Small steps in dynamics of information
Velazquez Quesada, F.R.

Citation for published version (APA):
As mentioned before, Logic itself has been traditionally identified with the study of inferences and inferential relations. Also traditionally, Logic has focused on truth-preserving inference: inference in which the truth of the premises guarantees the truth of the conclusion. Nevertheless, there are not so many approaches that study inference from an agent’s point of view. Our first step towards a framework for representing small steps in dynamics of information is the development of a framework in which we can represent the action of truth-preserving inference and reason about it.

But our goal is not to represent each one of the different discussed actions in isolation; our goal is to represent them as different components of the same framework, so we can study not only the particular effect of each one of them, but also the way they interact and work together. Then, in this chapter, we will present a framework in which we can represent not only the act of truth-preserving inference, but also a non-omniscient version of the act of observation, that is, explicit observation.

2.1 The Restaurant example

Consider the following situation, from van Benthem (2008a):

You are in a restaurant with your parents, and you have ordered three dishes: fish, meat, and vegetarian, for you, your father and your mother, respectively. Knowing that each person gets one dish, a new waiter comes from the kitchen with the full order. What can he do to get to know which dish corresponds to which person?

The waiter can ask “Who has the fish?”; then he can ask “Who has the meat?”. Now he does not have to ask anymore: “two questions plus one inference are all that is needed” (van Benthem 2008a).
This example shows the two mentioned logical processes at work. On one hand we have acts of observation, represented by the answers the second waiter receives for his questions. Acts of this kind reflect the agent’s interaction with her environment, and provide her with new arbitrary (and yet truthful) information. On the other hand, we also have an act of inference, more precisely, an act of deduction, represented by the reasoning step the waiter performs to realize that the remaining person should get the remaining dish. Acts of this kind are more ‘internal’, and allow the agent to derive new information based on what she already has.

Like van Benthem mentions, these two phenomena fall directly within the scope of modern Logic, since “asking a question and giving an answer is just as ‘logical’ as drawing a conclusion!” (van Benthem 2008b). Indeed, both processes are equally important in their own right, but so is their interaction. This chapter is devoted to the development of a logical framework that allows us to represent inference and observation together.

The approach of the present chapter results from combining ideas for representing inference in a modal framework with ideas for representing observations in the same setting. The key notions for the latter have been already introduced (the Observation Logic of Section 1.2.1), so now we will provide a brief summary of two modal approaches for inference: Dynamic Syntactic Epistemic Logic and Logic for Rule-Based Agents.

## 2.2 Modal truth-preserving inference

The two approaches discussed in this section emerged as proposals for solving the logical omniscience problem. Though most of the proposals for solving this problem focus on weakening the properties of the agent’s information, some authors (Drapkin and Perlis (1986); Duc (1995) among others) have argued that solutions of this kind are not acceptable. Their main reasons can be summarized in the following two points.

1. The agent’s information can be weakened in many ways, and there is no clear method to decide which restrictions produce reasonable agents and which ones make them too strong/weak.

2. These approaches do not look at the heart of the matter: they still describe the agent’s information at a single (probably final) stage, without looking at how such state is reached.

*Dynamic Syntactic Epistemic Logic* and *Logic for Rule-Based Agents* are based on the idea of dynamizing Epistemic Logic. As it is argued, by saying that an agent knows the laws of Logic, we do not mean that she knows some facts about the world, but rather that she is able to use these laws in the proper
situations to draw conclusions from what she already knows. This idea allows “a good trademark between logical omniscience and logical ignorance: the agent is surely not omniscient with respect to her actual or explicit knowledge, but neither is she logically ignorant” because she can always extend her information by means of the proper reasoning steps (Duc 1997).

2.2.1 Dynamic Syntactic Epistemic Logic

In Duc (1995, 1997, 2001), Ho Ngoc Duc proposes a Dynamic Syntactic Epistemic Logic to represent truth-preserving inference in a modal framework. His main goal is to represent an agent that is not logically omniscient, but nevertheless has enough reasoning abilities to extend what she knows. In other words, the agent’s knowledge does not appear automatically; it is the result of an action.

Duc proposes several languages, the main difference between them being the sub-language used for expressing what the agent can get to know (just propositional formulas, propositional and epistemic formulas, etc.). His approach is mainly syntactic in the sense that he mostly focuses on presenting different axiom systems and discussing how reasonable are some postulates about the agent’s reasoning abilities (he discusses questions like “Should the agent have perfect recall?”, “Can she reach an omniscient state?”, “Is the reasoning linear or branching?”). Among his proposals, the most interesting for us is the one in which he presents not only a language but also a semantic model: the logic $L_{BDE}$ (Duc 1995).

The language for this logic is built in two stages. There is an internal language, the propositional one, that allows us to express the knowledge the agent can get. Then there is another language to reason about this knowledge and how it evolves.

**Definition 2.1 (Language $L_{BDE}$)** Let $At$ denote the set of formulas of the form $\Box \gamma$ for $\gamma$ a propositional formula. The language $L_{BDE}$ is the smallest set of formulas that contains $At$ and is closed under negation, conjunction and the modal operator $\langle F \rangle$. More precisely, the language $L_{BDE}$ is given by

$$\gamma ::= p \mid \neg \gamma \mid \gamma \lor \delta$$
$$\phi ::= \Box \gamma \mid \neg \phi \mid \phi \lor \psi \mid \langle F \rangle \phi$$

Formulas of the form $\Box \gamma$ and $\langle F \rangle \phi$ are read as “$\gamma$ is known” and “after some course of thought of the agent, $\phi$ is true”, respectively. Note how the agent’s knowledge is restricted to propositional formulas and how $L_{BDE}$ itself does not allow us to talk about the real world.

A $BDE$-model is a small variation of a possible worlds model satisfying some special requirements.
Definition 2.2 (BDE-model) A general model \( M \) is a tuple \((W, R, V)\) where \( W \neq \emptyset \) is the set of possible worlds, \( R \subseteq (W \times W) \) is a transitive binary relation and \( V : W \rightarrow \mathcal{P}(\text{At}) \) associates a set of formulas of \( \text{At} \) to each possible world. Then, a \( \text{BDE-model} \) is a general model \( M \) in which

1. for all \( w \in W \), if \( \Box \gamma \in V(w) \) and \( Rwu \), then \( \Box \gamma \in V(u) \);

2. for all \( w \in W \), if \( \Box \gamma \) and \( \Box (\gamma \rightarrow \delta) \) are in \( V(w) \), then there is a world \( u \) such that \( Rwu \) and \( \Box \delta \in V(u) \).

3. if \( \gamma \) is a propositional tautology, then for all \( w \in W \) there is a world \( u \) such that \( Rwu \) and \( \Box \gamma \in V(u) \).

Definition 2.3 (Semantic interpretation) Given a \( \text{BDE-model} \), the semantic interpretation for negation and conjunctions is as usual. For formulas in \( \text{At} \) and ‘course-of-thought’ formulas, we have

\[
(M, w) \models \Box \gamma \text{ iff } \Box \gamma \in V(w) \\
(M, w) \models \langle F \rangle \phi \text{ iff } \text{there is a } u \in W \text{ such that } Rwu \text{ and } (M, u) \models \phi
\]

Duc’s strategy is now clear. In order to avoid logical omniscience, he represents the agent’s knowledge in a syntactic form, with the function \( V \) returning those formulas the agent knows at each world. In order to avoid logical ignorance he introduces inference, representing it as a relation that stands for the reasoning steps the agent can perform in order to change her knowledge.

Given the semantic interpretation, we can now see the meaning of the three semantic requirements. The first guarantees that the agent’s knowledge will only grow as she reasons, and the second and third guarantee that this knowledge will be closed under modus ponens and will contain all tautologies at some point in the future. So though a \( L_{\text{BDE}} \)-agent is not omniscient, she has the logical resources to eventually derive all that follows logically from what she currently knows (i.e., she can reach an omniscient state).

Though the important part of the language, the modality \( \langle F \rangle \), is similar to the ‘future’ modality in Tense Logic (Prior 1957), Duc discusses a more interesting interpretation of these “course of thoughts”. If the set of actions the agent can perform is explicitly given by \( \{r_1, \ldots, r_n\} \), then \( F \) actually stands for \( (r_1 \cup \cdots \cup r_n)^+ \): the transitive closure of the non-deterministic application of actions in the set. This makes the approach closer to the ideas in Propositional Dynamic Logic (PDL; Harel et al. 2000). In fact, we could look at a more appealing PDL-style language that states explicitly which are the actions that the agent performs. To quote one of Duc’s examples, assume that the agent knows the conjunction of \( p \) and \( p \rightarrow q \), that is, \( \Box (p \land (p \rightarrow q)) \). In classical Epistemic Logic, it follows that the agent knows \( p \land q \), that is, \( \Box (p \land q) \). But there is no guarantee that a realistic agent will know \( p \land q \) automatically. What we should say instead is that if she
knows $p$ and $p \to q$, and she reasons appropriately, then she will get to know $p \land q$. In this concrete case, let $CE$, $MP$ and $CI$ stand for the rules of conjunction elimination, modus ponens and conjunction introduction, respectively. Then, by using PDL-style notation (‘;’ stands for sequential composition), instead of the omniscient $\square (p \land (p \to q)) \to \square (p \land q)$, we get the more realistic

$$\square (p \land (p \to q)) \to \langle CE; MP; CI \rangle \square (p \land q)$$

But semantically, in order to formalize this example, we need more than just the abstract relation $R$ of before. We need specific relations $R_{CE}$, $R_{MP}$, $R_{CI}$ and so on for each one of the inference steps the agent can perform. More importantly, we need to be sure that each one of these relations follows the intuition behind it: if $R_{MP}$ relates world $w$ with world $u$, then at $w$ the agent should know an implication and its antecedent, and at $u$ her knowledge should be extended with the implication’s consequent.

The approach that we will recall now attacks the second problem, providing formal definitions of what a relation should satisfy in order to represent properly a rule-application.

### 2.2.2 Logic for Rule-Based Agents

Based on the prominent case of rule-based agents of the Artificial Intelligence (AI) literature, Mark Jago proposes in Jago (2006a,b, 2009) a system for agents whose reasoning steps are given by a generalized version of modus ponens.

In his approach, the agent can have beliefs about two different entities: literals and rules. A literal $\lambda$ is an atomic proposition or its negation. A rule, denoted usually as $\rho$, has the form $\lambda_1, \ldots, \lambda_n \Rightarrow \lambda$, with $\lambda$ and all $\lambda_i$s literals. In particular $\lambda$, the rule’s conclusion, is usually denoted by $cn(\rho)$.

