Unit 1

1 Basics: the modal approach to knowledge

Introduction

The textbook by Meyer and van der Hoek treats epistemic logics, that is logics about *knowledge*. The first person who wrote about epistemic logic was the Finnish philosopher G.H. von Wright in his book “An Essay in Modal Logic” (North-Holland, 1953). His treatment is completely axiomatic, with no mention of possible semantics. Most philosophical work on epistemic logic has concentrated on defending certain axioms and denouncing others.

In fact, the subject of epistemic logic only started to flourish after Kripke’s invention of a semantics for modal logic in the early sixties. Kripke introduced a *possible worlds semantics* for modal logics. The name “possible world” is somewhat misleading, because, in the words of Hintikka [Hin86], “applications to entire universes are scarcely found outside philosophers’ speculations. The primary intended applications are to scenarios covering relatively small pieces of space-time”. In the context of epistemic logic, one can view worlds that are possible for a certain agent i in world w as *epistemic alternatives*, worlds that are compatible with what agent i knows at w. The precise definitions will be given in Chapter 1. The first book about epistemic logic, J. Hintikka’s “Knowledge and Belief” (Cornell University Press, 1962), applies these semantical ideas to epistemic logic, although the definitions are not quite the same as the standard ones used today. As Hintikka writes in a paper of 1986, the semantics of epistemic logic presents much more interesting problems and solutions than the axiomatic side of the subject. You will find that in the textbook, semantical questions are indeed predominant.

Researchers have found applications of epistemic logic within fields as diverse as economy (where it is important in game theory and in negotiation to reason about what the other person knows and doesn’t know), Artificial Intelligence and computer science. Since the eighties, there has been more communication than before between the researchers from different fields using and studying epistemic logic. The most important conference on epistemic logic and related subjects is TARK (since 1996 standing for “Theoretical Aspects of Rationality and Knowledge”), held every other year since 1986.

Chapter 1 reintroduces the systems **K** and **S5** that the reader first met in an introduction to modal logic, but this time the systems are treated in the context of knowledge. Also, more than one agent is considered, and every agent may know different things and see different epistemic alternatives than her colleagues do. This luckily does not make the systems $\mathbf{K}_{(m)}$ and $\mathbf{S5}_{(m)}$ (the systems **K** and **S5** for m agents) much more complicated than their single agent counterparts, because interactions between agents are not axiomatized. Also a beginning is made to apply epistemic logic to distributed systems, a hot topic in Computer Science.

Study goals

After studying this unit you are supposed to be able to

- **syntax:** make axiomatic derivations in $\mathbf{K}_{(m)}$ and in $\mathbf{S5}_{(m)}$
- **semantics:** interpret epistemic formulae in Kripke models
- **theory:** understand soundness and completeness of $\mathbf{K}_{(m)}$ and $\mathbf{S5}_{(m)}$ with respect to the intended Kripke models
- **semantics:** understand why for $\mathbf{S5}_{(1)}$ reduced, simple models suffice of the systems
- **syntax:** rewrite an $\mathbf{S5}_{(1)}$ formula as an equivalent formula in normal form
- **modelling:** understand how distributed systems can be represented as Kripke models
- **modelling:** understand how epistemic logic may be applied to prove the correctness of communication protocols in distributed systems.

General recommendations for studying

Both this unit and unit 2 (about knowledge within a group) go with the textbook by W. Van Der Hoek en J.-J. Ch. Meyer, *Epistemic Logic for AI and Computer Science* (Cambridge Tracts in Theoretical Computer Science, No 41), Cambridge University Press, 1995, ISBN: 0-52146014-X. You may find a list of errata for the textbook at
http://www.ai.rug.nl/mas/index.php?topic=literatuur&subtopic=errata.

Standard exercises have no special labels. You are supposed to be able to make the standard exercises without external help. Exercises marked EXERCISE^0 are optional, but they are recommended if you experience difficulties with the standard exercises. Some exercises have hints; in that case they are marked EXERCISE^{hint}. With EXERCISE 1.11$^{1.4}$ we mean that exercise 1.4 is good additional training if you experience difficulties in making exercise 1.1, to be done before continuing exercise 1.1. If you like challenges, you may try exercises marked EXERCISE*

Recommendations for studying this unit

This unit requires about 36 hours of study. The following contains some further explanations of parts of the text, some historical and philosophical background about epistemic logic and some extra training material. The unit goes with Chapter 1 of the textbook by Meyer and Van der Hoek, but part of section 1.7 (pp. 30,31,32 up to just above Corollary 1.7.4.6) is skipped because it is replaced by a shorter proof contained in this study guide.

Main text

2.1 Epistemic formulas and their semantics

Read sections 1.1, 1.2, and 1.3 of the textbook.

notation K_i page 7 line 12

An operator K_i works the same as the modal necessity operator $\Box$ from the *Introduction to Modal Logic* (for RUG students: *Voortgezette Logica*). The K stands for knowledge, and the i for agent i. You may view agents as humans, or as computer processors in a distributed system, or as other objects, according to the context. If the context does not specify with what kind of agent we are doing, we will often refer to an agent as "she", which is shorthand for "he, she or it".

