphi)_{\mathbf{M}}$	$=$	$\top_{\mathbf{D}}$ if $s \models_{\mathbf{M}} \varphi$, and
$s(\varphi)_{\mathbf{M}}$	$=$	$\bot_{\mathbf{D}}$ if $s \not\models_{\mathbf{M}} \varphi$.
$\mathbf{M} = (S, \sim, \underline{\bullet}(\bullet)), \mathbf{E} = (E, \sim, \underline{\bullet}(\bullet))$	$\mapsto$	$\mathbf{M} \bigotimes \mathbf{E} = (S \otimes E, \sim, \underline{\bullet}(\bullet))$, where:
$S \otimes E$	$=$	$\{(s,e) \in S \times E \mid s \models_{\mathbf{M}} pre_{\mathbf{e}}\}$
$(s,e) \sim_a (s', e')$	iff	$s \sim_{\underline{e}(a)} s'$ and $e \sim_a e'$
$\underline{(s,e)}(p)$	$=$	$s(post_{\mathbf{e}}(p))_{\mathbf{M}}$
$\underline{(s,e)}(v)$	$=$	$s(post_{\mathbf{e}}(v))_{\mathbf{M}}$

Extended assignment on sets of expressions We can 'lift' the assignment map to the level of *sets* of variables $X \subseteq Var$ and *sets* of formulas $\Phi \subseteq Fml$), by putting: $s(X)_{\mathbf{M}} := \{s(x)_{\mathbf{M}} : x \in X\}$; $s(\Phi)_{\mathbf{M}} := \{s(\varphi)_{\mathbf{M}} : \varphi \in \Phi\}$. Whenever the model is understood, we skip the subscript $\mathbf{M}$, writing simply $s \models \varphi$, $s(x)$, $s(X)$ and $s(\Phi)$. For sets of formulas Φ, we also write $s \models \Phi$ whenever $s \models \varphi$ for all $\varphi \in \Phi$.

Abbreviations. We define the usual Boolean connectives $true := (\top = \top)$, $false := (\top = \bot)$, $\neg\varphi := (\varphi \to false)$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \leftrightarrow \psi$, as well as the dual existential (Diamond) modalities $\langle K_A \rangle \varphi :=$

$\neg K_A\neg\varphi$, $\langle K_A^\theta\rangle\varphi := \neg K_A^\theta\neg\varphi$, $\langle \mathbf{e}\rangle\varphi := \neg[\mathbf{e}]\neg\varphi$. We also use the following abbreviations, for variables $x,y \in Var$, finite sets of variables $X,Y \subseteq Var$ and natural numbers n,N with $1 \leq n \leq N$:

$$x \in Y := \bigvee\{x = y : y \in Y\}, \quad ?_\varphi := \varphi{\to}\top|\bot, \quad \varphi{\to}x := \varphi{\to}x|\top,$$
$$K_A^\theta\varphi := K_A(\theta \to \varphi),\ K_A^\theta x := K_A^\theta(x = \mu_1 x_A^\theta),\ K_A x := K_A^{true}x,$$
$$1_N.x_A^\theta := \mu_N x_A^\theta,\ (n+1)_N.x_A^\theta := \mu_{N-1} x_A^{\theta \wedge x > n_N.x_A^\theta},$$
$$|x|_A^\theta \leq 0 := K_A\neg\theta, \quad |x|_A^\theta \leq n := K_A^\theta(x \in Var^n(x_A^\theta)), \quad \text{where } Var^n(x_A^\theta) := \{i_n.x_A^\theta : 1 \leq i \leq n\},$$
$$|x|_A^\theta = 0 := |x|_A^\theta \leq 0,\ |x|_A^\theta > n := |x|_A^\theta \not\leq n,$$
$$|x|_A^\theta = (n+1) := |x|_A^\theta > n \wedge |x|_A^\theta \leq (n+1) \text{ for } n \geq 0,$$
$$min\{x\} := x,\ min(X \cup \{y\}) := (minX \leq y){\to}(minX)|y.$$

Intuitively, $x \in Y$ says that x currently takes the same value as some term in Y. The term $?_\varphi$ is a Boolean variable, taking value $\top_{\mathbf{D}}$ if φ is true, and value $\bot_{\mathbf{D}}$ otherwise. The operator $\varphi{\to}x$ is the term analogue of material implication: it takes the value of x if φ is true, and value $\top_{\mathbf{D}}$ otherwise. Next, *'knowledge of value' is definable in our logic*, in both conditional and unconditional forms, for groups and individuals, via the abbreviations $K_A^\theta x$, $K_A x$, $K_A^\theta \bar{x}$, $K_A \bar{x}$. Note that, for the empty tuple $\lambda = ()$, we have $K_A^\theta\lambda = \bigwedge\emptyset = \top$. The term $n.x_A^\theta$ denotes the n^{th} value of x (in $\leq_{\mathbf{D}}$-order) considered possible by A, conditional on θ. The expressions $|x|_A^\theta = n$, $|x|_A^\theta \leq n$ and $|x|_A^\theta > n$ refer to the cardinality of the set of possible values of x, according to group A, given θ. Finally, $minX$ denotes the minimum value of all terms in X.

Distributed Location For any *expression*, i.e., any term, formula or event $\alpha \in Var \cup Fml \cup Events$, its *(distributed) location* $\mathscr{L}(\alpha) \subseteq \mathscr{A} \cup \{\varepsilon\}$ is given by the following recursive clauses:

$$\mathscr{L}(p) = \mathscr{L}(v) = \{\varepsilon\}, \qquad \mathscr{L}(K_A\varphi) = \mathscr{L}(\mu_N x_A^\varphi) = A,$$
$$\mathscr{L}(Px_1\ldots x_n) = \mathscr{L}(F(x_1,\ldots,x_n)) = \bigcup\{\mathscr{L}(x_i) : 1 \leq i \leq n\},$$
$$\mathscr{L}(\varphi \to \psi) = \mathscr{L}(\varphi) \cup \mathscr{L}(\psi), \qquad \mathscr{L}(\varphi{\to}x|y) = \mathscr{L}(x) \cup \mathscr{L}(\varphi) \cup \mathscr{L}(y),$$
$$\mathscr{L}(\mathbf{e}) = \mathscr{L}(pre_{\mathbf{e}}), \qquad \mathscr{L}([\mathbf{e}]\varphi) = \mathscr{L}(\mathbf{e}) \cup \underline{e}(\mathscr{L}(\varphi)), \qquad \mathscr{L}(\mathbf{e}(x)) = \mathscr{L}(\mathbf{e}) \cup \underline{e}(\mathscr{L}(x)),$$

where we used the extended access map $\underline{e}(A)$. In particular, this gives us that $\mathscr{L}(\bot) = \mathscr{L}(\top) = \mathscr{L}(true) = \mathscr{L}(false) = \emptyset$, $\mathscr{L}(\neg\varphi) = \mathscr{L}(\varphi)$, $\mathscr{L}(\varphi \wedge \psi) = \mathscr{L}(\varphi) \cup \mathscr{L}(\psi)$ and $\mathscr{L}(K_A^\theta x) = A$.

We can also extend the location map to *finite sets of terms* $X \subseteq Var$, by putting:

$$\mathscr{L}(X) := \bigcup_{x \in X} \mathscr{L}(x).$$

Locality The values of terms or propositions having a distributed location within a group $A \subseteq \mathscr{A}$ are *distributed knowledge among group A's members*, as shown by the results below.

Proposition 3.1. *(*Preservation*) Let s,w be states in a model $\mathbf{M}$ s.t. $s \sim_A w$ for some group $A \subseteq \mathscr{A}$, let φ be a static formula s.t. $\mathscr{L}(\varphi) \subseteq A$, and let $x \in Var$ be a static term s.t. $\mathscr{L}(x) \subseteq A$. Then we have:*

1. *$s \models_{\mathbf{M}} \varphi$ iff $w \models_{\mathbf{M}} \varphi$;*
2. *$s(x)_{\mathbf{M}} = w(x)_{\mathbf{M}}$.*

Corollary 3.2. *(*Local Knowledge*) For formulas φ and terms x, we have the following validities:*

$$\models \varphi \to K_A\varphi, \text{ whenever } \mathscr{L}(\varphi) \subseteq A; \qquad \models K_A x, \text{ whenever } \mathscr{L}(x) \subseteq A.$$

3.1 Examples of Data-Exchange Events and Event Models

In *'semi-public' event models* (with *only one event*), it is common knowledge who can read whose data.

Public Announcements For every formula φ, the event $!\varphi = (\mathbf{E}_{!\varphi}, !)$ *of publicly announcing* φ has an event model $\mathbf{E}_{!\varphi} = (\{!\}, \sim, \underline{\bullet}(\bullet))$ whose only event name is the special symbol !, with identity $\sim_a = \{(!,!)\}$ as accessibility relation for every agent, and where we put $\underline{!}(p) = p$, $\underline{!}(v) = v$, $\underline{!}(a) = \{a\}$ and $\underline{!}(\varepsilon) = \varphi$, hence $pre_{!\varphi} = \varphi$. This event encodes the dynamics of truthful public announcements [22]: the updated model $\mathbf{M} \otimes \mathbf{E}_{!\varphi}$ is (isomorphic to the one) obtained by deleting all the φ-worlds from the original model $\mathbf{M}$ (and keeping everything else the same).

Public and Semi-Public Sharing (*"Tell Us All You Know"*) For groups $A, B \subseteq \mathscr{A}$, $!(A:B) = (\mathbf{E}_{!(A:B)}, !)$ is a semi-public event whose model $\mathbf{E}_{!(A:B)} = (\{!\}, \sim, \underline{\bullet}(\bullet))$ has the same structure as $\mathbf{E}_{!\varphi}$, except for the access map and the precondition, which are given by putting: $\underline{!}(a) = B \cup \{a\}$ for all $a \in A$: $\underline{!}(a) = \{a\}$ for $a \notin A$; and $\underline{!}(\varepsilon) = true$ (so the precondition $pre_{!(A:B)} = true$ is tautological). In this action, it is common knowledge that *all agents in A gain access to the databases of all agents in B*. Its effect is to replace all the relations $\sim_a$ (with $a \in A$) from the original model $\mathbf{M}$ by the new relations $\sim_a^{!(A:B)} := \sim_a \cap \sim_B$ in the updated model $\mathbf{M} \otimes \mathbf{E}_{!(A:B)}$, while keeping everything else unchanged (including the relations $\sim_a$ with $a \notin A$). A special case is semi-public sharing $!(a:b)$ from b to a, obtained by taking $A = \{a\}$ and $B = \{b\}$: it is common that b shares all his knowledge with a. Another special case is *A-public sharing* $!B := !(\mathscr{A} : B)$, which is obtained by taking $A = \mathscr{A}$: all agents in B publicly share all their information. An even more special case is *b-public sharing* $!b = !\{b\} = !(\mathscr{A} : \{b\})$, obtained by taking $A = \{a\}$ for some designated agent a (and $B = \mathscr{A}$): agent a publicly "tells all she knows".[7]

Sharing *between* Multiple Groups For groups $A_1, B_1, \ldots, A_n, B_n$, the event $!(A_1 : B_1, \ldots, A_n : B_n)$ is a semi-public event, whose model $\mathbf{E}_{!(A_1:B_1,\ldots,A_n:B_n)} = (\{!\}, \sim, \underline{\bullet}(\bullet))$ has the same structure as $\mathbf{E}_{!(A:B)}$, except that the access map is given by $\underline{!}(a) = \{a\} \cup \bigcup \{B_i : i \leq n, a \in A_i\}$ for all a: it is common knowledge that *all agents in each group A_i gain access to the databases of all agents in the corresponding group B_i*.

Sharing *within* Groups For groups $A_1, \ldots, A_n \subseteq \mathscr{A}$, the event of *parallel sharing within different groups* $!(A_1, \ldots, A_n) = !(A_1 : A_1, \ldots, A_n : A_n)$ is the special case of $!(A_1 : B_1, \ldots, A_n : B_n)$ where $B_i = A_i$. In this action, it is common knowledge that *all agents in each group A_i simultaneously share all their knowledge with all other agents in the same group A_i*. A very special case is $n = 1$, which represents the *resolution* action $!(A)$: it is common knowledge that all the agents in A share their information with each other.[8]

Public Hacking $!H_{a:b}$ (as in the *WikiLeaks* case). It is common knowledge that *agent a 'hacks' agent b's database, using her knowledge of b's password* (represented by some variable v_b), *and makes all b's data public*. Everybody gets to see all b's data (including the value of his password v_b), but only a and b knew the password beforehand. The model is the same as for public sharing $!b$, except for the precondition, which is $pre_! := K_a v_b \wedge K_b v_b$: this event happens only if a knew b's password v_b before the event.

