Logics for Data Exchange and Communication

Alexandru Baltag [1] Sonja Smets [2]

University of Amsterdam
Institute for Logic, Language and Computation, NL

Abstract

We present a new family of dynamic logics that can model complex scenarios of data exchange and communication. In the context of multi-agent systems, these are acts by which individual agents or groups can publicly or privately access all the information stored at specific locations. In addition to having the full power of standard dynamic epistemic logics (DEL) to model propositional communication, our logics can handle the type of information exchanges in which the data that is being communicated is non-propositional. To axiomatize these logics, in the presence of common and distributed knowledge modalities, we first have to enrich them with polyadic modalities, representing various complex levels of conditional group knowledge. While expressible in the extension of PDL with intersection (which is decidable but does not have FMP), these logics are better behaved: we use an innovative method to show that they do have FMP. We axiomatize these static logics using PDL-type systems, then axiomatize the corresponding dynamic data-exchange logics and prove their decidability, by reducing them to their static base via Recursion/Reduction Axioms.

Keywords: Dynamic Epistemic Logic; Communication protocols; Common Knowledge; Distributed Knowledge; Distributed computing.

1 Introduction

In this paper, we provide a formal setting for reasoning about interactions in which agents gain (or are given) access to information sources, thus being potentially able to 'read' all the data available at that source. More specifically, we assume that at any given time information is locally stored (in both propositional and non-propositional form, e.g. numerical data) at specific locations, databases or 'sites' (e.g. files, folders, data sets, websites, etc.). Each such database can itself be thought of as being associated with an 'agent': it may actually represent the knowledge base of a (natural or artificial) agent, or else we can think of the source itself as an abstract 'agent' ('possessing' all data that is stored at that site, as well as all the information that is logically entailed by these data). The same piece of data can be stored at multiple locations,

[1] thealexandrubaltag@gmail.com.

[2] s.j.l.smets@uva.nl.

and each location can store multiple pieces of data. A 'reading' agent may gain access to another agent's data and 'copy' them in her own database, either because she is granted such access by the source-agent, or because she illegally gain such access ('hacking'). Several agents may simultaneously access multiple sources. So a data-exchange event can be *public* (when it is common knowledge that the information stored at a given location has become visible to everybody, cf. the WikiLeaks case), or *semi-public* (when it is accessible only to some agents, but this fact is common knowledge), or *semi-private* (when some databases are publicly accessed, others are privately accessed, and these facts are known to some but not to all of the agents), or even *fully private* (when both the information and the access are unknown to outsiders).

Standard temporal-epistemic logics [20,23], and dynamic approaches e.g. Public Announcement Logic (PAL) [24] and Dynamic Epistemic Logic (DEL) [7,18,15,9], deal with complex communication scenarios only by *always making explicit the specific sentences* that are being communicated. So all data is assumed to be propositional, which is not always convenient. Moreover, sometimes there might be *no sentences* in our language that capture all relevant data. And, even if such sentences exist, they may be huge, or may be state-dependent: there may be no single formula that uniformly captures (across all the states of the model) the information that is being learnt. [3]

In [12], we introduced a general setting for representing a large class of such data-exchange events (called 'reading events' in that paper), together with a number of dynamic-epistemic logics to reason about them. However, the setting and the results in [12] suffered from a number of limitations. First, the setting did *not* constitute a proper generalization of classical DEL, since it did not fully include it: unlike the DEL epistemic events in [7], the 'reading events' in [12] did not come with (propositional) *preconditions*. As such, these events could happen in *any* state, and thus they *did not carry any specific propositional information*! So, for instance, public announcements $!\varphi$ (of some proposition φ) were not representable as reading events: hence, the dynamic logics in [12] did not formally include the public announcement logic PAL [4], and could not embed the logic of public announcements and common knowledge PAC.

Second, completeness and decidability were proved in [12] only for an even more restricted class of events (-the *'semi-public' ones*). Admittedly, a more general axiomatization was proposed (with a Dynamic Induction Rule in the style of [7]), but its completeness was left as a conjecture. [5]

Third, though decidability was obtained for the logics in [12], this was done via establishing only a Finite 'Pseudo-Model' Property (and then unravelling

[3] E.g. when an agent a learns another agent's secret password x_b, the information that is being gained can be expressed by a formula of the form $x_b = n$, where n is the natural number corresponding to the actual password, but... this number varies across possible states.

[4] Though PAL is not formally included in any of the logics in [12], it is known to be co-expressive with basic modal logic, and it is thus embeddable in those logics

[5] While we believe this conjecture is correct, we chose a different path in this paper, by enlarging the static base to obtain a more transparent axiomatization.

the pseudo-model into an infinite model, to obtain completeness). The Finite Model Property remained open for these logics, and this was *not* a trivial issue: the only known logic that embeds all the logics in [12] is PDL with Intersection, which is known *not* to have FMP (though it is decidable).

In this paper, we address all these limitations. The logics considered here combine classical DEL and the setting in [12]: by endowing data-exchange events with preconditions, they can represent the acquisition of *both* propositional *and* non-propositional information. To obtain simple axiomatizations, we extend the static base logics, in a spirit similar to the work on 'Epistemic PDL' (E-PDL), used in [16] as a static base for classical DEL [6], as well as to the work on 'Group Epistemic PDL' (GE-PDL) [25]. [7] But we achieve this by adapting and extending a very recent approach to classical DEL ([5], unpublished manuscript, under review): the idea is to introduce static n-ary modalities for "event-conditional common knowledge" $C^e_A\overline{\varphi}$ (where $\overline{\varphi}$ is an n-tuple of formulas of the same length as the size $n = |E|$ of the corresponding event model), that generalize both the notion of 'relativized common knowledge' $C^\theta_A\varphi$ from [16] and the common distributed knowledge $Cd_{A_1,...,A_n}\varphi$ from [12]. These operators can represent infinite (regular) iterations of distributed knowledge and conditionalization, expressing *regular levels of conditional group knowledge*, that generalize the levels in [22]. Though expressible in GE-PDL (and thus also in PDL with Intersection, but *not* in E-PDL), we directly axiomatize them (as generalizations of common knowledge), and use them to reduce the dynamic logics to their static base, via recursion/reduction laws. In contrast to both E-PDL and GE-PDL [8], the recursion axioms obtained in this way are of great transparency and simplicity.

Finally, we prove FMP for the resulting logics, by using an innovative method, which to our knowledge has never been used before: while based on on the standard technique of making copies of the same worlds, we avoid infinite unraveling by using elementary notions of modular arithmetic (-congruences modulo a natural number) to directly convert every finite pseudo-model into a finite model (in which distributed knowledge is 'standard').

2 Data-exchange logics

We introduce a family of logics for data-exchange and communication. Following the approach to DEL syntax introduced in [6], and then adopted in a number of papers [10,8,5], our languages will be parametrized by data-exchange *signatures*, corresponding to naturally closed classes of data-exchange events.

---

[6] E-PDL is just PDL with an epistemic interpretation: the basic modalities $[a]\varphi$ are interpreted as "knowledge by agent a", based on some indistinguishability relation $\sim_a$, $[\bigcup_{a\in A} a]^*\varphi$ captures "common knowledge of φ in the group A, $[\bigcup_{a\in A}(a;?\theta)]^*\varphi$ captures conditional (or "relativized") common knowledge of φ given θ, etc.

[7] GE-PDL is an extension of E-PDL, based on relations $\sim_A$ for groups A of agents (obtained by taking intersections $\bigcap_{a\in A} \sim_a$), then closing them under the PDL operations.

[8] Though of great theoretical importance, the reduction axioms in [16] and [25] are extremely complex, taking several pages to state.

2.1 Data-exchange frames and signatures

Throughout this paper, we fix a finite set $\mathcal{A}$ of *agents* $a, b, \ldots$, and a finite set P of *atomic sentences* $p, q, \ldots$. We use capital letters $A, B, \ldots$ for *sets of agents*.

Epistemic frames and epistemic models An *epistemic frame* is a structure $\mathbf{S} = (S, \sim)$, consisting of a set S of 'states' $s \in S$, and a map $\sim$ assigning to each agent $a \in \mathcal{A}$ some equivalence relation $\sim_a \subseteq S \times S$, called *(epistemic) indistinguishability relation*. An *epistemic (state) model* $\mathbf{M} = (\mathbf{S}, \|\bullet\|)$ consists of an epistemic frame $\mathbf{S} = (S, \sim)$ and a *valuation map* $\|\bullet\| : P \to \mathcal{P}(S)$, associating to each atom p some set of states $\|p\| \subseteq S$.

Data-exchange frames A *data-exchange frame* is a structure $\mathbf{\Sigma} = (\Sigma, \sim, \bullet_\bullet)$, consisting of a *finite* epistemic frame $(\Sigma, \sim)$ (whose 'states' $\sigma \in \Sigma$ are called *action types*) and an *access map* $\bullet(\bullet)$, associating to each agent $a \in \mathcal{A}$ and action type $\sigma \in \Sigma$ some set of agents $\sigma(a) \subseteq \mathcal{A}$, satisfying two constraints:

(i) *agents access their own data*: $a \in \sigma(a)$ for all $a \in A$;

(ii) *agents know what databases they access*: $\sigma \sim_a \lambda$ implies $\sigma(a) = \lambda(a)$.

The notation for access maps can be naturally extended to *sets of agents*, by putting $\sigma(A) := \bigcup_{a \in A} \sigma(a)$, for any $A \subseteq \mathcal{A}$.

We follow here the approach in [6], by distinguishing between action 'types' and full-fledged 'events': action-types are *not* yet fully specified events (since the propositional information carried by the event is not yet specified). An action-type $\sigma \in \Sigma$ represents a 'type' of data-exchange, in which every agent a gains access to all data stored in the locations (or possessed by the agents) in $\sigma(a)$. The epistemic relation $\sim_a$ represents agent a's knowledge about the type of the on-going exchange: a cannot distinguish type σ from types $\sigma' \sim_a \sigma$.

We now proceed to give examples of data-exchange frames.

Example: Public Announcements The *public announcement frame* $\mathbf{\Sigma}_! = (\{!\}, \sim, \bullet_\bullet)$ is a frame with *only one action-type*, typically denoted by !, with *loops for all agents* (i.e. every $\sim_a$ is just the identity relation), and with the access map given by singletons $!(a) = \{a\}$ for all $a \in \mathcal{A}$.

Example: Public and Semi-Public Sharing (*"Tell Us All You Know"*) For sets $A, B \subseteq \mathcal{A}$, the *semi-public sharing frame* $\mathbf{\Sigma}_{!(A:B)}$ has only one action-type, denoted by $!(A : B)$, with loops for all agents and with access map given by: $!(A : B)(a) = B \cup \{a\}$ for all $a \in A$, and $!(A : B)(a) = \{a\}$ for $a \notin A$. It is common knowledge that all agents in A gain access to the databases of all agents in B. A special case is *B-public sharing* frame $\mathbf{\Sigma}_{!B} := \mathbf{\Sigma}_{!(\mathcal{A}:B)}$, obtained by taking $A = \mathcal{A}$ for the unique action-type $!B := !(\mathcal{A} : B)$: all agents in B publicly share their information (with everybody).

