A Logic of Revelation and Concealment

Wiebe van der Hoek   Petar Iliev   Michael Wooldridge
Department of Computer Science
University of Liverpool, UK
{wiebe, pvi, mjw}@liverpool.ac.uk

ABSTRACT

The last decade has been witness to a rapid growth of interest in logics intended to support reasoning about the interactions between knowledge and action. Typically, logics combining dynamic and epistemic components contain ontic actions (which change the state of the world, e.g., switching a light on) or epistemic actions (which affect the information possessed by agents, e.g., making an announcement). We introduce a new logic for reasoning about the interaction between knowledge and action, in which each agent in a system is assumed to perceive some subset of the overall set of Boolean variables in the system; these variables give rise to epistemic indistinguishability relations, in that two states are considered indistinguishable to an agent if all the variables visible to that agent have the same value in both states. In the dynamic component of the logic, we introduce actions $r(p, i)$ and $c(p, i)$: the effect of $r(p, i)$ is to reveal variable p to agent i; the effect of $c(p, i)$ is to conceal p from i. By using these dynamic operators, we can represent and reason about how the knowledge of agents changes when parts of their environment are concealed from them, or by revealing parts of their environment to them. Our main technical result is a sound and complete axiomatisation for our logic.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Modal Logic*; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Theory

Keywords

Modal logic, epistemic logic, dynamic epistemic logic, interpreted systems, knowledge and change

1. INTRODUCTION

Over the past decade, there has been a rapid growth of interest in logics intended for reasoning about the interaction between knowledge and action (see, e.g., [3] for extensive references). Such *Dynamic Epistemic Logics* make it possible to investigate many different dimensions along which action can interact with knowledge. For example, one recent and very active area of research at the interface of knowledge and action is the study of how communicative utterances such as public announcements change the knowledge of those that witness the utterance [3, 9].

Our aim in the present paper is to introduce a logic that is intended for representing an aspect of the relationship between knowledge and action that has not hitherto been considered: *how the knowledge of agents is changed by concealing parts of their environment from them, and revealing parts of their environment to them.* In more detail, we are concerned with scenarios in which we have a set of agents, where each agent i is associated with a set of Boolean variables V_i, the idea being that agent i can completely and correctly perceive the value of the variables in V_i. We refer to V_i as the *visibility set* of agent i. To represent what an agent knows in such a scenario, our logic uses conventional S5 epistemic modalities K_i, where $K_i\varphi$ means that agent i knows φ [4]. The semantics of epistemic modalities is defined via possible worlds, with the interpretation that two states s and s' are indistinguishable to agent i if the variables V_i have the same value in s and s'. To model revelation and concealment, our logic has dynamic modalites [7], in which we have atomic actions of the form $r(p, i)$ and $c(p, i)$, meaning *reveal p to i* and *conceal p from i*, respectively. The effect of performing the action $r(p, i)$ is that the variable p is added to i's visibility set; and the effect of performing the action $c(p, i)$ is that the variable p is removed from i's visibility set. For example, the following formula of our logic asserts that i doesn't know $p \wedge q$, but after revealing p to i, it does:

$$\neg K_i(p \wedge q) \wedge [r(p, i)]K_i(p \wedge q).$$

The remainder of the paper is structured as follows. We introduce the logic in the following section, beginning with an informal overview of the language, and some example formulae and their intended meaning. We then go on to present the formal syntax and semantics of the logic, and consider a detailed example, showing how the logic can be used to axiomatise properties of a multi-agent voting scenario. We then present the main technical result of the paper: an axiomatisation of the logic, for which we prove completeness using a type of canonical model construction. We conclude with some comments and issue for future work. Throughout the paper we assume some familiarity with modal, dynamic, and epistemic logics (see, e.g., [4, 6, 7, 2]).

2. THE LOGIC

We will begin with an informal overview of the logic, before presenting the formal syntax and semantics.

2.1 Overview

The *Logic of Revelation and Concealment* (LRC) is a combination

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

of the well-known multi-agent epistemic logic $S5_n$ [4] with a specialised dynamic logic component [7], which is intended to allow us to reason about the effects of revealing variables to agents, and concealing variables from them. The language of LRC is parameterised by the following basic sets:

- A set $\mathcal{N} = \{1, \ldots, n\}$ of *agents*;
- A set $\Phi = \{p, q, \ldots\}$ of *Boolean variables*;
- A finite set of $Vis = \{v_1, \ldots, v_m\}$ of *visibility variables*;
- A set $\mathcal{A} = \{\alpha, \alpha', \ldots\}$ of *state changing actions*;
- A set $RC = \{r(v,i), c(v,i)\}$, where $i \in \mathcal{N}$, $v \in Vis$, of *revelation and concealment actions*; and
- a set of variables $SEES = \{sees_i(v) \mid i \in \mathcal{N}, v \in Vis\}$.

The language contains the usual Boolean operators of classical logic ($\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$). When we refer to propositional atoms or variables we mean $\Phi \cup Vis \cup SEES$. Formulae over those variables using only Boolean operators are called *objective*. To represent the knowledge possessed by agents in the system, we use indexed unary modalities K_i, where $i \in \mathcal{N}$, so that $K_i\varphi$ is intended to mean "agent i knows φ". The semantics of epistemic modalities is based on the interpreted systems model of knowledge [4]. Each agent $i \in \mathcal{N}$ is associated with a subset V_i of the variables Vis, with the idea being that the agent i is able to completely and correctly perceive the values of the variables V_i, but is not able to perceive the values of any other variables. We call V_i the *visibility set* of agent i. Thus, in the terminology of interpreted systems, V_i represents the *local state* of i. So, more precisely:

> $K_i\varphi$ means that, given the background knowledge of the agent and the variables V_i that he currently sees, the agent i can infer φ.

Note that the fact that $v \in V_i$ does not imply that i "controls" or has "write access" to i (cf. [11, 5, 10]): it simply means that i is able to see the value of v. Thus it could be that two different agents are able to see some of the same variables (i.e., we might have $v \in V_i$ and $v \in V_j$ for $i \neq j$). We require Vis to be finite for two reasons. The first is technical and it is reflected in the use of a particular axiom, (Ax37), in our axiomatic system, described later in the paper. This axiom helps us to enforce a certain property of the canonical model for our logic. The second reason is purely philosophical. If the agents we are interested in modelling have bounded observational and reasoning capabilities, then they can surely observe only finitely many features of their environment.

Within the object language, we can refer to the variables that an agent sees by using the primitive operators $sees_i(v)$, with the obvious meaning.

To represent actions and the effect that actions have on the system, we use a dynamic logic component, with program modalities $[\pi]$ ("after all executions of program $\pi\ldots$") and $\langle\pi\rangle$ ("after some execution of program $\pi\ldots$"). Programs π within dynamic modalities are constructed from *atomic actions*. Atomic actions in our language are of two types. First, we have a set $\mathcal{A}$ of *state changing actions*, typically denoted $\alpha, \alpha', \ldots$, which are essentially the same as atomic actions in conventional propositional dynamic logic [7]. The effect of performing such an action is to change the state of the system; we allow for the possibility that state changing actions have multiple possible outcomes.