The important concept, that of a rule application, has the following form:

$$\lambda_1, \ldots, \lambda_n, (\lambda_1, \ldots, \lambda_n \Rightarrow \lambda)$$

In words, if the agent has the rule and all its premises, then she can apply it, obtaining the rule’s conclusion.

The language used to reason about the agent’s beliefs, called $ML$, is based on formulas of the form $B\lambda$ and $B\rho$ for $\lambda$ a literal and $\rho$ a rule, and it is closed under negation, conjunction and the existential modal operator $\Diamond$.

**Definition 2.4 (Language $ML$)** Let $P$ be a set of atomic propositions. The collection of literals $\lambda$ based on $P$ and rules $\rho$ based on such literals is called the agent’s internal language. Then, formulas $\phi$, $\vartheta$ of the $ML$ language are given by

$$\phi ::= B\lambda \mid B\rho \mid \neg \phi \mid \phi \lor \vartheta \mid \Diamond \phi$$
Formulas of the form $B\lambda$ ($B\rho$) are read as “the agent believes the literal $\lambda$ (the rule $\rho$)”. Formulas of the form $\Diamond \phi$ are read as “after a rule application, $\phi$ is true”. Similar to Duc’s $L_{BDE}$, the agent’s beliefs are restricted, this time to literals and rules based on them; also, the language can express the agent’s beliefs and how they change, but cannot express what happens in the real world.

A model for this language is again a small variation of a possible worlds model. Each world has now associated a subset of the agent’s internal language (i.e., a set of literals and rules) representing what she believes at it, and each transition between worlds represents a change in the agent’s belief state.

**Definition 2.5 ($ML$-model)** A $ML$-model for the $ML$ language is a tuple $M = \langle W, R, V \rangle$ where $W$ is a non-empty set of states, $R$ is a binary relation on $W$, and $V$ is a labelling function, assigning a subset of the agent’s internal language to each possible world.

The semantic interpretation is the usual one, with formulas of the form $B\lambda$ and $B\rho$ simply looking at the contents of the subset of the internal language associated to the evaluation point.

**Definition 2.6 (Semantic interpretation)** Let $(M, w)$ be a pointed model for the $ML$ language, with $M = \langle W, R, V \rangle$. The semantic interpretation for negation and disjunction are standard. For the rest,

- $(M, w) \vDash B\lambda$ if and only if $\lambda \in V(w)$
- $(M, w) \vDash B\rho$ if and only if $\rho \in V(w)$
- $(M, w) \vDash \Diamond \phi$ if and only if there is a $u \in W$ such that $Rwu$ and $(M, u) \vDash \phi$

The defined class of models is too general, so it needs to be restricted in order to faithfully represent rule-based reasoning. The following definitions are used to state formally the properties such models should satisfy.

**Definition 2.7 (Matching rule)** Let $w$ be a state of a $ML$-model $M = \langle W, R, V \rangle$. A rule $\rho$ of the form $\lambda_1, \ldots, \lambda_n \Rightarrow \lambda$ is $w$-matching if and only if at $w$ the agent believes the rule and all its premises but not its conclusion, that is, $\{\rho, \lambda_1, \ldots, \lambda_n\} \subseteq V(w)$ but $\lambda \notin V(w)$.

**Definition 2.8 ($\rho$-extension of a state)** Let $w$ and $u$ be states of a $ML$-model $M = \langle W, R, V \rangle$, and let $\rho$ be a rule. The state $u$ $\rho$-extends the state $w$ if and only if $V(u)$ extends $V(w)$ with $\rho$’s conclusion, that is, $V(u) = V(w) \cup \{\text{cn}(\rho)\}$.

**Definition 2.9 (Terminating state)** A state $w$ in a $ML$-model $M$ is said to be terminating if and only if no rule is $w$-matching.

With these concepts, we can now present the four requirements a $ML$-model should satisfy in order to represent rule-based reasoning.
2.2. Modal truth-preserving inference

1. For every state \(w\), if a rule \(\rho\) is \(w\)-matching, then there is a state \(u\) such that \(Rwu\) and \(u\) is a \(\rho\)-extension of \(w\).

2. For every terminating state \(w\) there is a state \(u\) such that \(Rwu\) and, moreover, \(V(w) = V(u)\).

3. For every states \(w, u\), we have \(Rwu\) only if \(w\) and \(u\) satisfy one of the previous two points, that is, either \(V(u) = V(w) \cup \{cn(\rho)\}\) for some rule \(\rho\), or else \(V(u) = V(w)\).

4. For all rules \(\rho\) and states \(w, u\), we have \(\rho \in V(w)\) if and only if \(\rho \in V(u)\).

The first requirement tells us that if the agent can apply a rule, then there should always be a state that results from the application. The second one says that if there are no applicable rules, the agent should still be able to perform reasoning steps, but they should not change her beliefs. The third one states that no other reasoning steps are allowed and the fourth one, following a standard AI practice, says that the rules the agent believes should not change: they are neither learnt nor forgotten.

The two discussed approaches have interesting proposals: the representation of inference as a modal relation (Duc’s *Dynamic Syntactic Epistemic Logic*), and the use of a generalized version of modus ponens and the formalization of the precondition and the effect of a rule application (Jago’s *Logic for Rule-Based Agents*). Within these two frameworks we can represent agents that are, indeed, non-omniscient. And not only that; the represented agents are also capable of extending their information by performing the adequate reasoning steps.

But there is still room for improvement. From an ‘inference’ perspective, we can be more precise about the agent’s reasoning steps by specifying, syntactically and semantically, which one is the action (i.e., the rule) that the agent is actually performing (i.e., applying). From a ‘meta-inference’ perspective, the rules the agent can apply do not need to be given by the same set at any time: we can dynamize once more by looking at possible ways in which the agent’s rules can change. Even from an ‘expressiveness’ perspective, we can extend the language in order to be able to express not only the agent’s information and how it evolves, but also what happens in the real world.

Still, the most important point has to do with the representation of inference. The two discussed works represent inference as a modal relation, and in order to get a proper representation, the relation should satisfy several requirements (Definition 2.2 for Duc’s case; the paragraph below Definition 2.9 for Jago’s one). All these requirements make the frameworks not as clear as we would like, and we could expect for it to be really confusing when we incorporate more actions to the picture.
There is another possibility. Actions can be represented not only as relations within the model (the Propositional Dynamic Logic style); they can also be represented as operations that change the model (the Dynamic Epistemic Logic style). Instead of defining a model that represents not only the agent’s information, but also all the possible paths the agent’s reasoning steps can follow (what Duc and Jago do), we can define a model that represents exclusively the information the agent has at a given stage, and then define operations that change this model (and therefore the agent’s information) in different ways. The advantage of the latter is that we do not need to ask for a relation to satisfy certain requirements; we just need to define reasonable operations. More importantly, representing inference as a model operation will facilitate the incorporation of our other relevant action, explicit observation.

2.3 Implicit and explicit information

Recall that our goal is to represent the way an agent’s information evolves through the use of truth-preserving inferences and observations. As mentioned before, the EL framework with possible worlds models is one of the most widely used for representing and reasoning about agents’ information. Nevertheless, in its traditional form, it is not fine enough for our purposes since, as we have discussed, agents whose information is represented with this framework are logically omniscient. Though this feature is useful in some applications, it is too much in some others and, more importantly, it hides the inference process. In fact, when representing the restaurant example with a standard possible worlds model, the answer to the second question makes the waiter know not only that your father should get the meat dish, but also that your mother should get the vegetarian one. In this case, the hidden inference is short and very simple, but in general this is not the case. Proving a theorem, for example, consists on successive applications of deductive inference steps to show that the conclusion indeed follows from the premises. Some theorems may be straightforward but, as we know, some are not. Moreover, the distinction does not correspond to immediate notions like the number of inference steps, and may be related with the ‘complexity’ of each one of them, whatever this ‘complexity’ is.

Now, truth-preserving inference over a notion of information that is already closed under logical consequence becomes irrelevant: it does not provide new information. So our goal should not be to represent inference over what the classical modal operator □ represents. In fact, this operator should not be understood as ‘full-blooded knowledge’, but as a more implicit notion, describing not the information the agent actually has, but rather the information she can eventually get. With this idea in mind, closure under logical consequence is not a problem anymore because we do expect for implicit information to have such property. What we need now is to extend EL to provide an adequate
representation for another ‘weaker’ notion in which truth-preserving inference is actually meaningful: explicit information. Only then we will be able to represent inferences and observations together in the proper way.

2.3.1 Formulas, rules and the implicit/explicit language

In our framework, the agent’s explicit information will be given by a set of formulas and rules, with these rules being the mechanism through which the agent will be able to increase this explicit information. In other words, our agent can have information not only about the way the world is (given by the formulas), but also about the actions she can perform to increase her explicit information (given by the rules).

We start by defining the language to represent the explicit information the agent can have, and by indicating what a rule in that language is.

**Definition 2.10 (Formulas and rules in \( L_P \))** Let \( P \) be a set of atomic propositions. Formulas \( \gamma, \delta \) of the propositional language \( L_P \) are given by the rule

\[
\gamma ::= p | \neg \gamma | \gamma \lor \delta
\]

with \( p \) an atomic proposition in \( P \).

A rule \( \rho \) based on the propositional language is given by

\[
\rho ::= ([\gamma_1, \ldots, \gamma_n], \delta)
\]

In words, a rule \( \rho \) is a pair, sometimes represented as \( \{\gamma_1, \ldots, \gamma_n\} \Rightarrow \delta \), where \( \{\gamma_1, \ldots, \gamma_n\} \) is a finite set of formulas and \( \delta \) is a formula, all of them in \( L_P \).

While formulas describe situations about the world, rules describe relations between such situations. Intuitively, the rule \( ([\gamma_1, \ldots, \gamma_n], \delta) \) tells us that if every \( \gamma \in \{\gamma_1, \ldots, \gamma_n\} \) is true, so is \( \delta \). We denote by \( R_{L_P} \) the set of rules based on formulas of \( L_P \), omitting the subindex when no confusion arises.

When dealing with rules, the following definitions will be useful.