Example: Semi-public Data-Exchange Frames We can generalize the previous examples to a wider class of frames: for any map $\alpha : \mathcal{A} \to \mathcal{P}(\mathcal{A})$, the associated *semi-public exchange frame* $\mathbf{\Sigma}_{!\alpha}$ consists of *only one-action type* $!\alpha$, with *loops for all agents* (i.e. every $\sim_a$ is just the identity relation), and access map given by putting $(!\alpha)(a) = \alpha(a) \cup \{a\}$ for all $a \in \mathcal{A}$. So, in semi-public

actions, it is common knowledge who can access/read whose database, but the actual data may not be necessarily seen by all agents: the *type* of data exchange is public, but data themselves may be only privately available. A more compact description of the semi-public frames $!\alpha$ is obtained by grouping together in separate groups all the agents a having the same $\alpha(a)$. For each $n \geq 0$ and each n-sized family $\{(A_i, B_i) : i \leq n\} = \{(A_1, B_1), \ldots, (A_n, B_n)\} \subseteq \mathcal{P}(\mathcal{A}) \times \mathcal{P}(\mathcal{A})$ consisting of pairs (A_i, B_i) of groups $A_i, B_i \subseteq \mathcal{A}$, the frame $\mathbf{\Sigma}_{!\{A_1:B_1,\ldots,A_n:B_n\}}$ is just the semi-public frame whose unique action-type, denoted by $!\{(A_i : B_i) : i \leq n\} = !\{A_1 : B_1, \ldots, A_n : B_n\}$, has the access map given by $!\{A_1 : B_1, \ldots, A_n : B_n\}(a) = \{a\} \cup \bigcup\{B_i : i \leq n, a \in A_i\}$ for all $a \in \mathcal{A}$. This is semi-public sharing *between multiple groups*: it is common knowledge that all agents in each group A_i gain access to all data of the agents in the corresponding B_i.

Special Case: Sharing *within* Groups For sets $A_1, \ldots, A_n \subseteq \mathcal{A}$, the *multiple in-group sharing frame* is the special case of the previous type of frame in which we take $B_i = A_i$ for all i. Its action-type, denoted by $!\{A_1, \ldots, A_n\} = !\{A_1 : A_1, \ldots, A_n : A_n\}$, describes a type of data exchange in which it is common knowledge that every agent in each group A_i shares all her knowledge with the agents in the same group A_i. The special case $n = 1$ is the so-called *resolution* action $!\{A\}$ (in which it is common knowledge that all agents in A share their information with each other), first introduced in [2]. [9]

Example: Private Announcement with Common Knowledge of Alternatives. Given $A \subseteq \mathcal{A}$ and $n \geq 1$, the *A-private announcement* frame $\mathbf{\Sigma}_{!_{A,n}} = (\{!^1_{A,n}, \ldots, !^n_{A,n}\}, \sim, \bullet_{\bullet})$ (with *common knowledge of n alternatives* has a domain consisting of n action-types; the access map is $!(a) = \{a\}$ for all a; and $\sim_a$ is the identity relation for $a \in A$, and the universal relation for $a \notin A$.

Example: Secret Hacking Suppose that it is common knowledge that the only possible data-collecting action at the moment is the one by which an agent a (the "hacker") tries to gain access to another agent b's information base (the "source"). Only the hacker knows whether or not he actually succeeds to gain access to b's data (though everybody else is aware of this possibility). We can represent this scenario in a data-exchange frame with two action-types $SH_{a:b}$ (successful, secret hacking) and $UH_{a:b}$ (unsuccessful hacking attempt). The access maps are: $SH_{a:b}(a) = \{a, b\}$, $SH_{a:b}(b) = \{b\}$, and $SH_{a:b}(c) = \{c\}$ for all other agents $c \neq a, b$; while $UH_{a:b}$ is just the identity map on agents. Besides loops for all agents, we have $SH_{a:b} \sim_b UH_{a:b}$, and $SH_{a:b} \sim_c UH_{a:b}$ for all 'outsiders' $c \neq a, b$ (i.e. only a can distinguish between the two possibilities).

Secret Detection of Hacking If we modify the previous scenario to allow the possibility that the source-agent b might detect the hacking attack (so that she might come to know that she is being hacked by a); to have an S5 frame, we assume that everybody is aware of this possibility, but only b knows whether she actually detects an attack or not. The data-exchange frame will now contain three action-types $DH_{a:b}$ (detected hacking), $SH_{a:b}$ (successful, i.e.

[9] Resolution was previously considered under different names in [13] and [11].

undetected, hacking) and $UH_{a:b}$ (unsuccessful hacking attempt). The access maps for $SH_{a:b}$ and $UH_{a:b}$ are the same as in the previous example, and the access map for $DH_{a:b}$ is the same as for $SH_{a:b}$. Besides loops for all agents, we also have $DH_{a:b} \sim_a SH_{a:b}$ (i.e. a doesn't know whether his hacking is detected or not) and $SH_{a:b} \sim_b UH_{a:b}$ (b can't distinguish between undetected hacking and unsuccessful hacking attempt), while $\sim_c$ is the universal relation for all outsiders $c \neq a, b$ (they can't distinguish between any of the three cases).

Secret Mutual Hacking Suppose that agents a and b simultaneously attempt to hack each other's database, though their attempts might be unsuccessful. We assume now that the attempts cannot be detected, so only the hacker knows whether a hacking attack is successful or not. The data-exchange frame contains four action-types $SH_{a:b,b:a}$ (both hacking attacks are successful), $SH_{a:b}$ (only a succeeds to hack b's database), $SH_{b:a}$ (only b succeeds in her hacking attack on a) and UH (all hacking is unsuccessful). The access map for $SH_{a:b}$ is the same as in previous examples, and by switching the roles of a and b in it we obtain the access map for $SH_{b:a}$; the access map for UH is the same as the one for $UH_{a:b}$ in the previous examples; finally, the access map for $SH_{a:b,b:a}$ is $SH_{a:b,b:a}(a) = SH_{a:b,b:a}(b) = \{a, b\}$, and $SH_{a:b,b:a}(c) = \{c\}$ in rest. Besides loops for all agents, we have $SH_{a:b,b:a} \sim_a SH_{a:b} \sim_b UH$ and $SH_{a:b,b:a} \sim_b SH_{b:a} \sim_a UH$, while $\sim_c$ is again the universal relation for $c \neq a, b$.

In order to introduce our key notion of data-exchange signature, we need to first define an operation (*sequential product*) on data-exchange frames.

Product of data-exchange frames Given two data-exchange frames $\mathbf{\Sigma} = (\Sigma, \sim_a, \bullet_a)$ and $\mathbf{\Sigma}' = (\Sigma', \sim'_a, \bullet'_a)$, their *product* is the frame $\mathbf{\Sigma} \times \mathbf{\Sigma}' = (\Sigma \times \Sigma', \sim''_a, \bullet''_a)$, where $\Sigma \times \Sigma'$ is the *Cartesian product* of the two sets of action-types; the epistemic relations are the *product of the underlying relations*: $(\sigma, \sigma') \sim_a (\lambda, \lambda')$ iff $\sigma \sim_a \lambda$ and $\sigma' \sim_a \lambda'$; and the access map is the *composition of the underlying access maps*: $(\sigma, \sigma')(a) := \sigma(\sigma'(a)) = \bigcup_{b \in \sigma'(a)} \sigma(b)$.

Distinguished sets of frames A *distinguished set* is a countable set $\mathcal{F}$ of data-exchange frames s.t. every two frames in $\mathcal{F}$ are mutually disjoint (i.e. have no action-types in common).

Frames up-to-isomorphism Given a class $\mathcal{F}$ of data-exchanges frames, *the class $\mathcal{F}^\dagger$ of all $\mathcal{F}$-frames up-to-isomorphism* is defined by first choosing a representative $\mathbf{\Sigma} = (\Sigma, \sim_a, \bullet_a) \in \mathcal{F}$ from each isomorphism class in the quotient $\mathcal{F}/\cong$ (of $\mathcal{F}$ modulo isomorphism), then replacing it by an isomorphic copy $\mathbf{\Sigma}^\dagger = (\Sigma^\dagger, \sim_a, \bullet_a)$, obtained by taking $\Sigma^\dagger := \{(\sigma, \mathbf{\Sigma}) : \sigma \in \Sigma\}$, and in rest isomorphically copying the relational and access-map structures.

Observation *For every class $\mathcal{F}$ of data-exchanges frames, the class $\mathcal{F}^\dagger$ of all $\mathcal{F}$-frames up-to-isomorphism is a distinguished set.*

Signatures A *data-exchange signature* is a distinguished set $\mathcal{S}$ of data-exchange frames, that contains the public announcement frame $\mathbf{\Sigma}_!$ and is closed under products of frames up-to-isomorphism (i.e., if $\mathbf{\Sigma}, \mathbf{\Sigma}' \in \mathcal{S}$ then there exists $\mathbf{\Sigma}'' \in \mathcal{S}$ s.t. $\mathbf{\Sigma} \times \mathbf{\Sigma}' \cong \mathbf{\Sigma}''$).

Examples are: the *universal signature* $\mathcal{S}_U$, consisting of all data-exchange frames up-to-isomorphism; the *public announcement signature* $\mathcal{S}_!$, which is just the singleton $\{\mathbf{\Sigma}_!\}$; the *semi-public signature* $\mathcal{S}_{SP} = \{\mathbf{\Sigma}_{!\alpha} | \alpha : \mathcal{A} \to \mathcal{P}(\mathcal{A})\}$, consisting of all semi-public frames $\mathbf{\Sigma}_{!\alpha}$. These classes are already closed under products, because of the isomorphisms $\mathbf{\Sigma}_! \times \mathbf{\Sigma}_! \cong \mathbf{\Sigma}_!$ and $\mathbf{\Sigma}_{!\alpha} \times \mathbf{\Sigma}_{!\beta} \cong \mathbf{\Sigma}_{!(\alpha\circ\beta)}$, where $\alpha, \beta : \mathcal{A} \to \mathcal{P}(\mathcal{A})$ are access maps and $\alpha \circ \beta$ is their functional composition, given by putting $(\alpha \circ \beta)(a) = \alpha(\beta(a))$ for all $a \in \mathcal{A}$.

In contrast, the *signature* $\mathcal{S}_{PriCom}$ *of private announcements with common knowledge of alternatives* requires an additional closure step: $\mathcal{S}_{PriCom}$ consists of all *finite products* of frames of the form $!_{A,n}$ (for all $n \in N$ and all $A \subseteq \mathcal{A}$). [10]

2.2 Syntax of the logics $LDC^e\langle e\rangle$

Given a signature $\mathcal{S}$, the *set of formulas* φ of the language $LDC^e\langle e\rangle_{\mathcal{S}}$ is defined by the following recursive syntax

$$\varphi \ ::= \ p \,|\, \neg\varphi \,|\, \varphi \wedge \varphi \,|\, D_A\varphi \,|\, C_A^{\sigma\overline{\varphi}}\overline{\varphi} \,|\, \langle\sigma\overline{\varphi}\rangle\varphi$$

where p are atomic sentences, a are agents, $A \subseteq \mathcal{A}$ are non-empty sets of agents, σ are action-type symbols coming from any data-exchange frame $\mathbf{\Sigma} = (\Sigma, \sim, \bullet_\bullet) \in \mathcal{S}$ in the given signature $\mathcal{S}$, φ are formulas and $\overline{\varphi} = (\varphi_\sigma)_{\sigma\in\Sigma}$ are Σ-indexed tuples of formulas.