In addition to the conventional PDL-style state-changing actions $\mathcal{A}$, in LRC we have a set RC of two additional types of atomic actions, $r(v,i)$ and $c(v,i)$, where $i \in \mathcal{N}$ and $v \in Vis$. The action $r(v,i)$ is read "reveal v to i", while $c(v,i)$ is read "conceal v from i". The effect of performing $r(v,i)$ is to *add the variable v to agent i's visibility set*, while the effect of $c(v,i)$ is to *remove v from i's visibility set*. These two atomic programs thus directly manipulate an agent's local state, and since what an agent knows is determined solely by its local state, they can also change what an agent knows. Notice, however, that actions $r(v,i)$ and $c(v,i)$ do *not* change the actual state of the system. In this sense, state changing actions $\mathcal{A}$ and visibility actions $r(v,i)$, $c(v,i)$ can be understood as causing changes to an agent's knowledge along two different dimensions: visibility actions $r(v,i)$ and $c(v,i)$ change an agent's visibility set but do not change the state of the system, while actions $\mathcal{A}$ change the state of the system but do not change an agent's visibility set.

Atomic actions are combined into complex programs using the usual program constructs of dynamic logic [7]: $\pi_1;\pi_2$ means "execute program π_1 and then execute program π_2" (sequence); and $\pi_1 \cup \pi_2$ means "either execute program π_1 or execute program π_2" (non-deterministic choice). LRC allows only a limited form of iteration, as follows. If π is a program that does not contain an element of $\mathcal{A}$ (i.e., all sub-programs of π are built from the basic reveal and conceal actions in RC), then π^* means "repeatedly execute π an undetermined number of times". For technical reasons, (discussed in more detail below), we choose to omit the standard dynamic logic "test" operator, $\varphi?$, from the logic LRC.

Finally, and again for somewhat technical reasons, we include within LRC a universal modality $\Box\varphi$. The expression $\Box\varphi$ means "in all states of the model, fixing the current visibility descriptions, φ holds".

2.2 Some Example Formulae

Let us see some examples of formulae of our logic.

$$p \rightarrow [r(p,i)]K_ip$$

This formula says that, if p is true, then after revealing the variable p to i, agent i will know that p is true. This is in fact a valid formula of LRC: the effect of revealing a Boolean variable to an agent will be that the agent knows the value of that Boolean variable.

$$[\alpha]p \rightarrow [r(p,i);\alpha]K_ip$$

This formula says that if after doing α, the variable p is true, then if we reveal p to i and then do α, then i will know that p is true. This is in fact also a valid valid formula of LRC, for all $\alpha \in \mathcal{A}$.

$$K_ir \wedge [c(p,i) \cup c(q,i)]\neg K_ir$$

This formula says that i knows r, but if we choose to conceal either p or q, then i will not know r.

$$\langle r(v,i)^*\rangle\varphi \leftrightarrow \varphi \vee \langle r(v,i)\rangle\varphi$$

This (valid) formula says that φ is true after some undetermined number of executions of the $r(v,i)$ action if, and only if, φ is true now or after at most one repetition of the action $r(v,i)$.

Note that the following is not a well-formed formula of LRC.

$$[(r(v,i) \cup (\alpha_1;\alpha_2))^*]\varphi$$

This is because the iteration operator "*" is here applied to a program that contains state changing actions, i.e., elements of $\mathcal{A}$. We impose this restriction on iteration mainly for technical reasons. Having such formulae would greatly complicate any completness proof for an axiomatic system for LRC, while hiding the main idea behind some difficult technical details; moreover, as is well known, a strong completeness proof is out of reach in this situation. Note also that we do not have programs of the form $\varphi?$. This is justified by the following reasoning. We want to formulate a logic that

π	::=	α	atomic state changing action
	|	$r(v,i)$	reveal v to i
	|	$c(v,i)$	conceal v from i
	|	$\pi;\pi$	sequence
	|	$\pi \cup \pi$	non-deterministic choice
	|	π^*	repeat π some finite number of times (π must contain no actions $\mathcal{A}$)
	|	*skip*	do nothing
φ	::=	$\top$	truth constant
	|	p	propositional atoms
	|	$sees_i(v)$	agent i sees variable v
	|	$\neg\varphi$	negation
	|	$\varphi \vee \varphi$	disjunction
	|	$K_i\varphi$	epistemic box modality
	|	$[\pi]\varphi$	dynamic box modality
	|	$\Box\varphi$	universal modality for fixed visibility structure

Figure 1: Syntax of programs (π) and formulae (φ). Terminal symbols are interpreted as follows: $\alpha \in \mathcal{A}$ is an atomic state changing action, $p \in \Phi \cup Vis$ is an arbitrary Boolean variable, $v \in Vis$ is a visibility variable, and $i \in \mathcal{N}$ is an agent.

can be used for reasoning about the knowledge that can be obtained *only* from directly revealing features of the environment. Since the agents "know" the program that is being executed, performing a test on the value of a variable that is not visible to a certain agent can increase the agent's knowledge. This increase, however, is not because of a direct observation of the variable.

We define the syntax of programs π and formulae φ of the logic LRC by mutual induction through the grammar in Figure 1.

2.3 Models

(The reader may benefit from reading this section together with Example 1, below.) In what follows, we will assume the sets $\mathcal{N}$, Φ, and *Vis* are fixed, with each respective set playing the role described above.

Now, as we explained earlier, every agent $i \in \mathcal{N}$ is assumed to be able to completely and correctly see a subset $V_i \subseteq Vis$. That is, agent i will *know the value of the variables in* V_i; if $p \notin V_i$, then i does not necessarily know the value of p. We refer to V_i as the *visibility set* for agent i, and we refer to a tuple $V = (V_1, \dots, V_n)$, in which we have one visibility set for each agent, as a *visibility structure*. Notice that we place no requirements on visibility sets V_i or visibility structures $(V_1, \dots, V_n)$. It could be that $V_i = V_j$, for example, or even that $V_1 = \cdots = V_n = \emptyset$ (although this latter case would not be very interesting). Let $\mathcal{V}$ denote the set of visibility structures.

Next, we assume a set $\mathcal{S} = \{s_1, \dots, s_m\}$ of *states*, and a standard Kripke valuation function $\theta : \mathcal{S} \to 2^{\Phi \cup Vis}$, which gives the set of variables $\theta(s)$ true in each state $s \in \mathcal{S}$. Notice that θ gives a value both for variables in Φ and variables in *Vis*.

For each state changing action $\alpha \in \mathcal{A}$, we are assumed to have a binary relation $R_\alpha \subseteq S \times S$, capturing the effects of α. The interpretation of this relation is more or less standard for dynamic logic [7]: if $(s, s') \in R_\alpha$, then this means that state s' could result as a possible effect of performing action α in state s.

Putting these components together, a *model*, m, (over the sets fixed above) is a structure

$$m = \langle \mathcal{S}, V, \{R_\alpha\}, \mathcal{R}_\diamond, \theta \rangle,$$

where $\mathcal{S}$ is a state set, $V \in \mathcal{V}$ is a visibility structure, $\{R_\alpha\}$ is a collection of accessibility relations for the state changing actions in $\mathcal{A}$, $\mathcal{R}_\diamond$ is the universal relation on $\mathcal{S}$, and θ is a Kripke valuation function. Let $\mathcal{M}$ denote the set of models.

