5

Epistemic Actions

5.1 Introduction

The previous chapter dealt with public announcements. There are various more complex 'updates', or, as we call them, epistemic actions. This chapter introduces a language and logic for epistemic actions. Public announcements are epistemic actions that convey the same information for all agents. They result in a restriction of the domain of the model and therefore in a restriction of the corresponding accessibility relations. More complex epistemic actions convey different information to different agents. They may result in the refinement of accessibility relations while the domain of the model remains unchanged, and they may even result in the enlargement of the domain of the model (and its structure), even when the complexity is measured by the number of non-bisimilar states. We start with a motivating example. Reconsider Example 4.1 on page 67 of the previous chapter 'public announcements'.

Example 5.1 (Buy or sell?) Consider two stockbrokers Anne and Bill, having a little break in a Wall Street bar, sitting at a table. A messenger comes in and delivers a letter to Anne. On the envelope is written "urgently requested data on United Agents". □

As before, we model this as an epistemic state for one atom p, describing 'the letter contains the information that United Agents is doing well', so that $\neg p$ stands for United Agents *not* doing well. And it is reasonable to assume that both Anne (a) and Bill (b) know what information on United Agents is due, as this was announced by the messenger in their presence. In other words: a and b are both uncertain about the value of p, and this is common knowledge. In fact, p is true. Unlike before, where we assumed Anne had read the letter, we start with the initial situation where both Anne and Bill are ignorant. The epistemic model for this we call *Letter*. Figure 5.1 presents the epistemic state $(\mathit{Letter}, 1)$.

In Figure 5.1 we choose mnemonically convenient names 0 and 1 for states where p is false and true, respectively. In Example 4.1 we assumed that the

$$0 \;\text{———}\; a, b \;\text{———}\; \underline{1}$$

Figure 5.1. Epistemic state $(\mathit{Letter}, 1)$ modelling the situation where Anne and Bill are ignorant about atom p that is in fact true.

letter-reading was not aloud, and with Bill watching Anne. Also we did not formally model that action. This time we will model the action, and also add some more variation. Consider the following scenarios. Except for the parts between brackets, of which only one or no agent is aware, Anne and Bill commonly know that these actions are taking place.

Example 5.2 (tell) Anne reads the letter aloud. United Agents is doing well. □

Example 5.3 (read) Bill sees that Anne reads the letter. (United Agents is doing well.) □

Example 5.4 (mayread) Bill leaves the table and orders a drink at the bar so that Anne may have read the letter while he was away. (She does not read the letter.) (United Agents is doing well.) □

Example 5.5 (bothmayread) Bill orders a drink at the bar while Anne goes to the bathroom. Each may have read the letter while the other was away from the table. (Both read the letter.) (United Agents is doing well.) □

After execution of the action tell it is common knowledge that p: in the resulting epistemic state $C_{ab}p$ holds. This is not the case when the action is read, but still, some common knowledge is obtained there, namely $C_{ab}(K_a p \vee K_a \neg p)$: it is commonly known that Anne knows the contents of the letter, irrespective of it being p or $\neg p$. This is yet different in the third action, mayread; after that action, Bill does not even know if Anne knows p or knows $\neg p$: $\neg K_b(K_a p \vee K_a \neg p)$. Still, Bill has learnt something: he now considers it possible that Anne knows p, or knows $\neg p$. For Bill, action mayread involves actual choice for Anne: whatever the truth about p, she may have learnt it, or not. In the case of bothmayread both $\neg K_b(K_a p \vee K_a \neg p)$ and $\neg K_a(K_b p \vee K_b \neg p)$ are postconditions. Each agent has a choice, from the perspective of the other agent. Actually, *both* agents learn p, so that p is generally known: $E_{ab}p$, but they are ignorant of each other's knowledge: $\neg E_{ab}E_{ab}p$ (from which follows, in particular, that p is not common knowledge: $\neg C_{ab}p$), and *that* is commonly known to them: $C_{ab}\neg E_{ab}E_{ab}p$. The state transitions induced by each of these actions are shown in Figure 5.2.[1]

From these actions, we can only model the tell action in public announcement logic: it is public announcement of p. In public announcement logic 'after

[1] In Example 4.1 the action read was implicit. Therefore, the two-point model in Figure 5.2 resulting from that action is the same as the two-point model in Figure 4.1 in which public announcements take place.

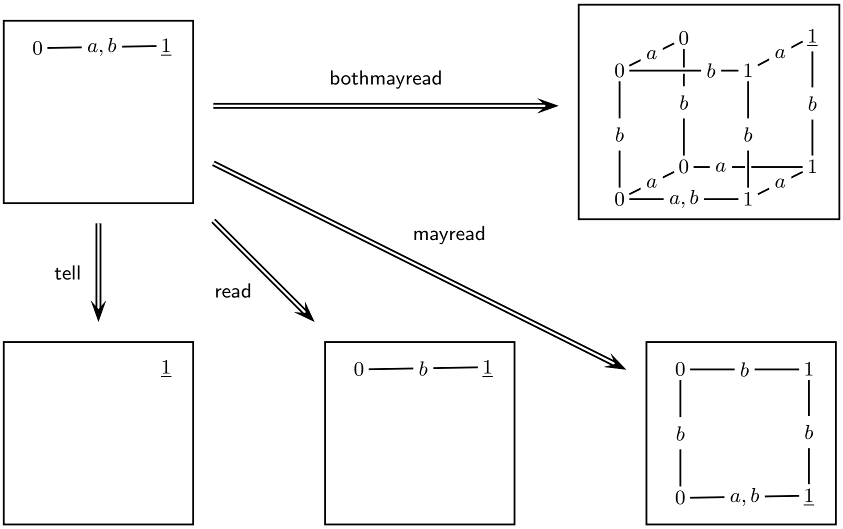

Figure 5.2. Epistemic states resulting from the execution of the four example actions. The top left figure represents $(Letter, 1)$. Points of epistemic states are underlined. Assume transitivity of access. For mayread and bothmayread only one of more executions is shown.

public announcement of φ, it holds that ψ' was described as $[\varphi]\psi$. In epistemic action logic this will be described as

$$[L_A?\varphi]\psi$$

Here, L stands for 'learning'. This is a dynamic counterpart of static common knowledge. We can paraphrase $[L_A?\varphi]\psi$ by: after the group A learns φ, it holds that ψ. One might also say: after it is common knowledge that the action 'that φ' takes place, it holds that ψ.

A simple example of an action involving learning by subgroups strictly smaller than the public, is the read action. It is described as

$$L_{ab}(!L_a?p \cup L_a?\neg p)$$

This stands for 'Agents a and b learn that a learns the truth about fact p, and actually agent a learns that p'; or, in yet other words 'Anne and Bill learn that Anne learns that p or that Anne learns that $\neg p$, and actually (this is the role of the exclamation mark in front of that alternative) Anne learns that p. In this case, the action results in a refinement of agent a's accessibility relations (corresponding to her increased knowledge). Although such knowledge refinement is related to non-deterministic choice, in this case Anne has no choice: either p or $\neg p$ is true, but not both.

In the mayread action instead, Anne has a choice from Bill's point of view. This action will be described by

$$L_{ab}(L_a?p \cup L_a?\neg p \cup\ !?\top)$$

which stands for 'Agents a and b learn that a *may* learn the truth about fact p, although actually nothing happens'. Or, in yet other words: 'Agents a and b learn that either a learns the truth about fact p, or not; and actually she does not'. Actions involving choice may increase the complexity of the epistemic state, as measured in the number of non-bisimilar states. This is indeed the case here: we go from two to four states, as one may check in Figure 5.2.

The bothmayread action is a more complex scenario also involving non-deterministic choice. It will be explained in detail after the formal introduction of the action language and its semantics.

A relational language for epistemic actions is introduced in Section 5.2. The semantics for this language is given in Section 5.3. This section contains subsections on valid principles for the logic. We do not give an axiomatisation for the logic—there are some difficulties with a principle relating knowledge and epistemic actions, summarily addressed near the end of Section 5.3. Sections 5.4 and 5.5 model multi-agent system features in the epistemic action logics: 'card game states', and 'spreading gossip', respectively.

5.2 Syntax

To the language $\mathcal{L}_{KC}$ for multi-agent epistemic logic with common knowledge for a set A of agents and a set P of atomic propositions, we add dynamic modal operators for programs that are called epistemic actions or just actions. Actions may change the knowledge of the agents involved, but do not change facts.

Definition 5.6 (Formulas, actions, group) The language $\mathcal{L}_!(A,P)$ is the union of the *formulas* $\mathcal{L}_!^{\text{stat}}(A,P)$ and the *actions* $\mathcal{L}_!^{\text{act}}(A,P)$, defined by

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

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

where $p \in P$, $a \in A$, $B \subseteq A$, and $\psi \in \mathcal{L}_!^{\text{stat}}(gr(\alpha),P)$, $\beta \in \mathcal{L}_!^{\text{act}}(B,P)$, and $\beta' \in \mathcal{L}_!^{\text{act}}(gr(\alpha),P)$. The *group* $gr(\alpha)$ of an action α is defined as: $gr(?\varphi) = \emptyset$, $gr(L_B\alpha) = B$, $gr(\alpha\ !\ \alpha') = gr(\alpha)$, $gr(\alpha\ ¡\ \alpha') = gr(\alpha')$, $gr(\alpha\ ;\ \alpha') = gr(\alpha')$, and $gr(\alpha \cup \alpha') = gr(\alpha) \cap gr(\alpha')$. □

As usual, we omit the parameters P and/or A unless this might cause ambiguity. So instead of $\mathcal{L}_!(A,P)$ we write $\mathcal{L}_!(A)$ or even $\mathcal{L}_!$. Unlike before, the parameter A is now often essential, and will in that case have to remain explicit. The dual of $[\alpha]$ is $\langle\alpha\rangle$. We omit parentheses from formula and action expressions unless ambiguity results.

Action $?\varphi$ is a *test*. Operator L_B is called *learning*, and the construct $L_B?\varphi$ is pronounced as 'group B learn that φ'. Action $(\alpha \mathbin{!} \alpha')$ is called *(left) local choice*, and $(\alpha \mathbin{¡} \alpha')$ is called *(right) local choice*. Action $(\alpha \mathbin{;} \alpha')$ is *sequential execution*—'first α, then α'', and action $(\alpha \cup \alpha')$ is *non-deterministic choice* between α and α'.

In epistemic action logic, 'the set of agents' occurring in formulas and in structures is a constantly changing parameter. The notion of *group gr* keeps track of the agents occurring in learning operators in actions. The constructs $[\alpha]\psi$, $L_B\beta$, and $(\alpha \mathbin{;} \beta')$ in the definition, wherein the group of an action is used as a constraint, guarantee that in an epistemic state for agents B that is the result of action execution, formulas containing modal operators for agents not in B are not in the language. This also explains the possibly puzzling clause $gr(\alpha \mathbin{;} \alpha') = gr(\alpha')$: the group of α' is by definition the smaller of $gr(\alpha)$ and $gr(\alpha')$.

Local choice $(\alpha \mathbin{!} \alpha')$ may be seen as 'from α and α', choose the first,' and local choice $\alpha \mathbin{¡} \alpha'$ as 'from α and α', choose the second'. Given that, why bother having those constructs? We need them, because the semantics of a learning operator that binds an action containing local choice operations, is defined in terms of those operators. We will see that in $L_B(\alpha \mathbin{!} \alpha')$, everybody in B but not in learning operators occurring in α, α', is unaware of the choice for α. That choice is therefore 'local'. The operations '!' and '¡' are dual; $\alpha \mathbin{!} \alpha'$ means the same as $\alpha' \mathbin{¡} \alpha$. Typically, we prove local choice properties for '!' only, and not for '¡'.