**Definition 2.11 (Premises, conclusion and translation)** Let \( \rho \) be a rule of the form \( ([\gamma_1, \ldots, \gamma_n], \delta) \). We define

\[
\text{pm}(\rho) := \{\gamma_1, \ldots, \gamma_n\} \quad \text{the set of premises of } \rho
\]

\[
\text{cn}(\rho) := \delta \quad \text{the conclusion of } \rho
\]

Moreover, we define a rule’s translation, \( \text{tr}(\rho) \), as an implication in \( L_P \) whose antecedent is the (finite) conjunction of the rule’s premises and whose consequent is the rule’s conclusion:

\[
\text{tr}(\rho) := \left( \bigwedge_{\gamma \in \text{pm}(\rho)} \gamma \right) \rightarrow \text{cn}(\rho)
\]
Chapter 2. Truth-preserving inference and observation

The rules we have defined are simple implications with a special notation. We could have defined them as rules schemas (based on meta-variables to be substituted by formulas) and then their application would be a non-deterministic operation (instantiating the rule and then accepting the instantiated rule’s conclusion). But our approach is not a limitation since, as we will see, rules will be applied in a generalized modus ponens way: if all the premises have been accepted, then the conclusion can be accepted. Then we can mimic the application of (instances of) other rules, like conjunction elimination \((p \land q \Rightarrow p)\) or disjunction introduction \((p \Rightarrow p \lor q)\).

Note also that we have defined the premises of a rule as a set, and not as a more general notion like an ordered sequence or a multi-set. With such a generalized definition, it would be possible to analyze inference in ‘resource-conscious’ sub-structural logics where order and multiplicity matters, like Linear Logic (Girard 1987) or Categorial Grammar (Moortgat 1997). Nevertheless, our restricted definition is good enough for dealing with the process we are interested in, truth-preserving inference, which will be explored in Section 2.4.

Finally, we could simplify the approach by defining formulas as rules with empty premises. Nevertheless, we will stick to the formulas-and-rules setting since it emphasizes the difference between ‘factual’ information (the formulas; what the agent already has) and ‘procedural’ information (the rules; the tools to perform derivations).

The language to reason about the agent’s information extends that of EL by adding two kinds of formulas: one for expressing the agent’s explicit information \((A \gamma)\) and the other for expressing the rules she can apply \((R \rho)\).

**Definition 2.12 (Language IE)** Let \(P\) be a set of atomic propositions. Formulas \(\varphi, \psi\) of the implicit/explicit language \(IE\) are given by

\[
\varphi ::= p \mid A \gamma \mid R \rho \mid \neg \varphi \mid \varphi \lor \psi \mid \Box \varphi
\]

with \(p \in P\), \(\gamma \in L_P\) and \(\rho \in R\). Formulas of the form \(A \gamma\), access formulas, are read as “the agent is explicitly informed about \(\gamma\)”, and formulas of the form \(R \rho\), rule formulas, are read as “the agent can apply rule \(\rho\)”. The universal modal operator, \(\Box\), is now interpreted as implicitly information, with formulas of the form \(\Box \varphi\) being read as “the agent is implicitly informed about \(\varphi\)”. Other boolean connectives (\(\land, \rightarrow\) and \(\leftrightarrow\)), logical constants (\(\top\) and \(\bot\)) as well as the existential modal operator \(\Diamond\) are defined as usual.

Our agent can have explicit information about facts, but not about her own (or, eventually, other agents’) information. This is indeed a limitation, but it allows us to define one of the two processes we are interested in: observation

\[\text{\textsuperscript{1}}\]In fact, sets are good enough even for dealing some forms of non-monotonic reasoning, as shown in Chapters 5 and 6.
2.3. Implicit and explicit information

In Section 2.5 we discuss the reasons for this limitation, leaving a deeper analysis and further proposals for the next chapters.

The semantic model extends a possible worlds model by assigning two new sets to each possible world: one indicating the formulas the agent is explicitly informed about, and other indicating the rules she can apply. We still have just one relation between worlds, the accessibility relation, indicating which the worlds the agent considers possible from a given one.

Definition 2.13 (Implicit/explicit model) Let \( \mathcal{P} \) be a set of atomic propositions. An implicit/explicit model is a tuple \( M = \langle W, R, V, A, R \rangle \) where \( \langle W, R, V \rangle \) is a possible worlds model over \( \mathcal{P} \) and

- \( A : W \rightarrow \wp(\mathcal{L}_p) \) is the access set function, indicating the agent’s explicit information at each possible world. The set \( A(w) \) will be called the agent’s access set at \( w \), and should be preserved by the accessibility relation: if \( \gamma \in A(w) \) and \( Rwu \), then \( \gamma \in A(u) \) (the coherence property for formulas);

- \( R : W \rightarrow \wp(\mathcal{R}) \) is the rule set function, indicating the rules the agent can apply at each possible world. The set \( R(w) \) will be called the agent’s rule set at \( w \), and should be also preserved by the accessibility relation: if \( \rho \in R(w) \) and \( Rwu \), then \( \rho \in R(u) \) (the coherence property for rules).

We denote by \( \text{IE} \) the class of implicit/explicit models. Note again how, just as in the definition of the premises of a rule, the agent’s explicit information about formulas and rules is given by a set.

The semantic interpretation of formulas in \( \text{IE} \) has an immediate definition.

Definition 2.14 (Semantic interpretation) Let \( (M, w) \) be a pointed \( \text{IE} \)-model with \( M = \langle W, R, V, A, R \rangle \). The semantic interpretation for negations and disjunctions is given as usual. The case of atomic propositions \( p \) and implicit information formulas \( \Box \varphi \) is just like in Epistemic Logic. For access and rule formulas, we just look at the corresponding sets:

\[
(M, w) \vDash A \gamma \quad \text{iff} \quad \gamma \in A(w)
\]
\[
(M, w) \vDash R \rho \quad \text{iff} \quad \rho \in R(w)
\]
Example 2.1 In the leftmost world $w_1$ of following model, the agent has (1) implicit and explicit information about $p$, (2) implicit but not explicit information about $q$ and (3) neither implicit nor explicit information about $r$, as indicated by the formulas on the right. Access sets are drawn below the corresponding world, and rule sets are not indicated.

\[
\begin{array}{c}
\xrightarrow{p,q,r} w_1 \\
\{p\}
\end{array} \quad \begin{array}{c}
\xrightarrow{p,q} w_2 \\
\{\} \\
\end{array}
\]

\begin{enumerate}
\item $(M, w_1) \models \Box p \land \neg A p$
\item $(M, w_1) \models \Box q \land \neg A q$
\item $(M, w_1) \models \neg \Box r \land \neg A r$
\end{enumerate}

Axiom system So which properties does the agent’s information get under this representation? A standard approach to find them is to look for those formulas that are \textit{valid} in the given class of models, that is, formulas that are true at every world of every model in the given class. There are several ways to look for such formulas, and one of the most commonly used is to look for their syntactic characterization, that is, a \textit{derivation} or \textit{axiom system}. Such system is a sort of calculus that gives us basic formulas and then operations to derive more formulas. An axiom system is interesting when it only derives formulas valid in a given class of models (a \textit{sound} axiom system), and it becomes even more interesting when, additionally, it derives every valid formula of the given class (a \textit{complete} axiom system).

In order to provide a sound and complete axiom system for formulas in $IE$ with respect to $IE$-models, it is helpful to look at that for the underlying system: Epistemic Logic (Subsection 1.1.1). The well-known axioms and rules of Table 2.1 provide us with a sound and strongly complete axiom system for the $EL$ language with respect to possible worlds models.

\begin{table}[h]
\centering
\begin{tabular}{ll}
\hline
Prop & $\vdash \varphi$ for $\varphi$ a propositional tautology \\
\hline
$K$ & $\vdash \Box (\varphi \rightarrow \psi) \rightarrow (\Box \varphi \rightarrow \Box \psi)$ \\
Dual & $\vdash \Diamond \varphi \leftrightarrow \neg \Box \neg \varphi$ \\
\hline
\end{tabular}
\end{table}

Table 2.1: Axiom system for $EL$ w.r.t. possible worlds models.

Now we can provide our particular axiom system.
2.3. Implicit and explicit information

**Theorem 2.1 (Axiom system for $\mathcal{IE}$ w.r.t. $\mathcal{IE}$)** The axioms and rules of Tables 2.1 and 2.2 form a sound and strongly complete axiom system for formulas in $\mathcal{IE}$ with respect to $\mathcal{IE}$-models. While axioms and rules of Table 2.1 provide us with validities of the language in possible worlds models, axioms $\text{Coh}_L$ and $\text{Coh}_R$ describe the particular requirements of access and rule formulas for $\mathcal{IE}$-models: the coherence property for formulas and rules, respectively. This axiom system is denoted by $\mathcal{IE}$.

\[
\begin{align*}
\text{Coh}_L & \vdash A \gamma \rightarrow \square A \gamma \\
\text{Coh}_R & \vdash R \rho \rightarrow \square R \rho
\end{align*}
\]

Table 2.2: Axioms for the coherence properties.

**Proof.** Soundness follows from axioms being valid and rules being validity-preserving. Completeness is proved by a standard modal canonical model construction with an adequate definition of the access and rule set functions. In order to obtain the crucial coherence properties, we use the $\text{Coh}_L$ and $\text{Coh}_R$ axioms. See Appendix A.1 for details. □

Observe how axioms of Table 2.1 say that the agent’s implicit information is omniscient: it contains all validities (rule $\text{Nec}$) and it is closed under logical consequence (axiom $K$). Nevertheless, the agent’s explicit information does not have these properties: the validity of $\gamma$ does not imply the validity of $A \gamma$, and $A (\gamma \rightarrow \delta) \rightarrow (A \gamma \rightarrow A \delta)$ is not valid.

### 2.3.2 The case of true information

The way it is defined, an $\mathcal{IE}$-model allows us to represent an agent whose information is not necessarily true. We do not ask for any property for the accessibility relation, so there are no constraints for implicit information, others than those given by the representation itself, like closure under modus ponens (the $K$ axiom) and the inclusion of validities (the $\text{Gen}$ rule). In particular, the actual world does not need to be among the ones the agent considers possible, so the agent can be implicitly informed about certain $\varphi$ without it being true. Moreover, formulas in access sets do not need to be true and rules in rule sets do not need to be truth-preserving at the corresponding world; therefore the agent can be explicitly informed about a formula $\gamma$ or a rule $\rho$ without they being true and truth-preserving, respectively.

By asking for the adequate model properties, we can represent different notions of information. Here we will focus on the case of *true* information, that is, knowledge.

---

2 For literature about information that can be true or false, we refer to Dretske (1981) and Floridi (2005).
Among models in \( \text{IE} \), we distinguish those where implicit and explicit information are true and the rules are truth-preserving. For implicit information, we consider equivalence accessibility relations, as it is usually done in \( \text{EL} \). For explicit information, we ask for every formula in an access set to be true in the corresponding world. Finally, for the case of rules, we ask for its translation to be true in the corresponding world.

**Definition 2.15 (The class \( \text{IE}_K \))** We denote by \( \text{IE}_K \) the class of models \( M = \langle W, R, V, A, \mathcal{R} \rangle \) in \( \text{IE} \) satisfying the following properties.