A *pointed model* is a pair (m, s), where $m \in \mathcal{M}$ is a model and s is a state in m. Below, we will define the satisfaction of formulae with respect to pointed models. Let $\mathcal{P}(\mathcal{M})$ be the set of all pointed models over the set of models $\mathcal{M}$. A *configuration* $f = \langle \mathcal{S}, \{R_\alpha\}, \mathcal{R}_\diamond, \theta \rangle$ abstracts away from the specific visibility structure V. The set M_f is the set of all models over the configuration f.

Where s and s' are two states in $\mathcal{S}$, we write $s \sim_i s'$ to mean that the states s and s' agree on the values of variables V_i, i.e.,

$$s \sim_i s' \qquad \text{iff} \qquad \theta(s) \cap V_i = \theta(s') \cap V_i.$$

The reader will note that the relations $\sim_i$ defined in this way are equivalence relations, and we will later use these relations to define a conventional (S5) interpretation for knowledge modalities, cf. [4].

2.4 Dynamic Accessibility Relations

We must now define the accessibility relations R_π, used to give a semantics to dynamic modalities $[\pi]$ (cf. [6, p.87]). In Propositional Dynamic Logic (PDL), program accessibility relations R_π are binary relations over the set $\mathcal{S}$ of system states. In our logic, they are slightly more complex: they are binary relations over the set $\mathcal{P}$ of pointed models (m, s). No actions in our framework will change a configuration: state changing actions (as the name suggests) change a state, while revealing and concealing actions *RC* change the visibility structure, and hence the model. We will define these relations in three stages: first we define the relations for atomic revelation and concealment programs $r(p,i)$ and $c(p,i)$, then we define the form of accessibility relations for state changing actions, and then finally, we define the accessibility relations for complex programs with respect to these.

Assume $m = (\mathcal{S}, V, \{R_\alpha\}, \mathcal{R}_\diamond, \theta)$ and $m' = (\mathcal{S}', V', \{R'_\alpha\}, \mathcal{R}'_\diamond, \theta')$ are models and s, s' are states such that $s \in \mathcal{S}$ and $s' \in \mathcal{S}'$. Then

$$((m,s),(m',s')) \in \mathcal{R}_{r(v,i)} \text{ iff:}$$

1. $s' = s, \mathcal{S}' = \mathcal{S}, \mathcal{R}' = \mathcal{R}, \mathcal{R}'_\diamond = \mathcal{R}_\diamond$ and $\theta' = \theta$
   Revealing v to i does not change the current point, the state set, any of the accessibility relations, or the truth of atomic propositions.
2. $V'_i = V_i \cup \{v\}$ and for all $j \neq i$, $V'_j = V_j$
   Atom v becomes visible for i after $r(v,i)$ has been executed, but otherwise, i's visibility set is unchanged and the visibility set of every other agent remains unchanged.

We define the relations $\mathcal{R}_{c(v,i)}$ in a similar way:

$$((m,s),(m',s')) \in \mathcal{R}_{c(v,i)} \text{ iff:}$$

1. $s' = s, \mathcal{S}' = \mathcal{S}, \mathcal{R}' = \mathcal{R}, \mathcal{R}'_\diamond = \mathcal{R}_\diamond$ and $\theta' = \theta$
   Concealing v from i does not change the current point, the state set, any of the accessibility relations, or the truth of atomic propositions.
2. $V'_i = V_i \setminus \{v\}$ and for all $j \neq i$, $V'_j = V_j$
   Atom v becomes invisible for i after $c(v,i)$ has been executed, but otherwise, i's visibility set is unchanged and the visibility set of every other agent remains unchanged.

The relation $\mathcal{R}_{skip}$ is the identity, i.e.,:

$$\mathcal{R}_{skip} = \{((m,s),(m,s))\}.$$

Let $\mathcal{R}_{r(p,i)^*} = (\mathcal{R}_{r(p,i)})^*$ and $\mathcal{R}_{c(p,i)^*} = (\mathcal{R}_{c(p,i)})^*$, i.e., $\mathcal{R}_{r(p,i)^*}$ is the reflexive and transitive closure of the relation $\mathcal{R}_{r(p,i)}$; similarly for $\mathcal{R}_{c(p,i)^*}$.

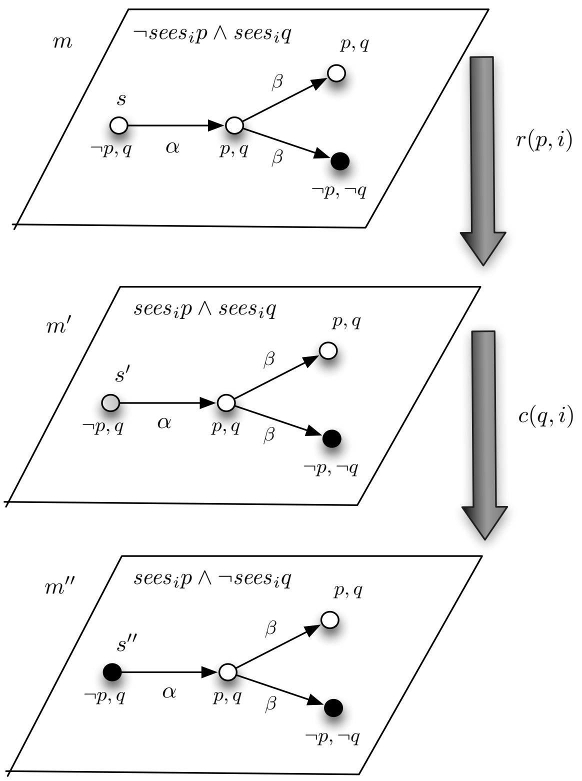

Figure 2: Three models.

EXAMPLE 1. *Consider the models m, m' and m'' from Figure 2, where $\mathcal{N} = \{i\}$ and $Vis = \{p, q\}$. The three models are connected by a $r(p,i)$ and a $c(q,i)$ transition, respectively. Note they are all models over the same configuration (i.e., the only difference between the models is the visibility structure). Points in a model in the same $\sim_i$-equivalence class have the same 'colour' (in m', the state s' induces the equivalence class $\{s'\}$). Since the visibility descriptions are global, they are given at the top of each model. We have the following (for all φ), where the reader may like to take a peek at the definition of entailment at the end of this section:*

1. $(m, s) \models K_i q \wedge \neg K_i p \wedge (\varphi \leftrightarrow [\alpha] K_i \varphi)$
   The agent knows q but not p, and the action α does not change his knowledge.

2. $(m, s) \models \langle r(p, i)\rangle (K_i \neg p \wedge [\alpha] K_i p)$
   After revealing p to i, the agent knows p is false, but also that p would become true after α.

3. $(m, s) \models \langle r(p, i); \alpha; c(q, i)\rangle K_i [\beta] K_i (p \leftrightarrow q)$
   It is possible to reveal p to i, then do α and then conceal q from i so that afterwards, the agent knows that all executions of β lead to states where the agent is sure that p and q are equivalent.