Semi-public Hacking This time, it is common knowledge that *agent a hacks agent b's database (using her prior knowledge of his password v_b) and can read all his data; but the others cannot read these data*. The event model is similar to the one for semi-public sharing $!(a:b)$ (and in particular, the access map is the same), except that the precondition is $K_a v_b \wedge K_b v_b$ (as in the case of public hacking $!H_{a:b}$).

Semi-public Change of Password $!(v_a := F(v_a, v'_a))$. It is common knowledge that *a changes her password v_a to a value $F(v_a, v'_a)$, that is a function of her current password v_a and of another (secret) local*

[7] This corresponds to a dynamics introduced and axiomatized in this form in [2], though technically isomorphic to the action of public resolution of an issue/question [24].

[8] The special case $!(\mathscr{A})$ was introduced in [2], and axiomatized in different contexts in [15, 19, 4]. The general case $!(\mathbf{G})$ was studied (as "*group resolution*") in [1].

variable v'_a. The encryption function F is common knowledge, but the values of the former password v_a and of the other secret number v'_a are known (hopefully) only by a. The event model is similar for $!\varphi$, except that the precondition is $pre_! = K_a v_a \wedge K_a v'_a$, and the postcondition for v_a is $\underline{!}(v_a) = F(v_a, v'_a)$.

Using larger event models, we can represent various forms of *private and semi-private data-exchanges*.

Secret Hacking Agent a may be hacking b's database iff she got hold of b's *password* v_b. Only the hacker (a) knows whether or not she succeeded to get v_b. We represent this event $SH_{a:b} = (\mathbf{E}, !)$ using a model $\mathbf{E} = (\{!, \tau\}, \sim, \underline{\bullet}(\bullet))$ with two events ! (for *successful hacking*) and τ (*unsuccessful hacking*). The preconditions are $pre_! := K_a v_b \wedge K_b v_b$ and $pre_\tau := \neg K_a v_b \wedge K_b v_b$. The access maps are $\underline{!}(a) := \{a, b\}$, $\underline{\tau}(a) = \{a\}$, and $\underline{!}(c) = \underline{\tau}(c) = \{c\}$ for all $c \neq a$; all postconditions are given by identity; a's accessibility is the identity relation, and the others' relations are the universal relation.

Conditional Change of Password $!(K_a|v_a|_b^{true} \leq N / v_a := F(v_a, v'_a))$. It is common knowledge that *a changes her password* v_a (as in the semi-public change) *only iff she knows that b has succeeded to narrow down her possible passwords to N possibilities*. This is an event $(\mathbf{E}, !)$, whose event model $\mathbf{E} = (\{!, ?\}, \sim, \underline{\bullet}(\bullet))$ has two event names ! (for *changing a's password*) and ? (for *no change*). The preconditions are $pre_! := K_a|v_a|_b^{true} \leq N$ and $pre_? := \neg K_a|v_a|_b^{true} \leq N$. The postconditions for v_a are $\underline{!}(v_a) := F(v_a, v'_a)$ and $\underline{?}(v_a) := v_a$, while all other postconditions and access map are given by identity: nobody gains access to others' databases (since the hacking is prevented by password-change). Finally, agent a's accessibility is the identity relation, while all others' accessibility is the universal relation.

Secret Detection of Hacking We can modify the secret hacking example $SH_{a:b}$ to allow the possibility that b might *detect* a's hacking attack (so that he might come to know that he is being hacked). Only b knows whether he actually detects an attack. This event $(\mathbf{E}, !)$ has a model $E = \{!, ?, \tau\}$ with *three* event names ! (detected hacking), ? (successful, undetected hacking) and τ (unsuccessful hacking). The access maps for ! and τ are as in the model for $SH_{a:b}$, while the access map for ? is the same as for !; and the same goes for $pre_!$, pre_τ and $pre_?$. Besides loops for all agents, we also have $! \sim_a ?$ (i.e. a doesn't know whether his hacking is detected or not) and $? \sim_b \tau$ (b can't distinguish between undetected hacking and unsuccessful hacking), while $\sim_c$ is the universal relation for all outsiders $c \neq a, b$.

4 Axiomatization and Decidability

We first look at the *static fragment LDA*: its proof system **LDA** is in Table 1.

Proposition 4.1. *The following theorems are provable in the system* **LDA***:*

1. *(Propositional Introspection)* $K_A\varphi \to K_A K_A \varphi$; $\neg K_A \varphi \to K_A \neg K_A \varphi$;
2. *(Term Introspection)* $K_A x$, *for* $\mathscr{L}(x) \subseteq A$;
3. *(Group Monotonicity)* $K_A \varphi \to K_B \varphi$, *for* $A \subseteq B$.

Theorem 4.2. *(*Soundness, Completeness and Decidability of **LDA***) The proof system* **LDA** *is complete for the static fragment LDA. Moreover, the static logic LDA is decidable.*

Proof. Completeness is shown in Section 5, using Prop 4.1. *Soundness* is trivial, except for the 'Reaching the Minimum' axiom, whose soundness we sketch here. Suppose that $s \models \varphi \wedge K_A^\varphi(x \in Y)$ for some $Y \subseteq Var$ s.t. $|Y| \leq N$ and $\mathscr{L}(Y) \subseteq A$. To show that $s \models \langle K_A^\varphi \rangle \left(\mu_N x_A^\varphi = x\right)$, we first prove the following:

Claim: $Val(x_A^\varphi)_{s,\mathbf{M}} \subseteq s(Y)_{\mathbf{M}}$.

To show this, let $d \in Val(x_A^\varphi)_{s,\mathbf{M}}$, i.e. there is some $w \sim_A s$ s.t. $w \models \varphi$ and $w(x) = d$. Since $s \models K_A^\varphi(x \in Y)$, we infer that $w \models (x \in Y)$, hence $w(x) \in w(Y)$. But, given that $w \sim_A s$ and $\mathscr{L}(Y) \subseteq A$, we have that $w(Y) = s(Y)$ (by Proposition 3.1). So we obtain $d = w(x) \in w(Y) = s(Y)$. Since this holds for any $d \in Val(x_A^\varphi)_{s,\mathbf{M}}$, we conclude that $Val(x_A^\varphi)_{s,\mathbf{M}} \subseteq s(Y)$, thus establishing our Claim.

(I)	**Axioms and rules of classical propositional logic (CPL):**
Modus Ponens rule	& all instances of CPL axioms in the language of *LDAE*
(II)	**Axioms for equality and order**:
(Indiscernability)	$\overline{x}=\overline{y}\to(P\overline{xz}\leftrightarrow P\overline{yz})$
(Functionality)	$\overline{x}=\overline{y}\to F(\overline{x})=F(\overline{y})$
(Definition by Cases)	$\varphi\to(x=(\varphi{\to}x|y)),\qquad \neg\varphi\to(y=(\varphi{\to}x|y))$
(Transitivity)	$(x\leq y\wedge y\leq z)\to x\leq z$
(Anti-symmetry)	$(x\leq y\wedge y\leq x)\to x=y$
(Totality)	$x\leq y\vee y\leq x$
(Top and Bottom)	$\bot\leq x\leq\top$
(Non-trivial Constants)	$\top\neq\bot$
(III)	**Axioms and rules for (distributed) knowledge**:
(Necessitation)	From φ, infer $K_A\varphi$
(Distribution)	$K_A(\varphi\to\psi)\to(K_A\varphi\to K_A\psi)$
(Veracity)	$K_A\varphi\to\varphi$
(Local Knowledge)	$\varphi\to K_A\varphi,\qquad$ for $\mathscr{L}(\varphi)\subseteq A$
(IV)	**Axioms for (cutoff) minimum value**:
(Lower Bound)	$\varphi\to\mu_N x_A^\varphi\leq x$
(Trivial Minimum)	$K_A\neg\varphi\to\mu_N x_A^\varphi=\top$
(Undefined Minimum)	$|x|_A^\varphi>N\to\mu_N x_A^\varphi=\bot$
(Reaching the Minimum)	$\left(\varphi\wedge K_A^\varphi(x\in Y)\right)\to\langle K_A^\varphi\rangle\left(\mu_N x_A^\varphi=x\right),\quad$ for $|Y|\leq N$ s.t. $\mathscr{L}(Y)\subseteq A$

Table 1: The proof system **LDA** for the static fragment, using abbreviations $K_A^\theta\varphi$, $\langle K_A^\theta\rangle\varphi$, $x\in Y$, $|x|_A^\varphi>N$.

Using the above Claim and the fact that $|Y|\leq N$, we have that $|Val(x_A^\varphi)_{s,\mathbf{M}}|\leq N$. Since we also have that $s(x)\in Val(x_A^\varphi)_{s,\mathbf{M}}\neq\emptyset$ (since $s\models\varphi$), we obtain that $s(\mu_N x_A^\varphi)=min_N Val(x_A^\varphi)_{s,\mathbf{M}}=min\,Val(x_A^\varphi)_{s,\mathbf{M}}\in Val(x_A^\varphi)_{s,\mathbf{M}}$ (by the semantic clause for μ_N and the definition of min_N), and hence $s(\mu_N x_A^\varphi)=w(x)$ for *some* $w\sim_A s$ with $w\models\varphi$. Applying again Proposition 3.1, we have $w(\mu_N x_A^\varphi)=s(\mu_N x_A^\varphi)=w(x)$ (since $\mathscr{L}(\mu_N x_A^\varphi)=A$). This, together with $w\sim_A s$ and $w\models\varphi$, yields $s\models\langle K_A^\varphi\rangle\left(\mu_N x_A^\varphi=x\right)$, as desired. □

As for the full *dynamic logic LDAE*, we need a few more notations and results to state our axioms.

Possible Values after an Event Recall that $Val(x_A^\varphi)_{s,\mathbf{M}}=\{w(x)_{\mathbf{M}}:w\sim_A s,w\models_{\mathbf{M}}\varphi\}$ is the set of possible values at state s according to A given φ. The following (easily checked) result gives us a characterization of the corresponding set of possible values of x (according to A given φ) *after an event* $\mathbf{e}$:

Lemma 4.3. $Val(x_A^\varphi)_{(s,\mathbf{e}),\mathbf{M}\otimes\mathbf{E}}=\bigcup_{f\sim_A e}Val\left(\mathbf{f}(x)_{\underline{f}(x)}^{\langle\mathbf{f}\rangle\varphi}\right)_{s,\mathbf{M}}$

Counting the Possible Values after an Event We want a formula expressing the fact that *there will be at most N possible values of x* (according to A given φ) *after the event* $\mathbf{e}$. Moreover, we want to express this *at the current state* (before the event). Put now $Var_e^N(x_A^\varphi):=\{n_N\mathbf{f}(x)_{\underline{f}(A)}^{\langle\mathbf{f}\rangle\varphi}:f\sim_A e,n\leq N\}$. Semantically, given Lemma 4.3, it should be clear that, *if* $|Val(\mathbf{f}(x)_{\underline{f}(A)}^{\langle f\rangle\varphi})_{s,\mathbf{M}}|\leq N$ *holds for all events* $f\sim_A e$, *then* $Val(x_A^\varphi)_{(s,\mathbf{e}),\mathbf{M}\otimes\mathbf{E}}\subseteq s(Var_e^N(x_A^\varphi))_{\mathbf{M}}$. However, *this inclusion might be strict*: it can happen that *not all of the values in* $s(Var_e^N(x_A^\varphi))_{\mathbf{M}}$ *are possible values* of x according to A (given φ, after the event e). The problem is that whenever $|Val(\mathbf{f}(x)_{\underline{f}(A)}^{\langle f\rangle\varphi})_{s,\mathbf{M}}|<N$ for *some* $f\sim_A e$, we get a possibly 'fake' x-value $N_N.\mathbf{f}(x)_{\underline{f}(A)}^{\langle\mathbf{f}\rangle\varphi}=\top_{\mathbf{D}}$. So, for any $z\in Var_e^N(x_A^\varphi)$, its condition of possibility is given by the formula:

$$\Diamond z := \left(z = \top \to \bigvee_{f \sim_A e} \langle K_{f(A)}^{\langle f \rangle \varphi} \rangle \mathbf{f}(x) = \top \right).$$

Thus, for any *subset* $Z \subseteq Var_e^N(x_A^\varphi)$, the formula

$$|Z|^\Diamond \leq N := \bigvee_{Y \subseteq Z, |Y| \leq N} \bigwedge_{z \in Z} \left(\Diamond z \to \bigvee_{y \in Y} z = y \right)$$

says that *the number of possible values of x in Z (according to A given* φ*) is at most N*. Finally, by applying this to the whole set $Z = Var_e^N(x_A^\varphi)$ above, we obtain the desired formula:

Lemma 4.4. *Let s be a state in a state model* $\mathbf{M}$*, let* $\mathbf{e}$ *be an event in an event model* $\mathbf{E}$ *with* $s \models_{\mathbf{M}} pre_{\mathbf{e}}$*, and let x,A,* φ *be s.t.* $|Val(f(x)_{\underline{f}(A)}^{\langle \mathbf{f} \rangle \varphi})_{s,\mathbf{M}}| \leq N$ *holds for* every $f \sim_A e$*. Then we have the equivalences:*

$$s \models_{\mathbf{M}} |Var_e^N(x_A^\varphi)|^\Diamond \leq N \;\; \textit{iff} \;\; |Val(x_A^\varphi)_{(s,\mathbf{e}),\mathbf{M}\otimes\mathbf{E}}| \leq N \;\; \textit{iff} \;\; (s,e) \models_{\mathbf{M}\otimes\mathbf{E}} |x_A^\varphi| \leq N.$$

Using these notations, the proof system **LDAE** for our full dynamic logic is given in Table 2.