Instead of $(\alpha \mathbin{!} \alpha')$ we generally write $(!\alpha \cup \alpha')$. This expresses more clearly that given choice between α and α', the agents involved in those actions choose α, whereas that choice remains invisible to the agents that learn about these alternatives but are not involved. Similarly, instead of $(\alpha \mathbin{¡} \alpha')$ we generally write $(\alpha \cup\, !\alpha')$. There is a simple relation between 'local choice' and 'non-deterministic choice', but before we explain that, a few examples.

Example 5.7 The action read where Bill sees that Anne reads the letter is described as $L_{ab}(L_a?p \mathbin{!} L_a?\neg p)$, also written as $L_{ab}(!L_a?p \cup L_a?\neg p)$. It can be paraphrased as follows: 'Anne and Bill learn that either Anne learns that United Agents is doing well or that Anne learns that United Agents is not doing well; and actually Anne learns that United Agents is doing well.' The part $(L_a?p \mathbin{!} L_a?\neg p)$ means that the first from alternatives $L_a?p$ and $L_a?\neg p$ is chosen, but that agent b is unaware of that choice. □

Example 5.8 The descriptions in $\mathcal{L}_!^{\mathrm{act}}(\{a,b\},\{p\})$ of the actions in the introduction are (including read again):

tell	$L_{ab}?p$
read	$L_{ab}(!L_a?p \cup L_a?\neg p)$
mayread	$L_{ab}(L_a?p \cup L_a?\neg p \cup\, !?\top)$
bothmayread	$L_{ab}(!L_a?p \cup L_a?\neg p \cup\, ?\top) \mathbin{;} L_{ab}(!L_b?p \cup L_b?\neg p \cup\, ?\top)$

In the last two actions, instead of $?\top$ (for 'nothing happens') we may as well write $(?p\cup?\neg p)$. We abused the language somewhat in those actions: 'from several actions, choose the first' is a natural and obvious generalisation of local choice. A description $L_{ab}(L_a?p \cup L_a?\neg p\cup\ !?\top)$ is formally one of the following three

$$L_{ab}(L_a?p \ ¡\ (L_a?\neg p \ ¡\ \ ?\top))$$
$$L_{ab}((L_a?p \ ¡\ L_a?\neg p) \ ¡\ \ ?\top)$$
$$L_{ab}((L_a?p \ !\ L_a?\neg p) \ ¡\ \ ?\top)$$

□

A non-deterministic action contains $\cup$ operators and may have more than one execution in a given epistemic state. In terms of relations between epistemic states: the relation is not functional. The only difference between an action with $\cup$ and one where this has been replaced by '!' or '¡' is, in terms of its interpretation, a restriction on that relation that makes it (more) functional.

Definition 5.9 (Type of an action) The *type*

$$\alpha_\cup$$

of action α is the result of substituting '$\cup$' for all occurrences of '!' and '¡' in α, except when under the scope of '?'. If $\alpha_\cup = \beta_\cup$ we say that α and β are the same type of action. An *instance* of an action α is the result of substituting either '!' or '¡' for all occurrences of '$\cup$' in α, except, again, when under the scope of '?'. For an arbitrary instance of an action α we write

$$\alpha_!$$

Let $\sim_!$ be the least congruence relation on actions such that $(\alpha\ !\ \alpha') \sim_! (\alpha\ ¡\ \alpha')$. If $\alpha \sim_! \beta$, then α and β are *comparable*. An $\mathcal{L}_!$ action is *deterministic* iff it does not contain $\cup$ operators, except when under the scope of '?'. □

In other words, if $\alpha \sim_! \beta$, then α and β are the same except for swaps of '!' for '¡' and vice versa. All instances of an action are comparable among each other, all instances of an action are deterministic, and the type of an action is itself an action. Local choice (nor other) operators in the scope of a test operator '?' do not matter: tests occurring in action descriptions represent *preconditions* for action execution; whatever is tested, is irrelevant from that perspective.

Example 5.10 The type $\mathsf{read}_\cup$ of the action read is $L_{ab}(L_a?p \cup L_a?\neg p)$. □

The action $\mathsf{read}_\cup$ is different from read, because another instance (and therefore possible execution) of $\mathsf{read}_\cup$ is $L_{ab}(L_a?p \ ¡\ L_a?\neg p)$, also written as $L_{ab}(L_a?p\cup\ !L_a?\neg p)$. Actions $L_{ab}(!L_a?p \cup L_a?\neg p)$ and $L_{ab}(L_a?p\cup\ !L_a?\neg p)$ are *comparable*, i.e., $L_{ab}(!L_a?p \cup L_a?\neg p) \sim_!\ L_{ab}(L_a?p\cup\ !L_a?\neg p)$.

Example 5.11 The types of the actions in the introduction are:

$\mathsf{tell}_\cup$	$L_{ab}?p$
$\mathsf{read}_\cup$	$L_{ab}(L_a?p \cup L_a?\neg p)$
$\mathsf{mayread}_\cup$	$L_{ab}(L_a?p \cup L_a?\neg p \cup\ ?\top)$
$\mathsf{bothmayread}_\cup$	$L_{ab}(L_a?p \cup L_a?\neg p \cup\ ?\top)\ ;\ L_{ab}(L_b?p \cup L_b?\neg p \cup\ ?\top)$ $\square$

The action mayread is one of three of that type, and bothmayread one of 9. If we are more formal, this amounts to one of 4, and 16, respectively. The difference is easily explained. With $L_{ab}((L_a?p$ ¡ $L_a?\neg p)$ ¡ $?\top)$ as the chosen precision of mayread, that action has the same type as the other instance $L_{ab}((L_a?p\ !\ L_a?\neg p)$ ¡ $?\top)$. But both are precisions of mayread, wherein the choice between the first two alternatives is invisible.

There is a certain symmetry in action descriptions that appears broken in tell. If we had described tell as $(!L_{ab}?p \cup L_{ab}?\neg p)$ instead of $L_{ab}?p$, its type would have been $(L_{ab}?p \cup L_{ab}?\neg p)$. That description expresses that Anne tells the truth about p, whatever it is: a description that is independent from the actual state, so to speak. So why the shorter description? In an action $(!L_{ab}?p \cup L_{ab}?\neg p)$, the choice expressed in ! is visible 'only to the agents involved in the choice options and not to others': but in this case there are no other agents, so there is no point in using that description.

5.3 Semantics

5.3.1 Basic Definitions

The language $\mathcal{L}_!(A, P)$ is interpreted in multi-agent epistemic states. For an introduction to and overview of the structures relevant to this semantics, see Chapter 2. For the class of epistemic models for agents A and atoms P we write $\mathcal{S}5(A, P)$, for the class of epistemic states we write $\bullet\mathcal{S}5(A, P)$. In this chapter we often combine models for different groups of agents. If M is a multi-agent epistemic model for agents A, we write $gr(M) = A$ and say 'the group of model M is A', and, given some state s in the domain of M, similarly, $gr((M, s)) = A$. For the class of all epistemic models for subsets of agents A and atoms P we write $\mathcal{S}5(\subseteq A, P)$, and similarly for epistemic states. We abuse the language and write $M \in \mathcal{S}5(A, P)$ for an arbitrary epistemic model M in class $\mathcal{S}5(A, P)$, etc.

A peculiarity of this setting is that a model for the *empty* set of agents is an $S5$ model, i.e., an epistemic model. This is, because for each agent $a \in \emptyset$, the accessibility relation R_a associated with that agent (namely: none) is an equivalence relation. Note the difference between an *empty* accessibility relation $R_a = \emptyset$ and the *absence* of an accessibility relation for a given agent. A model with an empty accessibility relation for a given agent is not an $S5$ model, because access is not reflexive for that agent.

The semantics of $\mathcal{L}_!(A, P)$ has a clause for the interpretation of dynamic modal formulas of form $[\alpha]\varphi$, where α is an action and φ a formula. In public announcement logic this was $[\psi]\varphi$, for *two* arbitrary formulas, but now we have a whole range of epistemic actions, not just public announcement.

As was indicated in the introduction earlier, sometimes the domain of an epistemic state is enlarged due to an epistemic action. The question is where those extra states come from? And once we have acquired them, how do we determine which are indistinguishable for an agent? And how do we determine the valuation for these states? The idea is the following. If the agents in a group *learn* that an action α takes place, we consider the epistemic states that result from executing α or comparable actions. These *epistemic states* are taken to be the *factual states* in the model that results from executing learning α. We obtain extra states by having actions with non-deterministic choice. The accessibility relation is then determined by the internal structure of the epistemic states that are taken to be factual states. In order to accommodate this, we need a notion of equivalence between epistemic states. We therefore lift equivalence of factual states in an epistemic state, to equivalence of epistemic states. This notion will be used in Definition 5.13 of action interpretation.

Definition 5.12 (Equivalence of epistemic states) Let $M, M' \in \mathcal{S}5(\subseteq A)$, $s \in M$, $s' \in M'$, and $a \in A$. Then

$$(M, s) \sim_a (M', s') \text{ iff } \begin{array}{l} a \notin gr(M) \cup gr(M') \text{ or} \\ M = M' \text{ and } s \sim_a s' \text{ or} \\ \text{there is a } t \in M : (M, t) \leftrightarrows (M', s') \text{ and } s \sim_a t \end{array} \quad \square$$

In other words: if an agent does not occur in either epistemic state, the epistemic states are the same from that agent's point of view; otherwise, they are the same if the points of those epistemic states are the same for that agent, modulo bisimilarity (see Definition 2.14 of bisimilarity $\leftrightarrows$ on page 24 in Chapter 2). Note that we choose to overload the notation $\sim_a$: it applies equally to factual states and epistemic states. Exercise 5.16 demonstrates what happens when the bisimilarity clause is removed. From the above definition one can immediately observe[2] that for all $a \in gr(M) \cup gr(M')$:

$$\text{If } (M, s) \sim_a (M', s'), \text{ then } gr(M) = gr(M')$$

Definition 5.13 (Semantics of formulas and actions) Let $M = \langle S, \sim, V\rangle \in \mathcal{S}5(A, P)$ and $s \in S$. The semantics of $\mathcal{L}_!^{\text{stat}}(A, P)$ formulas and $\mathcal{L}_!^{\text{act}}(A, P)$ actions is defined by double induction.