Note that the $\Box$ and $\Diamond$ operators enable us to quantify over states that are present in a model m. We have for instance in (m, s) that $\Box(\neg sees_i(p) \wedge \Diamond[\alpha \cup \beta]\bot)$: what an agent sees is the same in each snapshot, and in m, there is a state where neither α nor β can be performed.

We will now prove that the iteration operator, *, can in fact be eliminated from programs in LRC. This simplifies the semantics of the language, and greatly simplifies the completeness proof we give later. First, we prove a proposition which shows how reveal or conceal actions $\gamma(v, i)$ can be eliminated, or moved "along" a sequence *RC*-actions.

PROPOSITION 1. *The following are true. Let $\gamma(v,i), \hat{\gamma}(v,i)$ be either $r(v,i)$ or $c(v,i)$, such that $\gamma(v,i) = r(v,i)$ iff $\hat{\gamma}(v,i) = c(v,i)$.*

1. $\mathcal{R}_{\gamma(v,i)^*} = R_{skip} \cup \mathcal{R}_{\gamma(v,i)}$;
2. $\mathcal{R}_{\gamma(v,i)} \circ \mathcal{R}_{\gamma(v,i)} = \mathcal{R}_{\gamma(v,i)}$;
3. $\mathcal{R}_{\gamma(v,i)} \circ \mathcal{R}_{\hat{\gamma}(v,i)} = \mathcal{R}_{\hat{\gamma}(v,i)}$;
4. $\mathcal{R}_{r(v,i)} \circ \mathcal{R}_{r(w,k)} = \mathcal{R}_{r(w,k)} \circ \mathcal{R}_{r(v,i)}$;
5. $\mathcal{R}_{c(v,i)} \circ \mathcal{R}_{c(w,k)} = \mathcal{R}_{c(w,k)} \circ \mathcal{R}_{c(v,i)}$;
6. $\mathcal{R}_{r(v,i)} \circ \mathcal{R}_{c(w,k)} = \mathcal{R}_{c(w,k)} \circ \mathcal{R}_{r(v,i)}$*, where $i \neq k$ or $v \neq w$.*
7. $\mathcal{R}_{\gamma(v,i)} \circ \mathcal{R}_{\alpha} = \mathcal{R}_{\alpha} \circ \mathcal{R}_{\gamma(v,i)}$*, where $\alpha \in \mathcal{A}$.*

PROOF. Follows immediately from the definition of the relations $R_{r(v,i)}, R_{c(v,i)}, R_{skip}$. $\Box$

Using the above properties, we prove that for any program built from atomic programs in *RC* only, the following is true.

COROLLARY 1. *Let $\vec{\alpha}_i$ denote a sequence of RC-programs $\alpha_{i_1}; \alpha_{i_2}; \ldots; \alpha_{i_k}$. Let n sequences of RC-programs $\vec{\alpha}_1, \vec{\alpha}_2, \ldots, \vec{\alpha}_n$ be given, and define Σ as the set of sequences σ of RC-programs that are made by choosing an arbitrary number of sequences from $\vec{\alpha}_1, \vec{\alpha}_2, \ldots, \vec{\alpha}_n$ (each $\vec{\alpha}_i$ occurring at most once), and combining them in an arbitrary order using only the operator ";". So $\Sigma = \{skip, \vec{\alpha}_1, \ldots, \vec{\alpha}_n, (\vec{\alpha}_1; \vec{\alpha}_2), \ldots, (\vec{\alpha}_n; \vec{\alpha}_{n-1}), (\vec{\alpha}_1; \vec{\alpha}_2; \vec{\alpha}_3), \ldots, (\vec{\alpha}_n; \vec{\alpha}_{n-1}; \vec{\alpha}_{n-2}), (\vec{\alpha}_1; \vec{\alpha}_2; \ldots; \vec{\alpha}_n), \ldots, (\vec{\alpha}_n; \vec{\alpha}_2; \ldots; \vec{\alpha}_{n-1})\}$. Then*

$$(\vec{\alpha}_1 \cup \ldots \cup \vec{\alpha}_n)^* \quad = \quad \cup_{\sigma \in \Sigma} \sigma$$

PROOF. The statement is best understood via an example. We claim that if $\alpha, \beta \in RC$ then

$$(\alpha \cup \beta)^* = skip \cup \alpha \cup \beta \cup (\alpha; \beta) \cup (\beta; \alpha)$$

This follows from the fact that

$$\begin{aligned}(\alpha \cup \beta)^* \quad = \quad & skip \cup (\alpha \cup \beta) \cup (\alpha \cup \beta); (\alpha \cup \beta) \\ & \cup (\alpha \cup \beta); (\alpha \cup \beta); (\alpha \cup \beta) \ldots .\end{aligned}$$

Consider $(\alpha \cup \beta); (\alpha \cup \beta); (\alpha \cup \beta)$. It is equivalent to

$$\alpha; \alpha; \alpha \cup \alpha; \alpha; \beta \cup \ldots \cup \alpha; \beta; \beta \ldots \cup \beta; \beta; \beta$$

Using the equivalences from Proposition 1, we see that this is actually equivalent to

$$\alpha \cup (\alpha; \beta) \cup (\beta; \alpha) \cup \beta.$$

The proof of the general statement is similar. $\Box$

COROLLARY 2. *Every program is equivalent to a program without the operator *.*

PROOF. This follows from Proposition 1, together with the fact that we do not allow the star operator to be applied to programs containing elements from $\mathcal{A}$ and Corollary 1. $\Box$

Given the atomic relations R_α for state changing actions and $\mathcal{R}_{r(p,i)}$, $\mathcal{R}_{c(p,i)}$ for visibility actions, we obtain the accessibility relations $\mathcal{R}_\pi$ for arbitrary programs π as follows. Let the composition of arbitrary relations R_1 and R_2 be denoted by $R_1 \circ R_2$. Then the accessibility relations for complex programs are defined [7]:

$$\begin{array}{rcl} \mathcal{R}_\alpha &=& \{((m,s),(m,s')) \mid (s,s') \in R_\alpha \ \& \ m \in \mathcal{M}\} \\ \mathcal{R}_{\pi_1;\pi_2} &=& \mathcal{R}_{\pi_1} \circ \mathcal{R}_{\pi_2} \\ \mathcal{R}_{\pi_1 \cup \pi_2} &=& \mathcal{R}_{\pi_1} \cup \mathcal{R}_{\pi_2} \end{array}$$

At last we are ready to give the formal semantics for our logic. Assume that φ is a formula of our logic and that $(m,s) \in \mathcal{P}$ is a pointed structure. Let $\mathcal{M}$ be the class of all models. Then we write $\mathcal{M},(m,s) \models \varphi$ to mean that φ is true at (satisfied in) state s of m. Since $\mathcal{M}$ is fixed, we also write $(m,s) \models \varphi$ for this. The satisfaction relation "$\models$" is inductively defined by the following rules:

$(m,s) \models \top$

$(m,s) \models p$ iff $p \in \theta(s)$ (where $p \in \Phi$);

$(m,s) \models sees_i(v)$ iff $v \in V_i$ (where $v \in Vis$);

$(m,s) \models \neg\varphi$ iff not $(m,s) \models \varphi$

$(m,s) \models \varphi \vee \psi$ iff $(m,s) \models \varphi$ or $(m,s) \models \psi$

$(m,s) \models K_i\varphi$ iff $\forall s'$ with $s \sim_i s'$, we have $(m,s') \models \varphi$

$(m,s) \models \Box\varphi$ iff $\forall s'$ we have $(m,s') \models \varphi$

$(m,s) \models [\pi]\varphi$ iff $\forall (m',s')$ such that $((m,s),(m',s')) \in \mathcal{R}_\pi$ we have $(m',s') \models \varphi$.