(I)	**Static Axioms and rules of LDA**
All **LDA** rules & all	**LDA** axiom schemas extended to formulas of the full language *LDAE*
(II)	**Reduction axioms and rules for prop. formulas**:
($[e]$-Necessitation)	From φ, infer $[\mathbf{e}]\varphi$
(Change of Facts)	$[\mathbf{e}]p \leftrightarrow (pre_{\mathbf{e}} \to post_{\mathbf{e}}(p))$
(Change of Properties)	$[\mathbf{e}]P\bar{x} \leftrightarrow (pre_{\mathbf{e}} \to P\mathbf{e}(\bar{x}))$
(Distributivity)	$[\mathbf{e}](\varphi \to \psi) \leftrightarrow ([\mathbf{e}]\varphi \to [\mathbf{e}]\psi)$
(Knowledge Update)	$[\mathbf{e}]K_A\varphi \leftrightarrow \big(pre_{\mathbf{e}} \to \bigwedge\{K_{\underline{e}(A)}[\mathbf{f}]\varphi : f \sim_A e\}\big)$
(III)	**Reduction axioms for data terms**:
(Basic Value Change)	$\mathbf{e}(v) = (pre_{\mathbf{e}} \to post_{\mathbf{e}}(v))$
(Functional Change)	$\mathbf{e}(F(\bar{x})) = (pre_{\mathbf{e}} \to F(\mathbf{e}(\bar{x})))$
(Change of Cases)	$\mathbf{e}(\varphi \to x|y) = ([\mathbf{e}]\varphi \to \mathbf{e}(x)|\mathbf{e}(y))$
(Cutoff-Min. Change)	$\mathbf{e}(\mu_N x_A^\varphi) = \Big(pre_{\mathbf{e}} \to \Big(|Var_e^N(x_A^\varphi)|^\Diamond \leq N \to min\{\mu_N \mathbf{f}(x)_{\underline{e}(A)}^{\langle \mathbf{f} \rangle \varphi} : f \sim_A e\} | \bot\Big)\Big)$

Table 2: The proof system **LDAE**, where $\mathbf{e} = (\mathbf{E}, e), \mathbf{f} = (\mathbf{E}, f) \in Events$.

Proposition 4.5. *The following derived reduction laws are provable in the system* **LDAE***:*

- *(Impossible Change)* $[\mathbf{e}]false \leftrightarrow \neg pre_e$*; and* $[\mathbf{e}]true \leftrightarrow true$*;*
- *(Negation & Conjunction Reduction)* $[\mathbf{e}]\neg\varphi \leftrightarrow (pre_e \to \neg[\mathbf{e}]\varphi)$*; and* $[\mathbf{e}](\varphi \wedge \psi) \leftrightarrow ([\mathbf{e}]\varphi \wedge [\mathbf{e}]\psi)$*;*
- *(Preservation of Constants)* $\mathbf{e}(\top) = \top$*; and* $\mathbf{e}(\bot) = (pre_e \to \bot)$*;*
- *(Minimum Reduction)* $\mathbf{e}(min X) = min\{\mathbf{e}(x) : x \in X\}$.

Theorem 4.6. *(Soundness, Completeness, Expressivity and Decidability of* **LDAE***) The proof system* **LDAE** *in Table 2 is sound and complete for the dynamic logic LDAE. Moreover, LDAE is provably co-expressive with its static fragment LDA, and thus it is decidable.*

Proof. Completeness is shown in Section 6. *Soundness* is trivial, except for the 'Cutoff-Min' reduction axiom, which we sketch here. Let $\mathbf{M}$ be any state model, and s be any state. We consider *two cases*:

Case 1: $s \not\models pre_{\mathbf{e}}$. In this case, both terms of the equality claimed in the axiom take value $\top_{\mathbf{S}}$ at s in $\mathbf{M}$.

Case 2: $s \models pre_{\mathbf{e}}$. In this case, $s(\mathbf{e}(\mu_N x_A^\varphi))_{\mathbf{M}} = (s,e)(\mu_N x_A^\varphi))_{\mathbf{M}\otimes\mathbf{E}} = min_N \left(Val(x_A^\varphi)_{(s,e),\mathbf{M}\otimes\mathbf{E}}\right)$, and there are *two subcases* to consider:

Subcase (2A): there exists some $f^0 \in E$ s.t. $f^0 \sim_A e$ and $|Val(\mathbf{f^0}(x)^{\langle \mathbf{f^0}\rangle\varphi}_{\underline{f^0}(A)})_{s,\mathbf{M}}| > N$. Then we have by definition that $s(\mu_N \mathbf{f^0}(x)^{\langle \mathbf{f^0}\rangle\varphi}_{\underline{e}(A)}) = \bot_{\mathbf{D}}$ (given that $f_0 \sim_A e$ implies $\underline{f^0}(A) = \underline{e}(A)$), and thus $min\{\mu_N \mathbf{f}(x)^{\langle \mathbf{f}\rangle\varphi}_{\underline{e}(A)} : f \sim_A e\}$ takes value $\bot_{\mathbf{D}}$ at s. So the right-hand side term of the equality in the 'Cutoff-Min' reduction axiom takes value $\bot_{\mathbf{D}}$ at s (regardless of whether we have $s \models_{\mathbf{M}} |Var^N_e(x^\varphi_A)|^\Diamond \leq N$ or not). On the other hand, the left-hand side also evaluates to $\bot_{\mathbf{D}}$ at s (since by Lemma 4.3, $|Val(\mathbf{f^0}(x)^{\langle \mathbf{f^0}\rangle\varphi}_{\underline{f^0}(A)})_{s,\mathbf{M}}| > N$ implies that $|Val(x^\varphi_A)_{(s,\mathbf{e}),\mathbf{M}\otimes\mathbf{E}}| > N$, hence $(s,e)(\mu_N x^\varphi_A))_{\mathbf{M}\otimes\mathbf{E}} = min_N\left(Val(x^\varphi_A)_{(s,e),\mathbf{M}\otimes\mathbf{E}}\right) = \bot_{\mathbf{D}}$), as desired.

Subcase (2B): we have $|Val(\mathbf{f}(x)^{\langle \mathbf{f}\rangle\varphi}_{\underline{f}(A)})_{s,\mathbf{M}}| \leq N$ for all $f \in E$ s.t. $f \sim_A e$. We are in the conditions of Lemma 4.4, and there are again *two subcases* to consider:

Subcase (2B1): $s \not\models_{\mathbf{M}} |Var^N_e(x^\varphi_A)|^\Diamond \leq N$. In this case, we can use Lemma 4.4 to check that both sides of the equality in the 'Cutoff-Min' reduction axiom evaluate to $\bot_{\mathbf{D}}$ at s.

Subcase (2B2): $s \models_{\mathbf{M}} |Var^N_e(x^\varphi_A)|^\Diamond \leq N$. In this case, we can use Lemma 4.3 to show that $s(\mathbf{e}(\mu_N x^\varphi_A))_{\mathbf{M}} = (s,e)(\mu_N x^\varphi_A))_{\mathbf{M}\otimes\mathbf{E}} = min_N\left(Val(x^\varphi_A)_{(s,e),\mathbf{M}\otimes\mathbf{E}}\right) = min_N\left(\bigcup_{f\sim_A e} Val\left(\mathbf{f}(x)^{\langle \mathbf{f}\rangle\varphi}_{\underline{f}(x)}\right)_{s,\mathbf{M}}\right)$. So the left-hand side of the equality in the 'Cutoff-Min' axiom evaluates at state s (in $\mathbf{M}$) to $min_{f\sim_A e} min_N Val\left(\mathbf{f}(x)^{\langle \mathbf{f}\rangle\varphi}_{\underline{f}(x)}\right)_{s,\mathbf{M}}$. On the other hand, we can use Lemma 4.4 to check that the right-hand side of the equality in the 'Cutoff-Min' reduction axiom evaluates to the same expression at s. □

5 Completeness and Decidability Proofs for *LDA*

Throughout this section, *we fix a formula* $\varphi_0 \in Fml$. We prove Theorem 4.2 by the method of *quasi-models*. But, to obtain an appropriate analogue of Fischer-Ladner closure, we need two auxiliary notions:

Restricted Vocabulary For any finite set $\Sigma \subseteq Fml$, the Σ-*restricted vocabulary* $\mathscr{V}_\Sigma := (\mathscr{A}_\Sigma, V_\Sigma, Prop_\Sigma, Pred_\Sigma, Funct_\Sigma, ar_\Sigma, \varepsilon)$ is formed as follows: $\mathscr{A}_\Sigma$ is the set of agents occurring (inside terms or modalities in formulas) in Σ; V_Σ is *the set of basic variables occurring (as subterms of any term) in formulas of* Σ; $Prop_\Sigma := Prop \cap \Sigma$ is the set of atomic propositions in Σ; $Pred_\Sigma$ is the set of predicates occurring in (formulas of) Σ; $Funct_\Sigma$ is the set of function symbols occurring in (formulas of) Σ; ar_Σ is the restriction of ar to $Pred_\Sigma \cup Funct_\Sigma$. Clearly, if Σ is finite, then all the sets in $\mathscr{V}_\Sigma$ are finite.

The Σ-Restricted Set of Terms For any finite set of formulas $\Sigma \subseteq Fml$, the Σ-*restricted set of terms* Var_Σ is the smallest set of terms satisfying the following closure conditions: Var_Σ contains $\bot$ and $\top$, as well as all terms occurring in any formula of Σ (hence $V_\Sigma \subseteq Var_\Sigma$); Var_Σ is closed under subterms; if $i_N.x^\varphi_A \in Var_\Sigma$ for some $i \leq N$, then $j_N.x^\varphi_A \in Var_\Sigma$ for all $j \leq N$. Note that Var_Σ is only *a finite subset* of the (typically infinite) set $Var(\mathscr{V}_\Sigma)$ of terms of the language $LDA(\mathscr{V}_\Sigma)$.

We now proceed to introduce the appropriate notions of Fisher-Ladner Closure, syntactic types, and special sets of types called quasi-models.

Fisher-Ladner Closure Given now our fixed formula φ_0, the *closure of* φ_0 is the smallest set of formulas $\Sigma = \Sigma(\varphi_0)$ satisfying the following closure conditions: $\varphi_0 \in \Sigma$; $true, false \in \Sigma$; Σ is closed under subformulas and single negations $\sim\varphi$; if $P \in Pred_\Sigma$ has arity $ar(P) = n$ and $\bar{x} = (x_1, \ldots, x_n)$ is an n-tuple with all $x_1, \ldots, x_n \in Var_\Sigma$, then $P\bar{x} \in \Sigma$; if $(K_A\varphi) \in \Sigma$, θ is a subformula of $\sim\varphi$ and $B \subseteq \mathscr{A}_\Sigma$, then $(K_B\theta) \in \Sigma$ (hence also $\langle K_A\rangle\theta \in \Sigma$); if $(\varphi \to x|y) \in Var_\Sigma$, then $\varphi \in \Sigma$; if $x, y \in Var_\Sigma$, then $(x = y), (x \leq y) \in \Sigma$; if $(\mu_N x^\varphi_A), z \in Var_\Sigma$ and $Y \subseteq Var_\Sigma$, then $K^\varphi_A(z \in Y), \langle K^\varphi_A\rangle(z \in Y) \in \Sigma$.