- **Equivalence**: \( R \) is an equivalence relation.
- **Truth for formulas**: for every world \( w \), if \( \gamma \in A(w) \), then \( (M, w) \vDash \gamma \).
- **Truth for rules**: for every world \( w \), if \( \rho \in \mathcal{R}(w) \), then \( (M, w) \vDash \text{tr}(\rho) \).

Recall the coherence property for formulas and rules: if \( \gamma \in A(w) \) (\( \rho \in \mathcal{R}(w) \)) and \( Rwu \), then we have \( \gamma \in A(u) \) (\( \rho \in \mathcal{R}(u) \)). Note how, when the accessibility relation is an equivalence relation, we get the same information and rule set for all the worlds that belong to the same equivalence class.

In the rest of this chapter, we will use the term “information” for the general class of \( \text{IE} \)-models, and the term “knowledge” for the class of models with true information, that is, \( \text{IE}_K \)-models.

**Axiom system** In order to provide a sound and complete axiom system with respect to the just defined class of \( \text{IE}_K \)-models, we just need to provide axioms that characterize the properties of models in the class.

**Theorem 2.2 (Axiom system for \( \text{IE} \) w.r.t. \( \text{IE}_K \))** The axioms of Tables 2.1, 2.2 and 2.3 form a sound and strongly complete axiom system for formulas in \( \text{IE} \) with respect to \( \text{IE}_K \)-models. In particular, axioms of Table 2.3 characterize the properties that distinguish models in \( \text{IE}_K \) from models in \( \text{IE} \): equivalence and truth for formulas and rules. This axiom system is denoted by \( \text{IE}_K \).

<table>
<thead>
<tr>
<th>Table 2.3: Extra axioms for ( \text{IE}_K )-models</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 ( \vdash \Box \varphi \rightarrow \varphi )</td>
</tr>
<tr>
<td>( T \vdash \Box \varphi \rightarrow \varphi )</td>
</tr>
<tr>
<td>( T_{\text{th}_L} \vdash A \varphi \rightarrow \gamma )</td>
</tr>
</tbody>
</table>

\( ^3 \)Given our understanding of knowledge as true information, we actually just need for the relation to be reflexive. Nevertheless, we will stick to the standard approach of using equivalence relations that give the agent (implicit) positive and negative introspection.
2.4. Inference

Proof. Soundness is again simple. Completeness is proved by showing that the canonical model for $\text{IE}_K$ satisfies equivalence (from axioms T, 4, 5), truth for formulas (from axiom Tth$_{L_L}$) and truth for rules (from axiom Tth$_R$). Details can be found in Appendix A.2. ■

The new axioms provide new properties. Axioms T, 4 and 5 tell us that implicit knowledge is true ($T$) and has the positive and negative introspection properties (4 and 5). Axioms Tth$_{L_L}$ and Tth$_R$ indicate that the agent’s explicit knowledge about formulas and rules is also true. Note that from the coherence and the truth axioms, Coh$_{L_L}$ and Tth$_{L_L}$, we get the following validity, expressing that the agent’s explicit knowledge is also implicit knowledge.

\[ \forall \gamma \rightarrow \square \gamma \]

It is now time to turn our attention to the dynamics of implicit and explicit knowledge. In the following sections we will define our intended informational actions: truth-preserving inference and observation.

2.4 Inference

The agent can extend her explicit information by applying the rules she has available. Intuitively, a rule $(\Gamma, \delta)$ indicates that if every $\gamma \in \Gamma$ is true, so is $\delta$. However, so far, we have not indicated or restricted the way the agent can use a rule. She can use it to get the conclusion without having all the premises, or even deriving the premises whenever she has the conclusion. In the previous section we focused on models for true information; in the same spirit, this section will deal with truth-preserving inference.

2.4.1 Truth-preserving inference

The inference process aims at extending the agent’s explicit information by means of a rule-application. Technically, this boils down to adding formulas to the agent’s access sets. In order to represent truth-preserving inference, we will restrict the way in which the rule can be applied.

**Definition 2.16 (Deduction operation)** Let $M = \langle W, R, V, A, R \rangle$ be an IE-model, and let $\sigma$ be a rule in $\mathcal{R}$. The model $M_{\sigma} = \langle W, R, V, A', R \rangle$ differs from $M$ just in the access set function, which is given for every $w \in W$ by

\[ A'(w) := \begin{cases} A(w) \cup \{ \text{cn}(\sigma) \} & \text{if } \text{pm}(\sigma) \subseteq A(w) \text{ and } \sigma \in R(w) \\ A(w) & \text{otherwise} \end{cases} \]

The operation $(\cdot)_{\sigma}$ is called the deduction operation with rule $\sigma$. ■
Chapter 2. Truth-preserving inference and observation

The conclusion of the rule will be added to a world only when all the premises and the rule are already present. In other words, after the inference the agent’s explicit information will be extended with the rule’s conclusion only if she already has the rule and all its premises.

Note how the deduction operation preserves models in $\text{IE}_K$.

**Proposition 2.1** Let $\sigma$ be a rule. If $M$ is an $\text{IE}_K$-model, so is $M \hookrightarrow \rightarrow \sigma$.

**Proof.** Equivalence and both properties of rules are immediate since neither the accessibility relation nor the rule set function are modified. The properties of formulas can be verified easily; details can be found in Appendix A.3. ■

The language $\text{IE}_D$ extends $\text{IE}$ by closing it under existential deduction modalities $\langle \hookrightarrow \rightarrow \sigma \rangle$ for $\sigma$ a rule: if $\phi$ is a formula in $\text{IE}_D$, so is $\langle \hookrightarrow \rightarrow \sigma \rangle \phi$. These new formulas are read as “there is a deductive inference with $\sigma$ after which $\phi$ is the case” and their universal duals, defined as usual,

$$[\leftrightarrow \sigma] \phi := \neg(\leftrightarrow \sigma) \neg \phi$$

are read as “after any deductive inference with $\sigma$, $\phi$ is the case”.

For the semantic interpretation, note that the agent cannot preform a truth-preserving inference with $\sigma$ in any situation. In order to do it, she should know explicitly the rule and its premises. The abbreviation

$$\text{Pre}_{\leftrightarrow \sigma} := \bigwedge_{\gamma \in \text{pm}(\sigma)} (A \gamma) \land R \sigma$$

indicates precisely this requirement.

**Definition 2.17 (Semantic interpretation)** Let $(M, w)$ be a pointed $\text{IE}$-model.

$$(M, w) \vDash [\leftrightarrow \sigma] \phi \iff (M, w) \vDash \text{Pre}_{\leftrightarrow \sigma} \text{ and } (M \hookrightarrow \rightarrow \sigma, w) \vDash \phi$$

By unfolding the definition of the universal deduction modality, we get

$$(M, w) \vDash [\leftrightarrow \sigma] \phi \iff (M, w) \vDash \text{Pre}_{\leftrightarrow \sigma} \text{ implies } (M \hookrightarrow \rightarrow \sigma, w) \vDash \phi$$

The semantic interpretation of deduction modalities simply reflects our intuition about a rule application: the agent can perform a deductive inference with $\sigma$ after which $\phi$ is the case, $(M, w) \vDash \langle \leftrightarrow \sigma \rangle \phi$, if and only if she knows explicitly the rule and its premises, $(M, w) \vDash \text{Pre}_{\leftrightarrow \sigma}$, and, after the inference, $\phi$ is the case, $(M \hookrightarrow \rightarrow \sigma, w) \vDash \phi$. Note how the new pointed model $(M \hookrightarrow \rightarrow \sigma, w)$ is defined even if the precondition of the operation $\text{Pre}_{\leftrightarrow \sigma}$ does not hold at the original $(M, w)$. This is different from the observation operation (Definition 1.5) that makes $(M, \chi!w)$ undefined if the precondition $\chi$ fails at $(M, w)$. Nevertheless, the behaviour of the modalities is the same in the two operations: the existential
2.4. Inference

ones, $\langle \leftarrow_{e} \rangle \varphi$ and $\langle \chi! \rangle \varphi$, are true if and only if the respective actions can be executed and the (unique) resulting pointed model satisfies $\varphi$, and the universal ones, $\left[ \leftarrow_{u} \right] \varphi$ and $\left[ \chi! \right] \varphi$, are true if and only if either the action can be executed with the specified results, or else the action cannot be executed at all.

Now we can see that in IEK-models, the deduction operation behaves as expected: if the agent can perform a truth-preserving inference with $\sigma$, then after doing it she will know explicitly $\sigma$’s conclusion. This is because if she can perform the inference at $w$ in $M$, then the precondition tells us that she knows explicitly the rule and its premises. Because of the truth properties, the premises are true and the rule is truth-preserving at $w$ in $M$, so the conclusion is also true at $w$ in $M$. But since the conclusion is a propositional formula, its truth-value depends only on the atomic valuation of $w$; since $w$’s atomic valuation in the new model is exactly the same as in the original one, the conclusion will also be true at $w$ in $M_{\leftarrow_{e}}$. Moreover, this conclusion will be in the access set of $w$ in $M_{\leftarrow_{e}}$; therefore, the agent will know explicitly the rule’s conclusion. The following validity express this:

$\langle \leftarrow_{e} \rangle A \text{cn}(\sigma)$

Note also how, in IEK-models, if the agent can apply a $\sigma$-inference, then $\sigma$’s conclusion is already in the agent’s implicit information. This is because if $\sigma$ is applicable at $w$ in $M$, then the agent knows the rule and all its premises, that is, $A \text{pm}(\sigma) \land R \sigma$ holds at $w$. Then, the coherence properties put $\sigma$ and its premises in all worlds $R$-reachable from $w$. But then, because of the truth properties, $\sigma$’s conclusion holds in all of them. Hence, $\Box \text{cn}(\sigma)$ is true in the current world. The following validity express this:

$\langle \leftarrow_{e} \rangle \top \rightarrow \Box \text{cn}(\sigma)$

In other words, an act of deduction extends the agent’s explicit knowledge by making explicit what was already implicit.

**Axiom system** In order to provide a sound and complete axiom system for formulas in IE plus deduction modalities, we will review the sound and complete axiom system for Observation Logic (Definition 4).

Recall that the idea of a sound and complete axiom system is to characterize validities of a language with respect to a given class of models. We already have a characterization of the validities of the ‘static’ epistemic language in possible worlds models (Table 2.1); what we need now are axioms and rules describing the relevant properties of the observation modalities.