EXERCISE[0] 1.1

Let $\mathbf{P} = \{p, q, r\}$. Which of the following are formulas in $L_K^3(\mathbf{P})$? Here you may use the same conventions about parentheses and abbreviations (such as $p \rightarrow q$ for $\neg p \vee q$) as in the textbook. Explain your answers.

a $K_5(p \vee q)$

b $K_1 K_2 p \rightarrow K_1 p$

c $M_3 \neg(p \vee M_1)$

d $K_1 K_1 (Q(r) \vee Q(q))$

worlds and states page 8 line 4

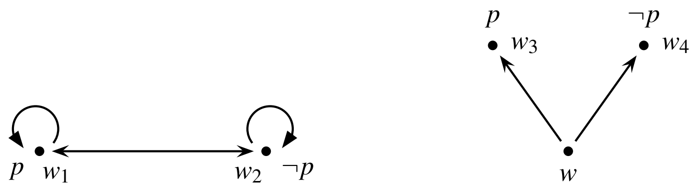
Those who have been introduced to modal logic by other books than this one, might wonder about the distinction between worlds and states. Isn't a state a world anyway? In those other books, the states of our textbook are called "worlds". The distinction of our textbook may be motivated in the following way. A world corresponds to a full description of what relevant propositions are true and false in that world. The set of all propositions includes atomic propositions, so that a world depends on a local truth assignment. But the set of all propositions also includes modal propositions such as $K_i p$, so that a world depends on the accessibility relations R_i as well. The description of a Kripke model includes both local truth assignments and the accessibility relations. Thus, as the textbook states, a world (M, s) consists of a Kripke model M and a distinguished state s.

epistemic alternatives page 8 line 7

The intuitive idea behind epistemic alternatives is that, in addition to the real state of affairs, there could be more states of affairs. For a given agent, some of these other states of affairs (or worlds) are indistinguishable from the real world, and she thinks that these are still possible. The more an agent knows, the better she can distinguish other worlds from the real one, so the fewer worlds are epistemic alternatives for her! This corresponds to the intuitive idea that information means elimination of uncertainty. We say that an agent *knows* φ if φ is true in all worlds that the agent thinks possible.

How to interpret knowledge in practice? Sometimes one can identify 'knowledge' with sensory perception, as in the case of a child seeing a dot of mud on another child's forehead. We now identify 'knowing' with 'seeing', and say that the first child knows that the second child is muddy. Similarly for hearing, feeling, etc.

Knowledge can also be quite conceptually oriented, where direct perception is not the only means of knowing. A real life example - for most readers - is the case of knowing whether the Argentine Open University is situated in Buenos Aires. Your knowledge of the world is likely to be consistent with it both being there and it not being there. In terms of epistemic logic: $\neg K \neg p$ and $\neg K p$. Both a world where the Argentine Open University is in Buenos Aires and a world where it isn't in Buenos Aires are epistemic alternatives for you. Picture (a) below gives the relevant states and accessibility relations. You may wonder why a picture like (b) is not correct. In fact, it is a common mistake to identify worlds with actors, as happened here with world w.



(a) correct picture (b) incorrect picture

Picture (a) illustrates an important feature of epistemic logic. Whereas $p \wedge \neg p$ is absurd, $\neg Kp \wedge \neg K \neg p$ is true in both worlds above. Similarly, $p \vee \neg p$ is trivially true, while $Kp \vee K\neg p$ ("the agent knows whether p") is false in both worlds.

Kripke semantics of epistemic formulas page 8 line 11

Mostly, only those few propositional constants relevant for the application are put in **P**. See the example above, where **P** contains only the propositional constant p. Thus, a "possible world" is not a complete world, as science fiction stories and some philosophers imagine them, at all; it is only complete relative to the set of propositional constants **P**.

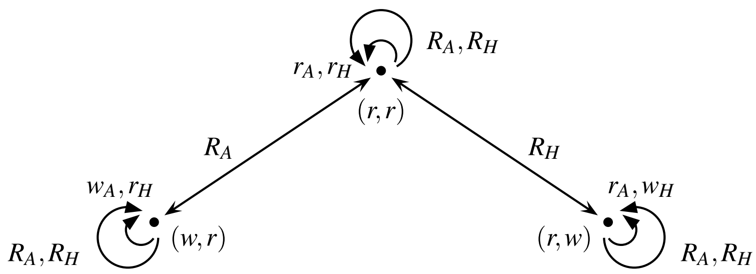
Quoting the textbook, "We define the relation $w \models \varphi$". Here, w is shorthand for the world (M,s) (see page 8, line 4 of the textbook). The new relation $w \models \varphi$ is inspired by the same notation $w \models \varphi$ on line 10 of page 6 of the textbook, where w is a classical truth valuation, and φ a formula in the language of propositional logic. In the sequel we deal almost exclusively with the new possible worlds relation $w \models \varphi$, which makes confusion unlikely.

Let's give an example of how Kripke models can be used to model a puzzle situation. We take a simplified version of the "Wise Persons" puzzle, which goes as follows. There are two wise persons, Abelard (A) and Heloise (H). It is known to everyone that there are three hats: two red ones and one white one. The king puts a hat on the head of each of the two wise persons, who cannot see their own hat but can see the other person's hat (and they both know this). The king asks them sequentially if they know the color of the hat on their own head. The first person, Abelard, says that he does not know; the second person, Heloise, says that she knows.