[2] Definition 5.12 gives three ways to achieve $(M, s) \sim_a (M', s')$. The case $a \notin gr(M) \cup gr(M')$ is ruled out by the assumption. The case $M = M'$ makes $gr(M) = gr(M')$ trivial. In the remaining case M is bisimilar to M', so that $gr(M) = gr(M')$ is also trivial.

$$
\begin{array}{lll}
M, s \models p & \text{iff} & s \in V_p \\
M, s \models \neg\varphi & \text{iff} & M, s \not\models \varphi \\
M, s \models \varphi \wedge \psi & \text{iff} & M, s \models \varphi \text{ and } M, s \models \psi \\
M, s \models K_a\varphi & \text{iff} & \text{for all } s' \in S : s \sim_a s' \text{ implies } M, s' \models \varphi \\
M, s \models C_B\varphi & \text{iff} & \text{for all } s' \in S : s \sim_B s' \text{ implies } M, s' \models \varphi \\
M, s \models [\alpha]\varphi & \text{iff} & \text{for all } M', s' : \\
 & & (M, s)[\![\alpha]\!](M', s') \text{ implies } M', s' \models \varphi
\end{array}
$$

$$
\begin{array}{lll}
(M, s)[\![?\varphi]\!](M', s') & \text{iff} & M' = \langle [\![\varphi]\!]_M, \emptyset, V|[\![\varphi]\!]_M \rangle \text{ and } s = s' \\
(M, s)[\![L_B\alpha]\!](M', s') & \text{iff} & M' = \langle S', \sim', V' \rangle \text{ and } (M, s)[\![\alpha]\!]s' \text{ (see below)} \\
[\![\alpha \ ; \ \alpha']\!] & = & [\![\alpha]\!] \circ [\![\alpha']\!] \\
[\![\alpha \cup \alpha']\!] & = & [\![\alpha]\!] \cup [\![\alpha']\!] \\
[\![\alpha \ ! \ \alpha']\!] & = & [\![\alpha]\!]
\end{array}
$$

In the clause for $[\alpha]\varphi$, $(M', s') \in \bullet\mathcal{S}5(\subseteq A, P)$. In the clause for $?\varphi$, $(V|[\![\varphi]\!]_M)_p = V_p \cap [\![\varphi]\!]_M$. In the clause for $L_B\alpha$, $(M', s') \in \bullet\mathcal{S}5(B, P)$ such that

$$S' = \{(M'', s'') \mid \text{there is a } t \in S : (M, t)[\![\alpha_\cup]\!](M'', s'')\} \ ;$$

if $(M, s)[\![\alpha_\cup]\!](M''_1, s'')$ and $(M, t)[\![\alpha_\cup]\!](M''_2, t'')$, then for all $a \in B$

$$(M''_1, s'') \sim'_a (M''_2, t'') \quad \text{iff} \quad s \sim_a t \text{ and } (M''_1, s'') \sim_a (M''_2, t'')$$

where the rightmost $\sim_a$ is equivalence of epistemic states; and for an arbitrary atom p and state (M'', u) (with valuation V'') in the domain of M':

$$(M'', s'') \in V'_p \quad \text{iff} \quad s'' \in V''_p$$

We call all the validities under this semantics the logic EA.. □

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

$$M, s \models \langle\alpha\rangle\varphi \quad \text{iff} \quad \text{there is a } (M', t) : (M, s)[\![\alpha]\!](M', t) \text{ and } M', t \models \varphi$$

A test results in an epistemic state without access for any agent. This is appropriate: knowledge changes as the result of ‘learning’; therefore, before we encounter a learn operator we cannot say anything at all about the knowledge of the agents in the resulting epistemic state: no access. (Note that ‘no access’ is different from ‘empty access’—see the explanation at the beginning of this subsection.) In other words, the computation of agents’ knowledge is postponed until, working one’s way upward from subactions that are tests, L operators are encountered in the action description.

To execute an action $L_B\alpha$ in an epistemic state (M, s), we do not just have to execute the (proper part of the) *actual* action α in the *actual* epistemic state (M, s), but also any *other* action of the same type of α in any *other* epistemic state (M, t) with the same underlying model M. The resulting set of epistemic states is the domain of the epistemic state that results from executing $L_B\alpha$ in (M, s). In other words: these epistemic states represent factual states. Such factual states cannot be distinguished from each other by an agent $a \in B$ iff their origins are indistinguishable, and if they are also indistinguishable as epistemic states. (In fact, if the agent occurs in those epistemic states, the first follows from the last. See Lemma 5.17, later.)

The semantics of learning is the raison d'être for local choice operators. Even though $[\![\beta \; ! \; \beta']\!] = [\![\beta]\!]$, so that—from a more abstract perspective—$[\![\beta \; ! \; \beta']\!]$ is computed from $[\![\beta]\!]$ *only*, $[\![L_B(\beta \; ! \; \beta')]\!]$ is computed from both $[\![\beta]\!]$ *and* $[\![\beta']\!]$. That is because the semantics of $[\![L_B(\beta \; ! \; \beta')]\!]$ is defined in terms of $[\![(\beta \; ! \; \beta')_\cup]\!]$, which is $[\![\beta_\cup \cup \beta'_\cup]\!]$, which is $[\![\beta_\cup]\!] \cup [\![\beta'_\cup]\!]$: a function of $[\![\beta]\!]$ *and* $[\![\beta']\!]$.

The semantics is complex because epistemic states serve as factual states. It is important to realise that this is merely a complex *naming device*. What counts in the model that results from a learning action, is the valuation of atoms on the domain and the access between states of the domain—whatever the names of such states.

If the interpretation of epistemic action α in epistemic state (M, s) is not empty, we say that α is *executable* in (M, s). If the interpretation is functional as well, write $(M, s)[\![\alpha]\!]$ for the unique (M', s') such that $(M, s)[\![\alpha]\!](M', s')$. Various properties of this semantics can be more conveniently formulated for actions whose interpretation is always functional. For those, see Section 5.3.4. First, an elaborate example of the semantics.

5.3.2 Example of Epistemic Action Semantics

The interpretation of the action $\mathsf{read} = L_{ab}(!L_a?p \cup L_a?\neg p)$ in epistemic state $(\mathit{Letter}, 1)$ (see Example 5.3 and Figure 5.2) is defined in terms of the interpretation of its type $(L_a?p \cup L_a?\neg p)$ on $(\mathit{Letter}, 1)$ and $(\mathit{Letter}, 0)$. To interpret $(L_a?p \cup L_a?\neg p)$ on $(\mathit{Letter}, 1)$ we may either interpret $L_a?p$ or $L_a?\neg p$. Only the first can be executed. The interpretation of $L_a?p$ on $(\mathit{Letter}, 1)$ is defined in terms of the interpretation of $?p$ on any epistemic state (Letter, x) where $?p$ can be executed, i.e. where p holds, that is on $(\mathit{Letter}, 1)$ only. Epistemic state $(\mathit{Letter}, 1)[\![?p]\!]$ is the singleton epistemic state consisting of factual state 1 without access, and where p is true. This epistemic state is therefore the single factual state in the domain of $(\mathit{Letter}, 1)[\![L_a?p]\!]$. Because the epistemic state that it stands for lacks access for a, and because $1 \sim_a 1$ in *Letter* (or quite simply because $\sim_a$ is an equivalence relation for all agents in the group of the model, currently $\{a\}$), that factual state has reflexive access for a:

$$(\mathit{Letter}, 1)[\![?p]\!] \sim_a (\mathit{Letter}, 1)[\![?p]\!]$$

In the next and final stage of the interpretation, where we construct the effect of L_{ab}, note that (as factual states)

$$(\mathit{Letter}, 1)[\![L_a?p]\!] \sim_b (\mathit{Letter}, 0)[\![L_a?\neg p]\!]$$

because agent b does not occur in those epistemic states and $1 \sim_b 1$ in *Letter*, but that

$$(Letter, 1)[\![L_a?p]\!] \not\sim_a (Letter, 0)[\![L_a?\neg p]\!]$$

because a does occur in both epistemic states and $(Letter, 1)[\![L_a?p]\!]$ is not bisimilar to $(Letter, 0)[\![L_a?\neg p]\!]$. As in the previous step of the construction, p is true in $(Letter, 1)[\![L_a?p]\!]$ and false in $(Letter, 0)[\![L_a?\neg p]\!]$. Finally, the epistemic state $(Letter, 1)[\![L_a?p]\!]$ is also the point of the resulting model. See Figure 5.3 for an overview of these computations.

Exercise 5.14 Compute the interpretation of $\mathsf{tell} = L_{ab}?p$ on $(Letter, 1)$ (see Example 5.8). □

Exercise 5.15 The action $\mathsf{bothmayread}$ (see Example 5.8) is described as $L_{ab}(!L_a?p \cup L_a?\neg p \cup\ ?\top)\ ;\ L_{ab}(!L_b?p \cup L_b?\neg p \cup\ ?\top)$. Compute the result of executing the second part of this action, $L_{ab}(!L_b?p \cup L_b?\neg p \cup\ ?\top)$, in the epistemic state $(Letter, 1)[\![L_{ab}(!L_a?p \cup L_a?\neg p \cup\ ?\top)]\!]$ resulting from executing its first part. □

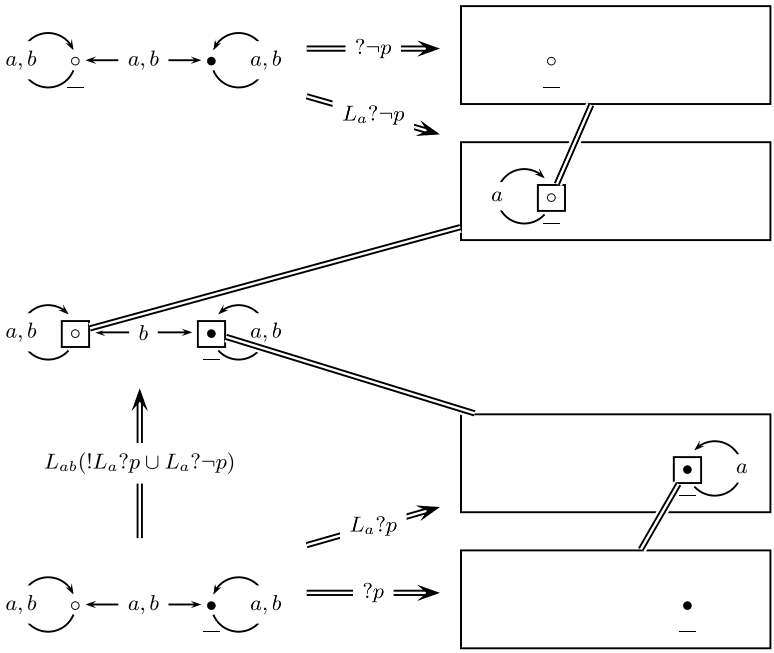

Figure 5.3. Details of the interpretation of action read in $(Letter, 1)$. All access is visualised. Atom p holds in • states, and does not hold in ○ states. Linked boxes are identical. See also Figure 5.2.

Exercise 5.16 In the second clause of Definition 5.12, bisimilarity to an $\sim_a$-equal epistemic state is a sufficient condition for $\sim_a$-equivalence of epistemic states. Without the bisimilarity 'relaxation', not enough epistemic states would be 'the same'.

Consider the epistemic state $(\mathit{Letter}, 1)$. Show that without the bisimilarity condition, agent b can distinguish the effects of action

$$L_{ab}(!(L_{ab}L_{ab}?q\ ;\ L_{ab}(!L_a?p \cup L_a?\neg p)) \cup (L_{ab}?q\ ;\ L_{ab}(L_a?p\cup\ !L_a?\neg p)))$$

from those of action

$$L_{ab}((L_{ab}L_{ab}?q\ ;\ L_{ab}(!L_a?p \cup L_a?\neg p))\cup!(L_{ab}?q\ ;\ L_{ab}(L_a?p\cup\ !L_a?\neg p)))$$

Explain also why this is undesirable. □

5.3.3 Semantic Properties

Lemma 5.17 Suppose that $(M, s)[\![\alpha]\!](M', s')$ and $(M, t)[\![\beta]\!](M'', t'')$, and that $a \in gr(M') \cup gr(M'')$. If $(M', s') \sim_a (M'', t'')$, then $s \sim_a t$. □

Proof Equivalence of $(M', s') \sim_a (M'', t'')$ is established by equivalence of the points of those epistemic states, modulo bisimilarity. But the *only* place where access between factual states, such as points, is ever constructed in the semantics of actions, is in the clause for 'learning'. In that part of the semantics, access can only be established if the 'origins' of those states are already the same, i.e., if $s \sim_a t$. □

The more intuitive contrapositive formulation of this Lemma says, that if an agent can distinguish states from each other, they will never become the same. In a temporal epistemic context this is known as the property of *perfect recall*.