First, note that the observation operation preserves possible worlds models, that is, if $M$ is a possible worlds model, so is $M_{\chi!}$. This is important because any formula valid in $M$ will still be valid after the operation. Then we have the following rule:

$\chi! N \quad \text{From } \vdash \varphi, \text{ infer } \vdash \left[ \leftarrow_{e} \right] \varphi$
Now, note that the observation modality has the behavior described in the validities of Table 2.4.

\[
\begin{align*}
!p & \vdash \langle \chi! \rangle p \iff (\chi \land p) \\
!\neg & \vdash \langle \chi! \rangle \neg \varphi \iff (\chi \land \neg \langle \chi! \rangle \varphi) \\
!\lor & \vdash \langle \chi! \rangle (\varphi \lor \psi) \iff (\langle \chi! \rangle \varphi \lor \langle \chi! \rangle \psi) \\
!\Box & \vdash \langle \chi! \rangle \Box \varphi \iff (\chi \land \Box [\langle \chi! \rangle \varphi])
\end{align*}
\]

Table 2.4: Validities for observation modality.

Let us read some of them. The first indicates that the agent can observe \( \chi \) and after doing it \( p \) will be true if and only if both \( \chi \) and \( p \) are true before the observation. More interestingly, the last one tells us that the agent can observe \( \chi \) and after doing it she will be (implicitly) informed about \( \varphi \) if and only if \( \chi \) is true and the agent is (implicitly) informed that after any observation of \( \chi \), \( \varphi \) will be true.

These validities give us more than just properties of the observation operation: they provide a way of translating any formula with observation modalities into a semantically equivalent one without them. For example, consider the formula \( \langle (p \land q)! \rangle \Box p \), expressing that \( p \land q \) can be observed and, after doing it, the agent will be informed about \( p \). By a repeated application of the validities, we can eliminate the observation modalities in the following way:

\[
\langle (p \land q)! \rangle \Box p \iff (p \land q) \land \Box [(p \land q)!!] p \\
\iff (p \land q) \land \Box ((p \land q) \rightarrow p)
\]

So \( \langle (p \land q)! \rangle \Box p \) is semantically equivalent to \( (p \land q) \land \Box ((p \land q) \rightarrow p) \). Even if we have a formula with nested occurrences of observation modalities, we can eliminate all of them by following a ‘deepest-first’ order. For example, eliminating the deepest observation modality from \( \langle (p \lor q)! \rangle (\neg p! \Box q \) gives us \( \langle (p \lor q)! \rangle (\neg p \land \Box (\neg p \rightarrow q)) \); then we can eliminate the remaining one. Note how the existence of these validities imply that the ‘static’ language, the one without observation modalities, is actually expressive enough to encode the way the model will change after the operation.

How are these validities useful when looking for an axiom system? By stating them as axioms, reduction axioms, a formula with observation modalities and its translation are not just semantically but also provably equivalent. Then, completeness of the language with the observation modality follows from the completeness of the basic ‘static’ system, since each formula with these modalities can be effectively translated into a provably equivalent one without them. For a more detailed explanation of this technique, we refer to Section 7.4 of van Ditmarsch et al. (2007).
2.4. Inference

Now we can provide an axiom system for $I\mathcal{E}_D$ with respect to $\mathbb{IE}_K$-models. By Proposition 2.1, this class is closed under the deduction operation, so we can rely on the axiom system $\mathbb{IE}_K$. We provide reduction axioms, expressing how deduction operation affect the truth-value of formulas of the language.

**Theorem 2.3 (Reduction axioms for the deduction modality)** Table 2.5 provides reduction axioms for the deduction modality. Together with $\mathbb{IE}_K$ (Theorem 2.2), they form a sound and complete axiom system for language $I\mathcal{E}_D$ with respect to $\mathbb{IE}_K$-models.

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Leftarrow p, \frac{\langle \rightarrow_o \rangle \varphi}{\langle \rightarrow_o \rangle \varphi \land \langle \rightarrow_o \rangle \neg p}$</td>
<td>by $\Leftarrow_v$</td>
</tr>
<tr>
<td>$\Leftarrow p, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by $\Leftarrow_p, \Leftarrow_\neg$, and Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>def. of $\neg$</td>
</tr>
<tr>
<td>$\Leftarrow \langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by $\neg \varphi$, $\Leftarrow_\neg$, and Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>def. of $\neg$</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by $\neg \varphi$, $\Leftarrow_\neg$, and Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>def. of $\neg$</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by $\neg \varphi$, $\Leftarrow_\neg$, and Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>def. of $\neg$</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by $\neg \varphi$, $\Leftarrow_\neg$, and Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>def. of $\neg$</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by $\neg \varphi$, $\Leftarrow_\neg$, and Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>def. of $\neg$</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by $\neg \varphi$, $\Leftarrow_\neg$, and Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>def. of $\neg$</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by $\neg \varphi$, $\Leftarrow_\neg$, and Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by Prop. logic</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>def. of $\neg$</td>
</tr>
<tr>
<td>$\Leftarrow \neg \varphi, \frac{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}{\langle \rightarrow_o \rangle \varphi \lor \langle \rightarrow_o \rangle \neg p}$</td>
<td>by $\neg \varphi$, $\Leftarrow_\neg$, and Prop. logic</td>
</tr>
</tbody>
</table>

Table 2.5: Axioms and rule for the deduction modality.

**Proof.** Soundness follows from the validity of the new axioms and the validity-preserving property of the new rule, just as before. Strong completeness follows from the fact that, by a repeated application of the reduction axioms, any deduction operation formula can be reduced to a formula in $I\mathcal{E}_D$ for which $\mathbb{IE}_K$ is strongly complete with respect to $\mathbb{IE}_K$.

The interesting reduction axioms, indicating how access and rule sets are affected by deduction, appear on the right column of Table 2.5. Axioms $\Leftarrow_A$ indicate that $\text{cn} (\sigma)$ is the unique formula added to access sets, and axiom $\Leftarrow_R$ indicates that rule sets are not modified.

As an example of what can be derived with the axiom system, consider the formula $\langle \rightarrow_o \rangle \mathcal{T} \rightarrow \Box \text{cn} (\sigma)$. Its validity was already justified by a semantic argument, but it can also be justified syntactically.\footnote{Through the whole text, this and other syntactic derivations make a slight abuse of notation.}
Comparison with previous works

It is illustrative to make a brief comparison between our proposal and the approaches for inference described in Section 2.2.

Dynamic Syntactic Epistemic Logic From Duc’s work we have inherited the syntactic representation of the agent’s explicit information and the spirit of inference as rule-application. There are small syntactic differences: (1) while Duc uses a single modality $\langle F \rangle$, standing for “a course of thought”, our language is more precise about what this ‘course of thought’ is by explicitly stating the applied rule; (2) Duc’s language does not allow us to talk about the real world, something our language can do. There is also the fact that while Duc’s framework just focuses on one notion, explicit information, our framework represents explicit and also implicit information.

But the most important difference is semantic. While Duc represents inference as a relation between worlds, we represent it as an operation over the model. Consequences of this will be discussed below, but first we will look at how our work relates to the other reviewed approach.

Logic for Rule-Based Agents From Jago’s work we have inherited the formal definition of the requirements and the consequences of a rule application. There are small language-related differences, like the ones with respect to Duc’s approach plus the fact that Jago’s system limits the agent’s information to literals and rules built from them. Also, his framework represents only one notion of information.

But again, the main difference is the representation of inference as a modal relation, different from our model operation approach.

Modal relation vs. model operation Following the two described approaches, inference was represented in our earlier proposals [Velázquez-Quesada 2008a] as a modal relation between worlds with a relation $R_{\sigma}$ for each inference rule $\sigma$ the agent can apply. But then, consider the following natural requirements for truth-preserving inference.

1. Inference steps should not modify the ontic (factual) situation.
2. In order to apply a rule, the agent needs the premises and the rule.
3. The application of a rule should preserve explicit factual information the agent had before.
4. Explicit information should be increased by the conclusion of the rule.
5. There should be no other difference between explicit information before and after the rule application.
2.4. Inference

If we represent inference as a modal relation, several restrictions are required in the semantic model in order to satisfy these requirements, making the treatment somehow confusing. But if we represent inference as a modal operation, we do not need to ask for these properties anymore: they are a consequence of the representation. The deduction operation preserves world-valuation, so the ontic situation is not affected. But not only that: we get automatically the four remaining properties, as the validity of the following formulas shows.

2. \((\leftrightarrow_\sigma) \top \rightarrow \text{Pre}_{\leftrightarrow_\sigma}\)
3. \(A \gamma \rightarrow [\leftrightarrow_\sigma] A \gamma\)
4. \([\leftrightarrow_\sigma] A \text{cn}(\sigma)\)
5. \((\leftrightarrow_\sigma) A \gamma \rightarrow A \gamma\) for \(\gamma \neq \text{cn}(\sigma)\)

There is another important consequence. In our representation, inference is functional: the agent can perform an inference step with \(\sigma\) every time she has the rule and all its premises. With a relational representation of inference, this property has to be explicitly required (as Jago does) and, more importantly, is not preserved by model operations: adding formulas to the set can make applicable a rule that was not applicable before. This is relevant for us because the second informational action we want to deal with, observation, is semantically represented by a model operation.

2.4.2 Dynamics of truth-preserving inference

Just as the agent’s explicit knowledge changes, her inferential abilities can also change. This may be because she gets to know a new rule (by means of an observation; Section 2.5), but it may be also because she builds new rules from the ones she already has. For example, from the rules \(\{p\} \Rightarrow q\) and \(\{q\} \Rightarrow r\), it is possible to derive the rule \(\{p\} \Rightarrow r\). It takes one step to derive the new rule, but it will save intermediate steps in future inferences.

In fact this situation, a form of transitivity, represents the application of cut over the mentioned rules. In general, inference relations can be characterized by structural rules, indicating how to derive new rules from the ones already present. In the case of deduction, we have the structural rules of Table 2.6.

In our setting, each application of a structural rule produces a rule that can be added to the rule set. Note that neither contraction nor permutation yield a new rule, since the premises of our rules are given by a set. On the other hand, reflexivity, monotonicity and cut can produce rules that were not present before.

**Definition 2.18 (Structural operations)** Let \(M = \langle W, R, V, A, R \rangle\) be an IE-model. The **structural operations** defined below return a model that differs from \(M\) just in the rule set function, which in each case is defined in the following way.