We define the remaining connectives of classical logic ("$\perp$" – falsum, "$\wedge$" – and, "$\rightarrow$" – implies, "$\leftrightarrow$" – if, and only if), the diamond dual M_i ("maybe") of the epistemic modality K_i, the diamond dual $\langle\pi\rangle$ of the dynamic box modality, and the diamond dual $\Diamond$ of the universal modality $\Box$ can be defined as abbreviations in the expected way:

$$\begin{array}{rcl} \perp &\hat{=}& \neg\top \\ \varphi \wedge \psi &\hat{=}& \neg(\neg\varphi \vee \neg\psi) \\ \varphi \rightarrow \psi &\hat{=}& (\neg\varphi) \vee \psi \\ \varphi \leftrightarrow \psi &\hat{=}& (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi) \\ M_i\varphi &\hat{=}& \neg K_i \neg\varphi \\ \langle\pi\rangle\varphi &\hat{=}& \neg[\pi]\neg\varphi. \\ \Diamond\varphi &\hat{=}& \neg\Box\neg\varphi. \end{array}$$

2.5 A Detailed Example

EXAMPLE 2. *Suppose we have three members of a committee, a, b, and c who are going to vote to elect a new committee chair. The standing chair is c and the new chair will be chosen from the three of them, by the three of them. Let $p^i_j \in \Phi$ denote that agent i's vote is for agent j (p^i_j is read as "i prefers j"', or "i votes for j"). We assume that votes are fair, in the sense that every agent votes exactly for one agent:*

$$\mu : \bigwedge_{i \in \mathcal{N}} \bigwedge_{k \neq j \neq m \neq k} p^i_j \leftrightarrow \neg p^i_k \wedge \neg p^i_m$$

The rule used for electing a winner is as follows: any agent will be elected if it has a majority of the votes; if there is no majority (everybody gets one vote each), then the standing chair, c, is elected.

Given this rule, let us define abbreviations w_i, denoting that i is the winner:

$$\begin{array}{rcl} w_a &\hat{=}& (\bigvee_{i,j \in \mathcal{N}: i \neq j}(p^i_a \wedge p^j_a)) \\ w_b &\hat{=}& (\bigvee_{i,j \in \mathcal{N}: i \neq j}(p^i_b \wedge p^j_b)) \\ w_c &\hat{=}& (\neg w_a \wedge \neg w_b) \end{array}$$

Let ω collect these three definitions as a conjunction. Finally, we specify who initially sees what: agents initially see only their own vote.

$$\sigma : \bigwedge_{i,j \in \mathcal{N}} sees(p^i_j, i) \wedge \bigwedge_{k \neq i, i,j,k \in \mathcal{N}} \neg sees(p^k_j, i)$$

To express the background information in this scenario, we find it convenient to define a common knowledge *operator, C. This is a standard construction, and we refer the reader to, e.g., [4] for details. Formally, given the individual knowledge operators K_i, we first define an "everyone knows" operator, E, as follows: $E\varphi \hat{=} \bigwedge_{i \in \mathcal{N}} K_i\varphi$. We then define the common knowledge operator $C\varphi$ as the maximal fixed point solution to the expression $E(\varphi \wedge C\varphi)$.*[1]

Now, let the background information χ be $\mu \wedge \omega \wedge \sigma$. We will assume that χ is common knowledge among $\{a,b,c\}$: all agents know (and they know that they know, etc) that they vote for only one candidate, what the definition of winning is, and which variables are initially seen.

Let us consider the vote $\nu = p^a_b \wedge p^b_c \wedge p^c_b$, which of course is not commonly known. We then have

$$(C\chi \wedge \nu) \rightarrow [r(p^a_b, c)]K_c w_b \tag{1}$$

If a's vote is revealed to c, then c knows who the winner is (it is b).

Agents a and b do not know that c in this case knows who the winner is, even if they know that c learns b's vote:

$$(C\chi \wedge \nu) \not\rightarrow (K_a[r(p^a_b, c)]K_aK_c w_b \vee K_b[r(p^a_b, c)]K_bK_c w_b) \tag{2}$$

Likewise, we have

$$(C\chi \wedge \nu) \rightarrow [r(p^b_c, a) \cup r(p^c_b, a)]K_a \neg w_a \tag{3}$$

If a learns the vote of one of the other committee members, he knows that he has not won the election.

Let now α be $(r(p^b_c, a) \cup r(p^c_b, a)); (c(p^b_c, a) \cup c(p^c_b, a))$ (randomly reveal one of the variables p^b_c and p^c_b to a, and then conceal randomly one of them). Then

$$(C\chi \wedge \nu) \rightarrow (\langle\alpha;\alpha\rangle K_a p^b_c \wedge \langle\alpha;\alpha\rangle \neg K_a p^b_c) \tag{4}$$

That is, there is a choice of revealing and concealing the variables so that a finds out he is winning, and there is a choice that he is not. This should be contrasted with the program $\alpha' = (r(p^b_c, a); c(p^b_c, a)) \cup (r(p^c_b, a); c(p^c_b, a))$, for which we obtain

$$(C\chi \wedge \nu) \rightarrow [\alpha';\alpha']\neg K_a p^b_c \tag{5}$$

(If the agent gets a random variable revealed and then concealed, he does not learn anything).

Our example easily illustrates how agents can know atoms without seeing them:

$$(C\chi \wedge \nu) \rightarrow \neg K_c \neg p^a_c \wedge [r(p^a_b, c)](\neg sees(p^a_c, c) \wedge K_c \neg p^a_c) \tag{6}$$

In words, given the initial constraints $(C\chi \wedge \nu)$, agent c does not know that p^a_c is false. However, after p^a_b is revealed to c, although he

[1] In the following, the C-operator does not appear in the consequent of any of the example formula, and hence can be omitted: for this example it suffices to think of it as an abbreviation of mutual knowledge of depth three.

still does not see p_c^a, he now knows its true value! This is because c knows that a only votes for one agent, i.e., he knows that certain constraints between variables exist.

When building or verifying intelligent agents, one might wonder what the benefit is of having conceal-actions, in which agents just 'forget' the truth value of certain atomic propositions. This makes sense in many scenario's where agents need at some time sufficient, maybe sensitive information to take an appropriate decision, where they should not 'accumulate' too much of this information. An example of this might be an agent who grants users access to a sensitive website, and the users posess an n-character password, where for each login session, they only are required to reveal k positions of this password ($k < n$). The agent deciding whether the user is allowed access should in such cases at every login attempt know whether the user has provided the right k characters, but when the login session ends, it should be ensured that this information is not remembered, since otherwise he in the end would learn the complete password. Instead of formalising this additional scenario, let us now show how, even in the voting setting, the possibility of concealing can be useful.