EXERCISE 1.2

Before reading the analysis below, find out for yourself what must be the color of Heloise's hat.

Let us begin to analyze the situation just after the king has put the hats on the two wise persons' heads but before he has asked any questions. In our case, we have three worlds that are characterized by the color of the hat of *both* persons. For example, in the world represented as (r,w) Abelard wears a red hat and Heloise a white one. Similarly for the other two possible worlds (w,r) and (r,r). (Note that according to the conditions given above, (w,w) is definitely not a possible situation, so we don't include it). The epistemic alternative relations R_A and R_H are given in the picture below. The relevant propositional constants are r_A, r_H, w_A and w_H with obvious meanings. In pictures of Kripke models, the convention is to write down only the propositional constants that are *true* in a world. We follow this convention below. Note that we do not introduce propositional constants for propositions such as "Abelard can see Heloise" and "there are three hats, two red ones and one white one": Abelard's and Heloise's knowledge about these facts, that belong to the *background theory*, is implicitly built into th e choice of worlds and the accessibility relations.



EXERCISE 1.3 We are still in the situation just after the king has put the hats on the two wise persons' heads but before he has asked any questions. Explain why, given the puzzle story, the relations R_A and R_H in the picture above are appropriate for this situation .

EXERCISE 1.4 Determine the value of the following propositions at the given worlds, and explain your answers using the truth definition on page 8 of the textbook.

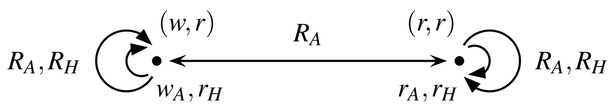
a $(M,(r,w)) \models K_H r_A$

b $(M,(r,w)) \models \neg K_A r_A$

c $(M,(r,r)) \models \neg K_A r_A \wedge \neg K_A w_A$

d $(M,(r,r)) \models K_H(\neg K_A r_A \wedge \neg K_A w_A)$

After the king has asked Abelard whether he knows the color of his hat and Abelard has answered "no", the relevant worlds with the new accessibility relations are given by the following picture:



EXERCISE 1.5 Explain why the relations R_A and R_H in the picture above are appropriate for the situation. Also, explain why Heloise now knows the color of her hat, namely ....

EXERCISE 1.6* We have assumed in the story that Abelard and Heloise can see each other. Now assume that Heloise is blind but that Abelard can see her (and they both know this). Can Heloise still figure out the color of her hat after Abelard's first answer "no"? (Draw new Kripke models if necessary.)

EXERCISE 1.7 Exercise 1.3.1.1 from the textbook.

EXERCISE 1.8 Exercise 1.3.2 from the textbook.

EXERCISE 1.9 John can see Mary and Will, Will can see Mary and hear everybody. John and Grace are deaf, and cannot read Mary's lips. Mary can hear John and sees Grace and Will. Grace can see everybody except herself. John and Grace have white spots on their foreheads. Mary shouts "Grace has a white spot on her forehead". Formalize the given information and draw two Kripke models with the corresponding accessibility relations, one for the situation before and one for the situation after Mary's utterance.

Valid formulas page 11 Prop. 1.3.5 (i) The textbook mentions an "instance of a propositional tautology". Examples of such instances are tautologies such as $(p \to q) \to (\neg q \to \neg p)$, but also tautologies where some propositional constants are replaced by formulas of the form $K_i\varphi$. An example of an instance of the propositional tautology above is $(K_1 p \to K_2(q \vee r)) \to (\neg K_2(q \vee r) \to \neg K_1 p)$.

Closure under logical consequence page 12 Remark 1.3.5.2

This remark discusses properties (ii) and (iv) of Proposition 1.3.5. The possible worlds approach seems to commit us to these two properties. However, the properties force us to treat agents as if they where ideal: they know all valid formulas (iv), and know all logical consequences of their knowledge (ii, iv). This is not a realistic demand on human agents. Even if the agents are computational artefacts such as processors or knowledge bases, they do not have unlimited time or memory space for computing and storing all the knowledge that they should have according to (ii) and (iv). Together the two properties cause the so-called logical omniscience problem, which is treated at length in sections 2.5, 2.6 and 2.7 of the textbook in the context of a logic for belief.

EXERCISE 1.10

Find two real-life (not formal) examples that illustrate the omniscience problem: one example where property (iv) of Proposition 1.3.5 does not hold, and one example where property the property that if $\models \varphi \to \psi$ then $\models K_i\varphi \to K_i\psi$ (which follows from (ii) and (iv) together) does not hold.

2.2 THE AXIOM SYSTEM **K**

Read section 1.4 of the textbook.

Definition of derivation page 13, def. 1.4.1

Because the definition of derivation is a bit abstract, we will give some background and a few examples. First, note that there is a difference between $\mathbf{K_{(m)}}$-derivations and the natural deduction derivations that you met in *Inleiding Logica*. In natural deduction derivations, you are allowed to have premises on the left side of the $\vdash$-sign, as in $p, \neg q \vdash p \vee q$. In the axiomatic approach that is taken in the textbook, this is not the case. Here you can only have a formula on the right-hand side of the $\vdash$-sign.