Types Let Σ be the closure of φ_0. A Σ-*type* is a subset of Σ with the following properties:

1. for every $\varphi \in \Sigma$: $(\sim\varphi) \in \Delta$ iff $\varphi \notin \Delta$;

2. for every $(\varphi \wedge \psi) \in \Sigma$: $(\varphi \wedge \psi) \in \Delta$ iff $\varphi \in \Delta$ and $\psi \in \Delta$;
3. for every $P\overline{xz} \in \Sigma$: if $(\overline{x} = \overline{y}), P\overline{xz} \in \Delta$ then $P\overline{yz} \in \Delta$;
4. for $F(\overline{x}), F(\overline{y}) \in Var_\Sigma$: if $(\overline{x} = \overline{y}) \in \Delta$, then $(F(\overline{x}) = F(\overline{y})) \in \Delta$;
5. if $(x \leq y), (y \leq z) \in \Delta$, then $(x \leq z) \in \Delta$;
6. if $(x \leq y), (y \leq x) \in \Delta$, then $(x = y) \in \Delta$;
7. either for all $x, y \in Var_\Sigma$, we have either $(x \leq y) \in \Delta$ or $(y \leq x) \in \Delta$;
8. for all $x \in Var_\Sigma$, we have $(\bot \leq x), (x \leq \top) \in \Delta$;
9. $(\top \neq \bot) \in \Delta$;
10. for every $(\varphi{\rightarrow}x|y) \in \Sigma$: $\varphi \in \Delta$ implies $(x = (\varphi{\rightarrow}x|y)) \in \Delta$; and $\varphi \notin \Delta$ implies $(y = (\varphi{\rightarrow}x|y)) \in \Delta$;
11. if $(K_A\varphi) \in \Delta$ then $\varphi \in \Delta$;
12. for all $\mu_N x_A^\varphi \in Var_\Sigma$: $(\mu_N x_A^\varphi \leq x) \in \Delta$;
13. if $\varphi, K_A^\varphi(\mu_N x_A^\varphi \neq x) \in \Delta$ and $Y \subseteq Var_\Sigma$ is s.t. $\mathscr{L}(Y) \subseteq A$ and $|Y| \leq N$, then $\langle K_A^\varphi\rangle(x \notin Y) \in \Delta$;
14. for all $\mu_N x_A^\varphi \in Var_\Sigma$: if $(K_A{\neg}\varphi) \in \Delta$, then $(\mu_N x_A^\varphi = \top) \in \Delta$;
15. for all $\mu_N x_A^\varphi \in Var_\Sigma$: if $(|x|_A^\varphi > N) \in \Delta$, then $(\mu_N x_A^\varphi = \bot) \in \Delta$.

Observation *Types are closed under modus ponens*: if Δ is a type and $(\varphi \rightarrow \psi), \varphi \in \Delta$, then $\psi \in \Delta$.

Accessibility relations on types For types Δ, Δ' and group $A \subseteq \mathscr{A}_\Sigma$, we put:

$$\Delta \sim_A \Delta' \text{ iff } \varphi \in \Delta \Leftrightarrow \varphi \in \Delta' \text{ for all } \varphi \in \Sigma \text{ s.t. } \mathscr{L}(\varphi) \subseteq A.$$

Proposition 5.1. *The relations $\sim_A$ are equivalence relations on types, satisfying Group Monotonicity: $\Delta \sim_A \Delta'$ and $B \subseteq A$ imply $\Delta \sim_B \Delta'$.*

Proof. This follows directly from the definition of relations $\sim_A$ on types. □

Proposition 5.2. *If $\Delta \sim_A \Delta'$ and $(K_A\varphi) \in \Delta$, then $\varphi \in \Delta'$.*

Proof. Since $\mathscr{L}(K_A\varphi) = A$ and $\Delta \sim_A \Delta'$, we use the definition of $\sim_A$ on types and the fact that $(K_A\varphi) \in \Delta$ to infer that $(K_A\varphi) \in \Delta'$. This together with condition 11 on types, gives us the desired conclusion. □

Hat notation. For every type Δ, we put $\widehat{\Delta} := \bigwedge\Delta$ for the *conjunction of all formulas in* Δ.

Proposition 5.3. *Given types Δ and Λ, if $\widehat{\Delta} \wedge \langle K_A\rangle\widehat{\Lambda}$ is consistent, then $\Delta \sim_A \Lambda$.*

Proof. Let Δ and Λ be types as above, and suppose towards a contradiction that we have $\Delta \not\sim_A \Lambda$. Then there must exist $\varphi \in \Sigma$ s.t. $\mathscr{L}(\varphi) \subseteq A$ and $\varphi \in \Delta$, but $(\sim\varphi) \in \Lambda$. From this together with the assumption that $\widehat{\Delta} \wedge \langle K_A\rangle\widehat{\Lambda}$ is consistent, we infer that $\varphi \wedge \langle K_A\rangle{\neg}\varphi$ is consistent, and thus $\varphi \wedge {\neg}K_A\varphi$ is consistent. But this contradicts the fact that $\vdash \varphi \rightarrow K_A\varphi$ is an **LDA**-theorem for formulas φ with $\mathscr{L}(\varphi) \subseteq A$. □

Quasi-Models A *quasi-model for* φ_0 is a set S of types over Σ, with the following two properties: (*) $\varphi_0 \in \Delta_0$ for some type $\Delta_0 \in S$; (**) if $\langle K_A\rangle\varphi \in \Delta \in S$, then there is some $\Delta' \in S$ with $\Delta \sim_A \Delta'$ and $\varphi \in \Delta'$.

Proposition 5.4. *If $\Delta \in S$ is a type in a quasi-model S, then we have the following:*

(1) *if $K_A(\varphi \rightarrow \psi) \in \Delta$ and $(K_A\varphi) \in \Delta$, then $K_A\psi \in \Delta$;*

(2) *for every $(K_A\varphi) \in \Sigma$ s.t. $\mathscr{L}(\varphi) \in A$, if $\varphi \in \Delta$ then $(K_A\varphi) \in \Delta$;*

(3) *if $(K_A\varphi) \in \Delta$ and $B \subseteq A$, then $(K_B\varphi) \in \Delta$.*

The Σ-canonical quasi-model A Σ-*theory* is any maximally consistent subset of Σ. We denote by S_Σ the *set of all* Σ-*theories*. We will *show that* S_Σ *is a (finite) "canonical" quasi-model* for Σ.

Lemma 5.5. *Every* Σ-*theory is a* Σ-*type.*

Proof. This is easy to check, using the axioms of **LDA** and Proposition 4.1. □

Lemma 5.6. *For every* $\Delta \in S_\Sigma$*, if* $\langle K_A\rangle\varphi \in \Delta$ *then there is some* $\Delta' \in S_\Sigma$ *with* $\Delta \sim_A \Delta'$ *and* $\varphi \in \Delta'$.

Proof. Put $\Delta_A := \{\theta : \theta \in \Delta \text{ s.t. } \mathscr{L}(\theta) \subseteq A\} \cup \{\sim\theta : \theta \in (\Sigma - \Delta) \text{ s.t. } \mathscr{L}(\theta) \subseteq A\}$.

Claim: $\Delta_A \cup \{\varphi\}$ is consistent wrt the system **LDA**.

Proof of Claim: Suppose not. Then we have $\vdash \widehat{\Delta_A} \rightarrow \sim\varphi$. Applying Necessitation and Distribution, we obtain $\vdash K_A\widehat{D_A} \rightarrow K_A \sim \varphi$. On the other hand, by inspecting the structure of Δ_A, it is easy to see that $\mathscr{L}(\Delta_A) \subseteq A$, so by Strong Introspection (Proposition 4.1) we have $\vdash \widehat{\Delta_A} \rightarrow K_A\widehat{\Delta_A}$. Putting these together, we get $\vdash \widehat{\Delta_A} \rightarrow K_A \sim \varphi$. Since $\Delta_A \subseteq \Delta$ and Δ is closed under Σ-consequences, we have $(K_A \sim \varphi) \in \Delta$. But this contradicts the assumption that $\langle K_A\rangle\varphi \in \Delta$ (given the consistency of Δ).
Using our Claim and the standard Lindenbaum Lemma, we get that $\Delta_A \cup \{\varphi\}$ has a Σ-maximally consistent extension $\Delta' \in S'_\Sigma$. So we have $\varphi \in \Delta'$ and $\Delta_A \subseteq \Delta'$, which implies that $\Delta \sim_A \Delta'$. □

Proposition 5.7. *If* $\varphi_0 \in \Sigma$ *is consistent, then there exists a quasi-model for* φ_0.

Proof. Take the set S_Σ of all Σ-theories. By the Lindenbaum Lemma, there exists a maximally consistent subset $\Delta_0 \in S_\Sigma$, such that $\varphi_0 \in \Delta_0$. By Lemmas 5.5 and 5.6, S_Σ is a quasi-model for φ_0. □

Corollary 5.8. If φ_0 is satisfiable then there exists a quasi-model for φ_0.

The hard part is to prove the *converse* of this:

Proposition 5.9. *If S is a quasi-model for* φ_0*, then* φ_0 *is satisfiable.*[9]

The rest of this section is dedicated to the proof of Proposition 5.9.

Unravelling: the tree of histories Let us fix a quasi-model S, a formula φ_0 and a type $\Delta_0 \in S$ with $\varphi_0 \in \Delta_0$. We will construct a model for φ_0, based on an unravelling of S around Δ_0. A *history* is a finite sequence $h = (\Delta_0, A^1, \Delta_1, \ldots, A^n, \Delta_n)$ of any length $n \geq 0$, where $\Delta_1, \ldots, \Delta_n \in S$ are types and $A^1, \ldots, A^n \subseteq \mathscr{A}_\Sigma$ are groups, such that we have $\Delta_{i-1} \sim_{A^i} \Delta_i$ for all $i = 1, n$. Let H be the set of all histories. We denote by $last(h) := \Delta_n$ the last state in history h, and by $\rightarrow_A$ the natural *forward one-step relation* on histories in H, given by putting: $h \rightarrow_A h'$ iff $h' = (h, A, \Delta')$ (with $last(h) \sim_A \Delta' = last(h')$). We denote by $\leftarrow_A$ the *backward one-step relation*, defined as the converse of the forward relation: $h \leftarrow_A h'$ iff $h' \rightarrow_A h$. The one-step relations structure H into a *tree rooted at* Δ_0 (with the immediate successor relation given by $h \rightarrow h'$ iff $h \rightarrow_A h'$ for some group A). In particular, we have *the tree property* : *every two nodes* h, h' *of the tree are connected by a unique non-redundant path* $h = h_0 \leftarrow_{A^1} h_1 \leftarrow_{A^2} \ldots \leftarrow_{A^i} h_i \rightarrow_{A^{i+1}} \ldots \rightarrow_{A^n} h_n = h'$ (-in which neighboring nodes are immediate successors, in one order or another, and no nodes are repeated).

Epistemic relations on histories To make this tree into a model for our restricted vocabulary $\mathscr{V}_\Sigma$, we define our *single-agent indistinguishability relations* $\sim_a \subseteq H \times H$ on histories, by putting

$$\sim_a := \left(\bigcup_{A \ni a} \rightarrow_A \cup \bigcup_{A \ni a} \leftarrow_A \right)^*,$$

[9] Note that *this would immediately give us the completeness and decidability of* **LDA** (by Proposition 5.9, Proposition 5.7 and the soundness of the system **LDA**, as well as the finiteness of the closure $\Sigma = \Sigma(\varphi_0)$, and the decidability of checking that a subset of Σ is a quasi-model).

where $\leftarrow_A$ is the converse of $\rightarrow_A$, the unions range over groups $A \subseteq \mathscr{A}_\Sigma$ s.t. $a \in A$, and R^* is the reflexive-transitive closure of R. Since our goal is to build a standard model, the *group indistinguishability relations* $\sim_A \subseteq H \times H$ are taken to be simply the *intersections* $\sim_A := \bigcap_{a \in A} \sim_a$ of all the individual relations.

It is useful to give more concrete characterizations of the relations $\sim_A$ (and $\sim_a$) on histories:

Lemma 5.10. *The following are* equivalent*, for $A \subseteq \mathscr{A}_\Sigma$ and histories $h, h' \in H$:*

1. $h \sim_A h'$;
2. $A \subseteq A^i$, *for all groups A^i that appear as transition labels on the non-redundant path from h to h'.*

Proof. Use the definitions of $\sim_a$ and $\sim_A = \bigcap_{a \in A} \sim_a$ on H, and the uniqueness of non-redundant path. □

Lemma 5.11. *If $h \sim_A h'$, then $last(h) \sim_A last(h')$.*

Proof. The proof is by *induction on the length N of the non-redundant path* from h to h'.

For the *base case* $h = h'$, the conclusion follows trivially (given that $\sim_A$ are equivalence relations).