---

This is not to say that order or multiplicity of inference steps are irrelevant; given our dynamic approach, they definitely matter, as changes in order or number of inference steps can yield different results. We just mean that order and multiplicity of the premises are irrelevant because we represent them as a set, and therefore the two mentioned operations will only generate rules that were already considered.
Chapter 2. Truth-preserving inference and observation

<table>
<thead>
<tr>
<th>Reflexivity:</th>
<th>Contraction:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi \Rightarrow \varphi$</td>
<td>$\psi, \chi, \xi, \phi \Rightarrow \varphi$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Permutation:</th>
<th>Monotonicity:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi, \chi, \xi, \phi \Rightarrow \varphi$</td>
<td>$\psi, \phi \Rightarrow \varphi$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cut:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi \Rightarrow \xi$</td>
</tr>
</tbody>
</table>

Table 2.6: Structural rules for deduction.

**Reflexivity** Let $\delta$ be a formula in $L_P$ and consider the rule

$$\zeta_{\delta} := ([\delta], \delta)$$

The rule set function $R'$ of the model $M_{Ref\delta}$ is given, for every $w \in W$, by

$$R'(w) := R(w) \cup \{ \zeta_{\delta} \}$$

The operation $(\cdot)_{Ref\delta}$ is called the *reflexivity operation* with formula $\delta$.

**Monotonicity** Let $\delta$ be a formula in $L_P$ and $\zeta$ a rule over it. Consider the rule

$$\zeta' := (\text{pm}(\zeta) \cup \{ \delta \}, \text{cn}(\zeta))$$

extending $\zeta$ by adding $\delta$ to its premises. The rule set function $R'$ of the model $M_{Mon\zeta}$ is given, for every $w \in W$, by

$$R'(w) := \begin{cases} R(w) \cup \{ \zeta' \} & \text{if } \zeta \in R(w) \\ R(w) & \text{otherwise} \end{cases}$$

The operation $(\cdot)_{Mon\zeta}$ is the *monotonicity operation* with formula $\delta$ and rule $\zeta$.

**Cut** Let $\zeta_1, \zeta_2$ be rules over $L_P$ such that the conclusion of $\zeta_1$ appears in the premises of $\zeta_2$. Consider the rule

$$\zeta' := (\text{pm}(\zeta_2) \setminus \{ \text{cn}(\zeta_1) \}) \cup \{ \text{pm}(\zeta_1), \text{cn}(\zeta_2) \}$$

combining $\zeta_1$ and $\zeta_2$. The rule set function $R'$ of the model $M_{Cut_{\zeta_1,\zeta_2}}$ is given, for every $w \in W$, by

$$R'(w) := \begin{cases} R(w) \cup \{ \zeta' \} & \text{if } \{ \zeta_1, \zeta_2 \} \subseteq R(w) \\ R(w) & \text{otherwise} \end{cases}$$

The operation $(\cdot)_{Cut_{\zeta_1,\zeta_2}}$ is called the *cut operation* with rules $\zeta_1$ and $\zeta_2$.  ▼
2.4. Inference

Just like the deduction operation, the three structural operations preserve models in the class \( \text{IE}_K \).

**Proposition 2.2** If \( M \) is an \( \text{IE}_K \)-model, then so are \( M_{\text{Ref}_0} \), \( M_{\text{Mon}_{\delta,c}} \) and \( M_{\text{Cut}_{c_1,c_2}} \).

**Proof.** Coherence and truth for formulas as well as equivalence are immediate, since neither access sets nor accessibility relations are modified. For coherence and truth for rules, see Appendix A.4. \( \blacksquare \)

The language \( \mathcal{IE}_D^S \) extends \( \mathcal{IE}_D \) by closing it under existential modalities for structural operations: if \( \varphi \) is in \( \mathcal{IE}_D^S \), so are \( \langle \text{Ref}_0 \rangle \varphi \), \( \langle \text{Mon}_{\delta,c} \rangle \varphi \) and \( \langle \text{Cut}_{c_1,c_2} \rangle \varphi \). The formulas are read as “there is a way of applying the structural operation after which \( \varphi \) is the case”. In order to formally define their semantic interpretation, we define the following formulas, stating the precondition of each operation. For uniformity, we define a precondition for reflexivity; since this operation can be defined in any situation, we define it simply as \( \top \).

\[
\begin{align*}
\text{Pre}_{\text{Ref}_0} & := \top \\
\text{Pre}_{\text{Mon}_{\delta,c}} & := R \, \zeta \\
\text{Pre}_{\text{Cut}_{c_1,c_2}} & := R \, \zeta_1 \land R \, \zeta_2 \land \left( (\forall \gamma \in \text{pm}(c_2)) A \, \gamma \rightarrow A \, \text{cn}(\zeta_1) \right)
\end{align*}
\]

**Definition 2.19 (Semantic interpretation)** Let \( (M, w) \) be a pointed \( \text{IE} \)-model:

\[
\begin{align*}
(M, w) \Vdash \langle \text{Ref}_0 \rangle \varphi & \quad \text{iff} \quad (M, w) \not\Vdash \text{Pre}_{\text{Ref}_0} \land (M_{\text{Ref}_0}, w) \Vdash \varphi \\
(M, w) \Vdash \langle \text{Mon}_{\delta,c} \rangle \varphi & \quad \text{iff} \quad (M, w) \not\Vdash \text{Pre}_{\text{Mon}_{\delta,c}} \land (M_{\text{Mon}_{\delta,c}}, w) \Vdash \varphi \\
(M, w) \Vdash \langle \text{Cut}_{c_1,c_2} \rangle \varphi & \quad \text{iff} \quad (M, w) \not\Vdash \text{Pre}_{\text{Cut}_{c_1,c_2}} \land (M_{\text{Cut}_{c_1,c_2}}, w) \Vdash \varphi
\end{align*}
\]

Just as before, the universal modalities of the structural operations are defined as the dual of their corresponding existential versions. Just as before, the unfolding yields the following semantic interpretation

\[
\begin{align*}
(M, w) \Vdash [\text{Ref}_0] \varphi & \quad \text{iff} \quad (M, w) \not\Vdash \text{Pre}_{\text{Ref}_0} \land (M_{\text{Ref}_0}, w) \Vdash \varphi \\
(M, w) \Vdash [\text{Mon}_{\delta,c}] \varphi & \quad \text{iff} \quad (M, w) \not\Vdash \text{Pre}_{\text{Mon}_{\delta,c}} \land (M_{\text{Mon}_{\delta,c}}, w) \Vdash \varphi \\
(M, w) \Vdash [\text{Cut}_{c_1,c_2}] \varphi & \quad \text{iff} \quad (M, w) \not\Vdash \text{Pre}_{\text{Cut}_{c_1,c_2}} \land (M_{\text{Cut}_{c_1,c_2}}, w) \Vdash \varphi
\end{align*}
\]

**Axiom system** In order to provide an axiom system for the new formulas, Proposition 2.2 allows us to rely on the axiom system \( \text{IE}_K \) once again. Table 2.7 provides axioms indicating how the truth value of formulas after the structural operations depends on the truth value of formulas before them.

**Theorem 2.4 (Reduction axioms for structural modalities)** Let \( \text{STR} \) stand for either \( \text{Ref}_0 \), \( \text{Mon}_{\delta,c} \) or \( \text{Cut}_{c_1,c_2} \), and let \( \zeta' \) stand for the corresponding new rule in each case. Table 2.7 provides reduction axioms for the structural modalities. Together with \( \text{IE}_K \) (Theorem 2.3), they form a sound and complete axiom system for language \( \mathcal{IE}_D^S \) with respect to \( \text{IE}_K \)-models.
Chapter 2. Truth-preserving inference and observation

\[ \text{STR} \vdash (\text{Pre}_{\text{STR}} \land p) \leftrightarrow (\text{Pre}_{\text{STR}} \land A \gamma) \]

\[ \text{STR} \vdash (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \leftrightarrow (\text{Pre}_{\text{STR}} \land A \gamma) \]

\[ \text{STR} \vdash (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \leftrightarrow \text{Pre}_{\text{STR}} \]

\[ \text{STR} \vdash (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \leftrightarrow (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \]

\[ \text{STR} \vdash (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \leftrightarrow (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \]

\[ \text{STR} \vdash (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \leftrightarrow (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \]

\[ \text{STR} \vdash (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \leftrightarrow (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \]

\[ \text{STR} \vdash (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \leftrightarrow (\text{Pre}_{\text{STR}} \land \neg(\text{STR}) \varphi) \]

\[ \text{Table 2.7: Axioms and rules for the reflexivity, monotonicity and cut modalities.} \]

Proof. Just like the reduction axioms for the deduction modality, soundness follows from the validity of the new axioms and the validity-preserving property of the new rules. Strong completeness follows from the fact that, by a repeated application of such axioms, any structural operation formula can be reduced to a formula in \( \text{IE}_D \), for which we already have a sound and strongly complete axiom system with respect to \( \text{IE}_K \)-models. ■

The three structural operations have similar reduction axioms. The difference between them is the precondition for each one to take place, and the new rule each one introduces. While the reflexivity operation with \( \delta \) can be performed in any case, adding the rule \( \{\delta\} \Rightarrow \delta \), the monotonicity operation with \( \delta \) and \( \varsigma \) can be performed only if the agent has already the rule \( \varsigma \), producing a rule that extends \( \varsigma \)'s premises with \( \delta \). Finally, the cut operation with \( \varsigma_1 \) and \( \varsigma_2 \) can be performed only if \( \varsigma_2 \)'s premises include \( \varsigma_1 \)'s conclusion and the agent has already these both rules, producing a rule whose premises are those of \( \varsigma_2 \) minus \( \varsigma_1 \)'s conclusion plus those of \( \varsigma_1 \), and whose conclusion is that of \( \varsigma_2 \).

The relevant axioms of Table 2.7 are those expressing how rule sets are affected by structural operations; from them we can derive validities analogous to those given at the end of Section 2.4.1 for the case of access sets and deduction.

2.4.3 Combining dynamics

Strictly speaking, we do not need axioms relating deduction and structural operations. We can focus on the deepest occurrence of them, apply the corresponding reduction axioms to eliminate it and then proceed with the next until we remove all the operation modalities. Nevertheless, it is interesting to see how the operations interact between them; in particular, it is interesting to see how deduction is affected by structural operations.

We finish this section presenting the validities of Table 2.8, expressing how deduction after structural operations is related to deduction before them. For each structural operation, the first formula indicates that the operation does not affect deduction with a rule different from the new one, and the second indicates how deduction with the new rule changes. For this last case, the
formula presents a disjunction of two possibilities: the new rule was already in the original rule set (so just deduction is needed) or it was not (so we ask for some requisites). As an example, the second formula for monotonicity indicates that a sequence of this operation and then deduction with the generated rule $\varsigma'$ is equivalent to a single deduction with $\varsigma'$ (if $\varsigma'$ was already present) or to a sequence of deduction with $\varsigma$ and then monotonicity with the agent having explicitly knowledge about the added premise $\delta$ and the original rule $\varsigma$. See Appendix [A.5] for details about the validity proofs.