Proposition 5.18 (Action algebra) Let $\alpha, \alpha', \alpha'' \in \mathcal{L}_!^{\mathrm{act}}(A)$. Then:

- $[\![(\alpha \cup \alpha') \cup \alpha'']\!] = [\![\alpha \cup (\alpha' \cup \alpha'')]\!]$
- $[\![(\alpha\ ;\ \alpha')\ ;\ \alpha'']\!] = [\![\alpha\ ;\ (\alpha'\ ;\ \alpha'')]\!]$
- $[\![(\alpha \cup \alpha')\ ;\ \alpha'']\!] = [\![(\alpha\ ;\ \alpha'') \cup (\alpha'\ ;\ \alpha'')]\!]$
- $[\![L_B\alpha]\!] = [\![L_B L_B\alpha]\!]$ □

Proof The first three properties are proved by simple relational algebra. We show the third, the rest is similar: $[\![(\alpha\cup\alpha')\ ;\ \alpha'']\!] = [\![\alpha\cup\alpha']\!]\circ[\![\alpha'']\!] = ([\![\alpha]\!]\cup[\![\alpha']\!])\circ [\![\alpha'']\!] = ([\![\alpha]\!]\circ[\![\alpha'']\!])\cup([\![\alpha']\!]\circ[\![\alpha'']\!]) = [\![\alpha\ ;\ \alpha'']\!]\cup[\![\alpha'\ ;\ \alpha'']\!] = [\![(\alpha\ ;\ \alpha'')\cup(\alpha'\ ;\ \alpha'')]\!]$.

Concerning $[\![L_B\alpha]\!] = [\![L_B L_B\alpha]\!]$, this immediately follows from the semantics of actions: consider the computation of $L_B L_B\alpha$. Let (M, s) be arbitrary. We have that α is executable iff $L_B\alpha$ is executable, that (therefore) the number of epistemic states (M', s') such that $(M, s)[\![\alpha_\cup]\!](M', s')$ equals the number of states (M'', s'') with $(M, s)[\![L_B\alpha_\cup]\!](M'', s'')$—namely, each s'' in the latter corresponds to a (M', s') in the former. Note also that $(L_B\alpha)_\cup = L_B\alpha_\cup$. Therefore the domain an epistemic state resulting from executing $L_BL_B\alpha$ equals the domain of such an epistemic state resulting from executing $L_B\alpha$. Obviously, the valuation does not change either.

In the domain of the first, two factual states are the same for agent a ($\sim_a$) if their s, t origins in M are the same for that agent *and* if they are the same as epistemic states. But these epistemic states in the domain (of an epistemic state resulting from executing $L_BL_B\alpha$), that are the results of executing actions of type $L_B\alpha_\cup$, are the same for a if *their* points (resulting from executions of $\alpha_\cup$) are the same for that agent, i.e., if they are the same for a in the domain of the epistemic state resulting from executing $L_B\alpha$. $\Box$

Exercise 5.19 Prove the first two items of Proposition 5.18. $\Box$

The two main theorems of interest are that bisimilarity of epistemic states implies their modal equivalence, and that action execution preserves bisimilarity of epistemic states. We prove them *together* by simultaneous induction.

Theorem 5.20 (Bisimilarity implies modal equivalence)
Let $\varphi \in \mathcal{L}_!^{\text{stat}}(A)$. Let $(M, s), (M', s') \in \bullet\mathcal{S}5(A)$. If $(M, s) \leftrightarrows (M, s')$, then $(M, s) \models \varphi$ iff $(M', s') \models \varphi$. $\Box$

Proof By induction on the structure of φ. The proof is standard, except for the clause $\varphi = [\alpha]\psi$ that we therefore present in detail.

Assume $(M, s) \models [\alpha]\psi$. We have to prove $(M', s') \models [\alpha]\psi$. Let (N', t') be arbitrary such that $(M', s')[\![\alpha]\!](N', t')$. By simultaneous induction hypothesis (Theorem 5.21) it follows from $(M', s')[\![\alpha]\!](N', t')$ and $(M, s) \leftrightarrows (M', s')$ that there is a (N, t) such that $(N, t) \leftrightarrows (N', t')$ and $(M, s)[\![\alpha]\!](N, t)$. From $(M, s)[\![\alpha]\!](N, t)$ and $(M, s) \models [\alpha]\psi$ (given) follows that $(N, t) \models \psi$. From $(N, t) \leftrightarrows (N', t')$ and $(N, t) \models \psi$ it follows that $(N', t') \models \psi$. From the last and $(M', s')[\![\alpha]\!](N', t')$ it follows that $(M', s') \models [\alpha]\psi$. $\Box$

Theorem 5.21 (Action execution preserves bisimilarity)
Let $\alpha \in \mathcal{L}_!^{\text{act}}(A)$ and $(M, s), (M', s') \in \bullet\mathcal{S}5(A)$. If $(M, s) \leftrightarrows (M', s')$ and there is a $(N, t) \in \bullet\mathcal{S}5(\subseteq A)$ such that $(M, s)[\![\alpha]\!](N, t)$, then there is a $(N', t') \in \bullet\mathcal{S}5(\subseteq A)$ such that $(M', s')[\![\alpha]\!](N', t')$ and $(N, t) \leftrightarrows (N', t')$. $\Box$

Proof By induction on the structure of α, or, more accurately, by induction on the *complexity* of α defined as: α is more complex than any of its structural parts and $L_B\alpha$ is more complex than $\alpha_\cup$.

Case $?\varphi$: Suppose $\mathfrak{R} : (M, s) \leftrightarrows (M', s')$. By simultaneous induction (Theorem 5.20) it follows from $(M, s) \leftrightarrows (M', s')$ and $(M, s) \models \varphi$ that $(M', s') \models \varphi$. Define, for all $t \in (M, s)[\![?\varphi]\!]$, $u \in (M', s')[\![?\varphi]\!]$: $\mathfrak{R}^{?\varphi}(t, u)$ iff $\mathfrak{R}(t, u)$. Then $\mathfrak{R}^{?\varphi} : (M, s)[\![?\varphi]\!] \leftrightarrows (M', s')[\![?\varphi]\!]$, because (Points:) $\mathfrak{R}^{?\varphi}(s, s')$, (Back and Forth:) both epistemic states have empty access, and (Valuation:) $\mathfrak{R}^{?\varphi}(t, u)$ implies $\mathfrak{R}(t, u)$. In other words: $(M', s')[\![?\varphi]\!]$ is the required (N', t').

Case $L_B\alpha$: Suppose $\mathfrak{R} : (M,s) \leftrightarrows (M',s')$ and $(M,s)[\![L_B\alpha]\!](N,t)$. Let $(N'',t'') \in (N,t)$ be arbitrary (i.e., the former is an epistemic state that is a factual state in the domain of the latter). Then there is a $u \in (M,s)$ such that $(M,u)[\![\alpha_\cup]\!](N'',t'')$. Because $u \in (M,s)$ and $\mathfrak{R} : (M,s) \leftrightarrows (M',s')$, there is a $u' \in (M',s')$ such that $\mathfrak{R}(u,u')$ and obviously we also have that $\mathfrak{R} : (M,u) \leftrightarrows (M',u')$ (the domain of an epistemic state is the domain of its underlying model). By induction, using that the complexity of $\alpha_\cup$ is smaller than that of $L_B\alpha$, there is a (N''',t''') such that $(M',u')[\![\alpha_\cup]\!](N''',t''')$ and $(N'',t'') \leftrightarrows (N''',t''')$.

Now define (N',t') as follows: its domain consists of worlds (N''',t''') constructed according to the procedure just outlined; accessibility between such worlds is accessibility between those worlds as sets of epistemic states, and valuation corresponds to those in the bisimilar worlds of (N,t). Finally, the point of (N',t') is the result of executing α in (M',s') that is bisimilar to the point of (N,t). The accessibility on (N',t') corresponds to that on (N,t): if $(N_1,t_1) \sim_a (N_2,t_2)$, $(N_1,t_1) \leftrightarrows (N_1',t_1')$, and $(N_2,t_2) \leftrightarrows (N_2',t_2')$, then $(N_1',t_1') \sim_a (N_1',t_1')$. Therefore $(M',s')[\![L_B\alpha]\!](N',t')$ and $(N,t) \leftrightarrows (N',t')$.

Case $\alpha\ ;\ \beta$: Suppose $(M,s) \leftrightarrows (M',s')$ and $(M,s)[\![\alpha\ ;\ \beta]\!](N,t)$. Note that $[\![\alpha\ ;\ \beta]\!] = [\![\alpha]\!] \circ [\![\beta]\!]$. Let (N_1,t_1) be such that $(M,s)[\![\alpha]\!](N_1,t_1)$ and that $(N_1, t_1)[\![\beta]\!](N,t)$. By induction we have a (N_1',t_1') such that $(M',s')[\![\alpha]\!](N_1',t_1')$ and $(N_1,t_1) \leftrightarrows (N_1',t_1')$. Again, by induction, we have a (N',t') such that $(N_1',t_1')[\![\beta]\!](N',t')$ and $(N,t) \leftrightarrows (N',t')$. But then also $(M',s')[\![\alpha\ ;\ \beta]\!](N',t')$.

Case $\alpha \cup \beta$: Suppose $(M,s) \leftrightarrows (M',s')$ and $(M,s)[\![\alpha \cup \beta]\!](N,t)$. Then either $(M,s)[\![\alpha]\!](N,t)$ or $(M,s)[\![\beta]\!](N,t)$. If $(M,s)[\![\alpha]\!](N,t)$, then by induction there is a (N',t') such that $(M',s')[\![\alpha]\!](N',t')$ and $(N,t) \leftrightarrows (N',t')$. Therefore, also $(M',s')[\![\alpha \cup \beta]\!](N',t')$. Similarly if $(M,s)[\![\beta]\!](N,t)$.

Cases $\alpha\ !\ \beta$ and α ¡ β are similar to $\alpha \cup \beta$ but even simpler. □

5.3.4 Deterministic and Non-deterministic Actions

Actions may have more than one execution in a given epistemic state. Although the action mayread only has a single execution in a given epistemic state, the type $L_{ab}(L_a?p \cup L_a?\neg p \cup\ ?\top)$ of mayread always has *two* executions. When p is true, Anne can either read the letter, or not. In the first case the corresponding action instance is $L_{ab}(!L_a?p \cup L_a?\neg p \cup\ ?\top)$, in the other case it is $L_{ab}(L_a?p \cup L_a?\neg p \cup\ !?\top)$, i.e., mayread itself. The only source of non-determinism in the language is the non-deterministic action operator $\cup$. It turns out that this operator is superfluous in a rather strong sense: the language with $\cup$ is just as expressive as the language without; for details, see Proposition 8.56 in Chapter 8 on expressivity. This is useful, because properties of the logic and proofs in the logic may be more conveniently formulated or proved in the language without $\cup$. In this subsection we prove some elementary properties of deterministic and non-deterministic actions. A trivial observation is that

Proposition 5.22 Deterministic actions have a (partial) functional interpretation. $\square$

For non-deterministic actions we have the following validity:

Proposition 5.23 Valid is $[\alpha \cup \alpha']\varphi \leftrightarrow ([\alpha]\varphi \wedge [\alpha']\varphi)$. $\square$

A conceptually more intuitive formulation of the above is $\langle \alpha \cup \beta \rangle \varphi \leftrightarrow (\langle \alpha \rangle \varphi \vee \langle \beta \rangle \varphi)$. The proof of Proposition 5.23 is obvious when seen in this dual form, as $[\![\alpha \cup \beta]\!] = [\![\alpha]\!] \cup [\![\beta]\!]$. A stronger but also fairly obvious result holds as well: The interpretation of an action is equal to non-deterministic choice between all its instances:

Proposition 5.24
Let $\alpha \in \mathcal{L}_!^{\mathrm{act}}$. Then $[\![\alpha]\!] = \bigcup_{\alpha_!} [\![\alpha_!]\!]$. $\square$

Proof Note that $\alpha_!$ means an arbitrary instance of α, so that $\bigcup_{\alpha_!}$ is the union for all instances of α. The proof is by induction on the structure of α. The two cases of interest are $\alpha \cup \alpha'$ and $L_B\alpha$:

Case $\alpha \cup \alpha'$:

By definition, $[\![\alpha \cup \alpha']\!]$ equals $[\![\alpha]\!] \cup [\![\alpha']\!]$. By applying the inductive hypothesis to α and α', $[\![\alpha]\!] \cup [\![\alpha']\!]$ equals $\bigcup_{\alpha_!} [\![\alpha_!]\!] \cup \bigcup_{\alpha'_!} [\![\alpha'_!]\!]$. Again by definition, this is equal to $[\![\bigcup_{\alpha_!} \alpha_!]\!] \cup [\![\bigcup_{\alpha'_!} \alpha'_!]\!]$, which is equal to $[\![\bigcup_{\alpha_!} \alpha_! \cup \bigcup_{\alpha'_!} \alpha'_!]\!]$, which is $[\![\bigcup_{(\alpha \cup \alpha)_!} (\alpha \cup \alpha')_!]\!]$. The last equals $\bigcup_{(\alpha \cup \alpha)_!} [\![(\alpha \cup \alpha')_!]\!]$.