Inductive case: Suppose the non-redundant path from h and h' has length $N+1$, and let us look at the last transition on this path. Given Lemma 5.10, this transition can be either of the form $h_N \rightarrow_{A^N} h_{N+1} = h'$, or of the form $h_N \leftarrow_{A^N} h_{N+1} = h'$, with $A^N \supseteq A$. By definition of $\rightarrow_A$ on histories, we have either $h' = (h_N, A^N, last(h'))$ or $h_N = (h', A^N, last(h_N))$, with $last(h_N) \sim_{A^N} last(h')$ in both cases. By Monotonicity (Proposition 5.1) and the fact that $A \subseteq A^N$, we obtain $last(h_N) \sim_A last(h')$. On the other hand, we also have $last(h) \sim_A last(h_N)$ (-since the non-redundant path from h to h_N has length N, so by the induction hypothesis the pair (h, h_N) satisfies the conclusion of our Lemma, with h' replaced by h_N). Putting these two together (and using the transitivity of $\sim_A$), we conclude that $last(h) \sim_A last(h')$, as desired. □

Lemma 5.12. *If $(K_A \varphi) \in last(h)$ and $h \sim_A h'$, then $\varphi \in last(h')$.*

Proof. By Lemma 5.11, $h \sim_A h'$ implies $last(h) \sim_A last(h')$. This, together with $K_A \varphi \in last(h)$, implies by Proposition 5.2 that $\varphi \in last(h')$, as desired. □

Lemma 5.13. *(*Diamond Lemma*) If $\langle K_A \rangle \varphi \in last(h)$, then there exists some $h' \sim_A h$ s.t. $\varphi \in last(h')$.*

Proof. By Lemma 5.6, there exists some type $\Delta' \sim_A last(h)$ s.t. $\varphi \in \Delta'$. Take $h' = (h, A, \Delta')$. By Lemma 5.10 we have $h' \sim_A h$, and obviously $\varphi \in \Delta' = last(h')$, as desired. □

Corollary 5.14. *If $(K_A \varphi) \in \Sigma$ and $h \in H$, then: $(K_A \varphi) \in last(h)$ iff we have $\varphi \in last(h')$ for all $h' \sim_A h$.*

The Value Domain: a Quotient Construction The *set of values D* of our model will be a quotient of the Cartesian product $H \times Var_\Sigma$. We *define an equivalence relation* $\approx$ on pairs $(history, variable)$ in $H \times Var_\Sigma$ (telling us *when two such pairs represent the same value*), as well as *a total preorder* $\lessapprox$ on these pairs in $H \times Var_\Sigma$ (telling us *when the value of a pair is at most equal to another pair's value*). Then we take our canonical set of objects D to be *the quotient of $H \times Var_\Sigma$ with respect to* $\approx$, while the preorder $\lessapprox$ induces our desired *total order* $\leq$ on the quotient D.

For this, we first introduce another *equivalence relation* $\sim$ on $H \times Var_\Sigma$ (representing *identity of objects at a given node*), and another partial preorder $\lesssim$ on $\sim$ on $H \times Var_\Sigma$ (representing the *order relation on values at a given node*). This is given by putting:

$$(h,x) \sim (h',x') \text{ iff } h = h' \text{ and } (x = x') \in last(h),$$
$$(h,x) \lesssim (h',x') \text{ iff } h = h' \text{ and } (x \leq x') \in last(h).$$

Second, we define *(forward and backward) one-step relations $\rightarrow_=$ and $\rightarrow_\leq$ on pairs* in $H \times Var_\Sigma$:

$$(h,x) \rightarrow_= (h',x') \text{ iff } \exists y \in Var_\Sigma \exists A \supseteq \mathscr{L}(y) \text{ s.t. } h \rightarrow_A h', (x = y) \in last(h) \ \&\ (y = x') \in last(h');$$

$$(h,x) \leftarrow_{=} (h',x') \text{ iff } \exists y \in Var_\Sigma\, \exists A \supseteq \mathscr{L}(y) \text{ s.t. } h \leftarrow_A h', (x=y) \in last(h) \;\&\; (y=x') \in last(h');$$

$$(h,x) \rightarrow_{\leq} (h',x') \text{ iff } \exists y \in Var_\Sigma\, \exists A \supseteq \mathscr{L}(y) \text{ s.t. } h \rightarrow_A h', (x \leq y) \in last(h) \;\&\; (y \leq x') \in last(h');$$

$$(h,x) \leftarrow_{\leq} (h',x') \text{ iff } \exists y \in Var_\Sigma\, \exists A \supseteq \mathscr{L}(y) \text{ s.t. } h \leftarrow_A h', (x \leq y) \in last(h) \;\&\; (y \leq x') \in last(h').$$

Note that $\leftarrow_{=}$ is just the converse of $\rightarrow_{=}$, but $\leftarrow_{\leq}$ is *not* the converse of $\rightarrow_{\leq}$.

Value Identity and Order Finally, we define our main equivalence relation $\approx$ and our total preorder $\lessapprox$ on pairs $(history, variable)$ in $H \times Var_\Sigma$, by putting: $\approx := (\sim \cup \rightarrow_{=} \cup \leftarrow_{=})^*$; and $\lessapprox := (\lesssim \cup \rightarrow_{\leq} \cup \leftarrow_{\leq})^*$, where R^* is the reflexive-transitive closure of a relation R. It is useful to have a more concrete characterization of $\approx$ and $\lessapprox$, in terms of the non-redundant path from h to h':

Lemma 5.15. *("Path Lemma") Let $(h,x),(h',x') \in H \times Var_\Sigma$, and let $h = h_0 \leftarrow h_1 \leftarrow \ldots \leftarrow h_i \rightarrow \ldots \rightarrow h_n = h'$ be the non-redundant path from h to h'. Then the following are equivalent:*

- $(h,x) \approx (h',x')$;
- *either $(h,x) \sim (h',x')$ (if $n=0$), or else there exist $x_0,x_1,\ldots,x_i,\ldots,x_n \in Var_\Sigma$ s.t. we have: $(h,x) = (h_0,x_0) \leftarrow_{=} (h_1,x_1) \leftarrow_{=} (h_2,x_2) \leftarrow_{=} \cdots \leftarrow_{=} (h_i,x_i) \rightarrow_{=} \cdots \rightarrow_{=} (h_{n-1},x_{n-1}) \rightarrow_{=} (h_n,x_n) = (h',x')$.*

Similarly, the following are equivalent:

- $(h,x) \lessapprox (h',x')$;
- *either $(h,x) \lesssim (h',x')$ (if $n=0$), or else there exist $x_0,x_1,\ldots,x_i,\ldots,x_n \in Var_\Sigma$ s.t. we have: $(h,x) = (h_0,x_0) \leftarrow_{\leq} (h_1,x_1) \leftarrow_{\leq} (h_2,x_2) \leftarrow_{\leq} \cdots \leftarrow_{\leq} (h_i,x_i) \rightarrow_{\leq} \cdots \rightarrow_{\leq} (h_{n-1},x_{n-1}) \rightarrow_{\leq} (h_n,x_n) = (h',x')$.*

Corollary 5.16. *If $h,h' \in H$ and $x \in Var_\Sigma$ are s.t. $h \sim_A h'$ and $\mathscr{L}(x) \subseteq A$, then $(h,x) \approx (h',x)$.*

Proof. Let $h = h_0 \leftarrow_{A^1} h_1 \leftarrow_{A^2} \ldots \leftarrow_{A^i} h_i \rightarrow_{A^{i+1}} \ldots \rightarrow_{A^n} h_n = h'$ be the non-redundant path from h to h'. Since $h \sim_A h'$, we know that $A \subseteq A_k$ for all k (by Lemma 5.10). Using this together with $\mathscr{L}(x) \subseteq A$ (and the definition of the relation $(h,x) \rightarrow_{=} (h',x')$ on history-variable pairs), we obtain $(h,x) \leftarrow_{=} (h_1,x) \leftarrow_{=} \cdots (h_i,x) \rightarrow_{=} \cdots \rightarrow_{=} (h_n,x) = (h',x)$. By the Path Lemma, we have $(h,x) \approx (h',x)$. □

Corollary 5.17. *Let $x,x' \in Var_\Sigma$ and $h,h' \in H$, and suppose that the non-redundant path from h to h' is of the form $h = h_0 \rightarrow_A h_1 \rightarrow_A \ldots \rightarrow_A h_i \rightarrow_A \ldots \rightarrow_A h_n = h'$. Then we have $(h,x) \approx (h',x')$ iff there exists $y \in Var_\Sigma$ with $\mathscr{L}(y) \subseteq A$, $(x=y) \in last(h)$ and $(y=x') \in last(h')$.*

From Quasi-Model to Model We are first defining our *first-order data model* $\mathbf{D} = (D,I)$ for the restricted vocabulary $\mathscr{V}_\Sigma$: as announced, the *set of 'values' D* is the quotient

$$D := (H \times Var_\Sigma)/\approx = \{[h,x] : (h,x) \in H \times Var_\Sigma\},$$

where $[h,x]$ denotes the equivalence class of (h,x) modulo $\approx$, defined by

$$[h,x] := \{(h',x') \in H \times Var_\Sigma : (h,x) \approx (h',x')\} \text{ (for any given pair } (h,x) \in H \times Var_\Sigma).$$

The partial preorder $\lessapprox$ on pairs $(h,x) \in H \times Var_\Sigma$ induces a *partial order on the equivalence classes* $[h,x] \in D$, which in its turn can be extended to some *total order on D, that we will denote by $\leq_{\mathbf{D}}$*.[10]

The *interpretation function* I will map n-ary functional symbols $F \in Funct_\Sigma$ into n-ary functions $I(F) : D^n \rightarrow D$ given by: $I(F)([h,x_1],\ldots,[h,x_n]) := [h,F(x_1,\ldots,x_n)]$ if $F(x_1,\ldots,x_n) \in Var_\Sigma$; and $I(f)([h,x_1],\ldots,[h,x_n]) := \bot_{\mathbf{D}}$, otherwise; it will also map n-ary predicate symbols $P \in Pred_\Sigma \setminus \{\leq\}$ into n-ary relations $I(P) \subseteq D^n$ given by: $I(P) := \{([h,x_1],\ldots,[h,x_n]) \in D^n : h \in H, \overline{x} = (x_1,\ldots,x_n) \in Var_\Sigma^n \text{ s.t. } P\overline{x} \in last(h)\}$; while the interpretation $I(\leq)$ will be just the total order $\leq_{\mathbf{D}}$ constructed above.

[10] Though not unique, such a total order $\leq_{\mathbf{D}}$ exists by the Order-Extension Principle, a well-known consequence of the Axiom of Choice.

The Model Finally, *our epistemic state model* $\mathbf{M} = (H, \sim, \underline{\bullet}, \underline{\bullet}(\bullet))$ is given by taking: as set of states, the set H of all histories; the indistinguishability relations $\sim_a \subseteq H \times H$ are as defined above on histories[11]; the valuation/assignment map $\underline{\bullet}(\bullet) : H \times (V_\Sigma \cup Prop_\Sigma) \to D$ is given by putting: $\underline{h}(v) := [h, v]$ for $v \in V_\Sigma$; and $\underline{h}(p) := \top_{\mathbf{D}}$ iff $p \in last(h)$ (-else, $\underline{h}(p) := \bot_{\mathbf{D}}$) for $p \in Prop_\Sigma$.

Lemma 5.18. *("Interpretation Lemma") The interpretation I is well-defined, i.e. we have the following:*

1. $(h, \overline{x}) \approx (h', \overline{y})$ *implies* $(h, F(\overline{x})) \approx (h', F(\overline{y}))$;
2. $(h, \overline{x}) \approx (h', \overline{y})$ *implies that:* $(P\overline{x}\,\overline{z}) \in last(h)$ *iff* $(P\overline{y}\,\overline{z}) \in last(h')$;
3. $I(E)$ *is really the identity relation on D.*

Proof. Induction on the length of the non-redundant path from h to h', using our conditions on types. □

Lemma 5.19. *("Preservation of Values") If* $h \sim_A h'$ *and* $x \in Var_\Sigma$ *is s.t.* $\mathscr{L}(x) \subseteq A$, *then* $[h, x] = [h', x]$.
Proof. This follows immediately from Corollary 5.16 and the definition of $[h, x]$. □

The next results use the following notation, for $h \in H$, $x \in Var_\Sigma$, $\varphi \in \Sigma$ and $A \subseteq \mathscr{A}$:

$$Val_h(x_A^\varphi) \;:=\; \{[h', x] \in D \mid h' \sim_A h, \varphi \in last(h')\}$$

Lemma 5.20. *("Trivial-Minimum Lemma") If* $\mu_N x_A^\varphi \in Var_\Sigma$ *and* $h \in H$ *are s.t.* $Val_h(x_A^\varphi) = \emptyset$, *then* $[h, \mu_N x_A^\varphi] = \top_{\mathbf{D}}$.
Proof. First, we prove an auxiliary *Claim:* $(K_A \neg\varphi) \in last(h)$. To show this, suppose towards a contradiction that $(K_A \neg\varphi) \notin last(h)$. This implies that $\langle K_A \rangle \varphi \in last(h)$ (given the closure conditions on types and the fact that $\mu_N x_A^\varphi \in Var_\Sigma$ implies $\langle K_A \rangle \varphi \in \Sigma$). By the Diamond Lemma 5.13, there exists some $h' \sim_A h$ with $\varphi \in last(h')$, and thus $[h', x] \in Val_h(x_A^\varphi)$. But this contradicts the assumption that $Val_h(x_A^\varphi) = \emptyset$.