**Table 2.8: Formulas relating structural operations and deduction.**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reflexivity with $\varsigma_\delta$ the rule $[\delta] \Rightarrow \delta$</td>
<td>$\langle \text{Ref}<em>\delta \rangle \langle \Rightarrow \delta \rangle \varphi \leftrightarrow \langle \Rightarrow \delta \rangle \langle \text{Ref}</em>\delta \rangle \varphi$ for $\sigma \neq \varsigma_\delta$</td>
</tr>
<tr>
<td>Reflexivity with $\varsigma_\delta$ the rule $[\delta] \Rightarrow \delta$</td>
<td>$\langle \text{Ref}<em>\delta \rangle \langle \Leftarrow \varsigma</em>\delta \rangle \varphi \leftrightarrow \left( \langle \Leftarrow \varsigma_\delta \rangle \varphi \lor (A \varsigma \land \langle \text{Ref}_\delta \rangle \varphi) \right)$</td>
</tr>
<tr>
<td>Monotonicity with $\varsigma'$ the rule $\text{pm}(\varsigma) \cup [\delta] \Rightarrow \text{cn}(\varsigma)$</td>
<td>$\langle \text{Mon}<em>{\delta,\varsigma} \rangle \langle \Rightarrow \sigma \rangle \varphi \leftrightarrow \langle \Rightarrow \sigma \rangle \langle \text{Mon}</em>{\delta,\varsigma} \rangle \varphi$ for $\sigma \neq \varsigma'$</td>
</tr>
<tr>
<td>Monotonicity with $\varsigma'$ the rule $\text{pm}(\varsigma) \cup [\delta] \Rightarrow \text{cn}(\varsigma)$</td>
<td>$\langle \text{Mon}<em>{\delta,\varsigma} \rangle \langle \Leftarrow \varsigma' \rangle \varphi \leftrightarrow \left( \langle \Leftarrow \varsigma' \rangle \varphi \lor (A \varsigma \land \text{R} \varsigma \land \langle \Leftarrow \varsigma \rangle \langle \text{Mon}</em>{\delta,\varsigma} \rangle \varphi) \right)$</td>
</tr>
<tr>
<td>Cut with $\varsigma'$ the rule $(\text{pm}(\varsigma_2) \setminus {\text{cn}(\varsigma_1)}) \cup \text{pm}(\varsigma_1) \Rightarrow \text{cn}(\varsigma_2)$</td>
<td>$\langle \text{Cut}<em>{\varsigma_1,\varsigma_2} \rangle \langle \Rightarrow \sigma \rangle \varphi \leftrightarrow \langle \Rightarrow \sigma \rangle \langle \text{Cut}</em>{\varsigma_1,\varsigma_2} \rangle \varphi$ for $\sigma \neq \varsigma'$</td>
</tr>
<tr>
<td>Cut with $\varsigma'$ the rule $(\text{pm}(\varsigma_2) \setminus {\text{cn}(\varsigma_1)}) \cup \text{pm}(\varsigma_1) \Rightarrow \text{cn}(\varsigma_2)$</td>
<td>$\langle \text{Cut}<em>{\varsigma_1,\varsigma_2} \rangle \langle \Leftarrow \varsigma' \rangle \varphi \leftrightarrow \left( \langle \Leftarrow \varsigma' \rangle \varphi \lor (A \text{pm}(\varsigma_1) \land \text{R} \varsigma_1 \land \langle \Leftarrow \varsigma \rangle \langle \text{Cut}</em>{\varsigma_1,\varsigma_2} \rangle \varphi) \right)$</td>
</tr>
</tbody>
</table>

**2.5 Observation**

So far, our language can express just internal dynamics. We can express how deductive steps modify explicit knowledge, and even how structural operations extend the available rules, but we cannot express how knowledge is affected by external interaction. We now add the other fundamental source of information; we extend our framework to express the effect of observations.

This action has been already studied in a DEL setting: an observation is interpreted as an operation that removes those worlds where the observed fact does not hold (Definition [1.5]). In our framework we have a finer representation of the agent’s information: we distinguish between an implicit form, given by the accessibility relation, and an explicit one, given by the access sets. Then, even after fixing the effect of an observation over the agent’s implicit information, there are several possibilities for how the operation will affect the explicit part, each one of them representing a different way in which the
Chapter 2. Truth-preserving inference and observation

agent processes external information. Here, we present one of the possible definitions, what we call an \textit{explicit observation}.

2.5.1 Explicit observation

Different kinds of observations may affect the agent’s explicit information in different ways. For example, if the observation is a formula, one option is to keep the $A$-sets as before (an \textit{implicit} observation); another possibility is to add a rule without premises that will allow the agent to derive the observation one inferential step later (a \textit{semi-explicit} observation). An \textit{explicit} observation has the most intuitive effect: it adds the observed formula to its corresponding set.

\textbf{Definition 2.20 (Explicit observation operation)} Let $M = \langle W, R, V, A, R \rangle$ be an IE-model, and let $\chi$ be a formula of (a rule based on) $\mathcal{L}_p$. The model $M_{\chi^+} = \langle W', R', V', A', R' \rangle$ is given by

- $W' := \{ w \in W \mid (M, w) \models \chi \}$
- $W' := \{ w \in W \mid (M, w) \models \text{tr}(\chi) \}$,
- $R' := R \cap (W' \times W')$

and, for every $w \in W'$,

- $V'(w) := V(w)$,
- $A'(w) := A(w) \cup \{ \chi \}$ ($A'(w) := A(w)$),
- $R'(w) := R(w)$ ($R'(w) := R(w) \cup \{ \chi \}$).

The operation $(\cdot)_{\chi^+}$ is called the \textit{explicit observation operation} with $\chi$.  

The explicit observation operation behaves just like the observation operation with respect to worlds, accessibility relation and valuation. It removes worlds where the observation (its translation, in case the observation is a rule) does not hold, restricting the accessibility relation to the new domain and leaving unmodified the atomic valuation of the preserved worlds. With respect to access and rule sets, explicitly observing $\chi$ adds $\chi$ itself to the corresponding set of every world, so the agent will have the observation explicitly, as expected.

The explicit observation operation also preserves $\text{IE}_K$-models.

\textbf{Proposition 2.3} Let $M$ be an $\text{IE}_K$-model and let $\chi$ be a formula in (a rule based on) $\mathcal{L}_p$. If $M$ is in $\text{IE}_K$, so is $M_{\chi^+}$.

\textit{Proof.} Equivalence is immediate since we go to a sub-model, and the coherence properties are also simple because access (rule) sets are extended uniformly. The interesting property is truth for formulas (rules), and it follows from the fact that $\chi$ ($\text{tr}(\chi)$) is propositional, and that atomic valuations of the preserved worlds are not modified. See Appendix A.6 for details.  

\hfill \blacksquare
The language $IE^{SD!}_D$ extends $IE^S_D$ with existential modalities for explicit observations. The new formulas $⟨χ!⟩^+ϕ$ are read as “there is a way of explicitly observing $χ$ after which $ϕ$ is the case”. For the agent to observe the formula $χ$ we need for $χ$ to be true; for the agent to observe the rule $χ$ we need for $χ$ to be truth-preserving:

$$\text{Pre}_{χ!^+} := \begin{cases} χ & \text{if } χ \text{ is a formula} \\ \text{tr}(χ) & \text{if } χ \text{ is a rule} \end{cases}$$

The semantics of explicit observation formulas is given as follows.

**Definition 2.21 (Semantic interpretation)** Let $(M, w)$ be a pointed IE-model.

$$(M, w) \models ⟨χ!⟩^+ϕ \iff (M, w) \models \text{Pre}_{χ!^+} \text{ and } (M_{χ!^+}, w) \models ϕ$$

The case of its universal counterpart, defined as usual, is given by

$$(M, w) \models [χ!]^+ϕ \iff (M, w) \models \text{Pre}_{χ!^+} \text{ implies } (M_{χ!^+}, w) \models ϕ$$

In words, $⟨χ!⟩^+ϕ$ holds at $w$ in $M$ if and only if at $w$, the agent can observe $χ$ (i.e., $χ$ is true/truth-preserving) and, after explicitly doing it, $ϕ$ holds. ◀

**Axiom system** A sound and complete axiom system for the new language with respect to $IE_K$-models can be given based on those already provided and reduction axioms for the new modality.

**Theorem 2.5 (Reduction axioms for the explicit observation modality)** Table 2.9 provides reduction axioms for the explicit observation modality. Together with $IE_K$ (Theorem 2.4), they form a sound and complete axiom system for language $IE^{SD!}_D$ with respect to $IE_K$-models. ■

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
<th>If $χ$ is a formula:</th>
<th>If $χ$ is a rule:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Γ^+$</td>
<td>$χ ≡ (\text{Pre}_{χ!^+} ∧ ϕ)$</td>
<td>$Γ^+<em>A ≡ (χ!^+) A χ ≡ \text{Pre}</em>{χ!^+}$</td>
<td>$Γ^+<em>R ≡ (χ!^+) R ρ ≡ (\text{Pre}</em>{χ!^+} ∧ R ρ)$ for $ρ \neq χ$</td>
</tr>
<tr>
<td>$Γ^+$</td>
<td>$¬ϕ ≡ (\text{Pre}_{χ!^+} ∧ ¬χ!^+)ϕ$</td>
<td>$Γ^+<em>A ≡ (χ!^+) A γ ≡ (\text{Pre}</em>{χ!^+} ∧ A γ)$ for $γ \neq χ$</td>
<td></td>
</tr>
<tr>
<td>$Γ^+$</td>
<td>$(ϕ ∨ ψ) ≡ (Δ χ!^+) (ϕ ∨ ψ)$</td>
<td>$Γ^+<em>A ≡ (χ!^+) A γ ≡ (\text{Pre}</em>{χ!^+} ∧ A γ)$ for $γ \neq χ$</td>
<td></td>
</tr>
<tr>
<td>$Γ^+_R$</td>
<td>$(ϕ ∨ ψ) ≡ (Δ χ!^+) (ϕ ∨ ψ)$</td>
<td>$Γ^+<em>R ≡ (χ!^+) R ρ ≡ (\text{Pre}</em>{χ!^+} ∧ R ρ)$ for $ρ \neq χ$</td>
<td></td>
</tr>
<tr>
<td>$Γ^+_N$</td>
<td>From $ϕ$, infer $[χ!]ϕ$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.9: Axioms and rules for explicit observation modality.
The relevant axioms are those indicating how explicit information about formulas and rules is affected, and appear on the right column of the table. The agent is always informed about the observation explicitly after observing it, and any other explicit information was already present before the observation. The axioms look similar to those for deduction and structural operations, but again the important difference is the precondition. While in the case of deduction and structural operations the agent needs to have enough explicit information to extract the new piece, an observation is a more radical informational process: it just need for the observation to be true (truth-preserving).