EXAMPLE 3. *Continuing with the voting example, let us introduce new atoms δ_i ($i \in \mathcal{N}$) meaning that i is the declared winner of the vote. Initially, we have*

$$\delta : \bigwedge_{i,j\in\mathcal{N}} \neg\delta_i \wedge sees(\delta_i, j) \tag{7}$$

That is, initially nobody is the declared winner, and the fact who is a declared winner is visible for each agent. We also assume an action β_i, which models that the chair c can declare that i is the winner. In order for i being enabled to be declared the winner, the pre-condition is $K_c w_i$ and the post-condition is δ_i.

$$(\neg K_c w_i \rightarrow \neg\langle\beta_i\rangle\top) \wedge (K_c w_i \rightarrow \langle\beta_i\rangle\delta_i) \tag{8}$$

Before defining our procedure for declaring the winner, let r_c (reveal to c) be short for

$$r_c : r(p_a^a, c); r(p_b^a, c); r(p_c^a, c); r(p_a^b, c); r(p_b^b, c); r(p_c^b, c)$$

Similarly, c_c is like r_c, but rather than revealing a's and b's votes to c, they are concealed from them. Define

$$\gamma = r_c; (\beta_a \cup \beta_b \cup \beta_c); c_c$$

Then γ has the following properties:

1. $C(\chi \wedge \delta) \rightarrow [\gamma](w_i \leftrightarrow K_a\delta_i \wedge K_b\delta_i \wedge K_c\delta_i)$
   After γ, any winner is known to be a declared winner
2. $C(\chi \wedge \delta) \rightarrow [\gamma]\neg(K_c p_a^a \vee K_c p_b^a \vee K_c p_c^a)$
   That is, after every execution of γ, agent c does not know (does not remember) a's vote (the same is of course true for b's vote). It is in fact easy to see that after execution of γ, we have $\neg K_j p_k^i$, for any $i \neq j$.

3. AXIOMS

We now present an axiomatization for LRC: the main technical result of our paper is that this axiomatization is sound and complete. We first present the axiomatization and discuss the properties the various axioms are capturing, before describing the completeness proof in the following subsection. The axiomatization is presented in Tables 1 and 2. Table 1 deals with the knowledge axioms, the axioms for $\Box$ and the inference rules of the logic. This is all fairly standard: the axioms say that both K_i and $\Box$ are $S5$-operators,

Propositional Logic:	
(Ax1)	propositional tautologies
S5 Axioms for Knowledge:	
(Ax2)	$K_i(\varphi \rightarrow \psi) \rightarrow (K_i\varphi \rightarrow K_i\psi)$
(Ax3)	$K_i\varphi \rightarrow \varphi$
(Ax4)	$\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$
S5 Axioms for State of Revelation:	
(Ax5)	$\Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$
(Ax6)	$\Box\varphi \rightarrow \varphi$
(Ax7)	$\neg\Box\varphi \rightarrow \Box\neg\Box\varphi$
Inference Rules:	
(IR1)	From $\vdash \varphi \rightarrow \psi$ and $\vdash \varphi$ infer $\vdash \psi$
(IR2)	From $\vdash \varphi$ infer $\vdash K_i\varphi$
(IR3)	From $\vdash \varphi$ infer $\vdash [\pi]\varphi$

Table 1: Inference rules for LRC and some aixoms.

which is standard for knowledge (see e.g., [4]) and for the universal modality (see, e.g., [2]). Notice that the positive introspection axiom, ($K_i \rightarrow K_iK_i\varphi$), follows from the other axioms, and similarly for the $\Box$ modality.

The axioms of Table 2 relate to the dynamic component and the interaction between our modalities. Of the dynamic logic axioms:

(Ax8) and (Ax9) say that actions conceal and reveal are deterministic: they lead to a unique outcome.

(Ax10) and (Ax11) say that reveal and conceal are idempotent: repeating them has no effect.

(Ax12) and (Ax13) explain that when doing a reveal and a conceal action in sequence, it is the last performed action that determines the result.

(Ax14), (Ax15) and (Ax18) say that two actions from *RC* commute with each other, as long as they concern different agents or different variables.

According to (Ax16) and (Ax17), atomic actions from *RC* and from $\mathcal{A}$ commute. Semantically, this is illustrated by Figure 2: "horizontal" and "vertical" steps can be taken in arbitrary order.

(Ax19)–(Ax21) are a standard set of axioms for the dynamic logic constructs of our logic, with a direct correspondence to PDL [7, p.173].

Finally, we have 16 interaction axioms in LRC:

(Ax22) and (Ax23) 'generalise' axioms (Ax16) and (Ax17), and are again illustrated by Figure 2.

Axioms (Ax24)–(Ax27) are *persistence properties* of *sees* and $\neg$*sees*. (Ax24) and (Ax25) for instance say that the fact that agent i sees variable v is not undone by concealing either another variable from i, or concealing a variable from another agent (Ax24), and persists through any reveal action.

(Ax28) and (Ax29) say that who sees what is a global property in a model m (the only way to change this is to perform an *RC*-action, which leads to a model m').

(Ax30) and (Ax31) give the non-persistence of *sees* (it can become false through an apropriate conceal action) and $\neg$*sees* (which can become false through an appropriate reveal action).