The set of *event expressions* (or 'events', for short) of the language $LDC^e\langle e\rangle_{\mathcal{S}}$ consists of all the expressions of the form $e ::= \sigma\overline{\varphi}$, occurring in any of the above formulas. Using letters e for events in $LDC^e\langle e\rangle_{\mathcal{S}}$, the above syntax can be restated in the more familiar form:

$$\varphi \ ::= \ p \,|\, \neg\varphi \,|\, \varphi \wedge \varphi \,|\, D_A\varphi \,|\, C_A^{e}\overline{\varphi} \,|\, \langle e\rangle\varphi$$

We read D_A as *distributed knowledge* (among the group A), and read C_A^e as *event-conditional common knowledge* (among the group A, conditional on event e), while $\langle e\rangle$ are dynamic modalities for events. Note that the operators C_A^e are *polyadic* modalities, which can be applied to corresponding tuples of formulas $\overline{\varphi}$, to produce new formulas $C_A^e\overline{\varphi}$, representing complex "levels of conditional group knowledge" (about the propositions in the tuple $\overline{\varphi}$). These levels can be thought of in terms of regular patterns of infinite nestings of knowledge and conditional operators, that generalize the 'regular levels of knowledge" investigated in [22]. Their *conditional* nature becomes more easily apparent when we think of event expressions $e := \sigma\overline{\theta}$ as being the result of plugging a tuple of formulas $\overline{\theta}$ into an action-type σ (coming from a data-exchange frame $\mathbf{\Sigma} \in \mathcal{S}$). The formulas $C_A^{\sigma\overline{\theta}}\overline{\varphi}$ could thus be seen as *(group epistemic) conditionals* $\overline{\theta}\,\Box\!\!\rightarrow_A^\sigma \overline{\varphi}$, in which the antecedent formulas $\overline{\theta}$ play the role of 'conditions' (or hypothesis) for knowledge, while the formulas in the tuple $\overline{\varphi}$ play the role of epistemic 'conclusions' (that are conditionally known). [11]

[10] More generally, every distinguished set $\mathcal{C}$ generates a signature $\mathcal{S}_{\mathcal{C}}$, obtained by adding the frame $\mathbf{\Sigma}_!$ to $\mathcal{C}$ and closing under finite products.

[11] The alternative conditional notation $C_A^{\sigma\overline{\theta}}\overline{\varphi}$ fits well the above-mentioned interpretation in terms of "levels of conditional group knowledge" within group A.

The *static language* $LDC^e_{\mathcal{S}}$ is the fragment obtained by dropping dynamic modalities $\langle e\rangle\varphi$ from the above recursive grammar of $LDC^e\langle e\rangle_{\mathcal{S}}$. Note that the static fragment is indeed 'static', despite its use of syntactic event expressions: its semantics will only use the evaluation of formulas in the *current model*. In contrast, when occurring in dynamic modalities $\langle e\rangle\theta$, the event expressions $e = \sigma\overline{\varphi}$ will really denote *epistemic events*, living in a DEL-style *event model* (and the semantics of $\langle e\rangle\theta$ will use the evaluation of θ in an *updated model*).

Example: the language of public announcements This is just the special case $LDC^e\langle e\rangle_{\mathcal{S}_!}$, where $\mathcal{S}_! = \{\mathbf{\Sigma}_!\}$ is the public-announcement signature. Unfolding our syntax in this case, we obtain the format:

$$\varphi \;::=\; p \,|\, \neg\varphi \,|\, \varphi \wedge \varphi \,|\, D_A\varphi \,|\, C_A^{!\varphi}\varphi \,|\, \langle !\varphi\rangle\varphi$$

The construct $C_A^{!\varphi}\psi$ is traditionally written $C_A^{\varphi}\psi$, and it is a 'factive' version of van Benthem's *relativized common knowledge* [16] (represented a form of 'conditional' common knowledge).

Example: the language of semi-public data-exchange events Another special case is obtained by taking the semi-public signature $\mathcal{S}_{SP} = \{!\alpha | \alpha : \mathcal{A} \to \mathcal{P}(\mathcal{A})\}$. In a more explicit form, this language is given by the recursive format:

$$\varphi \;::=\; p \,|\, \neg\varphi \,|\, \varphi \wedge \varphi \,|\, D_A\varphi \,|\, C_A^{!\{A_1:B_1,\ldots,A_n:B_n\}\varphi}\varphi \,|\, \langle !\{A_1 : B_1, \ldots, A_n : B_n\}\varphi\rangle\varphi$$

for all sets of pairs $\{(A_1, B_1), \ldots, (A_n, B_n)\} \subseteq \mathcal{P}(\mathcal{A}) \times \mathcal{P}(\mathcal{A})$, of any finite size $n \geq 0$.[12] Here, $!\{A_1 : B_1, \ldots, A_n : B_n\}\varphi$ represents a semi-public exchange event, having a precondition φ (so that this event can only happen when φ is true), and in which it is common knowledge that all agents in each group A_i gain access to all data of the agents in the corresponding group B_i.

Other examples include the *language of private announcements with common knowledge of alternatives* (based on the signature $\mathcal{S}_{PriCom}$), and the universal data-exchange language (based on the universal signature $\mathcal{S}_U$).

Event Models An $\mathcal{S}$*-event model* $\mathbf{E} = (E, \sim_a, \bullet(\bullet), pre)$ is a data-exchange frame $(E, \sim_a, \bullet(\bullet)) \in \mathcal{S}$ together with a 'precondition' map pre, associating to each action-type $e \in E$ some formula pre_e in $LDC^e\langle e\rangle_{\mathcal{S}}$, called the *precondition* of e. When considered within an event model, we refer to the action types $e \in E$ as *events*. Intuitively, pre_e gives the *condition of possibility* for event e.

Event models can be considered as semantic objects in their own respect, see e.g. [7,6,18]. However, in this paper, we will only consider them as *meta-syntactic notations*, that refer to event expressions $\sigma\overline{\varphi}$, and encode their relations to other such event expressions in a conveniently succinct way. We refer to these as *syntactic event models* (though often dropping the 'syntactic' qualification for brevity):

[12] The case $n = 0$ allows us to embed the logic public announcements: for $n = 0$, we get the event $!\emptyset\varphi$, which is the same as the public announcement $!\varphi$.

Syntactic Event Models Given any data-exchange frame $\mathbf{\Sigma} = (\Sigma, \sim, \bullet_\bullet) \in \mathcal{S}$ in our signature, and any Σ-indexed tuple of formulas $\overline{\varphi} = (\varphi_\sigma)_{\sigma\in\Sigma}$, we consider the *corresponding (syntactic) event model* $\mathbf{\Sigma}\overline{\varphi} = (\Sigma\overline{\varphi}, \sim_a, \bullet(\bullet), pre)$ defined as follows: the set of events is $\Sigma\overline{\varphi} := \{\sigma\overline{\varphi} : \sigma \in \Sigma\}$ consists of all event expressions $\sigma\overline{\varphi}$, with $\sigma \in \Sigma$; the accessibility relations and access maps on events are inherited from the underlying data-exchange frame: $\sigma\overline{\varphi} \sim_a \lambda\overline{\varphi}$ iff $\sigma \sim_a \lambda$, and $(\sigma\overline{\varphi})(a) := \sigma(a)$; while the precondition function pre maps each event $\sigma\overline{\varphi} \in \Sigma\overline{\varphi}$ into the corresponding σ-formula $pre_{\sigma\overline{\varphi}} := \varphi_\sigma$ in the Σ-indexed tuple $\overline{\varphi}$.

Notation We will denote by $\mathcal{E}(\mathcal{S})$ the *set of syntactic event models associated to the language* $LDC^e\langle e\rangle_{\mathcal{S}}$, consisting of all event models of the form $E = \mathbf{\Sigma}\overline{\varphi}$ with $\mathbf{\Sigma} \in \mathcal{S}$ and $\overline{\varphi} = (\varphi_\sigma)_{\sigma\in\Sigma}$ with all formulas φ_σ in $LDC^e\langle e\rangle_{\mathcal{S}}$.

Event-indexed tuples Given the natural bijection $\sigma \mapsto \sigma\overline{\varphi}$ between the action-types in the frame $\mathbf{\Sigma}$ and the corresponding events in the syntactic event model $\mathbf{E} = \mathbf{\Sigma}\overline{\varphi}$, the Σ-indexed tuple $\overline{\varphi} = (\varphi_\sigma)_{\sigma\in\Sigma}$ can also be thought of as being *indexed by events* $e \in E = \Sigma\overline{\varphi}$, i.e., written as $\overline{\varphi} = (\varphi_e)_{e\in E}$.

Sequential composition of event models. Given two (syntactic) event models $\mathbf{E} = (E, \sim_a, \bullet(\bullet), pre)$ and $\mathbf{F} = (F, \sim_a, \bullet(\bullet), pre)$, based respectively on data-exchange frames $\mathbf{\Sigma}, \mathbf{\Lambda} \in \mathcal{S}$ and on tuples of formulas $\overline{\varphi} = (\varphi_\sigma)_{\sigma\in\Sigma}$ and $\overline{\theta} = (\theta_\lambda)_{\lambda\in\Lambda}$ we form their *sequential product* $\mathbf{E};\mathbf{F}$, by taking as the underlying frame (the frame in $\mathcal{S}$ that is isomorphic to) the *product* $\mathbf{\Sigma} \times \mathbf{\Lambda}$ of the two frames, and as underlying tuple the $\Sigma \times \Lambda$-indexed tuple

$$\langle e\rangle\overline{\theta} \;=\; (\langle\sigma\overline{\varphi}\rangle\theta_\lambda)_{(\sigma,\lambda)\in\Sigma\times\Lambda}$$

Using the natural bijection between $\mathbf{E} \times \mathbf{F}$ and $\mathbf{\Sigma} \times \mathbf{\Lambda}$, we can identify the events in $\mathbf{E};\mathbf{F}$ with pairs $(e, f) \in \mathbf{E}\times\mathbf{F}$. We denote by $e; f$ the event expression corresponding to the pair (e, f) (via the bijection), and note that our choice of the above $\Sigma \times \Lambda$-indexed tuple of formulas gives us:

$$pre_{e;f} \;=\; \langle e\rangle pre_f$$

Observation. *The set $\mathcal{E}(\mathcal{S})$ of syntactic event models associated to each language $LDC^e\langle e\rangle_{\mathcal{S}}$ is closed under sequential composition.*

Subformulas of a formula or event expression. The set of *proper subformulas* of a given formula φ or event e is recursively defined by putting: $Sub(p) := \emptyset$; $Sub(\neg\varphi) = Sub(D_A\varphi) := \{\varphi\} \cup Sub(\varphi)$; $Sub(\varphi \wedge \psi) := \{\varphi, \psi\} \cup Sub(\varphi)\cup Sub(\psi)$; $Sub(C^e_A\varphi) = Sub(\langle e\rangle\varphi) := Sub(e)\cup\{\varphi\}\cup Sub(\varphi)$; and finally, for events $e = \sigma\overline{\varphi}$, where $\sigma \in \Sigma \in \mathcal{S}$ is an action type and $\overline{\varphi} = (\varphi_1, \ldots, \varphi_n)$ is an n-tuple of formulas, we put $Sub(e) := \{\varphi_i : 1 \leq i \leq n\} \cup \bigcup_{1\leq i\leq n} Sub(\varphi_i)$.