Case $L_B\alpha$:

Let M and $s \in \mathcal{D}(M)$ be arbitrary. Suppose $(M, s)[\![L_B\alpha]\!](M', s')$ and $M', s' \models \varphi$. By definition (of the semantics of actions) the point s' of (M', s') is an epistemic state such that $(M, s)[\![\alpha]\!]s'$. By induction, there must be a instance $\alpha_!$ of α such that $(M, s)[\![\alpha_!]\!]s'$. By the semantics of 'learning', we now immediately have $(M, s)[\![L_B\alpha_!]\!](M', s')$ for some instance $L_B\alpha_!$ of $L_B\alpha$.

The other direction is trivial. $\square$

Proposition 5.24 can also be expressed as the validity

$$\langle \alpha \rangle \varphi \leftrightarrow \bigvee_{\alpha_!} \langle \alpha_! \rangle \varphi \ ,$$

or its dual form

$$[\alpha]\varphi \leftrightarrow \bigwedge_{\alpha_!} [\alpha_!]\varphi \ .$$

Execution of deterministic actions of group B results in epistemic states of group B. This validates the overload of the *gr* operator for both actions and structures:

Proposition 5.25 Given $M \in \mathcal{S}5(A)$, $s \in \mathcal{D}(M)$, and $\alpha \in \mathcal{L}_!^{\mathrm{act}}(A)$. Suppose α is deterministic and executable in (M, s). Then $gr((M, s)[\![\alpha]\!]) = gr(\alpha)$. □

Proof By induction on action structure.

Case '$?\varphi$': By definition of gr, $gr((M, s)[\![?\varphi]\!]) = \emptyset = gr(?\varphi)$.

Case 'L': By definition of gr, $gr((M, s)[\![L_B\alpha]\!]) = B = gr(L_B\alpha)$.

Case ';': By definition of action semantics, $(M, s)[\![\alpha \ ; \ \beta]\!] = ((M, s)[\![\alpha]\!])[\![\beta]\!]$. By definition of gr, $gr(\alpha \ ; \ \beta) = gr(\beta)$. By inductive hypothesis, we have that $gr(((M, s)[\![\alpha]\!])[\![\beta]\!]) = gr(\beta)$.

Case '!': By definition of gr, $gr((M, s)[\![\alpha \ ! \ \beta]\!]) = gr((M, s)[\![\alpha]\!])$. By inductive hypothesis, $gr((M, s)[\![\alpha]\!]) = gr(\alpha)$, and also by definition of gr, the last equals $gr(\alpha \ ! \ \beta)$.

Case '¡' is as case '!'. □

Exercise 5.26 Show that Proposition 5.25 does not hold for all $\mathcal{L}_!^{\mathrm{act}}$ actions. □

For deterministic actions α, Theorem 5.21 can be formulated more succinctly:

Corollary 5.27 Let $(M, s), (M', s') \in \bullet\mathcal{S}5(A)$, and let $\alpha \in \mathcal{L}_!^{\mathrm{act}}(A)$ be a deterministic action that is executable in (M, s). If $(M, s) \leftrightarroweq (M', s')$, then $(M, s)[\![\alpha]\!] \leftrightarroweq (M', s')[\![\alpha]\!]$. □

5.3.5 Valid Properties of the Logic

This subsection lists some relevant validities of the logic EA. We remind the reader that such validities are candidate axioms for a proof system for EA, but that we do not give (nor have) a complete axiomatisation. The matter will be addressed at the end of this subsection.

Public announcements can only be executed when true. Similarly, more complex epistemic actions are sometimes executable, and sometimes not. A direct way to express that an epistemic action α is executable in an epistemic state (M, s) is to require that $M, s \models \langle\alpha\rangle\top$: this expresses that *some* epistemic state can be reached. Alternatively, one can express the condition of executability as the *precondition* of an epistemic action. In the case of a public announcement, the precondition is the announcement formula. For arbitrary epistemic actions, the notion is somewhat more complex but fairly straightforward.

Definition 5.28 (Precondition) The precondition $\mathsf{pre} : \mathcal{L}_!^{\mathrm{act}} \to \mathcal{L}_!^{\mathrm{stat}}$ of an epistemic action is inductively defined as

$$\begin{aligned}
\mathsf{pre}(?\varphi) &= \varphi \\
\mathsf{pre}(\alpha \ ; \ \beta) &= \mathsf{pre}(\alpha) \land \langle\alpha\rangle\mathsf{pre}(\beta) \\
\mathsf{pre}(\alpha \cup \beta) &= \mathsf{pre}(\alpha) \lor \mathsf{pre}(\beta) \\
\mathsf{pre}(\alpha \ ! \ \beta) &= \mathsf{pre}(\alpha) \\
\mathsf{pre}(L_B\alpha) &= \mathsf{pre}(\alpha)
\end{aligned}$$

□

One can now prove that $\models \mathsf{pre}(\alpha) \leftrightarrow \langle\alpha\rangle\top$. The notion of precondition will make it easier to compare the properties in the following proposition with the axioms in the proof system for the action model logic in Chapter 6.

Exercise 5.29 Show that $\models \mathsf{pre}(\alpha) \leftrightarrow \langle\alpha\rangle\top$. □

Proposition 5.30 All of the following are valid:

- $[?\varphi]\psi \leftrightarrow (\varphi \to \psi)$
- $[\alpha\ ;\ \alpha']\varphi \leftrightarrow [\alpha][\alpha']\varphi$
- $[\alpha \cup \alpha']\varphi \leftrightarrow ([\alpha]\varphi \wedge [\alpha']\varphi)$
- $[\alpha\ !\ \alpha']\varphi \leftrightarrow [\alpha]\varphi$
- $[\alpha]p \leftrightarrow (\mathsf{pre}(\alpha) \to p)$
- $[\alpha]\neg\varphi \leftrightarrow (\mathsf{pre}(\alpha) \to \neg[\alpha]\varphi)$ □

Proof We prove two, one has been proved as Proposition 5.23, and the remaining are left to the reader.

$\models [?\varphi]\psi \leftrightarrow (\varphi \to \psi)$

Note that in $[?\varphi]\psi \leftrightarrow (\varphi \to \psi)$, ψ is a propositional formula ($\psi \in \mathcal{L}_!^{\text{stat}}(\emptyset, P)$), because $gr(?\varphi) = \emptyset$. (In Definition 5.6 of the syntax of actions and formulas, see the restriction on ψ in the clause $[\alpha]\psi$.) The truth of propositional formulas is unaffected by action execution.

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

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

$\models [\alpha]p \leftrightarrow (\mathsf{pre}(\alpha) \to p)$

Suppose $M, s \models [\alpha]p$, and assume that $M, s \models \mathsf{pre}(\alpha)$. Let M', s' be such that $(M, s)[\![\alpha]\!](M', s')$. From $M, s \models [\alpha]p$ and $(M, s)[\![\alpha]\!](M', s')$ follows $M', s' \models p$. As epistemic actions do not change the valuation of atoms, $M, s \models p$.

For the converse direction, note that $M, s \not\models \mathsf{pre}(\alpha)$ trivially implies $M, s \models [\alpha]p$. Else, $M, s \models \mathsf{pre}(\alpha)$ implies $M, s \models \langle\alpha\rangle\top$—and whenever α can be executed, atoms do not change their value. □

Exercise 5.31 Prove the remaining cases of Proposition 5.30. □

This brings us to the formulation of a principle relating actions and knowledge. For public announcements the principle was $[\varphi]K_a\psi \leftrightarrow (\varphi \to K_a[\varphi]\psi)$. This principle no longer holds when we replace an announcement φ by an arbitrary action α, because not all agents may have full access to the postconditions of the action α! A typical example is the action read where Anne reads the letter containing p while Bill watches her: $L_{ab}(!L_a?p \cup L_a?\neg p)$.

After execution of this action Anne knows that p is true: $K_a p$. As read is a deterministic action, this is therefore true after *every* execution of the action: $[\mathsf{read}]K_a p$. And Bill knows that too: he considers it possible that p is false and that p is true; in the first case $[\mathsf{read}]K_a p$ is trivially true, in the second case it is true by the argument above. Therefore Bill *knows* that $[\mathsf{read}]K_a p$ is true, in other words $K_b[\mathsf{read}]K_a p$ is true. But on the other hand, it is not the case that after read's execution Bill knows that Anne knows p: $[\mathsf{read}]K_b K_a p$ is false. So, $[\alpha]K_a\psi \leftrightarrow (\mathsf{pre}(\alpha) \rightarrow K_a[\alpha]\psi)$ does not hold for arbitrary epistemic actions.

We need a generalisation of this principle that takes into account that some agents may not know what action is actually taking place. Consider again the action read. Bill cannot distinguish that action from the action $L_{ab}(L_a?p\cup\,!L_a?\neg p)$ where Anne learns $\neg p$ instead. Whatever Bill *knows* after the action read, should therefore also be true if in fact the other action had taken place. In case of these two actions there is no 'interesting' postcondition. But suppose the actions had been $\mathsf{this} = L_{ab}(!L_a?(q \wedge p) \cup L_a?(q \wedge \neg p))$ and $\mathsf{that} = L_{ab}(L_a?(q \wedge p)\cup\,!L_a?(q \wedge \neg p))$ instead. After both actions q is true. In other words: q is true after this, but also after any other action that Bill cannot distinguish from this, namely that. He therefore *knows* that after executing either action, q is true: $K_b[\mathsf{this}]q$ and $K_b[\mathsf{that}]q$, which we temptingly write as $\bigwedge_{\beta\sim_b\mathsf{this}} K_b[\beta]q$. This is sufficient to conclude that $[\mathsf{this}]K_b q$. The general principle on 'actions and knowledge' should then be

$$[\alpha]K_a\varphi \leftrightarrow (\mathsf{pre}(\alpha) \rightarrow \bigwedge_{\beta\sim_a\alpha} K_a[\beta]\varphi)$$

Unfortunately, we do not know of a *general* notion of the syntactic action accessibility $\beta \sim_a \alpha$. A similar problem occurs for a principle relating actions and common knowledge. In Chapter 6 another language for epistemic actions is introduced. In that logic the notion of accessibility between actions is a primitive. The principle relating actions and knowledge is then indeed precisely the principle that we describe above. For that logic we provide an axiomatisation.