Second, in $\mathbf{K_{(m)}}$-derivations, you get propositional logic for free, so to speak. Thus, you don't have to derive propositional tautologies, but have all of them to your disposal at once by axiom (A1). Here follows an example derivation, in which the purpose is to prove $\mathbf{K_{(1)}} \vdash K_1(p \to q) \to (K_1 p \to K_1 q)$.

EXAMPLE 1

1. $\mathbf{K_{(1)}} \vdash (K_1 p \wedge K_1(p \to q)) \to K_1 q$ by (A2).

2. $\mathbf{K_{(1)}} \vdash ((K_1 p \wedge K_1(p \to q)) \to K_1 q) \to (K_1(p \to q) \to (K_1 p \to K_1 q))$ by (A1), using the propositional tautology $((\varphi \wedge \psi) \to \chi) \to (\psi \to (\varphi \to \chi))$.

3. $\mathbf{K_{(1)}} \vdash K_1(p \to q) \to (K_1 p \to K_1 q)$ by (R1) on 1,2.

Note that the above derivation can be applied to any sentences φ, ψ instead of p, q. Thus, we have derived rule A2$'$ (xii on page 242 of the book by Meyer and Van der Hoek).

Derived rules page 14, exercise 1.4.1.1

When you make a lot of derivations in $\mathbf{K_{(m)}}$, you will soon notice that some patterns appear in many different proofs. One of these patterns is the following:

EXAMPLE 2

1,...,i. (a number of derivation steps, may be empty in some cases).

i+1. $\mathbf{K_{(m)}} \vdash \varphi \to \psi$ by a rule from the previous steps or by an axiom.

i+2. $\mathbf{K_{(m)}} \vdash K_i(\varphi \to \psi)$ by (R2) on i.

i+3. $\mathbf{K_{(m)}} \vdash K_i(\varphi \to \psi) \to (K_i\varphi \to K_i\psi)$ by Example 1 above, derived rule (A2$'$).

i+4. $\mathbf{K_{(m)}} \vdash K_i\varphi \to K_i\psi$ by (R1) on i+2, i+3.

From this pattern you may generalize to a so-called *derived rule*: if $\mathbf{K_{(m)}} \vdash \varphi \to \psi$, then $\mathbf{K_{(m)}} \vdash K_i\varphi \to K_i\psi$. Here you do not assume the formula

$\varphi \to \psi$, but you assume that $\varphi \to \psi$ has been *derived* in $\mathbf{K_{(m)}}$. This particular derived rule is called K-distribution (abbreviation KD), and may be used in proofs. Here follows such a derivation, where the purpose is to prove $\mathbf{K_{(1)}} \vdash (K_1 p \vee K_1 q) \to K_1(p \vee q)$.

EXAMPLE 3

1. $\mathbf{K_{(1)}} \vdash p \to (p \vee q)$ by (A1).

2. $\mathbf{K_{(1)}} \vdash q \to (p \vee q)$ by (A1).

3. $\mathbf{K_{(1)}} \vdash K_1 p \to K_1(p \vee q)$ by (KD) on 1.

4. $\mathbf{K_{(1)}} \vdash K_1 q \to K_1(p \vee q)$ by (KD) on 2.

5. $\mathbf{K_{(1)}} \vdash (K_1 p \to K_1(p\vee q)) \to ((K_1 q \to K_1(p\vee q)) \to ((K_1 p \vee K_1 q) \to K_1(p\vee q))$ by (A1), using the propositional tautology $(\varphi \to \chi) \to ((\psi \to \chi) \to ((\varphi \vee \psi) \to \chi))$.

6. $\mathbf{K_{(1)}} \vdash (K_1 q \to K_1(p \vee q)) \to ((K_1 p \vee K_1 q) \to K_1(p \vee q))$ by (R1) on 3,5.

7. $\mathbf{K_{(1)}} \vdash (K_1 p \vee K_1 q) \to K_1(p \vee q)$ by (R1) on 4,6.

Now it is time to try an exercise. You may use all derived rules from the appendix of the textbook, pp. 241–245 with reference to their abbreviated names.

EXERCISE 1.11

Prove the following:

a. $\mathbf{K_{(1)}} \vdash (K_1 p \wedge K_1 q) \to K_1(p \wedge q)$

b. $\mathbf{K_{(1)}} \vdash K_1(p \wedge q) \to (K_1 p \wedge K_1 q)$

c. $\mathbf{K_{(1)}} \vdash (K_1 p \wedge K_1 \neg p) \leftrightarrow K_1 \bot$

EXERCISE 1.12[hint]

The proof of Theorem 1.4.6 on p. 18 of the textbook is a bit short. Formulate the proof in a more precise way, by induction on the length (or the structure) of the derivation.

Hint: You may find inspiration on inductive proofs in Chapter 5 of J.F.A.K. van Benthem et al., *Logica voor Informatici*, second edition, Addison Wesley, 1994. Half of the work in making an inductive proof consists in formulating an appropriate inductive hypothesis!