Note that this definition has the consequence that: for all events $e, f \in E$ and $|E|$-tuples of formulas $\overline{\varphi}$, *pre_f and φ_f are proper subformulas of $C^e_A\overline{\varphi}$.*

Some abbreviations. We have the usual Boolean notions $\varphi \vee \psi$, $\varphi \to \psi$, $\varphi \leftrightarrow \psi$, the Diamond-type duals of our static epistemic modalities $\langle D_A\rangle\varphi$, as well as the Box-type dual $[e]\varphi$ of our dynamic operator $\langle e\rangle\varphi$. In addition, we adopt the following abbreviations:

$$\begin{array}{lcll} K_a\varphi & := & D_{\{a\}}\varphi & (\textit{individual knowledge}) \\ D^e_A\varphi & := & D_{e(A)}(pre_e \to \varphi) & (\textit{event-conditional distributed knowledge}) \\ K^e_a\varphi & := & D^e_{\{a\}}\varphi & (\textit{event-conditional knowledge}) \\ C_A\varphi & := & C^{!\top}_A\varphi & (\textit{common knowledge}) \\ C^\theta_A\varphi & := & C^{!\theta}_A\varphi & (\textit{relativized common knowledge}) \end{array}$$

where ! is the public announcement action-type (living in the public announcement frame $\mathbf{\Sigma}_!$), and $\top$ is any propositional tautology. Also, for two $\mathbf{E}$-indexed tuples of formulas $\overline{\varphi} = (\varphi_e)_{e\in\mathbf{E}}$ and $\overline{\psi} = (\psi_e)_{e\in\mathbf{E}}$, we put

$$\overline{\varphi \to \psi} \;:=\; (\varphi_e \to \psi_e)_{e\in\mathbf{E}}$$

for the tuple of implications between the corresponding formulas.

Action of a data-exchange model over indexed tuples Finally, here is another useful piece of meta-syntactic notation: given event models $\mathbf{E}$ and $\mathbf{F}$, and an $\mathbf{F}$-indexed tuple $\overline{\varphi} = (\varphi_f)_{f\in\mathbf{F}}$, we put

$$\langle\mathbf{E}\rangle\overline{\varphi} \;:=\; (\langle e\rangle\varphi_f)_{e\in\mathbf{E},f\in\mathbf{F}}$$

Note that we can think of $\langle\mathbf{E}\rangle\overline{\varphi}$ as *an* $\mathbf{E};\mathbf{F}$*-indexed tuple* $(\langle e\rangle\varphi_f)_{(e;f)\in\mathbf{E};\mathbf{F}}$.

2.3 Semantics

We interpret formulas on epistemic state models $\mathbf{M}$, and we interpret event expressions in syntactic $\mathcal{S}$-event models $\mathbf{E}$. As usual in DEL, we simultaneously define (i) a *satisfaction relation* $s \models_{\mathbf{M}} \varphi$ between states in a model and formulas, and (ii) an *update product operation* taking any state model $\mathbf{M}$ and any $\mathcal{S}$-based event model $\mathbf{E}$ into an updated state model $\mathbf{M}\bigotimes\mathbf{E}$ (their update product). The definition is by mutual recursion. For (i), the truth clauses for propositional logic are as usual, while for the modal operators, we put:

$$s \models_{\mathbf{M}} D_A\varphi \;\text{ iff }\; w \models \varphi \text{ for all states } w \sim_A s,$$

where $\sim_A := \bigcap_{a\in A} \sim_a$ is the intersection of all the relations $\sim_a$ with $a \in A$;

$$s \models_{\mathbf{M}} C^e_A\overline{\varphi} \;\text{ iff }\; s \models_{\mathbf{M}} \varphi_e,\, K^f_a\varphi_f,\, K^f_aK^h_b\varphi_h, \ldots,$$

for all finite chains $e \sim_a f \sim_b h \ldots$ with $a, b \ldots \in A$;[13] and

$$s \models_{\mathbf{M}} \langle e\rangle\varphi \quad\text{iff}\quad (s,e) \models_{\mathbf{M}\bigotimes\mathbf{E}} \varphi.^{14}$$

[13] Note the analogy with the semantic definition of common knowledge, in terms of a nested iteration of knowledge operators. But also note the differences: event-conditional common knowledge is in fact a *nested iteration of distributed knowledge operators and conditionals*.

[14] The statement on the right-hand side presupposes that $(s,e) \in M\bigotimes E$, hence this definition gives us $s \not\models \langle e\rangle\varphi$ whenever $(s,e) \notin M\bigotimes E$: this is the *existential* version of the dynamic modality.

For (ii): the update product $\mathbf{M} \bigotimes \mathbf{E} = (M \bigotimes E, \sim_a, \| \bullet \|)$ of a state model $\mathbf{M} = (M, \sim_a, \| \bullet \|^M)$ and an event model $\mathbf{E} = (E, \sim_a, \bullet(\bullet), pre)$ is given by:

$$M \bigotimes E = \{(s, e) \in M \times E : s \models_{\mathbf{M}} pre_e\}$$

$$(s, e) \sim_a (s', e') \quad \text{iff} \quad s \sim_{e(a)} s' \text{ and } e \sim_a e'$$

$$\|p\|^{\mathbf{M} \bigotimes \mathbf{E}} = \{(s, e) \in M \bigotimes E : s \in \|p\|^{\mathbf{M}}\}$$

Intuitively, this definition can be justified as follows. The pair (s, e) denotes the *output-state* produced by performing data-exchange action e on input-state s: *this output-state exists only if action e can be executed at s*, i.e. if s satisfies the precondition of e. In that case, the output state (s, e) is *unique*: our data-exchange events are *deterministic*, changing the state in a fully-determined way. The *new valuation is inherited from the old one*: these are *pure informational events*, which do not change the ontic facts p.

The crucial difference with standard DEL lies in the definition of the new epistemic relations $\sim_a$ on output-states: note the key role played in this definition by the relation $\sim_{e(a)}$ on input-states. This says that agent a's new knowledge after a data-exchange event e is the result of putting together the knowledge gained by accessing the prior information of all agents in $e(a)$ and her knowledge about the event e itself.

As noted in [6], we can think of event models $\mathbf{E}$ as "acting" on state models $\mathbf{M}$ to produce the product models $\mathbf{M} \bigotimes \mathbf{E}$, and we can then see that the 'sequential composition' operation $\mathbf{E}; \mathbf{F}$ defined above behaves indeed like a sequential composition of actions: the map $((s, e), f) \mapsto (s, (e; f))$ is a natural isomorphism between $(\mathbf{M} \bigotimes \mathbf{E}) \bigotimes \mathbf{F}$ and $\mathbf{M} \bigotimes (\mathbf{E}; \mathbf{F})$.

3 Axiomatization, completeness and FMP

Let $\mathbf{LDC^e}\langle \mathbf{e} \rangle_{\mathcal{S}}$ be the proof system in Table 1, and let $\mathbf{LDC^e}_{\mathcal{S}}$ be the system for the static logic $LDC^e_{\mathcal{S}}$ obtained by removing the recursion axioms (group IV) from Table 1 (and restricting all instances of remaining axioms to $LDC^e_{\mathcal{S}}$).

Theorem 3.1 *Given the above notations, we have the following:*

(i) *the system $\mathbf{LDC^e}_{\mathcal{S}}$ is sound and complete for the static fragment $LDC^e_{\mathcal{S}}$;*

(ii) *the logic $LDC^e_{\mathcal{S}}$ is decidable and has the FMP;*

(iii) *the full dynamic logic $LDC^e\langle e \rangle_{\mathcal{S}}$ has the same expressivity as its static fragment $LDC^e_{\mathcal{S}}$ (and hence it is also decidable and has the FMP);*

(iv) *the system $\mathbf{LDC^e}\langle \mathbf{e} \rangle_{\mathcal{S}}$ is sound and complete for the logic $LDC^e\langle e \rangle_{\mathcal{S}}$.*

The meaning of the axioms We now give some explanations of the axioms and rules of $\mathbf{LDC^e}\langle \mathbf{e} \rangle_{\mathcal{S}}$. The ones in Group (II) are just the standard laws of distributed knowledge (having as a special case the standard $S5$ axioms for individual knowledge). As for the axioms and rules for C^e_A (Group III), note the analogy with the standard axiomatization for common knowledge.

(I) Axioms and rules of classical propositional logic	
(II) Axioms and rules for (distributed) knowledge:	
(D-Necessitation)	From φ, infer $D_A\varphi$
(D-Distribution)	$D_A(\varphi \to \psi) \to (D_A\varphi \to D_A\psi)$
(Veracity)	$D_A\varphi \to \varphi$
(Pos. Introspection)	$D_A\varphi \to D_AD_A\varphi$
(Neg. Introspection)	$\neg D_A\varphi \to D_A\neg D_A\varphi$
(Monotonicity)	$D_A\varphi \to D_B\varphi$, for sets $A \subseteq B$
(III) Axioms and rules for event-conditional common knowledge	
Given	$\mathbf{E} = (E, \sim_a, \bullet(\bullet), pre) \in \mathcal{E}(\mathcal{S})$, $e \in E$ and $\overline{\varphi} = (\varphi_{e'})_{e' \in E}$:
(C^e-Necessitation)	From $\{\varphi_e\}_{e\in E}$, infer $C^e_A\overline{\varphi}$
(C^e-Distribution)	$C^e_A\overline{\varphi \to \psi} \to (C^e_A\overline{\varphi} \to C^e_A\overline{\psi})$
(C^e-Mix)	$C^e_A\overline{\varphi} \to \Big(\varphi_e \wedge K^{e'}_a C^{e'}_A\overline{\varphi}\Big)$, for $e' \in E$, $a \in A$ s.t. $e \sim_a e'$
(C^e-Induction Rule)	From $\{\varphi_f \to K^{f'}_a \varphi_{f'} : f, f' \in E, a \in A \text{ s.t. } f \sim_a f'\}$, infer $\varphi_e \to C^e_A\overline{\varphi}$
(IV) Recursion/reduction axioms and rules	
Given	$\mathbf{E} = (E, \sim_a, \bullet(\bullet), pre), \mathbf{F} = (F, \ldots) \in \mathcal{E}(\mathcal{S})$, $e \in E$, $f \in F$, $\overline{\varphi} = (\varphi_f)_{f\in F}$:
($[e]$-Necessitation)	From φ, infer $[e]\varphi$
($[e]$-Distribution)	$[e](\varphi \to \psi) \to ([e]\varphi \to [e]\psi)$
(Atomic Permanence)	$\langle e\rangle p \leftrightarrow (pre_e \wedge p)$
(Partial Functionality)	$\langle e\rangle\neg\varphi \leftrightarrow (pre_e \wedge \neg\langle e\rangle\varphi)$
(D-Action)	$\langle e\rangle D_A\varphi \leftrightarrow \Big(pre_e \wedge \bigwedge_{e'\sim_A e} D^{e'}_A\langle e'\rangle\varphi\Big)$
(C^f-Action)	$\langle e\rangle C^f_A\overline{\varphi} \leftrightarrow C^{e;f}_A\langle\mathbf{E}\rangle\overline{\varphi}$

Table 1

The proof system $\mathbf{LDC^e}\langle\mathbf{e}\rangle_{\mathcal{S}}$, where we used the abbreviations from Section 2.2: in particular, $\langle\mathbf{E}\rangle\overline{\varphi} := (\langle e\rangle\varphi_f)_{e\in E, f\in F}$, for event models $\mathbf{E}, \mathbf{F}$ and F-indexed tuples $\overline{\varphi} = (\varphi_f)_{f\in F}$. The system $\mathbf{LDC^e}_{\mathcal{S}}$ is obtained by removing the last group (IV).