We hope that the attention that we have given to the logic EA in this textbook, is sufficiently validated because it is (or at least, it seems to us) a convenient and flexible specification language for multi-agent system dynamics, properly backed up by a formal semantics—even though we have not provided a complete axiomatisation. We conclude this chapter with two in-depth investigations of such multi-agent systems.

5.4 Card Game Actions

We describe epistemic states involving players holding cards and exchanging information about their cards and their knowledge, including what they know about other players. First we model the actions of players picking up dealt cards. Then we model various actions in the situation where three players each know their own card. This is the setting of Example 4.2 in Chapter 4. For one of these actions, namely, the show action wherein Anne shows her card to Bill, we once more perform the semantic computation in great detail, to illustrate the semantics of actions. This subsection is followed by a short subsection on 'knowledge games' such as Cluedo. Finally comes yet another cards setting, now involving only two players.

5.4.1 Dealing and Picking Up Cards

Suppose there are three players Anne, Bill and Cath (a, b, c) and three cards 0, 1, and 2. Proposition 0_a expresses that Anne holds card 0, as before, etc. There are six possible deals of three cards over three players. This time we consider the model where the cards have been dealt but where the players have not picked up their card yet. A player's card is simply in front of that player on the table, facedown. Therefore, none of the three players can distinguish any deal from any other deal: their access on the domain of six deals is simply the universal relation; all deals are the same to them. This is the top-left model in Figure 5.4. Suppose the actual deal is 012 (Anne holds 0, Bill holds 1, and Cath holds 2). In this 'initial epistemic state' the following actions take place:

Example 5.32

- pickup_a: Anne picks up her card and looks at it. It is card 0.
- pickup_b: Bill picks up his card and looks at it. It is card 1.
- pickup_c: Cath picks up her card and looks at it. It is card 2. □

When Anne picks up her card, neither Bill nor Cath know which card that is. Publicly is only known that it must be one of the three possible cards 0, 1, 2. Therefore, the action needs a description where all agents learn that Anne learns one of three alternatives, and where the actual alternative is that she picks up card 0. Of course, Bill's and Cath's subsequent actions are quite similar. The descriptions in $\mathcal{L}_!^{\text{act}}$ of these actions are, therefore:

Example 5.33

- $\mathsf{pickup}_a = L_{abc}(!L_a?0_a \cup L_a?1_a \cup L_a?2_a)$
- $\mathsf{pickup}_b = L_{abc}(L_b?0_b \cup\, !L_b?1_b \cup L_b?2_b)$
- $\mathsf{pickup}_c = L_{abc}(L_c?0_c \cup L_c?1_c \cup\, !L_c?2_c)$ □

The resulting models are visualised in Figure 5.4. The final model is, of course, the model *Hexa* where each player only knows his own card.

5.4.2 Game Actions in Hexa

We proceed from the epistemic state $(\mathit{Hexa}, 012)$ where each player only knows his own card, and where the actual deal of cards is that Anne holds 0, Bill holds

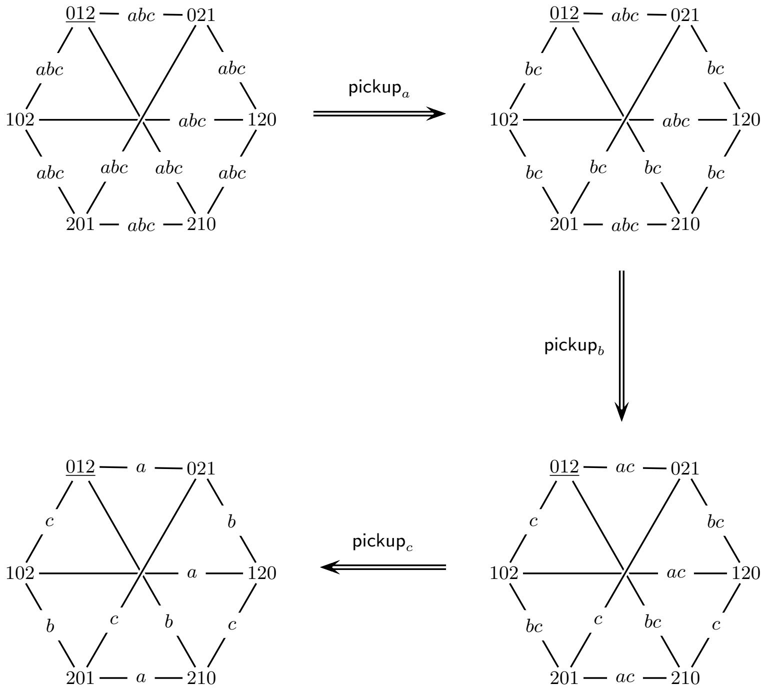

Figure 5.4. Three cards have been dealt over three players. The actual deal is 012: Anne holds 0, Bill holds 1, and Cath holds 2. This is pictured in the top-left corner. Anne now picks up her card (0). Then Bill picks up his card (1). Finally, Cath picks up her card (2). These three transitions are pictured. In fact, in all but the last model there are more links between states in the visualisation than strictly necessary—but the pictured transitions become more elegant that way.

1, and Cath holds 2. That each player only knows his own card induces for each player an equivalence relation on the domain.[3] We can imagine various actions to take place:

Example 5.34 (table) Anne puts card 0 (face up) on the table. □

Example 5.35 (show) Anne shows (only) Bill card 0. Cath cannot see the face of the shown card, but notices that a card is being shown. □

Example 5.36 (whisper) Bill asks Anne to tell him a card that she (Anne) does not have. Anne whispers in Bill's ear "I do not have card 2." Cath notices that the question is answered, but cannot hear the answer. □

[3] We can also see the model as an interpreted system for three agents, where each agent only knows his own local state, where each agent has three local state values—and where additionally there is some interdependence between global states, namely, that the value of your local state cannot be the value of the local state of another agent.

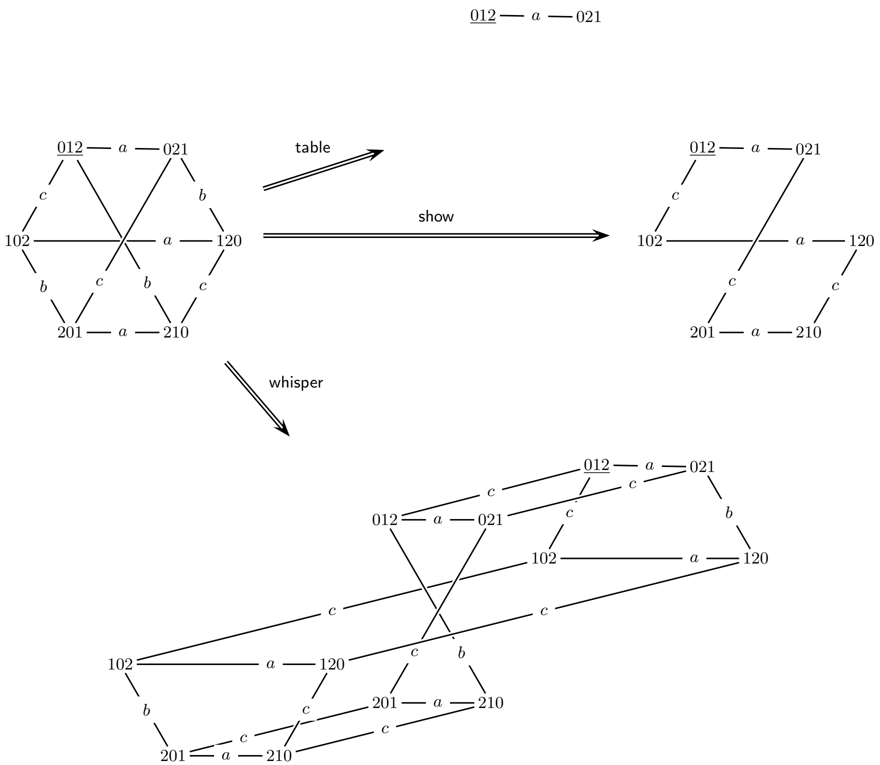

Figure 5.5. The results of executing table, show, and whisper in the state $(Hexa, 012)$ where Anne holds 0, Bill holds 1, and Cath holds 2. The points of the states are underlined. States are named by the deals that characterise them. Assume reflexivity and transitivity of access. The structures resulting from show and whisper are explained in great detail in the text.

We assume that only. the truth is told. In show and whisper, we assume that it is publicly known what Cath can and cannot see or hear.

Figure 5.5 pictures. $(Hexa, 012)$ and the epistemic states that result from executing the three actions. The action table is yet another appearance of a 'public announcement' and it therefore suffices to eliminate the states from the domain where Anne does not hold card 0. We can restrict access and valuation correspondingly, as it is publicly known that eliminated deals are no longer accessible. In show we cannot eliminate any state. After this action, e.g., Anne can imagine that Cath can imagine that Anne has shown card 0, but also that Anne has shown card 1, or card 2. However, some *links* between states have now been severed: whatever the actual deal of cards, Bill cannot consider any other deal after the execution of show. In whisper Anne can *choose* whether to whisper "not 1" or "not 2". The resulting epistemic state has therefore twice as many states as the current epistemic state. This is because for each deal of cards there are now two possible actions that can be executed.

We can paraphrase some more of the structure of the actions. In table, all three players learn that Anne holds card 0, where 'learning' is the dynamic equivalent of 'common knowledge'. Note that there is also a slight but interesting difference between Anne publicly *showing* card 0 to the other players and Anne *saying* that she holds card 0: the first is obviously the public announcement of 0_a whereas the second is more properly the public announcement of $K_a 0_a$, even though we have typically also described that as 0_a.

In the show action, Anne and Bill learn that Anne holds 0, whereas the group consisting of Anne, Bill and Cath learns that Anne and Bill learn which card Anne holds, or, in other words: that either Anne and Bill learn that Anne holds 0, or that Anne and Bill learn that Anne holds 1, or that Anne and Bill learn that Anne holds 2. The choice made by subgroup $\{a, b\}$ from the three alternatives is *local*, i.e., known to them only, because it is hidden from Cath. This is expressed by the 'local choice' operators '!' and '¡'. In fact, Cath knows that Anne can only possibly show card 0 or card 1, and not card 2, as Cath holds card 2 herself. But what counts is that this is not *publicly* known: Bill does not know (before the action takes place) that Cath knows that Anne cannot show card 2. The paraphrase describes the *publicly known* alternatives, and therefore all three.

The whisper action is paraphrased quite similar to the show action, namely as: "Anne and Bill learn that Anne does not hold card 2, and Anne, Bill, and Cath learn that Anne and Bill learn that Anne does not hold card 0, or that Anne and Bill learn that Anne does not hold card 1, or that Anne and Bill learn that Anne does not hold card 2". In the description of this action we also need the local choice operator.

In case of confusion when modelling an action: always first describe the type of an action, and only then the specific instance you want. From the description 'Anne whispers a card that she does not hold into Bill's ear' it is more immediately clear that no specific card can be excluded.