Using now the above Claim, and applying condition (14) on types, we obtain that $(\mu_N x_A^\varphi = \top) \in last(h)$, i.e. $[h, \mu_N x_A^\varphi] = \top_{\mathbf{D}}$, as desired. □

Lemma 5.21. *("Trivial-Value Lemma") If* $[h, \mu_N x_A^\varphi] = \top_{\mathbf{D}}$, *then* $Val_h(x_A^\varphi) \subseteq \{\top_{\mathbf{D}}\}$.

Proof. From $[h, \mu_N x_A^\varphi] = \top_{\mathbf{D}}$, we obtain that $(h, \mu_N x_A^\varphi) \approx (h, \top)$. By the Path Lemma 5.15, we must have $(\mu_N x_A^\varphi = \top) \in last(h)$. Suppose now (towards a contradiction) that $Val_h(x_A^\varphi) \not\subseteq \{\top_{\mathbf{D}}\}$, i.e. there exists some $h' \sim_A h$ with $\varphi, (x \neq \top) \in last(h')$. By Lemma 5.11, $h \sim_A h'$ implies that $last(h) \sim_A last(h')$. From this, together with the fact that $(\mu_N x_A^\varphi = \top) \in last(h)$ and that $\mathscr{L}(\mu_N x_A^\varphi = \top) = A$, = we derive that $(\mu_N x_A^\varphi = \top) \in last(h')$ (by the definition of $\sim_A$ on types). Using condition 12 on types, we get that $(\top \leq x) \in last(h')$, which together with conditions 8 and 6 on types, gives us that $(x = \top) \in last(h')$, contradicting the above assumption that $(x \neq \top) \in last(h')$. □

Lemma 5.22. *("Undefined-Minimum Lemma") Let* $\mu_N x_A^\varphi \in Var_\Sigma$ *be s.t.* $[h, \mu_N x_A^\varphi] = \bot_{\mathbf{D}}$. *Then we have either* $\bot_{\mathbf{D}} \in Val_h(x_A^\varphi)$ *or else* $|Val_h(x_A^\varphi)| > N$

Proof. Assume towards a contradiction that $[h, \mu_N x_A^\varphi] = \bot_{\mathbf{D}}$, but $\bot_{\mathbf{D}} \notin Val_h(x_A^\varphi)$; i.e.: $(\mu_N x_A^\varphi = \bot) \in last(h)$, but $(x \neq \bot) \in last(h')$ for all $h' \sim_A h$ with $\varphi \in last(h')$. By Corollary 5.14, $K_A^\varphi(x \neq \bot) \in last(h)$. This, together with $K_A(\mu_N x_A^\varphi = \bot) \in last(h)$ (which follows from $(\mu_N x_A^\varphi = \bot) \in last(h)$, by Proposition 5.4(2), and with the closure conditions on Σ and $\mathscr{L}(\mu_N x_A^\varphi = \bot) = A$), yield $K_A^\varphi(\mu_N x_A^\varphi \neq x) \in last(h)$.

To show that $|Val_h(x_A^\varphi)| > N$, we will construct a sequence

$$h_0 \to_A h_1 \to_A \ldots \to_A h_n \to_A \ldots \to_A h_N, \;\text{ with } h_n \sim_A h \text{ and } \varphi \in last(h_n) \text{ for all } n,$$

[11] Note that this is meant to be a (standard) model, so the group indistinguishability relations $\sim_A$ are just the intersections $\bigcap_{a \in A} \sim_a$, thus coinciding with the general relations $\sim_A$ introduced above.

together with a sequence of N 'witnesses' $y_1,\ldots,y_n,\ldots,y_N \in Var_\Sigma$, with $\mathscr{L}(y_n) \subseteq A$ for all n.

The construction is by recursion on $n \leq N$. For the base step ($n=0$), note that $(\langle K_A\rangle \varphi) \in last(h)$ (since otherwise we'd have $(\mu_N x_A^\varphi = \top) \in last(h)$ by condition (14) on types, contradicting the fact that $(\mu_N x_A^\varphi = \bot) \in last(h)$, given also condition (9) on types). By condition (**) on quasi-models, there exists $\Delta_0 \in S$ s.t. $last(h) \sim_A \Delta_0$ and $\varphi \in \Delta_0$. Take now $h_0 := (h, A, \Delta_0)$, which fulfills the desired specifications.

For the step n of the induction (with $1 \leq n \leq N$, assuming given h_{n-1} and $y_1,\ldots,y_{n-1}$ with the above properties), we first take the n^{th} witness y_n to be any term in Var_Σ with $\mathscr{L}(y_n) \subseteq A$ and $(x = y_n) \in last(h_{n-1})$, if such a term exists; and otherwise, we just put $y_n := \bot$. Next, we note that, by the induction hypothesis, we have $h_{n-1} \sim_A h$ and $\varphi \in last(h_{n-1})$, hence $last(h_{n-1}) \sim_A last(h)$. Together with the fact that $K_A^\varphi(\mu_N x_A^\varphi \neq x) \in last(h)$, this gives us that $\varphi, K_A^\varphi(\mu_N x_A^\varphi \neq x) \in last(h)$. By condition (13) on types (applied to $Y := \{y_1,\ldots,y_n\}$, where note that $\mathscr{L}(Y) \subseteq A$ and $|Y| \leq N$), we obtain $\langle K_A^\varphi\rangle \bigwedge_{1\leq i\leq n}(x \neq y_i) \in last(h_{n-1})$. Using again clause (**) on quasi-models, there exists some $\Delta_n \in S$ s.t. $last(h_{n_1}) \sim_A \Delta_n$ and $\varphi, \bigwedge_{1\leq i\leq n}(x \neq y_i) \in \Delta_n$. Take now $h_n := (h_{n-1}, A, \Delta_n)$, which obviously fulfills the desired specifications.

Given this construction, it is clear that $[h_n, x] \in Val_h(x_A^\varphi)$ for all $0 \leq n \leq N$. We will prove that *all these $N+1$ values are distinct* (which immediately gives us that $|Val_h(x_A^\varphi)| > N$, as desired):

Let $1 \leq n \leq N$. It is enough to show that $[h_n, x] \neq [h_m, x]$ for all $m < n$. Suppose not, i.e. assume towards a contradiction that we have $(h_n, x) \approx (h_m, x)$ for some $m < n$. Given that the non-redundant path from h_m to h_n has the shape $h_m \to_A h_{m+1} \to_A \ldots \to_A h_n$, we can apply Corollary 5.17 to conclude that there exists some $y \in Var_\Sigma$ with $\mathscr{L}(y) \subseteq A$, $(y = x) \in last(h_m)$ and $(y = x) \in last(h_n)$. On the other hand, we also have by construction that $(x = y_{m+1}) \in last(h_m)$ and $(x \neq y_{m+1}) \in last(h_n)$ (since $m+1 \leq n$). Putting all these together and using our conditions on equality, we obtain that $(y = y_{m+1}) \in last(h_m)$ and $(y \neq y_{m+1}) \in last(h_n)$. But this contradicts the fact that $last(h_m) \sim_A last(h_n)$ (since $h_m \sim_A h_n$ by construction), given that $\mathscr{L}(y = y_{m+1}) \subseteq A$ (and given the definition of the relation $\sim_A$ on types). □

Lemma 5.23. *("Least-of-N Values Lemma") If $\mu_N x_A^\varphi \in Var_\Sigma$ is s.t. $[h, \mu_N x_A^\varphi] \neq \bot_{\mathbf{D}}, \top_{\mathbf{D}}$, then:*

1. $(|x|_A^\varphi \leq N) \in last(h)$;
2. $|Val_h(x_A^\varphi)| \leq N$;
3. $[h, \mu_N x_A^\varphi] = minVal_h(x_A^\varphi)$; *i.e., there exists some history $h_0 \in H$, satisfying: $h_0 \sim_A h$; $\varphi \in last(h_0)$; and $[h, \mu_N x_A^\varphi] = [h_0, x] \leq_{\mathbf{D}} [h', x]$ for all $[h', x] \in Val_h(x_A^\varphi)$.*

Proof. For part 1, from $[h, \mu_N x_A^\varphi] \neq \bot_{\mathbf{D}}$ we obtain that $(\mu_N x_A^\varphi \neq \bot) \in last(h)$, so by condition (15) on types we have $(|x|_A^\varphi \leq N) \in last(h)$, as desired.

For part 2, we use part 1 and Lemma 5.12, to obtain that $(\bigvee_{1\leq i\leq n} x = i_N.x_A^\varphi) \in last(h')$ holds for all $[h', x] \in Val_h(x_A^\varphi)$; hence, for every $[h', x] \in Val_h(x_A^\varphi)$ there exists $i \in \{1,\ldots,N\}$ s.t. $[h', x] = [h', i_N.x_A^\varphi] = [h, i_N.x_A^\varphi]$ (where the last equality follows by the Corollary 5.16 from the fact that $\mathscr{L}(i_N.x_A^\varphi) = A$ together with $h' \sim_A h$). Thus, we have that $Val_h(x_A^\varphi) \subseteq \{[h, i_N.x_A^\varphi] \mid 1 \leq i \leq N\}$, which implies that $|Val_h(x_A^\varphi)| \leq N$.

For part 3, we use the assumption that $[h, \mu_N x_A^\varphi] \neq \top_{\mathbf{D}}$ and Lemma 5.20 to obtain $Val_h(x_A^\varphi) \neq \emptyset$; i.e., there exists $h_1 \sim_A h$ s.t. $\varphi \in last(h_1)$. Putting this together with the fact that $(|x|_A^\varphi \leq N) \in last(h_1)$ (which follows from part 1 and $h_1 \sim_A h$, using $\mathscr{L}(|x|_A^\varphi \leq N) = A$ and Lemma 5.11), and applying condition (13) on types (with $Y := Var^N(x_A^\varphi) = \{i_N.x_A^\varphi : 1 \leq i \leq N\}$, while recalling that by definition $|x|_A^\varphi \leq N := K_A^\varphi(x \in Var^N(x_A^\varphi))$, we conclude that $\langle K_A^\varphi\rangle(x = \mu_N x_A^\varphi) \in last(h_1)$. By the Diamond Lemma 5.13, there exists some $h_0 \sim_A h_1 \sim_A h$ with $\varphi, (x = \mu_N x_A^\varphi) \in last(h_0)$, hence $[h_0, x] = [h_0, \mu_N x_A^\varphi] = [h, \mu_N x_A^\varphi]$ (with the last equality due to $h_0 \sim_A h$ and $\mathscr{L}(\mu_N x_A^\varphi) = A$, by Lemma 5.19). Finally, to prove that $[h, \mu_N x_A^\varphi] = minVal_h(x_A^\varphi)$, let $[h', x] \in Val_h(x_A^\varphi)$, i.e., $h' \sim_A h$ with $\varphi \in last(h)$, and we need to show that $[h, \mu_N x_A^\varphi] \leq_{\mathbf{D}} [h', x]$. By condition (12) on types, we have $(\mu_N x_A^\varphi \leq x) \in last(h')$, hence $[h', \mu_N x_A^\varphi] \leq_{\mathbf{D}} [h', x]$. Since $h \sim_A h'$ and $\mathscr{L}(\mu_N x_A^\varphi) = A$, Lemma 5.19 gives us that $[h, \mu_N x_A^\varphi] = [h', \mu_N x_A^\varphi] \leq_{\mathbf{D}} [h', x]$, as desired. □

Lemma 5.24. *("Truth Lemma") For every $\varphi \in \Sigma$ and $x \in Var_\Sigma$, the following hold for all $h \in H$:*

1. $h \models_{\mathbf{M}} \varphi$ *iff* $\varphi \in last(h)$;
2. $\underline{h}(x)_{\mathbf{M}} = [h,x]$.

Proof. We prove 1.&2. *for all expressions $\alpha \in \Sigma \cup Var_\Sigma$ by induction on sub-expression complexity*.