Completeness of $\mathbf{K_{(n)}}$

page 14, def. 1.4.2 up to page 22, line 3

In Definition 1.4.2, a start is made to define concepts such as *maximally consistent set of formulas* that eventually play a role in Theorem 1.4.7, the completeness proof of $\mathbf{K_{(n)}}$. It is handy to have a clear outline of the whole proof in mind before working on all the definitions and lemmas that are needed.

In order to show completeness of $\mathbf{K_{(n)}}$ with respect to K_n, you have to prove the following (this follows from the contraposition of Definition 1.4.5 (ii) of the textbook):
If $\mathbf{K_{(n)}} \nvdash \varphi$, there is a model $M \in K_{(n)}$ and a $w \in M$ such that $(M, w) \not\models \varphi$.

There will be three main steps in the completeness proof:

1 A “Lindenbaum” lemma will be proved: a consistent set of sentences can always be extended to a set that is maximally consistent. This is lemma 1.4.3 (i) of the textbook; part (ii) of the same lemma gives useful properties of maximally consistent sets.

2 These maximally consistent sets will correspond to the states in the Kripke countermodel against φ, and appropriate accessibility relations and a valuation will be defined on these states. The model is called *canonical* because it is a single model that contains counterworlds not just against the formula φ in question, but against all formulas that are not provable from $\mathbf{K_{(n)}}$, at once! The construction of the canonical model is part of the proof of Theorem 1.4.7, and can be found on the upper half of page 19 of the textbook.

3 It will be shown that the model constructed in step 2 indeed contains a world in which φ is false. For this, the so-called Truth Lemma is proved using induction on all formulas. This is the most complex step in the proof. See p. 19, last 10 lines, up to p. 22, line 3 of the textbook.

Truth Lemma — page 19, Lemma 1.4.8

The definition of the canonical model M^c is constructed on purpose so that for all propositional atoms p and all maximally consistent sets Θ we have $(M^c, s_\Theta) \models p$ iff $p \in \Theta$. The Truth Lemma extends this equivalence to all formulas, not just propositional atoms. The proof by induction on the structure of the formula uses appropriate features of maximally consistent sets (see p. 20), as well as some tricky derivations in $\mathbf{K_{(n)}}$ (see p. 21).

Mixed theorems — page 22 after finishing Lemma 1.4.8

Even though the systems $\mathbf{K_{(n)}}$ do not contain explicit mixed axioms in which different K_i operators appear, it is definitely possible to prove mixed theorems. Here follows an example.

EXERCISE 1.13

Prove semantically that $\mathbf{K_{(2)}} \models K_2K_1p \wedge K_2K_1(p \rightarrow q) \rightarrow K_2K_1q$.
Note: by completeness of $\mathbf{K_{(2)}}$, this also implies that
$\mathbf{K_{(2)}} \vdash K_2K_1p \wedge K_2K_1(p \rightarrow q) \rightarrow K_2K_1q$.

2.3 Further properties of knowledge: the system S5

Read sections 1.5 and 1.6 of the textbook.

The axiom (A3) of **S5** — page 23, line 8

Axiom A3 says that an agent only knows things that are true. This axiom is the one used by most philosophers to distinguish knowledge from belief, for which it does not hold: you cannot know a fact that is false, although you may believe it. See section 2.4 of the textbook for more on a logic for belief.

The axiom (A4) and (A5) of **S5** — page 23, line 8

Axioms (A4) and (A5) intuitively say that an agent is introspective. It can look at its knowledge base (or memory in the case of a human agent) and will know what it knows and what it does not know. Many philosophical papers discuss the appropriateness of these two axioms for human agents. Most philosophers reject the introspective axioms, and especially (A5), for various reasons. However, in the area of computer science both axioms are usually accepted. This is reasonable in situations where the epistemic alternatives are viewed as “compatible with the *information* that the agent has” and the agent can check this. For example, if a database doesn’t know a basic fact p and you ask it whether p holds, it will check whether p is among the set of basic facts it contains, and answer ”no”. See Section 1.5 on applications of epistemic logic to distributed systems and protocol verification for an example in which one would certainly accept axioms (A4) and (A5).

EXERCISE 1.14

This exercise concerns the plausibility of axioms (A4) and (A5) for human agents.

a Find two real-life (not formal) examples in which human agents appear: one example where property (A4) does not hold, and one where (A5) does not hold.

b What is your own opinion about the axioms of **S5**, e.g. do you find axiom (A4) more acceptable than (A5) for human agents? (If you follow this course in a group, discuss these issues among groups of 2 or 3 students and write down your conclusions.)

Correspondence theory — page 25 and 26

If you want to know more about the fascinating subject of correspondence theory, you may read Johan van Benthem's "Correspondence theory" in D.M. Gabbay and F. Guenthner (eds.) *Handbook of Philosophical Logic, Vol. II*, pp. 167-247.