Example 5.37 The description of the actions table, show, and whisper in $\mathcal{L}_!(\{a, b, c\}, \{0_a, 1_a, 2_a, 0_b, \dots\})$ is:

$$\begin{aligned} \mathsf{table} &= L_{abc}?0_a \\ \mathsf{show} &= L_{abc}(!L_{ab}?0_a \cup L_{ab}?1_a \cup L_{ab}?2_a) \\ \mathsf{whisper} &= L_{abc}(L_{ab}?\neg 0_a \cup L_{ab}?\neg 1_a \cup\, !L_{ab}?\neg 2_a) \end{aligned}$$ □

As we assume associativity of $\cup$ in these descriptions, and as local choice between alternatives that have been ruled out is irrelevant, the description $L_{abc}(!L_{ab}?0_a \cup L_{ab}?1_a \cup L_{ab}?2_a)$ is formally one of

$$\begin{gathered} L_{abc}(L_{ab}?0_a \;!\; (L_{ab}?1_a \;!\; L_{ab}?2_a)) \\ L_{abc}(L_{ab}?0_a \;!\; (L_{ab}?1_a \;¡\; L_{ab}?2_a)) \\ L_{abc}((L_{ab}?0_a \;!\; L_{ab}?1_a) \;!\; L_{ab}?2_a) \end{gathered}$$

Example 5.38 The types $\mathsf{table}_\cup$, $\mathsf{show}_\cup$, and $\mathsf{whisper}_\cup$ of these actions are

$$\begin{array}{ll}\mathsf{table}_\cup & = L_{abc}?0_a \\ \mathsf{show}_\cup & = L_{abc}(L_{ab}?0_a \cup L_{ab}?1_a \cup L_{ab}?2_a) \\ \mathsf{whisper}_\cup & = L_{abc}(L_{ab}?\neg 0_a \cup L_{ab}?\neg 1_a \cup L_{ab}?\neg 2_a)\end{array}$$ □

Apparently, there are three actions of type $\mathsf{show}_\cup$ and three actions of type $\mathsf{whisper}_\cup$, representing Anne showing card 0, card 1, and card 2; and Anne whispering that she does not hold card 0, card 1, and card 2, respectively. There are four actions of that type in the more formal description, as the two actions of form $(x \,!\, (y \,!\, z))$ and $(x \,!\, (y \,¡\, z))$ are indistinguishable when written as $(!x \cup y \cup z)$.

Note that the table action $L_{abc}?0_a$ is, obviously, the same as its type. But instead we could have chosen to model that action as $(!L_{abc}?0_a \cup L_{abc}?1_a \cup L_{abc}?2_a)$, in which case its type would have been $(L_{abc}?0_a \cup L_{abc}?1_a \cup L_{abc}?2_a)$. That more properly describes the action "Anne puts her card face up on the table (whatever the card is)".

We continue with details on how to compute the interpretation of these card show actions in the model *Hexa*. To compute the interpretation of show is requested in Exercise 5.39, following immediately below. It is also instructive to actually compute the interpretation of table and see the semantics of public announcements reappear, even though the semantic computations are different. Observations on the computation of $\mathsf{whisper}$ follow after Exercise 5.39.

Exercise 5.39 Compute the interpretation of the show action in $(Hexa, 012)$ in detail. This exercise has a detailed answer. We recommend the reader to pay careful attention to this exercise. □

In the case of the action $\mathsf{whisper}$ $(L_{abc}(L_{ab}?\neg 0_a \cup L_{ab}?\neg 1_a \cup\, !L_{ab}?\neg 2_a))$, where Anne whispers into Bill's ear that she does not have card 2, Cath does not know what Anne has whispered, and should therefore not be able to distinguish in the resulting epistemic state $(Hexa, 012)[\![\mathsf{whisper}]\!]$ the state resulting from whispering "not 0" from the states resulting from "not 1" and "not 2". This is indeed the case. Consider access in $(Hexa, 012)[\![\mathsf{whisper}]\!]$ in Figure 5.5 (page 129). As before, we have named the states by the deals characterising their valuations, to improve readability. The state 012 'in front' in the picture is the epistemic state $(Hexa, 012)[\![L_{ab}?\neg 1_a]\!]$ and the state 012 'at the back' (that is also the point of the resulting structure) is the epistemic state $(Hexa, 012)[\![L_{ab}?\neg 2_a]\!]$. They cannot be distinguished from one another by Cath, because she does not occur in the groups of either epistemic state (so that they are indistinguishable from one another as epistemic states), and because, obviously, $012 \sim_c 012$ in *Hexa*.

In the 'back 012', that corresponds to the answer 'not 2', Bill knows that Anne holds 0. In the 'front 012', that corresponds to the answer 'not 1', Bill still considers 210 to be an alternative, so Bill does not know the card of Anne.

In both the 'back' and the 'front' 012, neither Anne nor Cath know whether Bill knows Anne's card.

But not just the 'front 012' and 'back 012' are indistinguishable for Cath. She also cannot distinguish the 'back 012' from the 'back 102' and the 'back 102' from the 'front 102'. Because of transitivity she cannot distinguish between any of those four: {back 102, back 102, front 012, back 012} is one of her equivalence classes, namely, the one that corresponds to Cath holding card 2.

In different words, Cath considers it possible that Anne holds card 0 and told Bill that she does not hold card 2, but she also considers it possible that Anne holds card 1 and told Bill that she does not hold card 0 (even though Anne actually holds card 0).

5.4.3 Knowledge Games

Actions such as showing and telling other agents about your card(s), occur in many card games. Such games can therefore with reason be called *knowledge games*. Of particular interest are the card games where the *only* actions are epistemic actions. In that case, the goal of the game is to be the first to know (or guess rightly) the deal of cards, or a less specific property such as the whereabouts of specific cards. In *Hexa*, "Bill knows the deal of cards" can be described as $\mathsf{win}_b = K_b\delta^{012} \vee K_b\delta^{021} \vee \ldots$. Here δ^{ijk} is the atomic description of world (deal) ijk, e.g., $\delta^{012} = 0_a \wedge \neg 0_b \wedge \neg 0_c \wedge \neg 1_a \wedge 1_b \wedge \neg 1_c \wedge \neg 2_a \wedge \neg 2_b \wedge 2_c$. The action of Bill winning is therefore described as the public announcement of that knowledge: $L_{abc}?\mathsf{win}_b$.

If the goal of the game is to be the first to guess the deal of cards, and if players are perfectly rational, then ending one's move and passing to the next player also amounts to an action, namely, (publicly) announcing that you do not yet have enough knowledge to win. This action is described as, for the case of Bill, $L_{abc}?\neg\mathsf{win}_b$. In the epistemic state $(Hexa, 012)[\![\mathsf{whisper}]\!]$, Bill knows the card deal. But saying so is still informative for the other players. For example, before Bill said so, Cath still considered it possible that Bill did not know the card deal. If Anne had whispered 'I do not have card 1' instead of 'I do not have card 2', indeed Bill would not have learnt the card deal. Implicitly 'moving on' in that game state amounts to such an implicit declaration $L_{abc}?\neg\mathsf{win}_b$ of not being able to win.

This talk about winning and losing makes more sense for a 'real' knowledge game. The 'murder detection game' Cluedo is an example. The game consists of 21 cards and is played by six players. There is also a game board to play with, but a fair and already most interesting abstraction of Cluedo is to model it as a card game only. Each player has three cards and there are three cards on the table. The first player to guess those cards wins the game. The following actions are possible in Cluedo (and *only* those actions): showing (only to the requesting player) one of three requested cards (of different types, namely, a murder suspect card, a weapon card, and a room card), confirming that you do not hold any of three requested cards (by public announcement), 'ending your move', i.e., announcing that you cannot win, and 'ending the game', i.e., correctly guessing the murder cards. Because each player now holds three cards, the action of showing a card may now involve real choice, such as we have already seen in the case of the somewhat artificial whisper action (*not* legal in Cluedo!). A play of the game Cluedo can therefore be seen as a sequence of different game actions that can all be described as epistemic actions in $\mathcal{L}_!^{\mathrm{act}}$, so in that sense it is represented by a single $\mathcal{L}_!^{\mathrm{act}}$ action.

5.4.4 Different Cards

Two players a, b (Anne, Bill) face three cards p, q, r lying face-down in two stacks on the table. Let p^2 be the atom describing 'card p is in the stack with two cards', so that $\neg p^2$ stands for that 'card p is the single-card stack'. Consider the following two actions:

Example 5.40

- independent
  Anne draws a card from the two-cards stack, looks at it, returns it, and then Bill draws a card from the two-cards stack and looks at it.
- dependent
  Anne draws a card from the two-cards stack, and then Bill takes the remaining card from that stack. They both look at their card. □

Action independent has nine different executions. Action dependent has only six different executions. It is more constrained, because the cards that Anne and Bill draw must be different in dependent, whereas they may be the same in independent.

Action independent is described by the sequence

$$L_{ab}(L_a?p^2 \cup L_a?q^2 \cup L_a?r^2) \; ; \; L_{ab}(L_b?p^2 \cup L_b?q^2 \cup L_b?r^2)$$

Action dependent can also be described as a sequence of two actions, in which case we have to express implicitly that the second card is different from the first. Because the card that Bill draws is different from the card that Anne has just drawn, Anne does not know which is the card that Bill draws. In other words, from the two cards on the stack, it must be the card that she does not know. We get:

$$\begin{array}{l} L_{ab}(L_a?p^2 \cup L_a?q^2 \cup L_a?r^2) \; ; \\ L_{ab}(L_b?(p^2 \wedge \neg K_a p^2) \cup L_b?(q^2 \wedge \neg K_a q^2) \cup L_b?(r^2 \wedge \neg K_a r^2)) \end{array}$$

For example, $L_b?(p^2 \wedge \neg K_a p^2)$ expresses that Bill only learns that card p is on the two-card stack when player a has not learnt that already.

5.5 Spreading Gossip

Example 5.41 Six friends each know a secret. They call each other. In each call they exchange all the secrets that they currently know of. How many calls are needed to spread all the news? □

We first present the solution of the riddle and related combinatorics, and after that we model it in dynamic epistemic logic.

It can be shown that for a number of n friends with $n \geq 4$, the minimum sufficient number of calls is $2n-4$. The correct answer for six friends is therefore 'eight calls'. A general procedure for communicating n secrets in $2n-4$ calls, for $n \geq 4$, is as follows:

Assume that both the secrets and the agents are numbered $1, 2, \ldots, n$—there is no reason to distinguish agents from secrets, just as we previously distinguished players from cards. Let ab mean 'agent a calls agent b and they tell each other all their secrets'. For $n = 4$, a sequence of length $2 \cdot 4 - 4 = 4$ is $12, 34, 13, 24$. For $n > 4$ we proceed as follows. *First* make $n-4$ calls from agent 1 to the agents over 4. This is the call sequence $15, 16, ..., 1n$. *Then*, let agents 1 to 4 make calls as in the case of $n = 4$. That is the call sequence $12, 34, 13, 24$. *Finally*, repeat the first part of the procedure. So we close with another sequence $15, 16, ..., 1n$. The first part of the procedure makes the secrets of 5 to n known to 1. The second part of the procedure makes all secrets known to $1, 2, 3, 4$. The last part of the procedure makes all secrets known to the agents 5 to n.

For six agents, the resulting sequence is $15, 16, 12, 34, 13, 24, 15, 16$. There are also (non-trivially different) other ways to communicate all secrets in eight calls to all six agents. A different sequence is $12, 34, 56, 13, 45, 16, 24, 35$. Table 5.1 shows in detail how the secrets are spread over the agents by this other sequence. We name this call sequence six. Note that in the first, general, procedure some calls occur twice, for example calls 15 and 16 occur twice, whereas in six all calls are different.

call	*between*	1	2	3	4	5	6
1	12	12	12	3	4	5	6
2	34	12	12	34	34	5	6
3	56	12	12	34	34	56	56
4	13	1234	12	1234	34	56	56
5	45	1234	12	1234	3456	3456	56
6	16	123456	12	1234	3456	3456	123456
7	24	123456	123456	1234	123456	3456	123456
8	35	123456	123456	123456	123456	123456	123456

Table 5.1. The protocol six, a minimal sequence for communicating six secrets.