(i) *Predicative Atoms*: $\varphi = P\bar{x}$, with $\bar{x} = (x_1,\ldots,x_n)$. For $P \in Pred \setminus \{\leq\}$, we have the equivalences: $h \models P\bar{x}$ iff $h(\bar{x}) \in I(P)$ iff (by the induction hypothesis for claim (2)) $([h,x_1],\ldots,[h,x_n]) \in I(P)$ iff $\exists h' \in H$ s.t. $((h,\bar{x}) \approx (h',\bar{x}) \& (P\bar{x}) \in last(h'))$ iff $(P\bar{x}) \in last(h)$ (by Lemma 5.18). For $\leq$, we have the equivalencies: $h \models x \leq y$ iff $(h,x) \lesssim (h,y)$ iff $(h,x) \lessapprox (h,y)$ iff $[h,x] \leq_{\mathbf{D}} [h,y]$ (by the Path Lemma).

(ii) *Propositional atoms*: $p \in Prop_\Sigma$. By definition, $h \models p$ iff $\underline{h}(p) = \top_{\mathbf{D}}$ iff $p \in last(h)$.

(iii) *Basic variables*: $v \in V_\Sigma$. By definition, $h(v)_{\mathbf{M}} = \underline{h}(v) = [h,v]$.

(iv) *Boolean Case*: $\varphi \to \psi$. These is trivial, using conditions 1 and 2 on types.

(v) *K_A-modal Case*: $(K_A\varphi) \in \Sigma$. From *left-to-right*: assume (towards a contradiction) that $h \models K_A\varphi$ but $(K_A\varphi) \notin last(h)$. By condition 1 on types, we have $(\langle K_A\rangle \sim \varphi) \in last(h)$, and so by the property (**) of quasi-models, there exists a type $\Delta' \in S$ s.t. $last(h) \sim_A \Delta'$ and $(\sim \varphi) \in \Delta'$. Take $h' := (h,A,\Delta')$: this is a well-defined history in H, satisfying $h \sim_A h'$ and $last(h') = \Delta'$. From $h \models K_A\varphi$ we infer that $h' \models \varphi$, and so by the induction hypothesis we have $\varphi \in last(h') = \Delta'$, in contradiction to $(\sim \varphi) \in \Delta'$.

For the *right-to-left* direction: we assume that $(K_A\varphi) \in last(h)$, and we have to prove that $h \models K_A\varphi$. For this, let $h' \in H$ be s.t. $h \sim_A h'$, and we need to show that $h' \models \varphi$. From $h \sim_A h'$ we obtain that $last(h) \sim_A last(h')$ (by Lemma 5.11), which together with $h \models K_A\varphi$ gives us $\varphi \in last(h')$ (by Proposition 5.2). Applying the induction hypothesis, we conclude that $h' \models \varphi$, as desired.

(vi) *Terms defined by cases $\varphi \to x|y$*. Given $(\varphi \to x|y) \in \Sigma$, we have $\varphi \in \Sigma$, so we have that either $\varphi \in last(h)$, or else $(\sim \varphi) \in last(h)$. In the first case, we have $(\varphi \to x|y = x) \in last(h)$ (by condition 10 on types), hence $[h, \varphi \to x|y] = [h,x]$; and on the other hand, $\varphi \in last(h)$ yields $h \models \varphi$ (by the induction hypothesis for Claim 1), so by definition we have $\underline{h}(\varphi \to x|y)_{\mathbf{M}} = \underline{h}(x)_{\mathbf{M}}$, and by the induction hypothesis for Claim 2 we have $\underline{h}(x)_{\mathbf{M}} = [h,x]$, thus obtaining $\underline{h}(\varphi \to x|y)_{\mathbf{M}} = \underline{h}(x)_{\mathbf{M}} = [h,x] = [h, \varphi \to x|y]$, as desired. The second case is similar: from $(\sim \varphi) \in last(h)$ we get $(\varphi \to x|y = y) \in last(h)$ (by condition 10 on types), hence $[h, x|_\varphi] = [h,y]$; and on the other hand, $(\sim \varphi) \in last(h)$ implies $\varphi \notin last(h)$ (by the consistency of types), which by the induction hypothesis for Claim 1 yields $h \not\models \varphi$, so by definition we have $\underline{h}(\varphi \to x|y)_{\mathbf{M}} = \underline{h}(y)_{\mathbf{M}}$, and by the induction hypothesis for Claim 2 we have $\underline{h}(y)_{\mathbf{M}} = [h,y]$, thus obtaining $\underline{h}(\varphi \to x|y)_{\mathbf{M}} = \underline{h}(y)_{\mathbf{M}} = [h,y] = [h, \varphi \to x|y]$, as desired.

(vii) *Functional terms $F(\bar{x}) \in Var_\Sigma$* for some $\bar{x} = (x_1,\ldots,x_n)$. We assume by the induction hypothesis that $\underline{h}(\bar{x})_{\mathbf{M}} = [h,\bar{x}]$, and using the semantics of functional terms and the definition of $I(F)$ in our history model, we obtain $\underline{h}(F(\bar{x}))_{\mathbf{M}} = (I(F))(h(\bar{x})) = (I(F))[h,\bar{x}] = [h,F(\bar{x})]$, as desired.

(viii) *Least-value terms $\mu_N x_A^\varphi \in Var_\Sigma$*. Using the notation $Val_h(x_A^\varphi) := \{[h',x] \in D : h' \sim_A h, \varphi \in last(h')\}$ and the induction hypothesis for Claims (1) and (2), we have $Val_h(x_A^\varphi) = Val(x_A^\varphi)_{h,\mathbf{M}}$, where $Val(x_A^\varphi)_{h,\mathbf{M}} := \{\underline{h'}(x)_{\mathbf{M}} : h' \sim_A h, h' \models_{\mathbf{M}} \varphi\}$ is the notation from Section 3. We distinguish *three subcases*:

Case 1: $[h, \mu_N x_A^\varphi] = \top_{\mathbf{D}}$. By the Trivial-Value Lemma 5.21, we have $Val_h(x_A^\varphi) \subseteq \{\top_{\mathbf{D}}\}$, and hence $min_N Val_h(x_A^\varphi) = \top_{\mathbf{D}}$ (since by definition we have $min_N\emptyset = \min_N\{\top_{\mathbf{D}}\} = \top_{\mathbf{D}}$). Given this, we obtain that $\underline{h}(\mu_N x_A^\varphi)_{\mathbf{M}} = min_N Val(x_A^\varphi)_{h,\mathbf{M}} = min_N Val_h(x_A^\varphi) = \bot_{\mathbf{D}} = [h, \mu x_A^\varphi]$, as desired.

Case 2: $[h, \mu_N x_A^\varphi] = \bot_{\mathbf{D}}$. By the Undefined-Minimum Lemma 5.22, we have either $\bot_{\mathbf{D}} \in Val_h(x_A^\varphi)$ or else $|Val_h(x_A^\varphi)| > N$. In both cases, we get $min_N Val_h(x_A^\varphi) = \bot_{\mathbf{D}}$ by definition, so we obtain $\underline{h}(\mu_N x_A^\varphi)_{\mathbf{M}} = min_N Val(x_A^\varphi)_{h,\mathbf{M}} = min_N Val_h(x_A^\varphi) = \bot_{\mathbf{D}} = [h, \mu_N x_A^\varphi]$, as desired.

Case 3: $[h, \mu_N x_A^\varphi] \neq \top_{\mathbf{D}}, \bot_{\mathbf{D}}$. By Lemma 5.23, we have both $|Val_h(x_A^\varphi)| \leq N$ and $[h, \mu_N x_A^\varphi] = min Val_h(x_A^\varphi)$, i.e., $[h, \mu_N x_A^\varphi] = min_N Val_h(x_A^\varphi)$, thus $\underline{h}(\mu_N x_A^\varphi)_{\mathbf{M}} = min_N Val(x_A^\varphi)_{h,\mathbf{M}} = min_N Val_h(x_A^\varphi) = [h, \mu_N x_A^\varphi]$, as desired. □

Proof of Proposition 5.9: If S is a quasi-model for φ_0, then by applying claim 1 of the Truth Lemma 5.24 to φ_0 and to the history $h_0 := (\Delta_0)$, we conclude that $h_0 \models \varphi_0$ in our history-based model $\mathbf{M}$ above.

6 Completeness and Decidability Proofs for *LDAE*

In this section we prove Theorem 4.6. We first establish our co-expressivity result: *LDAE and LDA are provably co-expressive*. For this, we need a few preliminary notions and results.

Reducible expressions A formula θ in *LDAE* is said to be *reducible* if it is provably equivalent to a 'static' formula; i.e., if there exists some formula θ' in the static fragment *LDA* s.t. $\vdash \theta \leftrightarrow \theta'$ is provable in **LDAE**. A term $x \in Var_{LDAE}$ is *reducible* if it is provably equal to a static term; i.e., if there exists some $x' \in Var_{LDA}$ s.t. $\vdash x = x'$ is provable in **LDAE**. Finally, an event $\mathbf{e} = (\mathbf{E}, e)$ is *reducible* if all preconditions pre_f and all post-conditions of the form $\underline{e}(p)$ or $\underline{e}(v)$ are reducible (for all $f \in E$, $p \in Prop$ and $v \in V$).

Lemma 6.1. (Replacement of Equivalents) *Suppose that* $\vdash \varphi \leftrightarrow \varphi'$, $\vdash \theta \leftrightarrow \theta'$, $\vdash x = x'$, $\vdash y = y'$ *and* $\vdash x_i = x_i'$ *(for all* $i = 1,n$*) are provable in* **LDAE**. *Then the following are also provable in* **LDAE**:

1. $\vdash \varphi{\rightarrow}x|y = \varphi{\rightarrow}x'|y'$
2. $\vdash F(x_1,\ldots,x_n) = F(x_1',\ldots,x_n')$
3. $\vdash \mu_N x_A^{\varphi} = \mu_N (x')_A^{\varphi'}$
4. $\vdash Px_1\ldots x_n \leftrightarrow Px_1'\ldots x_n'$
5. $\vdash (\varphi \rightarrow \theta) \leftrightarrow (\varphi' \rightarrow \theta')$
6. $\vdash K_A\varphi \leftrightarrow K_A\varphi'$
7. $\vdash [\mathbf{e}]\varphi \leftrightarrow [\mathbf{e}]\varphi'$
8. $\vdash \mathbf{e}(x) = \mathbf{e}(x')$

We first prove a preliminary "one-step reduction" result:

Lemma 6.2. *Let* $\mathbf{e}$ *be any reducible event. Then, for every 'static' formula* θ *in LDA and every 'static' term* $x \in Var_{LDA}$*, the formula* $[\mathbf{e}]\theta$ *and the term* $\mathbf{e}(x)$ *are also reducible.*

Proof. Since $\mathbf{e} = (\mathbf{E}, e)$ is reducible, its precondition pre_e is also reducible. Let us fix some static formula ρ s.t. $\vdash pre_e \leftrightarrow \rho$ is provable in **LDAE**. We'll prove both claims simultaneously, by induction on sub-expression complexity (-and we'll make liberal use of Lemma 6.1, without explicitly mentioning it):

For $\theta := p$: since e is reducible, there exists some static formula θ' s.t. $\vdash \underline{e}(p) \leftrightarrow \theta'$. Using the Change of Facts axiom, we can see that $[\mathbf{e}]p$ is provably equivalent to $\rho \rightarrow \theta'$.

For $\theta := Px_1\ldots x_n$: by the induction hypothesis, for all $i = 1,n$ we have $\vdash \mathbf{e}(x_i) = x_i'$ for some static terms $x_1',\ldots,x_n' \in Var_{LDA}$. Using also the Indiscernability axiom and the Property Change reduction axiom, $[\mathbf{e}]\theta$ is provably equivalent to $pre_e \rightarrow Px_1'\ldots x_n'$, and hence also to $\rho \rightarrow Px_1'\ldots x_n'$.

For $\theta := \varphi \rightarrow \psi$ is similar: by induction, there exist static formulas φ_e and ψ_e s.t. $\vdash [\mathbf{e}]\varphi \leftrightarrow \varphi'$ and $\vdash [\mathbf{e}]\psi \leftrightarrow \psi'$ are provable. Using the $[\mathbf{e}]$-Distributivity axiom, we obtain $\vdash [\mathbf{e}]\theta \leftrightarrow (\varphi' \rightarrow \psi')$.

For $\theta := K_A\varphi$: by induction, for each $f \in E$ there exists some static formula φ_f s.t. $\vdash [\mathbf{f}]\varphi \leftrightarrow \varphi_f$. Putting this together with the Knowledge Update axiom, we get $\vdash [\mathbf{e}]\theta \leftrightarrow \bigwedge\{\rho \rightarrow K_{\underline{e}(A)}\varphi_f : f \sim_A e\})$.