The increased complexity is due to the fact that C^e_A is a polyadic modality, so its Induction Rule requires multiple implications as premises to prove its conclusion. Another dis-analogy with standard common knowledge is 'buried' in the notation $K^f_a\psi$ for $D_{f(a)}(pre_f \to \psi)$: the role of individual knowledge K_a in the usual definition of C_A is now played by conditional distributed knowledge!

The first four axioms and rules in group (IV) are standard. The recursion axiom (D-Action) can be restated more explicitly as $\langle e\rangle D_A\varphi \leftrightarrow \big(pre_e \wedge \bigwedge_{e'\sim_A e} D_{e(A)}[e']\varphi\big)$, where $[e']\psi$ is the universal dual of $\langle e'\rangle\psi$ (and where we used the fact that $e' \sim_A e$ implies $e'(A) = e(A)$). This says that group A's distributed knowledge after an exchange e is pre-encoded by the prior distributed knowledge among the *larger* group $e(A)$ (of *agents whose data are being accessed by anybody in* A). And, as a special case of this, *individual knowledge after an exchange is pre-encoded by distributed knowledge*: $\langle e\rangle K_a\varphi \leftrightarrow \Big(pre_e \wedge \bigwedge_{f\sim_a e} D_{e(a)}[f]\varphi\Big)$

Finally, the simplicity of the recursion axiom for C_A^f illustrates the usefulness of our new polyadic modalities: *their recursion law simply amounts to event-composition*! Indeed, the law says that f-conditional common knowledge after an event e is pre-encoded by e; f-conditional common knowledge. [15]

3.1 Sketch of the proof of Theorem 3.1

While relegating many details to the Appendix, we briefly explain here the plan and main steps of the proof (and spelling out the most innovative parts).

Step 1: Pseudo-model completeness and FMP via filtration.

We need a detour through a more general type of models, called *pseudo-models*, in which distributed knowledge relations $\sim_A$ are treated as basic, undefined relations (rather than as intersections of individual knowledge relations).

Pseudo-model semantics. A *pseudo-model* is a structure $\mathbf{S} = (S, \sim_A, \|\bullet\|)_A$, where: S is a set of states; $\sim_A \subseteq S \times S$ are equivalence relations (one for each non-empty set $A \subseteq \mathcal{A}$) with the property that: $A \subseteq B$ and $s \sim_A w$ imply $s \sim_B w$; and $\| \bullet \| : Prop \to \mathcal{P}(S)$ is a valuation function, mapping atomic propositions p into sets of states $\|p\| \subseteq S$. Given a pseudo-model $\mathbf{S}$, we can define the *satisfaction* relation $s \models \varphi$, by using the standard clauses for atomic formulas and propositional connectives, defining D_A as the Kripke modality for $\sim_A$, and defining $C_A^e\varphi$ as in standard models.

Models are *standard* pseudo-models. A pre-model $\mathbf{S}$ is *standard* if $\sim_{A\cup B}$ is the intersection of $\sim_A$ and $\sim_B$, for all sets of agents A and B. It should be obvious that *every model comes with a standard premodel structure*, by defining $\sim_A$ (for each non-empty $A \subseteq \mathcal{A}$) to be the *intersection* $\bigcap_{a\in A} \sim_a$ of all relations $\sim_a$ with $a \in A$. Vice-versa, *every standard pre-model can be seen as a model*, by defining $\sim_a$ to be the same as $\sim_{\{a\}}$, for every $a \in \mathcal{A}$. Moreover, *pseudo-model semantics is indeed a generalization of our intended semantics*: when restricted to standard pseudo-models, the pseudo-model semantics amounts to the same as model semantics. So *models are essentially the same as standard pre-models*: from now on we will *identify* the two.

Proposition 3.2 *The system* $\mathbf{LDC^e}_{\mathcal{S}}$ *is sound and complete wrt pseudo-model semantics, and has the Finite Pseudo-Model Property.*

The *proof* uses a variant of the standard method of *filtration*, adapted to our polyadic modalities. We relegate the details to Appendix A.

Step 2: From finite pseudo-models to finite models.

First, note that pseudo-models *are* relational $S5$-models, in the usual sense, except that their basic relations are labelled by sets of agents instead of single agents. Hence, we can apply to them all the standard Modal Logic concepts and theory [17]. In particular, we get for free the notions of *bisimulation* and *p-morphism* (=functional bisimulation) between pseudo-models.

Moreover, we can easily check invariance under p-morphisms:

[15] Contrast this with the complexity of the recursion axiom for iteration π^* in E-PDL.

Lemma 3.3 *All formulas of our static logic LDC^e are preserved under pseudo-model p-morphisms.*

To prove this, we need a non-standard notion of complexity:

Lemma 3.4 *There exists $\gamma : LDC^e \to On$, associating to every formula φ in LDC^e some (countable) ordinal $\gamma(\varphi) \in On$, s.t. the associated well-founded strict partial order $<$ on formulas in LDC^e (called γ-complexity order, and defined by putting $\varphi < \psi$ iff $\gamma(\varphi) < \gamma(\psi)$) has the following properties:*

- *if φ is a proper subformula of ψ, then $\varphi < \psi$ (and so in particular, $pre_f < C^e_A\overline{\varphi}$ for all e and f living in the same event model E);*
- *$C^e_A\overline{\varphi}$ has higher complexity than all the formulas involved in its semantics:*

$$\varphi_e, K^f_a\varphi_f, K^f_a K^h_b \varphi_h, \ldots < C^e_A\overline{\varphi}$$

for all events $e, f, h, \ldots$ in a given event model, and all agents $a, b, \ldots \in A$.

Proof. We define γ by setting: $\gamma(p) = 0$; $\gamma(\neg\varphi) = \gamma(D_A\varphi) = \gamma(\varphi) + 1$; $\gamma(\varphi \wedge \psi) = max\{\gamma(\varphi), \gamma(\psi)\} + 1$; and $\gamma(C^e_A\overline{\varphi}) = max\{\gamma(pre_f) : f \in E\} \cup \{\gamma(\varphi_f) : f \in E\} + \omega$ (where E is the event model in which e lives).[16] □

Proof of Lemma 3.3. Let $\pi : \mathbf{S} \to \mathbf{S'}$ be a p-morphism between pseudo-models. We show that for all formulas φ in LDC^e, we have:

$$s \models_{\mathbf{S}} \varphi \text{ iff } \pi(s) \models_{\mathbf{S'}} \varphi,$$

for all $s \in S$. The proof is by *induction on the order* $<$. The atomic case and the inductive steps for Boolean connectives and modalities $D_A\varphi$, go as usual, using the Atomic Preservation and the back and forth clauses in the definition of p-morphism, as well as the induction hypothesis (where we use the fact that every formula has higher γ-complexity than its subformulas). Similarly, the inductive step for $C^e_A\overline{\varphi}$ uses the fact that $C^e_A\overline{\varphi}$ has higher γ-complexity than all the formulas involved in its semantics $(\varphi_e, K^f_a\varphi_f, K^f_a K^h_b\varphi_h, \ldots)$; so the induction hypothesis, those formulas are preserved by π, and hence $C^e_A\varphi$ is also preserved by π.

Given Proposition 3.2 and Lemma 3.3, it is clear that in order to prove completeness and FMP for our intended models, it is enough to show:

Lemma 3.5 ***(Intersection Lemma.)*** *Every finite pseudo-model $\mathbf{S} = (S, \sim_A, \| \bullet \|)$ is a p-morphic image of some finite model $\overline{\mathbf{S}} = (\overline{S}, \sim_a, \| \bullet \|)$.*

Proof. For every non-empty set of agents $A \subseteq \mathcal{A}$, let $[s]_A := \{s' \in S : s \sim_A s'\}$ be the equivalence class of any state $s \in S$ modulo $\sim_A$. We denote by $S/\sim_A := \{[s]_A : s \in S\}$ the quotient of S modulo $\sim_A$ (consisting of all equivalence classes). Clearly $S/\sim_A$ is *finite* (since S is finite), and so let N_A be its *cardinality* (=the *number of equivalence classes* modulo $\sim_A$).

[16] The map γ is well-defined, since the relevant event models E are finite, so at each step we are only taking the maximum of finitely many ordinals.

For each non-empty $A \subseteq \mathcal{A}$, consider some *fixed enumeration* (without repetitions) of the equivalence classes in $S/\sim_A$. For every state $s \in S$, denote by $|s|_A$ the *index* of its equivalence class $[s]_A$ in this enumeration. We obviously have that $|s|_A \in \{0, 1, \ldots, N_A - 1\}$, for all $s \in S$. Note that $|s|_A = |s'|_A$ implies $[s]_A = [s']_A$, i.e. $s \sim_A s'$ (since our enumeration has no repetitions).

We now construct our new model $\overline{\mathbf{S}}$. The set of states will be given by

$$\overline{S} \;:=\; S \times \prod_{A \subseteq \mathcal{A}, A \neq \emptyset \bigcup} \{0, 1, \ldots, N_A - 1\}^{\mathcal{A}}.$$

In other words, the new states are of the form $(s, \bar{i}) := (s, i_A)_{A \subseteq \mathcal{A}, A \neq \emptyset}$, consisting of an old state $s \in S$ and a tuple $\bar{i} = (i_A)_{A \subseteq \mathcal{A}}$ of assignments $i_A : \mathcal{A} \to \{0, 1, \ldots, N_A - 1\}$ (indexed by non-empty sets $A \subseteq \mathcal{A}$), that map agents $a \in \mathcal{A}$ to indeces $i_A(a) \in \{0, 1, \ldots, N_A - 1\}$ in the enumeration of $S/\sim_A$.

Restriction, Sum and Accessibility. We use the notation $i_A|B$ to denote the *restriction of the map i_A to the agents in B*. We also put

$$\bar{i}(A) \;:=\; \sum_{a \in A} i_A(a)$$

for the *sum of all the values of i_A over agents in A*. Finally, we define a *basic accessibility relation* $\sim$ on $\overline{S}$, by

$$(s, \bar{i}) \sim (s', \bar{i}') \;\text{ iff }\; |s|_A - \bar{i}(A) \equiv |s'|_A - \bar{i}'(A) \,(mod\, N_A) \;\text{ for all } A \subseteq \mathcal{A} \text{ s.t. } A \neq \emptyset,$$

where $\equiv$ denotes *congruence modulo N_A*.

Obviously, accessibility $\sim$ is an equivalence relation on $\overline{S}$. In our construction, $\sim$ will constitute a precondition for epistemic equivalence $\sim_a$.