Completeness of $\mathbf{S5}_{(\mathbf{m})}$ — page 27, Theorem 1.6.7

Note that $\mathbf{S5}_{(\mathbf{m})}$ does not contain any "mixed" axioms containing knowledge operators K_i for different agents. Still, one can prove useful mixed theorems in $\mathbf{S5}_{(\mathbf{m})}$ for $m \geq 2$, such as $\mathbf{S5}_{(\mathbf{2})} \vdash K_1K_2\varphi \rightarrow K_1\varphi$. Here follows a useful mixed meta-theorem: for all φ, $\mathbf{S5}_{(\mathbf{2})} \vdash \varphi$ iff $\mathbf{S5}_{(\mathbf{2})} \vdash K_1\varphi$ iff $\mathbf{S5}_{(\mathbf{2})} \vdash K_2\varphi$.

EXERCISE 1.15

This exercise is about mixed theorems.

a Show syntactically (i.e. by giving a formal proof) that $\mathbf{S5}_{(\mathbf{2})} \vdash K_1K_2\varphi \rightarrow K_1\varphi$;

b Find and prove at least two other mixed theorems of $\mathbf{S5}_{(\mathbf{2})}$.

2.4 The one agent case: remarks on the $\mathbf{S5}_{(\mathbf{1})}$-model

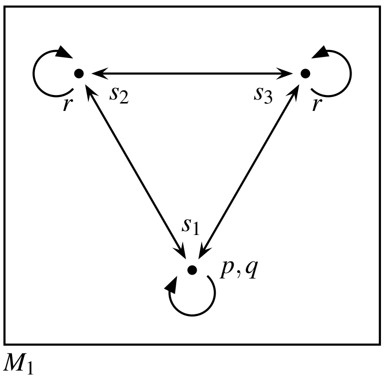
Read section 1.7 of the textbook,
but skip pp. 30, 31 and 32 up to just above
Corollary 1.7.4.6.

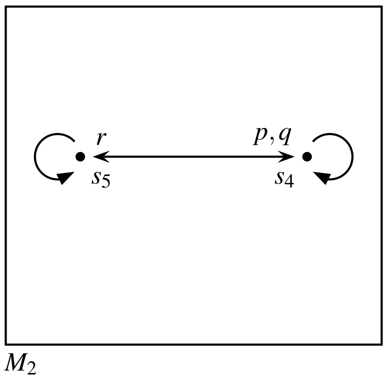
Equivalent worlds — page 28, Definition 1.7.1

The motivation to define equivalent worlds is the following. It turns out that, in $\mathbf{S5}_{(\mathbf{1})}$-models, you don't need to have two different equivalent worlds, but you can always throw away one of them without losing any information. This will be proved below.

*EXERCISE 1.16**

Look at the two models $M_1 = (S_1, \pi_1, R_1)$ and $M_2 = (S_2, \pi_2, R_2)$ below. Prove by formula-induction that for all formulas φ we have $(M_1, s_1) \models \varphi \Leftrightarrow (M_2, s_4) \models \varphi$.





Restricting to the equivalence class — page 28, Proposition 1.7.2

The proof of this proposition is Exercise 1.7.2.1. A hint for this exercise: you have to prove by induction that for all φ, $(M,s) \models \varphi \Leftrightarrow (M',s) \models \varphi$. But other worlds in S' play a role when determining whether formulas of the form $K\psi$ are true in (M,s) and (M',s). Therefore, it is better to prove something stronger, namely for all $t \in S'$ and for all φ, $(M,t) \models \varphi \Leftrightarrow (M',t) \models \varphi$. This trick of proving something stronger so that you can use a strong induction hypothesis is called "loading the induction hypothesis".

EXERCISE 1.17

Make Exercise 1.7.2.1 from the textbook.

Reduced simple models are sufficient — page 29, line 5 from bottom

By proposition 1.7.2 and the remarks below it, you can turn every $\mathbf{S5}_{(\mathbf{1})}$-model into a reduced model $M = < S, \pi, R >$, in which $(s,t) \in R$ for all $s,t \in S$. Once you have a reduced model you can find an equivalent model that is *simple*. This means that all states in such a model have different truth assignments. In order to prove this, we have to "put worlds together" that have the same truth assignment. The textbook needs a very long proof of 2,5 pages (that you have skipped when reading section 1.7) to do this, but in fact the following shortcut is possible.

Let the reduced model $M =< S, \pi, R >$ be given; we will show that there is a reduced *simple* model $M' =< S', \pi', R' >$ that satisfies the same formulas as M. This means that for every formula φ, there is an $s \in S$ such that $(M, s) \models \varphi$ iff there is an $s' \in S'$ such that $(M', s') \models \varphi$.