Dynamic Logic Axioms:		
(Ax8)	$\langle c(v,i)\rangle\varphi \leftrightarrow [c(v,i)]\varphi$	
(Ax9)	$\langle r(v,i)\rangle\varphi \leftrightarrow [r(v,i)]\varphi$	
(Ax10)	$\langle r(v,i); r(v,i)\rangle\varphi \leftrightarrow \langle r(v,i)\rangle\varphi$	
(Ax11)	$\langle c(v,i); c(v,i)\rangle\varphi \leftrightarrow \langle c(v,i)\rangle\varphi$	
(Ax12)	$\langle r(v,i); c(v,i)\rangle\varphi \leftrightarrow \langle c(v,i)\rangle\varphi$	
(Ax13)	$\langle c(v,i); r(v,i)\rangle\varphi \leftrightarrow \langle r(v,i)\rangle\varphi$	
(Ax14)	$\langle c(v,i); c(w,k)\rangle\varphi \leftrightarrow \langle c(w,k); c(v,i)\rangle\varphi$	if C_1
(Ax15)	$\langle r(v,i); r(w,k)\rangle\varphi \leftrightarrow \langle r(w,k); r(v,i)\rangle\varphi$	if C_1
(Ax16)	$\langle c(v,i)\rangle\langle\alpha\rangle\varphi \leftrightarrow \langle\alpha\rangle\langle c(v,i)\rangle\varphi$	if C_2
(Ax17)	$\langle r(v,i)\rangle\langle\alpha\rangle\varphi \leftrightarrow \langle\alpha\rangle\langle r(v,i)\rangle\varphi$	if C_2
(Ax18)	$\langle r(v,i); c(w,k)\rangle\varphi \leftrightarrow \langle c(w,k); r(v,i)\rangle\varphi$	if C_1
(Ax19)	$[\pi](\varphi \rightarrow \psi) \rightarrow ([\pi]\varphi \rightarrow [\pi]\psi)$	
(Ax20)	$[\pi_1 \cup \pi_2]\varphi \leftrightarrow [\pi_1]\varphi \wedge [\pi_2]\varphi$	
(Ax21)	$[\pi_1; \pi_2]\varphi \leftrightarrow [\pi_1][\pi_2]\varphi$	
Interaction Axioms:		
(Ax22)	$\Diamond\langle r(v,i)\rangle\varphi \leftrightarrow \langle r(v,i)\rangle\Diamond\varphi$	
(Ax23)	$\Diamond\langle c(v,i)\rangle\varphi \leftrightarrow \langle c(v,i)\rangle\Diamond\varphi$	
(Ax24)	$sees_i(v) \rightarrow [c(w,k)]sees_i(v)$	if C_1
(Ax25)	$sees_i(v) \rightarrow [r(w,j)]sees_i(v)$	
(Ax26)	$\neg sees_i(v) \rightarrow [c(w,j)]\neg sees_i(v)$	
(Ax27)	$\neg sees_i(v) \rightarrow [r(w,k)]\neg sees_i(v)$	if C_1
(Ax28)	$sees_i(v) \rightarrow \Box sees_i(v)$	
(Ax29)	$\neg sees_i(v) \rightarrow \Box\neg sees_i(v)$	
(Ax30)	$sees_i(v) \rightarrow \langle c(v,i)\rangle\neg sees_i(v)$	
(Ax31)	$\neg sees_i(v) \rightarrow \langle r(v,i)\rangle sees_i(v)$	
(Ax32)	$\varphi_0 \rightarrow [\pi]\varphi$	if C_3
(Ax33)	$\Box\varphi \rightarrow [\alpha]\varphi$	if C_2
(Ax34)	$\Box\varphi \rightarrow K_i\varphi$	
(Ax35)	$sees_i(v) \wedge v \rightarrow K_i v$	
(Ax36)	$sees_i(v) \wedge \neg v \rightarrow K_i\neg v$	
(Ax37)	$\big(\bigwedge_{u\in U}(u \wedge sees_i(u)) \wedge \bigwedge_{v\in V}(\neg v \wedge sees_i(v))$ $\wedge \bigwedge_{w\in W}\neg sees_i(w)\big) \rightarrow$ $(K_i\varphi \rightarrow \Box((\bigwedge_{u\in U} u \wedge \bigwedge_{v\in V}\neg v) \rightarrow \varphi))$	if C_4

Table 2: An axiomatization. The condition C_1 reads $i \neq k$ or $v \neq w$, condition C_2 is that $\alpha \in \mathcal{A}$, condition C_3 is φ_0 is objective and π contains no actions from $\mathcal{A}$, and, finally, C_4 is that $Vis = U \cup V \cup W$.

(Ax32) says that actions from RC do not change the value of atoms in a sate.

(Ax33) and (Ax34) explain that any action from $\mathcal{A}$ keeps us in m and an agent only considers states in m possible.

(Ax35) and (Ax36) capture the basic interaction between visibility sets and knowledge: it an agent i sees a variable v, then i correctly knows the value of v Notice that the converse implication does not hold: to see this, suppose the state set $\mathcal{S}$ was a singleton; then every agent would know the value of every variable, irrespective of whether they could see it or not.

Finally, (Ax37) splits up the visibility atoms in three sets U, V and W. The atoms in U are all true and seen by i. The atoms in V are false and seen by i. None of the atoms in W are seen by i. Then, if i knows that φ, then φ must be true in every state that agrees on the atoms in U and V. In other words, every state (in the same m) that agrees on the atoms that i sees is considered possible by him.

3.1 Completeness

The technical details of our completeness proof are rather involved, and so here we will simply describe the key steps on which these details are based. By Corollary 2, we can restrict ourselves to the language without the operator *. We work with the canonical model for our logic, which is built from maximal consistent sets (see [2] for the relevant notions): completeness of our logic follows from Lemma 2, below. In a nutshell, we take a consistent formula, include it in a maximal consistent set s, and in the canonical model, where states are consistent sets, truth in the model at state s and membership of a formula in s coincide, demonstrating that consistent formulas. The fact that we have eliminated the star operator from our language means that we do not have to introduce the machinery of the Fisher-Ladner closure that is used to deal with the non-compactness of the full Propositional Dynamic Logic. This, however does not mean that a completeness proof will be straightforward, because we have a technical problem of a different nature. In particular, the canonical model for our logic consists of maximal consistent sets of formulae (MCS) that are related via the canonical relations $R^c_\alpha, R^c_\Diamond, R^c_i$ in the usual way. Of course, we can prove a truth lemma with respect to this model but the model itself does not consist of pointed models of the form (m, s) that are related in the desired way. Therefore, this canonical model must be transformed so that it has the desired properties.

Definition 1. *The canonical model for our logic is*

$$M = \langle W, \mathcal{R}^c_\Diamond, \mathcal{R}^c_\alpha, \mathcal{R}^c_i, \theta^c\rangle, \quad \textit{where:}$$

- *W is the set of all maximal consistent sets of formulae;*
- *$\Gamma R^c_\alpha \Delta$ iff for all formulae φ: if $\varphi \in \Delta$, then $\langle\alpha\rangle\varphi \in \Gamma$;*
- *$\Gamma R^c_\Diamond \Delta$ iff for all formulae φ: if $\varphi \in \Delta$, then $\Diamond\varphi \in \Gamma$;*
- *$\Gamma R^c_i \Delta$ iff for all formulae φ: if $\varphi \in \Delta$, then $M_i\varphi \in \Gamma$;*
- *$\theta^c(p) = \{\Gamma \in W \mid p \in \Gamma\}$.*

It is a standard result in modal logic that the relations $R^c_\Diamond$ and R^c_i are reflexive symmetric and transitive. In addition, the axioms governing composition and union of programs are Sahlqvist formulae and, therefore, canonical. It is a standard exercise in modal logic to prove that R^c_α satisfies the regularity conditions:

- $R^c_{\alpha;\beta} = R^c_\alpha \circ R^c_\beta$
- $R^c_{\alpha\cup\beta} = R^c_\alpha \cup R^c_\beta$.

We begin the transformation of this canonical model by defining pointed models of the form (m, s).

Definition 2. *For every MCS $s \in W$ and all $\alpha \in \mathcal{A}$, $mod(s) = (m, s) = \langle S, V, R_\alpha, R_\Diamond, \theta\rangle$, where*

1. *$S = \{t \in W \mid sR^c_\Diamond t\}$;*
2. *$R_\alpha = R^c_\alpha \cap (S \times S)$;*
3. *$R_\Diamond = R^c_\Diamond \cap (S \times S)$;*
4. *$V = \langle V_1, \ldots, V_n\rangle$ is such that $v \in V_i$ iff $sees_i(v) \in s$;*
5. *$\theta(p) = \theta^c(p) \cap (S \times S)$.*

It follows from 1 that $R_\Diamond$ is the universal relation on S. We could also add an epistemic relation $R_i = R^c_i \cap (S \times S)$ to interpret knowledge: (Ax33) and (Ax34) ensure that $R_i \subseteq R_\Diamond$ and $R_\alpha \subseteq R_\Diamond$. Of course, we have that R_i and $R_\Diamond$ are equivalence relations because reflexivity, transitivity and symmetry are modally definable universal properties that are preserved under taking generated sub-models. We need to show that in fact this R_i relation captures exactly the truth-definition of knowledge in models based on the relation $\sim_i$: in fact, (Ax37), together with the axioms that relate knowledge with *sees* and $\Box$ are crucial here. In particular, Ax37 guarantees that within a fixed visibility structure, given a state s, there are no states s' for which $s \sim_i s'$ yet agent i would not consider them the same.