The model. We now structure the set $\overline{S}$ into a *model* $\overline{\mathbf{S}}$, as follows: the *valuation* is inherited from the pseudo-model $\mathbf{S}$, i.e. $\|p\|_{\overline{\mathbf{S}}} := \{(s, \bar{i}) \in S : s \in \|p\|_{\mathbf{S}}\}$; and we define the basic *indistinguishability relations* $\sim_a$ on $\overline{\mathbf{S}}$ by putting

$$(s, \bar{i}) \sim_a (s', \bar{i}') \;\text{ iff }\; (s, \bar{i}) \sim (s', \bar{i}') \text{ and } i_A(a) = i'_A(a) \text{ for all } A \neq \emptyset.$$

It should be clear that all $\sim_a$ are equivalence relations, so the structure $\overline{\mathbf{S}} = (\overline{S}, \sim_a, \| \bullet \|)$ is a *model*. Hence, if for every non-empty set $B \subseteq \mathcal{A}$ we take $\sim_B$ to be the *intersection* $\bigcap_{b \in B} \sim_b$, then $\overline{\mathbf{S}}$ is a standard pseudo-model.

It is obvious that, for every non-empty such set B of agents, we have:

$$(s, \bar{i}) \sim_B (s', \bar{i}') \;\text{ iff }\; (s, \bar{i}) \sim (s', \bar{i}') \text{ and } i_A|B = i'_A|B \text{ for all } A \neq \emptyset.$$

All is left to show is that $\mathbf{S}$ *is a p-morphic image of* $\overline{\mathbf{S}}$. For this, we take the *first-component projection* $\pi : \overline{S} \to S$, given by $\pi(s, \bar{i}) := s$, and we will prove that π *is a surjective p-morphism of pseudo-models.*

Surjectivity and Atomic Preservation are obvious.

To check the 'forth' clause, we assume that $(s, \bar{i}) \sim_B (s', \bar{i}')$, and we need to show that $s \sim_B^{\mathbf{S}} s'$. For this, we apply the above characterization of the relation

$\sim_B$ to the set $A := B$: our assumption implies both that $i_B|B = i'_B|B$, and hence that $\bar{i}(B) = \bar{i}'(B)$, and that $|s|_B - \bar{i}(B) \equiv |s'|_B - \bar{i}'(B)\,(mod\, N_B)$. Adding the last two equalities, we obtain that $|s|_B \equiv |s'|_B\,(mod\, N_B)$, i.e. $|s|_B = |s'|_B$, and thus $s \sim^{\mathbf{S}}_B s'$, as desired.

Finally, to check the 'back' clause, let $(s, \bar{i}) \in \overline{S}$ be such that $s \sim_B s'$ for some $s' \in S$. We need to find some $\bar{i}' = (i'_A)_{A\subseteq\mathcal{A}}$ satisfying $(s, \bar{i}) \sim_B (s', \bar{i}')$. For this, we simply put $i'_A := i_A$ for every $A \subseteq B$; while, for $A \not\subseteq B$, we choose some $a \in A - B$, then put $i'_A(b) := i_A(b)$ for all $b \in \mathcal{A} - \{a\}$, and $i'_A(a) := i_A(a) + |s'|_A - |s|_A\,(mod\, N_A)$. Using again the above characterization of the relations $\sim^{\overline{\mathbf{S}}}_B$, we can verify that we have $(s, i) \sim_B (s', i')$, as desired. □

This finishes the proof of parts 1 and 2 of Theorem 3.1 (completeness and decidability for the static fragments $LDC^e_{\mathcal{S}}$ of our signature-based logics).

Step 3: Reduction and completeness for the dynamic logics $LDC^e\langle e\rangle_{\mathcal{S}}$.

Checking the *soundness* of the axioms and rules of $\mathbf{LDC^e\langle e\rangle}_{\mathcal{S}}$ is a tedious, but easy verification, so we skip the details. For *completeness*, we use the reduction axioms to recursively show that each formula in $LDC^e\langle e\rangle_{\mathcal{S}}$ is provably equivalent to a 'static' formula in $LDC^e_{\mathcal{S}}$. This establishes parts 3 and 4 of Theorem 3.1. Once again, we relegate the details to Appendix B.

4 Conclusions and future work

The purpose of this paper was to axiomatize the logic of all data-exchange events (with preconditions), in the presence of both common and distributed knowledge. In order to have elegant and transparent recursion/reduction axioms, we extended the static base with new polyadic modalities, denoting complex levels of conditional group knowledge. This in itself is an original technical contribution to the field of Dynamic Epistemic Logic.

Furthermore, we proved decidability by directly establishing FMP, a strategy that was not previously available for logics with distributed knowledge (or intersection of programs). For this, we employed a highly innovative method, that uses notions of elementary number theory (congruences modulo a natural number) to control the proliferation of world-copies: this is the second original technical contribution of our paper. [17]

We should note that our congruence-based method can be extended to *prove FMP for GE-PDL, and even for 'Group Epistemic Mu-Calculus' GE_μ* (i.e. Mu-Calculus based on modalities $[A]\varphi = D_A\varphi$ for distributed knowledge). This is in sharp contrast to PDL with Intersection (which does *not* have FMP). We are planning to write up these proofs in a follow-up paper.

Other on-going work pertains investigating *non-S5 versions* of our data-exchange events. This would allow us to capture notions of *belief* and treat

[17] In addition, we adapted a number of notions (pseudo-models, filtration, Fischer-Ladner closure, bisimulations and p-morphisms etc), to yield variants that are appropriate for our polyadic generalizations of the common knowledge operator.

hidden, unsuspected, or deceiving actions: e.g., *fully secret hacking* (in which the hacked agent does not even suspect the hacking), *fully private announcements* a la Gerbrandy [21] (where the outsiders are unaware that the announcement may be happening), etc. For this, one may need to extend the static base with conditional beliefs and other ingredients borrowed from Belief Revision Theory, as it has already been done for classical DEL [14,10].

Yet another line of on-going research involves adding *explicit non-propositional data*, in the form of *variables* x (taking e.g. different numerical values $x(s)$ in different possible states s), and express the agents' possession of such data via epistemic operators for *(distributed) knowledge of the value of a variable* $K_A x$, as it was done in [26,19,3]. This allows us to explicitly formalize the data-acquisition effects of our data-exchange events: e.g., after accessing the database of some other agent b who already knew the value of a variable x, agent a comes to know the value of x as well. Proving completeness and decidability for these logics poses interesting technical challenges, but we are in the process of finishing a manuscript dealing with these challenges.

Appendix

A Completeness wrt finite pseudo-models

In this section, we prove Proposition 3.2. Soundness of $\mathbf{LDC^e}$ on pseudo-models is an easy verification. But completeness require a bit more work.

Fisher-Ladner Closure. Given a formula φ_0 in $LDC^e_{\mathcal{S}}$, its Fisher-Ladner closure is the smallest set Φ that contains φ_0, and is closed under subformulas and single negations [18] $\sim\varphi$, as well as under the following rule:

if $C^e_A\phi \in \Phi$ for some event model E, event $e \in E$ and set of agents A, then $K^f_a C^f_A \phi \in \Phi$ for all $a \in A$ and all $f \in E$ s.t. $e \sim_a f$,

One can easily check that *the Fisher-Ladner closure of any formula is finite.*

Finite Canonical Pseudo-Model. For a given formula φ_0, let $\Phi = \Phi(\varphi_0)$ be its Fisher-Ladner closure. The *finite canonical pseudo-model for* φ_0 is the structure $\mathbf{S}^c = (S^c, \sim_B, \|\bullet\|)$, where: S^c is the set of all maximally consistent theories $T \subseteq \Phi$ (over the finite sublanguage given by the Fisher-Ladner closure of φ_0); the group epistemic relations $\sim_A$ are given by putting, for all $T, W \in S^c$:

$$T \sim_A W \quad \text{iff} \quad \forall(D_A\varphi) \in \Phi \; (D_A\varphi \in T \leftrightarrow D_A\varphi \in W)\,;$$

and the valuation is given by putting, for all $p \in Prop$: $\|p\| = \{T \in S^c : p \in T\}$.

It is easy to check that S^c *is indeed a pseudo-model*: $\sim_A$ are obviously equivalence relations, and the other conditions are ensured by the axioms. It is also clear that S^c *is finite*, of size $|S^c| \leq 2^{|\Phi|}$.

For our next results, we need another abbreviation:

Hat Notation For every theory $T \in S^c$, we put $\widehat{T} := \bigwedge T$.

[18] The single negation $\sim\varphi$ is defined as: $\sim\varphi := \psi$ if φ is of the form $\neg\psi$; and $\sim\varphi := \neg\varphi$ if φ is *not* of the form $\neg\psi$ (for any ψ).

Lemma A.1 *("*Diamond Lemma*") For all theories $T, W \in S^c$, we have:*

(i) *If $\widehat{T} \wedge \langle D_A \rangle \widehat{W}$ is consistent, then $T \sim_A W$;*

(ii) *If $D_A\varphi \in \Phi - T$, then there exists $T' \in S^c$ s.t. $(\sim \varphi) \in T'$ and $T \sim_A T'$.*

The proof of this lemma is standard, so we skip the details.

Lemma A.2 *("*Truth Lemma*") Given a finite canonical pseudo-model S^c over some Fisher-Ladner closure Φ, we have for all $\varphi \in \Phi$:*

$$T \models_{\mathbf{S}^c} \varphi \;\; \textit{iff} \;\; \varphi \in T, \; \textit{for every } T \in S^c$$

Proof. Induction on subformulas. The atomic case is taken care by the canonical valuation, whole the Boolean cases are trivial.

Inductive step for $D_A\varphi$. *Left-to-right*: assume that $T \models D_A\varphi$ for some $(D_A\varphi) \in \Phi$, and suppose towards a contradiction that $(D_A\varphi) \notin T$, hence $(D_A\varphi) \in \Phi - T$. By the Diamond Lemma, there exists some $T' \in S^c$ with $(\sim \varphi) \in T'$ and $T \sim_A T'$. But $T \sim_A T'$ and $T \models D_A\varphi$ imply that $T' \models \varphi$, which by the induction hypothesis gives us $\varphi \in T'$, thus contradicting the fact that $(\sim \varphi) \in T'$ (given the consistency of T').

Right-to-left: Assume that $(D_A\varphi) \in T$. To prove that $T \models D_A\varphi$, let $W \in S^c$ be s.t. $T \sim_A W$; it is enough to show that $\varphi \in W$. For this, note that, by the definition of $T \sim_A W$ in our canonical pseudo-model, $(D_A\varphi) \in T$ implies $(D_A\varphi) \in W$, which in its turn implies that $\varphi \in W$ (by the Veracity axiom).

Inductive step for $C^e_A\overline{\varphi}$, with $e \in E$, where $\mathbf{E}$ is an event model (involving only preconditions in Φ of lower complexity than $C^e_A\overline{\varphi}$). To prove this case, we need some useful notation and two preliminary claims. For all $e \in E$, put

$$\eta_e := \bigvee\{\widehat{W} : W \in S^e_{\overline{\varphi}}\}, \quad \text{where } S^e_{\overline{\varphi}} := \{W \in S^c : W \models C^e_A\overline{\varphi}\},$$

and denote by $\overline{\eta} := (\eta_e)_{e \in E}$ the resulting E-tuple of formulas.