Proof For every $s \in S$, denote by $[s]_\pi$ the set of states in M which have the same truth assignment as s, thus $[s]_\pi = \{s' \mid \pi(s) = \pi(s')\}$. Now pick exactly one representant world $w_{[s]_\pi}$ from every set $[s]_\pi$. Define the new set of states S' as $\{w_{[s]_\pi} \mid s \in S\}$. Let π' be $\pi \mid S'$ and let R' be the total relation on S', namely $R' = \{(s,t) \mid s,t \in S'\}$. It is clear that $M' =< S', \pi', R' >$ is a reduced model. Moreover, for every two states $w, w' \in S'$ we have $\pi(w) \neq \pi(w')$ because w and w' are representants from different sets $[s]_\pi$; thus $M' =< S', \pi', R' >$ is a simple model. Finally, one can prove by induction on the formula φ that for all φ, $(M, s) \models \varphi \Leftrightarrow (M', w_{[s]_\pi}) \models \varphi$; this in turn immediately implies that M' satisfies the same formulas as M. The atomic step follows because $\pi(s) = \pi'(w_{[s]_\pi})$). The only other interesting case is $\varphi = K\psi$. In this case we have $(M, s) \models K\psi \Leftrightarrow$ for all $t \in S$ $(M, t) \models \psi \Leftrightarrow$ (by induction hypothesis) for all $t \in S (M', w_{[t]_\pi}) \models \psi \Leftrightarrow$ (because all worlds in S' are of the form $w_{[t]_\pi}$) for all $t' \in S'(M', t') \models \psi \Leftrightarrow (M', w_{[s]_\pi}) \models K\psi$.

This finishes the proof.

Computational Complexity — page 34 after Exercise 1.7.5.1

The subject of decidability and complexity of epistemic logics is an interesting and important one. If you would like to see a more extensive explanation, also giving the necessary background on complexity classes, we strongly recommend sections 3.5 and 3.6 of R. Fagin, J.Y. Halpern, Y. Moses and M.Y. Vardi, *Reasoning about Knowledge*, MIT Press, Cambridge (MA), 1995.

Normal forms — page 35, Lemma 1.7.6.3

This lemma is extremely general because it is used to prove a general theorem. Often in applications of the lemma, λ and/or π is taken to be $\top$ or $\bot$. Remember that for all ψ, we have

- $\top \wedge \psi$ is equivalent to ψ
- $\top \vee \psi$ is equivalent to $\top$
- $\bot \wedge \psi$ is equivalent to $\bot$
- $\bot \vee \psi$ is equivalent to ψ.

For practical cases, it is good to know the following equivalences that are all special cases of Lemma 1.7.6.3:

a. $\mathbf{S5}_{(\mathbf{1})} \vdash KK\varphi \leftrightarrow K\varphi$; this is a special case of Lemma 1.7.6.3 (a) by taking $\pi = \bot$, $\lambda = \top$ and $\beta = \varphi$.

b. $\mathbf{S5}_{(\mathbf{1})} \vdash KM\varphi \leftrightarrow M\varphi$; this is a special case of Lemma 1.7.6.3 (b) by taking $\pi = \bot$, $\lambda = \top$ and $\beta = \varphi$.

c. $\mathbf{S5}_{(\mathbf{1})} \vdash MK\psi \leftrightarrow K\psi$; this can be derived from *b* by taking $\varphi = \neg\psi$ and negating both sides of the equivalence.

d. $\mathbf{S5}_{(\mathbf{1})} \vdash MM\psi \leftrightarrow M\psi$; this can be derived from *a* by taking $\varphi = \neg\psi$ and negating both sides of the equivalence.

EXERCISE 1.18

Prove the first two equivalences above semantically without use of Lemma 1.7.6.3; i.e. show that

a. $\mathbf{S5_{(1)}} \models KK\varphi \leftrightarrow K\varphi$;

b. $\mathbf{S5_{(1)}} \models KM\varphi \leftrightarrow M\varphi$;

and then invoke completeness of $\mathbf{S5_{(1)}}$.

EXERCISE 1.19

Write the formula $K(Kp \wedge Mq)$ in normal form.

2.5 Applications to distributed systems and protocol verification

Read sections 1.8 and 1.9 of the textbook.

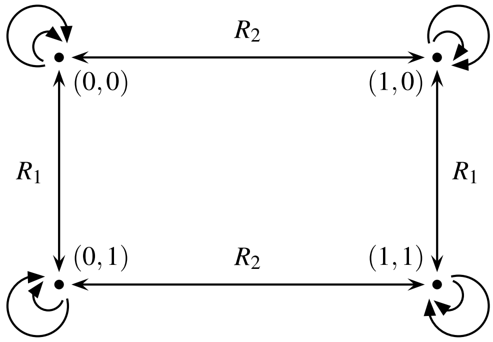
Distributed systems page 38, line 4–10

Epistemic logic can be used to reason about "knowledge" in many kinds of multi-agent systems: the Wise Persons from the puzzle, multi-agent systems that consist of both human and computer agents who cooperate towards a collective goal as studied in Artificial Intelligence, and distributed systems from Computer Science. The latter application concerns us here. A distributed system consists of a number of processors that can communicate with each other through links in a network. When modelling such systems, we make the assumption that at each point in time, each of the processors is in some state, which we refer to as its *local state*. All of these local states together form the system's *global state* at that point in time. These global states will be the possible worlds in our Kripke model. Thus, if you represent the global state as a vector of the local states, a system consisting of two processors may be in global state $s = (s_1, s_2)$, where s_1 and s_2 are the local states of processors 1 and 2.

Kripke models for distributed systems page 38, line 11–16

Let's give a simple example with two processors 1 and 2. Both of these may be in either of two local states, namely $s_i = 0$ or $s_i = 1$ for $i = 1, 2$. Here s_i refers to the contents of processor i's register. The four possible global states are $(1,1), (1,0), (0,1)$, and $(0,0)$, making the set of possible worlds $S = \{(1,1), (1,0), (0,1), (0,0)\}$.