LEMMA 1 (EXISTENCE LEMMA FOR POINTED MODELS). *For all pointed models (m, s) as defined above and all $t \in S$,*

$\Diamond\varphi \in t$ *iff there is a* $t_1 \in S$ *such that* $tR_\Diamond t_1$ *and* $\varphi \in t_1$*;*

$M_i\varphi \in t$ *iff* $\exists t_1 \in S$ *such that* $tR_i t_1$ *and* $\varphi \in t_1$*;*

$R_\alpha\varphi \in t$ *iff* $\exists t_1 \in S$ *such that* $tR_\alpha t_1$ *and* $\varphi \in t_1$.

PROOF. By induction on the structure of φ. $\Box$

Having defined our pointed models. We move to defining our transformed canonical model. We start with defining the following operation:

DEFINITION 3. *If* $(m, s) = \langle S, V, R_\alpha, R_\Diamond, \theta\rangle$*, then*

$r(v, i)(m, s) = mod(s')$ *as defined in Definition 2, where* $s' = s \setminus \{\neg sees_i(v)\} \cup \{sees_i(v)\}$

$c(v, i)(m, s) = mod(s')$ *as defined in Definition 2, where* $s' = s \setminus \{sees_i(v)\} \cup \{\neg sees_i(v)\}$

Intuitively, we form the set $r(v, i)(m, s)$ by collecting all maximal consistent sets that are related via the canonical relation $R^c_{r(v,i)}$ to the maximal consistent sets in S.

PROPOSITION 2. *For all pointed models* (m, s)*,* $r(v, i)(m, s)$ *and* $c(v, i)(m, s)$ *are pointed models.*

We now define a structure that collects all pointed models.

DEFINITION 4. $\mathbb{M} = \{\mathbb{W}, \mathbb{R}_\pi\}$*, where*

- $\mathbb{W} = \{(m, s) \mid s \in W\}$*, i.e.,* $\mathbb{W}$ *consists of all pointed models for every MCS in the canonical model M as defined in Definition 1.*
- $\mathbb{R}_\pi$ *is defined inductively as follows.*
  1. *If* π *is an atomic state changing action* $\alpha \in \mathcal{A}$*, then* $(m, s)\mathbb{R}_\pi(m_1, s_1)$ *iff* $m = m_1$ *and* $sR_\pi s_1$ *in the sense of item 2 from Definition 2.*
  2. *If* π *is* $r(v, i)$*, then*

     $$(m, s)\mathbb{R}_\pi(m_1, s_1) \text{ iff } (m_1, s_1) = r(v, i)(m, s);$$
  3. *If* π *is* $c(v, i)$*, then*

     $$(m, s)\mathbb{R}_\pi(m_1, s_1) \text{ iff } (m_1, s_1) = c(v, i)(m, s);$$
  4. $\mathbb{R}_{\alpha_1;\alpha_2} = \mathbb{R}_{\alpha_1} \circ \mathbb{R}_{\alpha_1}$*;*
  5. $\mathbb{R}_{\alpha_1\cup\alpha_2} = \mathbb{R}_{\alpha_1} \cup \mathbb{R}_{\alpha_1}$.

Now we can prove the desired Truth lemma

LEMMA 2 (TRUTH LEMMA FOR.). *For all* LRC *formulae* φ*:*

$$\mathbb{M}, (m, s) \models \varphi \text{ iff } \varphi \in s.$$

Completeness then follows via a standard argument. Note that the semantic structure $\mathbb{M}, (m, s)$ 'corresponds' to the set of all possible pointed models.

4. CONCLUSIONS

We developed a logic LRC, that allows us to reason about the effects of epistemic actions that reveal and conceal parts of an environment to an agent. In that sense, our logic is a direct 'dynamisation' of the interpreted systems approach to epistemic logic. Such epistemic actions seem very natural, and we believe that several applications of our logic are possible, for example in the area of security (LRC might for instance model situations where a user can access a secure website by only revealing part of his password). For future work, it will be interesting to consider the possibility of revealing and concealing *actions*, rather than variables, although this is likely to require a more elaborate semantic framework than that presented in this paper. Another natural extension would be to weaken the assumption that it is publicly known who sees which variables.

One of the few papers we are aware of that deal with ontic and epistemic actions is [12]. There, the state changing actions are assignments, and the epistemic actions are public announcements. In our framework, we can model public announcements of atoms (just reveal the value to all agents), but not directly disjunctions of them for instance, or epistemic formulas. On the other hand, [12] does not offer a logic for their framework: perhaps a synergy between their and our framework would lead to an interesting and well-understood framework that mixes ontic and epistemic actions. One might also expect that our approach is related to work on *awareness*, especially dynamic versions of it (cf. [1]). Although this warrants further investigation, a main difference is that if an agent i does not see p, we still have $K_i(p \vee \neg p)$ (the agent knows that p has some truth value), whereas, if i is not aware of p, the negation of this holds.

Interesting venues for further research are the connection with Dynamic (Epistemic) Logic, and decidability. As shown in [8], having a 'universal modality' may jeopardise decidability of a logic, but since our universal modality is 'restricted' to visibility structures, the logic might well be decidable.

5. REFERENCES

[1] J. van Benthem and F. R. Velázquez-Quesada. The Dynamics of Awareness. *Synthese*, **177**:1, pp. 5–27, 2010.

[2] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge UP, 2001.

[3] H. Ditmarsch, W. Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer, 2007.

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

[5] J. Gerbrandy. Logics of propositional control. In *Proc. AAMAS-2006*, Hakodate, Japan, 2006.

[6] R. Goldblatt. *Logics of Time and Computation*. CSLI, 1987.

[7] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.

[8] E. Hemaspaandra. The Price of Universality. *Notre Dame J. Formal Logic* **37**:2, pp. 174–203, 1996.

[9] C. Lutz. Complexity and succinctness of public announcement logic. In *Proc. AAMAS-2006*, Hakodate, Japan, 2006.

[10] W. van der Hoek, D. Walther, and M. Wooldridge. Reasoning about the transfer of control. *JAIR*, 37:437–477, 2010.

[11] W. van der Hoek and M. Wooldridge. On the logic of cooperation and propositional control. *Artificial Intelligence*, 64:81–119, 2005.

[12] H. van Ditmarsch, W. van der Hoek, and B. Kooi. Dynamic epistemic logic with assignment. *Proc. AAMAS-2005*, pages 141–148, New York, USA, 2005. ACM Inc.