Claim 1: For all $e, f \in E$, $a \in A$ with $e \sim_a f$, we have $\vdash \eta_e \to K^f_a\eta_f$.

Proof of Claim 1: Suppose not. Using the definition of K^f_a as an abbreviation, we must have that $\eta_e \wedge \langle D_{f(a)}\rangle(pre_f \wedge \neg\eta_f)$ is consistent (where $\langle D_{f(a)}\rangle\theta := \neg D_{f(a)}\neg\theta$ is the existential dual of $D_{f(a)}$). Given the definition of $\overline{\eta}$, and the easily proven theorem $\vdash \bigvee\{\widehat{V} : V \in S^c\}$, this means there exist $W \in S_c$, $V \in S^c$ such that $W \models C^e_A\overline{\varphi}$, $V \not\models C^f_A\overline{\varphi}$ and $\widehat{W} \wedge \langle D_{f(a)}\rangle(pre_f \wedge \widehat{V})$ is consistent. But this implies that $W \sim_{f(a)} V$ (using the first part of the Diamond Lemma) and that $pre_f \in V$ (since $pre_f \in \Phi$ is consistent with $\widehat{V}$, and V is a maximally consistent subset of Φ), and thus $V \models pre_f$ (by the induction hypothesis). But by (C^e)-Mix Axiom, $W \models C^e_A\overline{\varphi}$ implies $W \models K^f_a C^f_A\overline{\varphi}$, i.e. $W \models D_{f(a)}(pre_f \to C^f_A\overline{\varphi})$. From this together with $W \sim_{f(a)} V$ and $V \models pre_f$, we infer $V \models C^f_A\overline{\varphi}$, contradicting the previously established fact $V \not\models C^f_A\overline{\varphi}$.

Claim 2: For all $e \in E$, we have $\vdash \eta_e \to C^e_A\overline{\varphi}$.

Proof of Claim 2: First, by (C^e)-Mix, we have $\vdash C^e_A\overline{\varphi} \to \varphi_e$, hence $W \models \varphi_e$ (for all $e \in E$ and $W \in S^e_{\overline{\varphi}}$). By the induction hypothesis, we get that $\varphi_e \in W$,

and hence that $\vdash \widehat{W} \to \varphi_e$, for all $e \in E$ and $W \in S^e_{\overline{\varphi}}$. By the definition of η_e, we then obtain that $\vdash \eta_e \to \varphi_e$ is a theorem for all $e \in E$. By applying C^e-Necessitation, we derive $\vdash C^e_A \overline{\eta \to \varphi}$, and by C^e-Distribution we obtain $\vdash C^e_A \overline{\eta} \to C^e_A \overline{\varphi}$. On the other hand, we can also prove the theorem $\vdash \eta_e \to C^e_A \overline{\eta}$ for all $e \in E$ (by applying the C^e-Induction Rule to Claim 1 above). Putting these together, we obtain that $\vdash \eta_e \to C^e_A \overline{\varphi}$ for all $e \in E$, as desired.

We now proceed to prove the inductive case for $C^e_A \overline{\varphi}$:

Left-to-right implication: assume that $T \models C^e_A \overline{\varphi}$. Then (by the very definition of η_e), we have that $\vdash \widehat{T} \to \eta_e$. Together with Claim 2, this gives us that $\vdash \widehat{T} \to C^e_A \overline{\varphi}$, which implies that $C^e_A \overline{\varphi} \in T$ (since $C^e_A \overline{\varphi} \in \Phi$ and T is a maximally consistent subset of Φ), as desired.

Right-to-left: Assume that $C^e_A \overline{\varphi} \in T$. To prove that $T \models C^e_A \overline{\varphi}$, let $e = e_0 \sim_{a^1} e_1 \sim_{a^1} \dots e_{n-1} \sim_{a^n} e_n$ be any finite chain with all $a^i \in A$, and we need to show that $T \models K^{e_1}_{a^1} \dots K^{e_n}_{a^n} \varphi_{e_n}$, i.e. that $T \models D_{e_1(a^1)}(pre_{e_1} \to \dots D_{e_n(a^n)}(pre_{e_n} \to \varphi_{e_n}) \dots)$. For this, let $T = T^0 \sim_{e_1(a^1)} T^1 \dots T^{n-1} \sim_{e_n(a^n)} T^n$, with $T^k \models pre_{e_k}$ for all $k = 1, n$, and we need to prove that $T^n \models \varphi_{e_n}$. For that, we first need to establish a more general statement:

Claim 3: $C^{e_k}_A \overline{\varphi} \in T^k$ for all $1 \leq k \leq n$.

Proof of Claim 3: Induction on k. For $k = 1$, the claim is true by the assumption that $(Cd_{\mathcal{B}}\varphi) \in T$. For the inductive step: assume that $C^{e_{k-1}}_A \overline{\varphi} \in T^{k-1}$, and use the theorem $\vdash C^{e_{k-1}}_A \overline{\varphi} \to D_{e_k(a^k)}(pre_{e_k} \to C^{e_k}_A \overline{\varphi})$ (which is just the explicit unfolding of $\vdash C^{e_{k-1}}_A \overline{\varphi} \to K^{e_k}_{a^k} C^{e_k}_A \overline{\varphi}$, itself an instance of C^e-Mix), to obtain $D_{e_k(a^k)}(pre_{e_k} \to C^{e_k}_A \overline{\varphi}) \in T^{k-1}$. From this and $T^{k-1} \sim_{e_k(a^k)} T^k$ (and the canonical definition of $\sim_A$), we get that $D_{e_k(a^k)}(pre_{e_k} \to C^{e_k}_A \overline{\varphi}) \in T^k$. Using Veracity and the fact that $pre_{e_k} \in T^k$ (which follows from $T^k \models pre_{e_k}$ by the induction hypothesis), we conclude that $C^{e_k}_A \overline{\varphi} \in T^k$.

Applying Claim 3 to $k := n$, we obtain $C^{e_n}_A \overline{\varphi} \in T^n$. Using C^e-Mix, we infer $\varphi_{e_n} \in T^n$, which by the induction hypothesis yields $T^n \models \varphi_{e_n}$, as desired. □

Proof of Proposition 3.2 (*Completeness of* $\mathbf{LDC^e}_{\mathcal{S}}$ *wrt finite pseudo-models*) Given a consistent formula φ_0, the Lindenbaum Lemma gives us some $T_0 \in S^c$ s.t. $\varphi_0 \in T_0$. By the Truth Lemma, T_0 satisfies φ_0 in $\mathbf{S}^c$.

B Completeness and Reduction of Dynamic Logics

Lemma B.1 *The following are standard consequences of* $[e]$*-Necessitation and* $[e]$*-Distributivity (holding in general for all normal modalities* $[e]$*):*

(i) (Replacement of Equivalent Formulas) *If* $\vdash \varphi \leftrightarrow \psi$ *is a theorem in* $\mathbf{LDC^e}\langle \mathbf{e} \rangle$*, then* $\vdash \langle e \rangle \varphi \leftrightarrow \langle e \rangle \psi$ *is also a theorem.*

(ii) (Distributivity over Conjunction) *The following is a theorem in* $\mathbf{LDC^e}\langle \mathbf{e} \rangle$*:* $\vdash [e](\varphi \wedge \psi) \leftrightarrow ([e]\varphi \wedge [e]\psi)$

The second item, in combination with the reduction axiom for negation (i.e. the Partial Functionality axiom), gives us:

Lemma B.2 (∧-Reduction Law) *The following is a theorem in* $\mathbf{LDC^e\langle e\rangle}$*:*

$$\vdash \ \langle e\rangle(\varphi \wedge \psi) \ \leftrightarrow \ (\langle e\rangle\varphi \wedge \langle e\rangle\psi)$$

Lemma B.3 (Replacement of Equivalent Event Models) *Let* $\mathbf{\Sigma} \in \mathcal{S}$ *be a data-exchange frame, of size* $n = |\Sigma|$*, and let* $\overline{\theta}, \overline{\theta'}$ *be* n*-tuples of formulas, such that we have* $\vdash \theta_i \leftrightarrow \theta'_i$ *for all* $i = 1, n$*. Consider the event models* $E = \Sigma\overline{\theta}$ *and* $E' = \Sigma\overline{\theta'}$*, and the map* $' : E \to E'$*, given by* $e = \sigma\overline{\theta} \mapsto e' = \sigma\overline{\theta'}$*. Then, for all* $e \in E$*, agents* a*, sets of agents* A*, formulas* φ *and tuples* $\overline{\varphi}$*, we have:*

(i) $' : E \to E'$ *is an isomorphism of data-exchange frames, with the property that* $\vdash pre_e \leftrightarrow pre_{e'}$*;*

(ii) $\vdash K^e_a\varphi \leftrightarrow K^{e'}_a\varphi$*;*

(iii) $\vdash C^e_A\overline{\varphi} \leftrightarrow C^{e'}_A\overline{\varphi}$*.*

Proof. The first item is a trivial verification, while the second follows from the normality of D_A, together with the definition of $K^e_a\varphi := D_{e(a)}(pre_e \to \varphi)$ and the fact that events e and e' have the same access maps and equivalent preconditions. For the left-to-right implication in item 3: by the Mix Axiom, we have $\vdash C^e_A\overline{\varphi} \to K^f_a C^e_A\overline{\varphi}$, for all agents $a \in A$ and events $e, f \in E$ s.t. $e \sim_a f$; putting this together with $\vdash K^f_a C^e_A\overline{\varphi} \to K^{f'}_a C^e_A\overline{\varphi}$ (by item 2), we obtain $\vdash C^e_A\overline{\varphi} \to K^{f'}_a C^e_A\overline{\varphi}$ for all such and a; then apply the C^e_A-Induction Rule to conclude that $\vdash C^e_A\overline{\varphi} \to C^{e'}_A\overline{\varphi}$. The right-to-left implication is similar. □

We can now prove a preliminary "one-step reduction" result:

Lemma B.4 *Let* θ *be any 'static' formula in* LDC^e*, and* $E = \Sigma\overline{\theta}$ *be any event model over the static fragment (i.e. with all* θ_i *in* LDC^e*). Then, for every event* $e \in E$*, there exists some formula* θ_e *in the static fragment* LDC^e*, s.t.*

$$\vdash \langle e\rangle\theta \leftrightarrow \theta_e \qquad \textit{is provable in } \mathbf{LDC^e\langle e\rangle}.$$

Proof. Induction on the subformula complexity of the static formula θ:

For $\theta := p$, Atomic Permanence gives us the appropriate formula $\theta_e := p$.

For $\theta := \neg\psi$: by the induction hypothesis, there exists some 'static' formula ψ_e, s.t. $\vdash \langle e\rangle\psi \leftrightarrow \psi_e$. By the reduction axiom for negation (i.e. Partial Functionality), we get $\vdash \theta \leftrightarrow (pre_e \wedge \neg\psi_e)$. So we can take $\theta_e := pre_e \wedge \neg\psi_e$.

For $\theta := \phi \wedge \psi$: put $\theta_e := \phi_e \wedge \psi_e$ (using induction and Lemma B.2).