For $x := v$ (basic variable): since e is reducible, there exists some static term x' s.t. $\vdash \underline{e}(v) = x'$. Using this and the Value Change axiom, we obtain that $\vdash \mathbf{e}(x) = (\rho{\rightarrow}x'|\top)$, so $\mathbf{e}(x)$ is reducible.

For $x := F(x_1,\ldots,x_n)$: by the induction hypothesis, there exist static terms $x_1',\ldots,x_n'$ s.t. $\vdash \mathbf{e}(x_i) = x_i'$ for all $i \leq n$. Using the Functional Change reduction axiom, we obtain $\vdash \mathbf{e}(x) = (\rho{\rightarrow}F(x_1',\ldots,x_n'))$.

For $x := \varphi{\rightarrow}y|z$: by the induction hypothesis, there exist static terms y',z' and static formula φ', s.t. $\vdash \mathbf{e}(y) = y', \vdash \mathbf{e}(z) = z'$ and $\vdash [\mathbf{e}]\varphi \leftrightarrow \varphi'$ are provable. Using these, as well as the Case Change reduction axiom, we obtain $\vdash \mathbf{e}(x) = (\varphi'{\rightarrow}y'|z')$, and so $\mathbf{e}(x)$ is reducible.

For $x := \mu_N y_A^{\varphi}$: first note that by definition, the reducibility of $\mathbf{e} = (\mathbf{E}, e)$ implies the reducibility of *all* $\mathbf{f} = (\mathbf{E}, f)$ with $f \in E$. By the induction hypothesis and Proposition 4.5, for every $f \in E$ there exist a

static formula φ_f and a static term y_f, s.t. $\vdash \langle \mathbf{f} \rangle \varphi \leftrightarrow \varphi_f$ and $\vdash \mathbf{f}(y) = y_f$. Thus, if we put

$$\eta \ := \ \{n_N(y_f)^{\varphi_f}_{\underline{f}(A)} : f \sim_A e, n \leq N\}^{\Diamond} \leq N,$$

then $\vdash \eta \leftrightarrow Val^N_e(x_A\varphi)^{\Diamond} \leq N$ is provable in **LDAE**. Using the Min. Val. Change axiom, $\mathbf{e}(\mu_N y^{\varphi}_A)$ is provably equal to $\rho \rightarrow \Big(\eta \rightarrow min\{\mu_N(y_f)^{\varphi_f}_{\underline{f}(A)} : f \sim_A e\} | \bot\Big)$. □

Using this, we can establish our full reduction result:

Lemma 6.3. (Co-expressivity of *LDAE* and *LDA*) *All expressions (formulas, terms and events) α of LDAE are reducible. As a consequence, LDAE and LDA have the same expressive power.*

Proof. Induction on the subexpression-complexity of the expression α. The base cases $\alpha := p \in Prop$ and $\alpha := v \in V$ are trivial. The cases $\alpha := \varphi \rightarrow x|y$, $\alpha := F(\bar{x})$, $\alpha := \mu_N x^{\varphi}_A$, $\alpha := P\bar{x}$, $\alpha := \varphi \rightarrow \theta$ and $\alpha := K_A\varphi$ are straightforward: use the induction hypothesis and parts 1-6 of Lemma 6.1.

For $\alpha := [\mathbf{e}]\varphi$: by induction, $\mathbf{e}$ and φ are reducible, so there exists static φ' s.t. $\vdash \varphi \leftrightarrow \varphi'$. By Lemma 6.1.7, $\vdash \alpha \leftrightarrow [\mathbf{e}]\varphi'$, and by Lemma 6.2 $[\mathbf{e}]\varphi'$ is reducible (since $\mathbf{e}$ is reducible and φ' is static), so $\vdash [\mathbf{e}]\varphi' \leftrightarrow \varphi''$ for some static φ''. Using transitivity of equivalence, we get $\vdash \alpha \leftrightarrow \varphi''$, so α is reducible.

For $\alpha := \mathbf{e}(x)$: by induction, $\mathbf{e}$ and x are reducible, so there exists a static term x' s.t. $\vdash x = x'$. By Lemma 6.1.8, we have $\vdash \alpha = \mathbf{e}(x')$, and by Lemma 6.2 $\mathbf{e}(x')$ is reducible (since $\mathbf{e}$ is reducible and x' is static), so $\vdash \mathbf{e}(x') = x''$ for some static term x''. Using transitivity of equality, we obtain $\vdash \alpha = x''$.

Finally, the case $\alpha := \mathbf{e} = (\mathbf{E}, e)$ is straightforward: for all $f \in E$, $p \in Prop$ and $v \in V$, the preconditions pre_f and postconditions $\underline{f}(p)$ and $\underline{f}(v)$ are $< \mathbf{e}$ in the subexpression complexity order, so they are all reducible (by the induction hypothesis), and hence $\mathbf{e}$ is also reducible (by definition). □

Proof of Theorem 4.6: Completeness follows immediately from Lemma 6.3, the soundness of **LDAE**, and the completeness of the system **LDA** (Theorem 4.2); decidability follows from the decidability of the static logic *LDA* (Theorem 4.2) and the co-expressivity of *LDA* and *LDAE* (Lemma 6.3).

7 Conclusions and Future Work

In this paper we axiomatize a decidable but richly expressive logic, that deals with a large class of data-exchange events, while capturing group knowledge of both propositional and non-propositional data, as well as a group's ability to narrow down the values of a variable to finitely many possibilities.

There are a number of things still left to do. As already mentioned, one can add common knowledge operators $C_A\varphi$, but pre-encoding their dynamics requires a generalization to polyadic conditionals $C^e_A\varphi$ as in [11]. We leave this for a journal version, where we also plan to explore the relationships of our formalism with the Logic of Functional Dependence [13] and Graded (Multi)Modal Logic.

Our axioms are somewhat complicated due to the fact that we did *not* succeed to prove FMP (Finite Model Property) for our logics. This would have allowed us to replace cutoff-minimization $\mu_N x^{\varphi}_A$ by plain minimization $min\, x^{\varphi}_A$ (over all A-possible x-values given φ), which would greatly simplify our axioms. On the other hand, we *don't* have a counterexample to FMP, so this issue is still an open question.

References

[1] Thomas Agotnes & Yi N. Wang (2017): *Resolving Distributed Knowledge*. *Artificial Intelligence* 252, pp. 1–21, doi:10.1016/j.artint.2017.07.002.

[2] Alexandru Baltag (2010): *Presentation: "What is DEL good for?"*. In: *ESSLI Workshop on 'Logic, Rationality and Intelligent Interaction' (organized by J. van Benthem and E. Pacuit)*. Available at `https://cs.stanford.edu/~epacuit/lograt/wkshp-esslli2010.html`.

[3] Alexandru Baltag (2016): *To Know is to Know the Value of a Variable*. In: *Advances in Modal Logic 2016*, 11, College Publications, pp. 135–155. ISBN:978-1-84890-201-5.

[4] Alexandru Baltag, Rachel Boddy & Sonja Smets (2018): *Group Knowledge in Interrogative Epistemology*. In: *Outstanding Contributions to Logic*, 12, Springer, pp. 131–164, doi:10.1007/978-3-319-62864-6_5.

[5] Alexandru Baltag, Lawrence Moss & Slawomir Solecki (1998): *The Logic of Public Announcements, Common Knowledge, and Private Suspicions*. In: *Proceedings TARK 98*, Morgan Kaufmann, pp. 43–56.

[6] Alexandru Baltag, Lawrence Moss & Slawomir Solecki (2023): *Logics for epistemic actions: completeness, decidability, expressivity*. *Logics* 1(2), pp. 97–147, doi:10.3390/logics1020006.

[7] Alexandru Baltag & Lawrence S. Moss (2004): *Logics for Epistemic Programs*. *Synthese* 139(2), pp. 165–224, doi:10.1023/B:SYNT.0000024912.56773.5e.

[8] Alexandru Baltag, Lawrence S. Moss & Hans van Ditmarsch (2008): *Epistemic Logic and Information Update*. In P. Adriaans & J. van Benthem, editors: *Philosophy of Information, part of series: Handbook of the Philosophy of Science*, 8, Elsevier, pp. 361–465, doi:10.1016/C2009-0-16481-4.

[9] Alexandru Baltag & Bryan Renne (2016): *Dynamic Epistemic Logic*. In Edward N. Zalta, editor: *The Stanford Encyclopedia of Philosophy*, Winter 2016 edition, Metaphysics Research Lab, Stanford University. Available at `https://plato.stanford.edu/archives/win2016/entries/dynamic-epistemic/`.

[10] Alexandru Baltag & Sonja Smets (2020): *Learning what Others Know*. In L. Kovacs & E. Albert, editors: *LPAR23 proceedings, EPiC Series in Computing*, 73, pp. 90–110, doi:10.48550/arXiv.2109.07255.

[11] Alexandru Baltag & Sonja Smets (2024): *Logics for Data Exchange and Communication*. In: *Advances in Modal Logic 2024*, 15, College Publications, pp. 147–169. ISBN:978-1-84890-467-5.

[12] Alexandru Baltag & Sonja Smets (2025): *Group Knowledge of Hypothetical Values*. In: *Electronic Proceedings in Theoretical Computer Science (TARK 2025)*, 437, pp. 135–154, doi:10.4204/EPTCS.437.15.

[13] Alexandru Baltag & Johan van Benthem (2021): *A Simple Logic of Functional Dependence*. *Journal of Philosophical logic* 50, pp. 939–1005, doi:10.1007/s10992-020-09588-z.

[14] Johan van Benthem & Ştefan Minică (2009): *Toward a Dynamic Logic of Questions*. In: *LORI 2009 Proceedings, Lecture Notes in Computer Science*, 5834, Springer, pp. 27–41, doi:10.1007/978-3-642-04893-7_3.

[15] Rachel Boddy (2014): *Epistemic Issues and Group Knowledge*. Master of logic thesis, mol-2014-03, University of Amsterdam. Available at `https://eprints.illc.uva.nl/id/eprint/921/`.

[16] Yifeng Ding (2016): *The axiomatization and complexity of Knowing-What-Logic on model class K, Epistemic Logic with Functional Dependency Operator*. *arXiv 1609.07684*, doi:10.48550/arXiv.1609.07684.

[17] Hans van Ditmarsch, Wiebe van der Hoek & Barteld Kooi (2008): *Dynamic Epistemic Logic*. 337, Springer, doi:10.1007/978-1-4020-5839-4.

[18] Jan van Eijck, Malvin Gattinger & Yanjing Wang (2017): *Knowing Values and Public Inspection*. In: *Lecture Notes in Computer Science*, 10119, Springer, pp. 77–90, doi:10.1007/978-3-662-54069-5_7.

[19] Roosmarijn Goldbach (2015): *Modelling Democratic Deliberation*. Master of logic thesis, mol-2015-05, University of Amsterdam. Available at `https://eprints.illc.uva.nl/id/eprint/946/`.

[20] Tao Gu & Yanjing Wang (2016): *"Knowing value" logic as a normal modal logic*. In: *Advances in Modal Logic*, 11, College Pub., pp. 362–381, doi:10.48550/arXiv.1604.08709. ISBN:978-1-84890-201-5.

[21] Bo Hong (2023): *Knowing the Value of a Predicate*. In: *Proceedings of LORI 2023, Lecture Notes in Computer Science*, 14329, pp. 149–166, doi:10.1007/978-3-031-45558-2_12.

[22] Jan Plaza: *Logics of Public Communication*. *Synthese* 158, pp. 165–179, doi:10.1007/s11229-007-9168-7. Republication of J. Plaza (1989), Proceedings 4th International Symposium on Methodologies for Intelligent Systems, pp.201-216.

[23] Johan van Benthem (2011): *Logical Dynamics of Information and Interaction*. Cambridge University Press, UK, doi:10.1017/CBO9780511974533.

[24] Johan van Benthem & Ştefan Minică (2012): *Toward a Dynamic Logic of Questions*. Journal of Philosophical Logic 41, pp. 633–669, doi:10.1007/s10992-012-9233-7.

[25] Yanjing Wang (2018): *Beyond Knowing That: A New Generation of Epistemic Logics*. In: Outstanding Contributions to Logic, 12, Springer, pp. 499–533, doi:10.1007/978-3-319-62864-6_21.

[26] Yanjing Wang & Jie Fan (2013): *Knowing that, Knowing what, and Public Communication: Public Announcement Logic with Kv Operators*. In: Proceedings of IJCAI 2013, AAAI Press, pp. 1147–1154, doi:10.5555/2540128.2540293.

[27] Yanjing Wang & Jie Fan (2014): *Conditionally knowing what*. In: Advances in Modal Logic, 10, College Publications, pp. 569–587. ISBN:978-1-84890-151-3.