The accessibility relation is defined according to the following informal description of "knowledge" of a processor. The processor i "knows" φ (e.g. that processor j's register contains 0) if in every other global state which has the same local state as processor i, the formula φ holds. In particular processor i knows its own local state. In the picture below, the accessibility relations are drawn for our example case. The definition is natural because it is reasonable to suppose that a processor "only knows itself". (Note that this is opposite to Abelard's and Heloise's predicament before the king asked his first question: they only knew the color of the *other* person's hat.)



EXERCISE 1.20

This exercise concerns the example given above. Suppose proposition letter p stands for "the register of processor 1 contains 0" and q stands for "the register of processor 1 contains 1". Let $P = \{p, q\}$.

1 Give the valuation π by drawing p and q at appropriate worlds in the Kripke model above.

2 Now determine the value of the following propositions at the given worlds.

a $(M, (0,1)) \models K_1 p$
b $(M, (0,1)) \models K_2 p$
c $(M, (0,0)) \models K_2 K_1 p$
d $(M, (0,0)) \models M_2 K_1 p$

Protocol verification page 39, line 1 from bottom

Note that in this model knowledge is an external notion. We do not imagine that processors are really reasoning about their own and other processors' knowledge. Instead, the programmer or person who wants to prove that a protocol is correct reasons about the processors from the outside. For example, the programmer could say that "processor 1 knows that processor 2 is faulty" because processor 2 is faulty in all states consistent with the current state of processor 1.

alternating-bit protocol page 40, line 2 and 6

The alternating-bit protocol is mentioned twice on page 40. At this point it is too early to explain how it works, but see p. 44, last 6 lines, of the textbook.

deletion errors page 40, line 9 from bottom

The textbook doesn't say so until Theorem 1.9.1, but we have to demand that, even if deletion errors may occur, the process does not end up in an infinite loop of deletion errors. Thus, there will always be *some* later point at which *some* message arrives.

Under this assumption, we can use epistemic logic to specify a protocol that handles messages in the correct way, even if deletion errors occur. In the specification of the protocol we want to guarantee that it is *known* whether messages have been received by the other processor at a certain point.

Protocol A page 40, last line, and further

The idea behind this specification of protocol A is the following. First, processor S repeatedly sends its message x_i to R until S knows that x_i is known to R. Then, before sending the new message x_{i+1}, processor S sends a message to R that S knows that x_i is known to R, so that R will not be confused when receiving x_{i+1} into thinking that it is merely a repetition of x_i.

R, meanwhile, just has to send acknowledgments that it has received both types of messages from S, so that S knows that it can proceed.

Let's give an example of what may happen in a particular case, when the input tape is $< 0, 0, 1, \ldots >$.

1 S sets its counter i to 0 and proceeds into the while loop.

2 S reads x_0, in this case it is 0; it sends this to R. In our example, a deletion error occurs, and R doesn't receive the message.

3 S resends the message 0 to R.

4 R receives the message 0, makes its counter equal to 0, and proceeds into its while loop.

N.B. Because of our assumption that the process does not end in an infinite loop of deletion errors, processor R is bound to receive message x_0 at some point.

5 S, not having heard from R, resends the message 0 to R.

6 R writes the value of x_0, namely 0, at the first position of the output tape. (R ignores S's second attempt to send x_0). Then it sends the message "$K_R(x_0)$" to S in order to let S know that the first value has arrived. In our example, a deletion error occurs, and S does not receive the message "$K_R(x_0)$".

7 S, not having heard from R, resends the message 0 to R.

8 R receives another 0, again correctly interprets it as a repetition of x_0 and ignores it. Instead, it resends the message "$K_R(x_0)$" to S.

9 This time S does receive the message "$K_R(x_0)$", as is bound to occur at some point because of our assumption that there are no infinite deletion error loops. Now $K_SK_R(x_0)$ thus S jumps to the next line in its program and sends the message "$K_SK_R(x_0)$" back to R.

10 R receives the message "$K_SK_R(x_0)$", so $K_RK_SK_R(x_0)$, R jumps to the next line in its program and sends the message "$K_RK_SK_R(x_0)$".

11 S receives the message "$K_RK_SK_R(x_0)$", thus $K_SK_RK_SK_R(x_0)$ finally holds. S puts its counter to 1. It reads x_1, in this case it is 0, and sends this to R.

12 R receives the message 0 and correctly interprets this as x_1 (not as a repetition of x_0), so $K_R(x_1)$. It puts its counter to 1 and writes 0 on the second place of the output tape.

13 etc. etc.

other errors than just deletion — page 42, below Theorem 1.9.1

If you are interested to find out how to handle mutation and insertion errors, look at Halpern and Zuck's 1987 the paper mentioned in the textbook.

Protocol B — page 44, line 1

Here appears a subtlety in the implemented version of protocol B: x_{i+1} is a local variable of the receiver R, in which R stores the sender's colored message var($x_{i+1}, i+1$).

EXERCISE 1.21

Work out an example of what may happen using protocol B in a particular case, for input tape $< 0,1,0,\ldots >$, in about as much detail as the example given above for protocol A.