For $\theta := D_A\psi$, we apply the induction hypothesis to ψ; hence for every event $f \sim_A e$, there exists some static formula ψ_f s.t. $\vdash [f]\psi \leftrightarrow \psi_f$. Putting this together with the Reduction Axiom for D_A, we get $\vdash \theta \leftrightarrow (pre_e \wedge \bigwedge_{f\sim_B e} D^f_A\psi_f)$. So we can take $\theta_e := pre_e \wedge \bigwedge_{f\sim_B e} D^f_A\psi_f$.

For $\theta := C^f_A\overline{\varphi}$, where $f = \sigma\overline{\psi}$ comes from some event model $F = \Sigma\overline{\psi}$ over LDC^e: note that, since $\theta = C^f_A\overline{\varphi} = C^{\sigma\overline{\psi}}_A\overline{\varphi}$, all the formulas φ_i and ψ_j are subformulas of θ. So, by the induction hypothesis, for each $e_i \in E$ and each $f_j \in F$ with precondition $pre_{f_j} = \psi_j$, there exist static formulas $\varphi'_{ij}, \psi'_{ij} \in LDC^e$, s.t. we have $\vdash \langle e_i\rangle\varphi_j \leftrightarrow \varphi'_{ij}$ and $\vdash \langle e_i\rangle\psi_j \leftrightarrow \psi'_{ij}$. By the

Reduction Axiom for $C_A^f\overline{\varphi}$, we also have $\vdash \langle e_i\rangle C_A^{f_j}\overline{\varphi} \leftrightarrow C_A^{e_i;f_j}\langle E\rangle\overline{\varphi}$. Since each precondition $pre_{e_i;f_j} = \langle e_i\rangle\psi_j$ is equivalent to the corresponding static formula ψ'_{ij}, we can use Lemma B.3 to replace the event model $E;F = \{e_i;f_j\}_{i,j}$ by another event model $G = \{g_{i,j}\}_{i,j}$ having the same data-exchange structure but with the static formulas $\{\psi'_{ij}\}$ as preconditions of the corresponding $g_{i;j}$; so we obtain that $\vdash C_A^{e_i;f_j}\langle E\rangle\overline{\varphi} \leftrightarrow C_A^{g_{i,j}}\langle E\rangle\overline{\varphi}$. Using this, the Normality of C^e and the fact that each $\langle e_i\rangle\varphi_j$ is equivalent to the corresponding φ'_{ij}, we obtain that $\vdash C_A^{e_i;f_j}\langle E\rangle\overline{\varphi} \leftrightarrow C_A^{g_{i,j}}\overline{\varphi'}$, where $\overline{\varphi'} := \{\varphi_{i,j}\}_{i,j}$. Since G is an event model with preconditions ψ'_{ij} in the static fragment LDC^e, and also all $\varphi'_{ij} \in LDC^e$, the formulas $C_A^{g_{i,j}}\overline{\varphi'}$ are all in LDC^e as well. Applying this to the specific i and j for which $e_i := e \in E$ and $f_j := f \in F$ and recalling that $\theta = C_A^f\overline{\varphi}$, we obtain that $\vdash \langle e\rangle\theta \leftrightarrow \theta_e$, for $\theta_e := C_A^{g_{i,j}}\overline{\varphi'}$ belonging to LDC^e, as desired. □

Now we can establish our full reduction result:

Lemma B.5 *For every formula θ of the dynamic logic $LDC^e\langle e\rangle$, there exists some formula θ' of the static language LDC^e, s.t.*

$$\vdash \theta \leftrightarrow \theta' \qquad \textit{is provable in } \mathbf{LDC^e\langle e\rangle}.$$

Proof. Induction on the subformula complexity of the dynamic formula θ:

For $\theta := p$, we can take $\theta' = \theta$ (since this is already in LDC^e).

The Boolean cases $\theta := \neg\psi$ and $\theta := \phi \wedge \psi$ are also trivial.

For $\theta := D_A\psi$, apply the induction hypothesis to ψ; so there exists some 'static' formula ψ', such that $\vdash \psi \leftrightarrow \psi'$. By D_A-Necessitation and D_A-Distribution, we get that $\vdash D_A\psi \leftrightarrow D_A\psi'$ (so we can take $\theta' := D_A\psi'$).

For $\theta := C_A^e\overline{\psi}$, with $e \in E$, where E is an event model of size $|E| = n$, and $\overline{\psi} = \{\psi_i\}_{i\leq n}$ is an n-tuple of formulas: apply the induction hypothesis to obtain an n-tuple $\overline{\psi'} = \{\psi'_i\}_{i\leq n}$ of static formulas $\psi'_i \in LDC^e$, s.t. we have $\vdash \psi_i \leftrightarrow \psi'_i$ for all i. Using the Normality of C^e, we obtain $\vdash C_A^e\overline{\psi} \leftrightarrow C_A^e\overline{\psi'}$. Since the preconditions of the events in E are subformulas of θ, we can use the induction hypothesis to obtain 'static' formulas in LDC^e that are provably equivalent to each precondition of an event in E; then use Lemma B.3 to replace the event model E by another event model $E' = \{e' : e \in E\}$, having the same data-exchange structure, but with static preconditions $pre_{e'} \in LDC^e$ s.t. $\vdash pre_e \leftrightarrow pre_{e'}$ for all e, obtaining $\vdash C_A^e\overline{\psi'} \leftrightarrow C_A^{e'}\overline{\psi'}$ for all $e \in E$. Putting all these together, we conclude that $\vdash C_A^e\overline{\psi} \leftrightarrow \theta'$, where $\theta' := C_A^{e'}\overline{\psi'}$.

For $\theta := \langle e\rangle\psi$: by the induction hypothesis, there exists some 'static' formula ψ', such that $\vdash \psi \leftrightarrow \psi'$. By Lemma B.1, we get that $\vdash \langle e\rangle\psi \leftrightarrow \langle e\rangle\psi'$, and by Lemma B.4 we obtain another static formula $\psi'_e \in LDC^e$, s.t. $\vdash \langle e\rangle\psi' \leftrightarrow \psi'_e$. Putting these together, we get $\vdash \langle e\rangle\psi \leftrightarrow \psi'_e$ (so we can take $\theta' := \psi'_e$). □

Finally, we can prove the last part of Theorem 3.1:

Proof of completeness of $\mathbf{LDC^e\langle e\rangle}$: Let θ be a consistent formula of $LDC^e\langle e\rangle$. By Lemma B.5, there exists some θ' in LDC^e s.t. $\vdash \theta \leftrightarrow \theta'$ is a theorem in $\mathbf{LDC^e\langle e\rangle}$. So θ' is consistent. By Theorem 3.1(1) (completeness for $\mathbf{LDC^e}$), θ' is satisfiable. Since $\vdash \theta \leftrightarrow \theta'$, θ is also satisfiable.

References

[1] Adriaans, P. and J. van Benthem, *Handbook of Philosophy of Information*, (2008), Elsevier.

[2] Agotnes, T. and Y.N. Wang, *Resolving Distributed Knowledge*, in: *Artificial Intelligence*, (2017), 252, pp. 1-21.

[3] Baltag, A., *To Know is to Know the Value of a Variable*, in: *Adv. in Modal Logic 2016*, (2016), pp. 135-155.

[4] Baltag, A. and J. van Benthem, *A Simple Logic of Functional Dependence*, in: *Journal of Philosophical logic*, (2021), vol. 50, pp. 939-1005.

[5] Baltag, A. and J. van Benthem, *Updates, Generalized p-Morphisms, and (Co-)Recursive Equations*, To appear in van Benthem, J. and F. Liu (eds.), *Graph Games and Logic Design - Recent developments and further directions*, Springer, (2024).

[6] Baltag, A. and L. S. Moss, *Logics for Epistemic Programs*, in: Symons, J. and J. Hintikka. (eds.), W. van der Hoek (special section editor), Synthese, (2004), 139(2), pp. 165-224.

[7] Baltag, A.; L. Moss and S. Solecki, *The Logic of Public Announcements, Common Knowledge, and Private Suspicions*, in: *Proceedings TARK 98*, (1998), pp. 43-56.

[8] Baltag, A.; L. Moss and S. Solecki, *Logics for epistemic actions: completeness, decidability, expressivity*, in: *Logics* (2023), 1 (2), pp. 97-147.

[9] Baltag, A. and B. Renne, *Dynamic Epistemic Logic*, in: *Stanford Encyclopedia of Philosophy*, (2016).

[10] Baltag, A. and S Smets, *A qualitative theory of dynamic interactive belief revision*, in *Texts in logic and games* (2008), vol. 3, pp. 9–58), Amsterdam University Press.

[11] Baltag, A. and S. Smets, *Protocols for Belief Merge: Reaching Agreement via Communication*, in: *Logic Journal of the IGPL*, (2013), 21(3), pp. 468-487.

[12] Baltag, A. and S. Smets, *Learning what Others Know*, in: Kovacs, L. and E. Albert (eds.), *LPAR23 proceedings of the International Conference on Logic for Programming AI and Reasoning, EPiC Series in Computing*, (2020), Volume 73, pp. 90-110.

[13] van Benthem, J., *One is a lonely number*, in: Koepke, P.; Z. Chatzidakis and W. Pohlers, (eds.) *Logic Colloquium 2002*, (2002), pp. 96-129, ASL and A.K. Peters, Wellesley MA.

[14] van Benthem, J., *Dynamic logic of belief revision*, in: *Journal for Applied Non-Classical Logics* (2007), 17(2), pp. 129–155.

[15] van Benthem, J., *Logical Dynamics of Information and Interaction*, (2011), Cambridge University Press, Cambridge UK.

[16] van Benthem, J.; J. van Eijck and B. Kooi, *Logics of Communication and Change*, in: *Information and Communication*, (2006), 204:11, pp. 1620-1662.

[17] Blackburn, P.; M. de Rijke and Y. Venema, *Modal Logic*, (2000), Cambridge University Press, Cambridge.
[18] van Ditmarsch, H.; W. van der Hoek and B. Kooi, *Dynamic Epistemic Logic*, (2007), Springer, Dordrecht.
[19] van Eijck, J; M. Gattinger and Y. Wang, *Knowing Values and Public Inspection*, in: *Proceedings 7th Indian Conference on Logic and its Applications* (2007), Kanpur, 77–90.
[20] Fagin, R.; J. Halpern, Y. Moses and M. Vardi, *Reasoning About Knowledge*, (1995), The MIT Press, Cambridge MA.
[21] Gerbrandy, J., *Bisimulations on Planet Kripke*, (1998), Doctoral Dissertation, ILLC, UvA. ILLCS DS Series DS-1999-01. ISBN: 9057760193, 9789057760198
[22] Parikh, R., *Levels of Knowledge, Games and Group Action*, in: *Research in Economics*, (2003), 57, pp. 267-281.
[23] Parikh, R. and R. Ramanujam, *A Knowledge-Based Semantics of Messages*, in: *Journal of Logic, Language and Information*, (2003), 12, pp. 453-467.
[24] Plaza, J., *Logics of Public Communication*, in: *Proceedings 4th International Symposium on Methodologies for Intelligent Systems*, (1989), pp. 201-216.
[25] van Wijk, S., *Coalitions in Epistemic Planning.MSc Thesis*, (2015), ILLC, UvA. *https://www.illc.uva.nl/Research/Publications/Reports/MoL/*
[26] Wang, Y., *Beyond Knowing That: A New Generation of Epistemic Logics*, in *Hintikka Volume* (2018), Springer, Dordrecht, 499-533.