We finish this section like the previous one, by presenting some validities expressing how the two informational processes considered in this chapter, truth-preserving inference and observation, interact with each other (details about the proof of their validity can be found in Appendix A.7). Table 2.10 presents two cases, according to whether the observed χ is a formula or a rule. Then we make a further difference, this time according to whether the observation enables the application of a rule or not. The first formula indicates that an observation does not affect deduction when the observation is not part of what the agent needs to perform the inference; the second formula presents the disjunction of two possibilities: the observation was already explicit information or it was not. These principles, together with those of Table 2.8, indicate how external and internal dynamics intertwine when we process information, as it will be shown when reviewing the restaurant example (Section 2.6).

If χ is a formula:
\[
\langle \chi^+ \rangle \langle \to_o \rangle \varphi \leftrightarrow \langle \to_o \rangle \langle \chi^+ \rangle \varphi \quad \text{for } \chi \notin \text{pm}(\sigma)
\]
\[
\langle \chi^+ \rangle \langle \to_o \rangle \varphi \leftrightarrow (\langle \to_o \rangle \langle \chi^+ \rangle \varphi \lor (A \chi \land \langle \to_o \rangle \langle \chi^+ \rangle \varphi)) \quad \text{for } \chi \in \text{pm}(\sigma)
\]

If χ is a rule:
\[
\langle \chi^+ \rangle \langle \to_o \rangle \varphi \leftrightarrow \langle \to_o \rangle \langle \chi^+ \rangle \varphi \quad \text{for } \chi \neq \sigma
\]
\[
\langle \chi^+ \rangle \langle \to_\chi \rangle \varphi \leftrightarrow (\langle \to_\chi \rangle \langle \chi^+ \rangle \varphi \lor (R \chi \land \langle \to_o \rangle \langle \chi^+ \rangle \varphi)) \quad \text{for } \chi = \sigma
\]

Table 2.10: Formulas relating explicit observation and deduction.

2.6 Back to the Restaurant

Let us represent the restaurant example with our framework. The new waiter’s initial information can be given by a model M with six possible worlds, each one of them indicating a possible distribution of the dishes, and all of them
indistinguishable from each other. For the language, consider atomic propositions of the form \( p_d \) where \( p \) stands for a person (father, mother or you) and \( d \) stands for some dish (meat, fish or vegetarian). The waiter explicitly knows each person will get only one dish, so we can put the rules

\[
\rho_1 : \{ yf \} \Rightarrow \neg yv \\
\rho_2 : \{ fm \} \Rightarrow \neg fv
\]

and similar ones in each world. Moreover, he explicitly knows that each dish corresponds to one person, so we can add the following rule, among any others

\[
\sigma : \{ \neg yv, \neg fv \} \Rightarrow mv
\]

Let \( w \) be the real world, where \( yf, fm \) and \( mv \) are true. In this initial situation, the waiter does not know neither explicitly nor implicitly that your mother has the vegetarian dish:

\[
(M, w) \models \neg \Box mv \land \neg A mv
\]

While approaching to the table, the waiter can increase the rules he knows. This does not give him new explicit facts, but allows him to reduce the number of inference steps he will need later. He has \( \rho_1 \) and \( \sigma \), and the conclusion of the first is in the premises of the second, so he can apply cut over them, getting

\[
\varsigma_1 : \{ yf, \neg fv \} \Rightarrow mv
\]

Then, we have

\[
(M, w) \models \langle \text{Cut}_{\rho_1, \sigma} \rangle \left( \neg \Box mv \land \neg A mv \land R \varsigma_1 \right)
\]

Moreover, he can apply cut again, this time with \( \rho_2 \) and \( \varsigma_1 \), obtaining the rule

\[
\varsigma_2 : \{ yf, fm \} \Rightarrow mv
\]

Now we have

\[
(M, w) \models \langle \text{Cut}_{\rho_1, \sigma} \rangle \langle \text{Cut}_{\rho_2, \varsigma_1} \rangle \left( \neg \Box mv \land \neg A mv \land R \varsigma_2 \right)
\]

After the answer to the first question, “Who has the fish?”, the waiter explicitly knows that you have the fish. Four possible worlds are removed, but he still does not know (neither explicitly nor implicitly) that your mother has the vegetarian dish. Then,

\[
(M, w) \models \langle \text{Cut}_{\rho_1, \sigma} \rangle \langle \text{Cut}_{\rho_2, \varsigma_1} \rangle \langle yf \rangle \left( \neg \Box mv \land \neg A mv \land R \varsigma_2 \land A yf \right)
\]

Then he asks “Who has the meat?”, and the answer not only gives him explicit knowledge about the fact that your father has the meat, but also gives him implicit knowledge about the fact that your mother has the vegetarian dish.

\[
(M, w) \models \langle \text{Cut}_{\rho_1, \sigma} \rangle \langle \text{Cut}_{\rho_2, \varsigma_1} \rangle \langle yf \rangle \langle fm \rangle \left( \Box mv \land \neg A mv \land R \varsigma_2 \land A yf \land A fm \right)
\]

Now he can perform the final inference step:

\[
(M, w) \models \langle \text{Cut}_{\rho_1, \sigma} \rangle \langle \text{Cut}_{\rho_2, \varsigma_1} \rangle \langle yf \rangle \langle fm \rangle \left( \Box mv \land R \varsigma_2 \land A yf \land A fm \land \langle \rightarrow \varsigma_2 \rangle A mv \right)
\]

Two structural operations, two explicit observations and one truth-preserving inference are all that is needed.
2.7 Remarks

By extending the possible worlds model with a set of formulas and a set of rules at each possible world, the framework presented in this chapter allows us to make a finer distinction in an agent’s information. We can represent not only ‘epistemic’ implicit information, but also explicit information. With this semantic model, explicit information can be defined in several ways, and the present chapter has explored the option in which the agent’s explicit information is given by the set of formulas she has in the evaluation point (A $\varphi$). Then we have asked for extra requirements that produce true implicit and explicit information, that is, implicit and explicit knowledge. This merging of syntax and semantics provides us a fine grained structure that allows us to represent the information of non-ideal (i.e., non-omniscient) agents. A list of the static notions introduced in this chapter, including their definition and the relevant properties the model should satisfy, is presented in Table 2.11.

<table>
<thead>
<tr>
<th>Notion</th>
<th>Definition</th>
<th>Relevant model requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit information</td>
<td>$\Box \varphi$</td>
<td>——</td>
</tr>
<tr>
<td>Explicit information</td>
<td>$A \gamma$</td>
<td>Coherence (Definition 2.13).</td>
</tr>
<tr>
<td>Implicit knowledge</td>
<td>$\Box \varphi$</td>
<td>Equivalence accessibility relation.</td>
</tr>
<tr>
<td>Explicit knowledge</td>
<td>$A \gamma$</td>
<td>Coherence and truth (Definition 2.15).</td>
</tr>
</tbody>
</table>

Table 2.11: Static notions of information.

On the dynamic side, we have provided a notion of explicit observation, similar in its effects to the observation act in DEL. But our focus is not on explicit versions of acts already defined and studied; providing a finer representation of an agent’s information highlights informational acts hidden before. In particular the notion of truth-preserving inference, an act that is irrelevant in standard DEL due to the omniscient nature of the represented agents, becomes now significant and, moreover, gets an intuitive and clear representation. But there is more. Once our act of rule-based inference has been defined, we have shown how structural operations allow the agent to extend the rules she can apply. In all the cases we have provided the model operation, the corresponding modalities for the language, and reduction axioms that express how the operations modify the truth-value of formulas in the language, therefore allowing us to derive how the notions of information are affected. We have also presented validities describing how deduction, structural operations and observations interact with each other. A list of the reviewed actions and a brief description of their effect is presented in Table 2.12.
2.7. Remarks

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truth-preserving inference.</td>
<td>Turns implicit knowledge into explicit knowledge.</td>
</tr>
<tr>
<td>Structural operations.</td>
<td>Add truth-preserving rules the agent can apply.</td>
</tr>
<tr>
<td>Explicit observation.</td>
<td>Changes the agent's implicit and explicit knowledge.</td>
</tr>
</tbody>
</table>

Table 2.12: Actions and their effects.

Still, there are some not completely satisfactory points in our proposal. Among them, the most important is the one that limits the agent’s explicit information to only propositional formulas, leaving out high-order information (information about her own and, eventually, other agent’s information) and information about how actions affect her and other agent’s information. The reason for restricting access sets to purely propositional formulas is that, in general, the truth-value of formulas of the full implicit/explicit language is not preserved by the explicit observation operation, and then the operation does not preserve the truth property for formulas. Under our definition of explicit information, this property is needed to deal with the case of true information, that is, knowledge.

Let us look at the problem in more detail. In general, a true observed formula of the full implicit/explicit language cannot be simply added to an access set because it may become false after being observed. Classical examples of such cases are Moore sentences of the form $p \land \neg \Box p$. Intuitively, an observation of “$p$ is the case and the agent does not know it (implicitly)” will make the agent to know (implicitly) $p$, and therefore the observation is not true anymore. Technically, an explicit observation of $p \land \neg \Box p$ keeps only those worlds in which the observation is true in the original model, but the operation affects the accessibility relation so $\neg \Box p$ will not be true in the model that results from the operation.

A first attempt to solve this limitation would be to change the definition of the new access set function in order to keep only those formulas that are still true in the new model. Nevertheless, such definition faces circularity. The new access set should contain only those formulas of the original one that are still true in the new one; but, in particular, in order to decide whether an explicit information formula $A \gamma$ is true or not in the new model, we need the new access set, precisely the one we are just defining.

Yang (2009) suggests another possibility. Though it is reasonable to ask for our non-omniscient agent to have true propositional information, maybe it is too much to ask for her to have also true high-order information. He suggests to allow arbitrary formulas of the full language in access sets, but restrict the truth property to purely propositional ones. As he mentions, our non-omniscient agent does not need to realize automatically all the high-order
consequences of the observation. Nevertheless, she should be able to realize eventually that some explicit (high-order) information she held correctly before has been ‘outdated’ by an informational act, and therefore she should be able to correct herself.

There is another option. As we mentioned, the definition of explicit information that we have used is not the only possibility. We will discuss some other alternatives in the following chapter, when we will focus on another action logical omniscience hides: changes in awareness.