This curious interest for the *minimum* number of calls seems to be more fuelled by a bunch of academics aiming for efficient communication, than by a group of friends wishing to *prolong* the pleasure of such gossip as much as possible. What is the *maximum* number of calls where each time something new is learnt? (Subject to the 'rule' that all shared information is always exchanged.) For n secrets this is $\binom{n}{2}$. For $n = 6$ the maximum is obtained in the call sequence $12, 13, 14, 15, 16, 23, 24, 25, 26, 34, 35, 36, 45, 46, 56$. Fifteen all different calls! In general: agent 1 calls all other agents, agent 2 calls all other agents except agent 1, etc. $\Sigma_1^{n-1} = \frac{1}{2} \cdot n \cdot (n-1)$. The maximum number of *informative* calls between agents is therefore also the maximum number of *different* calls between 2 from n persons. This is not obvious, because not every sequence of informative calls consists of all different calls, and not every sequence of all different calls consists of all informative calls.

Exercise 5.42 Show that not every sequence of informative calls consists of all different calls, and not every sequence of all different calls consists of all informative calls. □

Thus far we have described this system from the viewpoint of an observer that is registering all calls. The telephone company, so to speak. If we describe it from the viewpoint of the callers themselves, i.e., as a multi-agent system, other aspects enter the arena as well. What sort of epistemic action is such a telephone call? Are the secrets generally or publicly known after the communications? Answers to such questions depend on further assumptions about the communications protocol and other 'background knowledge'. To simplify matters, we assume that all communication is faultless, that it is common knowledge that at the outset there are six friends each having one secret, and that always all secrets are exchanged in a call. If no further assumptions are made, after an effective call sequence (where 'effective' means: after which all secrets are exchanged) the secrets are generally known, but they are not commonly known. For example, after six, agent 3 knows that agent 5 knows all secrets, but (s)he does not know this for any other agent. For example, he has no reason to assume that the 35 call was the last in the sequence of eight calls, and that the 24 call had already taken place. Some knowledge of the protocol used to communicate the secrets *may* make a difference. In fact it is a bit unclear how much makes enough of a difference, and this might merit further investigation: e.g., note that on the one hand 35, 16, and 24 can be swapped arbitrarily while guaranteeing general knowledge of the secret, but that on the other hand the number of secrets known by 3 and 5 prior to the moment of the 35 call (four and four) is different from the numbers for 1 and 6, and 2 and 4 (four and two)—information that may reveal part of the executed protocol and therefore knowledge about other agents' information state.

If the agents know that a length 8 protocol is executed, *and* if time is synchronised, then the secrets are commonly known after execution of the protocol. But without either of these, it becomes problematic again. Suppose that time is synchronised but that the length of the protocol is unknown. By agreeing on a protocol 'keep calling until you have established that all other agents know all secrets'—this obviously includes non-informative calls—general knowledge of general knowledge of the secrets can be established, but again, this falls short of common knowledge.

Exercise 5.43 Give a protocol that achieves general knowledge of general knowledge of the secrets, for six agents. □

In the remainder we assume that everybody knows which calls have been made and to whom, but that the secrets that have been exchanged in a call are unknown. After that, the secrets are common knowledge at the completion of the protocol. A more realistic setting for this scenario is where the six agents are seated around a table, where all have a card with their secret written on it. A 'call' now corresponds to two persons showing each other their cards, and both adding the secrets that are new to them on their own card. The other players notice which two players show each other their cards, but not what is written on those cards.

Exercise 5.44 In the two example protocols, no agent learns the *source* of all secrets. But there are protocols where an agent does learn that. Give an example. Prove also that at most one agent knows the source of all secrets. (We also conjecture that after a minimal call sequence no agent knows the source of all secrets.) □

We now proceed to describe the call sequence six as a $\mathcal{L}_!$ epistemic action. We keep naming the six agents $1, 2, 3, 4, 5, 6$ but call the secrets (the value of six propositions) $p_1, \ldots, p_6$. The initial epistemic state is one in which each agent only knows 'his own' secret, in other words, it is an interpreted system where agents only know their own local state. The action "agent a and agent b learn each others' secrets" can be paraphrased as "all agents learn that agent a and agent b learn each others' secrets" and this can be further specified as "for each p_n (of all six atomic propositions) all agents learn that a and b learn whether a knows p_n, and all agents learn that a and b learn whether b knows p_n". Formally, the (type of this) action is call_{ab}, defined as the sequence

$$\mathsf{call}_{ab}(p_1) \ ; \ \ldots \ ; \ \mathsf{call}_{ab}(p_6)$$

where $\mathsf{call}_{ab}(p_n)$ is defined as:

$$\begin{array}{rl} \mathsf{call}_{ab}(p_n) = L_{123456}(& (L_{ab}?K_a p_n \cup L_{ab}?K_a \neg p_n \cup L_{ab}?\neg(K_a p_n \vee K_a \neg p_n)) \\ & ; \\ & (L_{ab}?K_b p_n \cup L_{ab}?K_b \neg p_n \cup L_{ab}?\neg(K_b p_n \vee K_b \neg p_n)) \\ &) \end{array}$$

It will now be clear that the type of action describing the six protocol is

$$\mathsf{call}_{12} \ ; \ \mathsf{call}_{34} \ ; \ \mathsf{call}_{56} \ ; \ \mathsf{call}_{13} \ ; \ \mathsf{call}_{45} \ ; \ \mathsf{call}_{16} \ ; \ \mathsf{call}_{24} \ ; \ \mathsf{call}_{35}$$

For a more concrete example, given the actual distribution of secrets where p_4 is false, in call call_{16} of six agent 6 learns from agent 1 that the value of the secret p_4 is 0. This happens in the part $\mathsf{call}_{16}(p_4)$ of call call_{16}, which is described as:

$$L_{123456}(L_{16}?K_1p_4 \cup !L_{16}?K_1\neg p_4 \cup L_{16}?\neg(K_1p_4 \cup K_1\neg p_4))\ ;\ \ldots$$

The exclamation mark, or local choice operator, points to the choice known by 1 and 6, but not (publicly known) by the remaining agents, as agent 2 does not yet know the value of p_4 at this stage.

5.6 Notes

Epistemic action logic The history of this logic is mainly the academic birth of van Ditmarsch. It achieved the goal to generalise the dynamic epistemic logic of public announcements by Plaza [168], Gerbrandy [75], and Baltag, Moss, and Solecki [11], which was described in detail in the Notes of Chapter 4. Van Ditmarsch' efforts to generalise the 'card show' action of Example 5.35 played a major part in the development of this framework. The relational action semantics of Section 5.3 found its way to the community in van Ditmarsch' publications [42, 43, 46]. Later developments on this logic (see below) include shared work with van der Hoek and Kooi.

Alternative syntax Alternatively to the primitives of the language $\mathcal{L}_!^{\mathrm{act}}$ one can stipulate a clause $L_B(\alpha, \alpha')$, meaning $L_B(\alpha\ !\ \alpha')$, and remove the clauses for local choice. The difference seems to be 'syntactic sugar' with some conceptual consequences: by doing this, the notion of the 'type of an action' disappears. Instead of modelling 'Anne reads the content of the letter in the presence of Bill' as an action type $L_{ab}(L_a?p \cup L_a?\neg p)$, with two instances $L_{ab}(!L_a?p \cup L_a?\neg p)$ and $L_{ab}(L_a?p \cup\ !L_a?\neg p)$, we would now have two deterministic actions $L_{ab}(L_a?p, L_a?\neg p)$ and $L_{ab}(L_a?\neg p, L_a?p)$, respectively, and the former action type $L_{ab}(L_a?p \cup L_a?\neg p)$ then corresponds to non-deterministic choice between those two: $L_{ab}(L_a?p, L_a?\neg p) \cup L_{ab}(L_a?\neg p, L_a?p)$. A similar approach to the language is followed, for a more general setting, by Economou in [56].

Concurrent epistemic action logic Concurrent epistemic action logic was proposed by van Ditmarsch in [45] and a proof system for this logic was proposed by van Ditmarsch, van der Hoek, and Kooi in [49]. The completeness proof in [49] was—in retrospect—based on a flawed notion of 'syntactic accessibility between actions'. We have chosen not to include a detailed treatment of this material. The treatment of concurrency for dynamic operators in [45, 49] is similar to that in the logic cPDL—for 'concurrent propositional dynamic logic'—proposed by Peleg [167] and also mentioned in, e.g., Goldblatt [79] and Harel *et al.* [93]. This treatment of concurrency is known as 'true concurrency': the result of executing an action $\alpha \cap \beta$ is the *set* of the results from executing just α and just β. For epistemic states, this means that execution of a concurrent action consisting of two parts each resulting in an epistemic state, results in a set of two epistemic states. The modelling solution is to see the corresponding state-transforming relation no longer as a relation between epistemic states, but as a relation between an epistemic state and a set of epistemic states.

Such 'true concurrency' is an alternative to another approach to concurrency, namely intersection concurrency. The dynamic logic IPDL (intersection PDL) is briefly presented in [93], for a detailed analysis see [8]. In that case, we take the intersection of the respective binary relations that are the interpretation of the two 'intersection-concurrent' actions. For an intersection concurrency sort of epistemic action, see the logic ALL in Kooi's Ph.D. thesis [114]. Versions of unpublished manuscripts by Baltag also contained that feature.

For an example of a concurrent action description in the setting of concurrent epistemic action logic in [45], consider again the action bothmayread, where both Anne and Bill may have read the letter (learnt the truth about p). It is described in this language as

$$\begin{aligned} L_{ab}(\ & (L_a?p \cap L_b?p) \cup (L_a?\neg p \cap L_b?\neg p) \\ & \cup L_a?p \cup L_a?\neg p \cup L_b?p \cup L_b?\neg p \cup\ ?\top\ ) \end{aligned}$$

Executing the part $(L_a?\neg p \cap L_b?\neg p)$ of this action indeed results in a set of two epistemic states. The effect of the L_{ab} operator binding this subaction and other subactions, results after all in the single epistemic state that we are already familiar with (see Figure 5.2). The *set of epistemic states* resulting from executing $(L_a?\neg p \cap L_b?\neg p)$ functions as a (single) *factual state* in the domain of the epistemic state resulting from executing the entire action. For details, see [45, 49].

Playing cards The results on modelling card games have previously appeared in publications by van Ditmarsch, and by Renardel de Lavalette [43, 44, 46, 174].

Spreading gossip The riddle on spreading gossip formed part of the 1999 Dutch Science Quiz. The original version (in Dutch), of which we presented a gender-neutral translation, was

"Zes vriendinnen hebben ieder 'n roddel. Ze bellen elkaar. In elk gesprek wisselen ze alle roddels uit die ze op dat moment kennen. Hoeveel gesprekken zijn er minimaal nodig om iedereen op de hoogte te brengen van alle zes de roddels?"

The answer options in the (multiple choice) quiz were 7, 8, and 9. The answer was, of course, 8. In the aftermath of that science quiz—partly reported in the Dutch media—academics generalised the answer. The given procedure for communicating n secrets (for $n \geq 4$) in $2n - 4$ calls was suggested by Renardel de Lavalette. Hurkens [108] proved (independently) that this is also the minimum. The observations on general and common knowledge and on how to model the spreading of gossip in epistemic logic are partly found in van Ditmarsch's Ph.D. thesis [43].