Dynamic Variations: Update and Revision for Diverse Agents[1]

Fenrong Liu[2]
The Institute for Logic, Language and Computation
University of Amsterdam

August 20, 2004

[1]Supervisor: Prof. Dr. Johan van Benthem
[2]Email: liu@science.uva.nl or fenrong@hss.ustb.edu.cn

Contents

Acknowledgements

There are many people I would like to thank for a huge variety of reasons.

I would like to thank my supervisor, Prof. Johan van Benthem, for his consistent support and assistance. He is the best supervisor and teacher I could have wished for. Without his perceptiveness, I would never have finished this thesis. His insightful conversations and countless helpful comments always made me realize that everything could be better and better... In particular, I much appreciate that he gave me the opportunity to execute a joint text (it will be published in *Philosophia Scientiae* in a shortened form.) with him and agreed to put it here as my Chapter 3. From the collaboration, I learned how to formulate our intuitions precisely and how to deepen our exploration step by step. I enjoyed the whole process very much! I am indebted to him for his always making time for me, even during his Stanford Spring quarter, his effort of teaching me how to do research, and especially his attempt of helping me realize my dream in logic. Thank you for pushing me!

Also I want to thank my study advisor, Prof. Dick de Jongh, for his generous help and encouragement. Without his extremely helpful advice on my course arrangement before and after I came to Netherlands, I would never have finished the Master program within one year. He is one of the nicest person that I have ever met. His help is always there when it is most required. I really appreciate his understanding and patience through all the times I have been frustrated. In particular, I am more than grateful for his English correction to any pieces that I got, even ordinary emails. I learned a lot from these corrections. On the personal side, he did not hesitate to invite his students to become an extended part of his family. Christmas 2003 was the most wonderful time that I have spent here. I appreciated such activities immensely, it was really a joy!

Besides my supervisor, I would like to thank the rest of my thesis committee, Prof. Dick de Jongh and Prof. Jan van Eijck, thank you both for your managing to read the whole thing thoroughly during the summer holiday.

I thank Guillaume Aucher for his extensive comments on my draft of Chapter 2, as well as for the discussions via email. I also thank Josh Synder for his penetrating comments on the draft of Chapter 3, his insightful emails. I benefited from such discussions, which has made me further think of many issues of interest.

In addition to my thesis work, I would like to express my gratitude to my teachers at Amsterdam for their excellent lectures and correction of my homework: Joost Joosten, Johan van Benthem, Balder ten Cate, Dick de Jongh, Barteld Kooi, Paul Dekker, Frank Veltman, Michiel van Lambalgen, Yde Venema, Nick Bezhanishvili. Time after time, their easy grasp of logic at its most fundamental level helped me in the struggle for my own understanding. I came to realize that it is their successful work that has made and continues to make the 'Dutch school' so prosperous.

I would also like to thank all the rest of the academic and support staff of the ILLC, with special thanks to Peter van Emde Boas for his talks on *Sun Tzu*, *Stratego* and complexity; Tanja Kassenaar, Marjan Veldhuisen are also especially thanked for their care and attention.

Living in Amsterdam was a very exciting and rewarding experience. Thanks to my classmates and other friends at Amsterdam, they have been a part of my wonderful life. I also thank my teachers, colleagues, friends and family in China, their emails and e-cards always made me feel absolutely supported. I just cannot put the whole list here, it would be too long. My special thanks to Chunlai Zhou for what he had done for me before I came to Amsterdam.

Perhaps most importantly, I am grateful for the financial support from the Huygens Scholarship, the Beth Foundation and the China Scholarship Council, which made my Amsterdam trip and life possible.

Chapter 1

Introduction

1.1 Motivating Example and Research Problem

Logical structures in conversation Logical analysis of information flow and interaction starts with very simple phenomena. Our knowledge and belief are in a continuous flux. Consider this Mosaic dialogue in the NWO elevator on 18 March 2004, where four people, two younger, A, B, and two older, $A1, B1$, met going up in the morning:

A: Are you a Mosaic candidate?
B: Yes.
A: Is he ($B1$) your promotor?
B: No.

Some hours later, 05:00 PM, A and B met again:

A: Are you a Mosaic candidate?
B: Yes, we talked this morning.
A: Sorry. Do you have a concrete idea for your proposal due on April 5?
B: No, I have no idea.

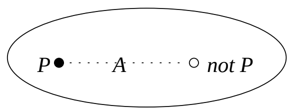
Behind the scenes of this simple daily question-answer scenario, flow information occurs. In the first question-answer pair. A's asking indicates to B that A does not know the answer. B's answer makes A learn that B is a candidate (P, for short). This can be expressed with the epistemic-logical operator K for 'knows that'. Before the conversation we have

$$\neg K_A P \wedge \neg K_A \neg P, \quad K_B P, \quad K_A(K_B P \vee \neg K_B P).$$

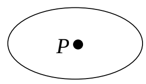
But more is true after the answer. B knows that A knows, A knows that B knows that A knows, and so on to any depth of iteration: we get so-called *common knowledge.* In formulas we have:

$$K_A P, \quad K_B K_A P, \quad \text{etc. and } C_{\{A,B\}} P.$$

We can also express the initial knowledge of A in a mathematical model. The black dot stands for the actual 'world' (with P), the white one for another world (with not-P) that A considers possible. The dotted line with A indicates that A does not know whether P,



Now we see the process character of information in conversation. B's answer triggers an *update* of this information model, it eliminates A's option not-P, to yield the one-point diagram:



At this stage, P has become common knowledge between A and B, and not only that, $A1$ and $B1$ are also in the know, if they have paid attention to the conversation.

Let us now look at the second question-answer pair:

A: Is he your promotor?
B: No.

A has reason to *believe* that if B is a candidate, $B1$ is his promotor, because of the NWO rules (the promotor may attend the workshop on March 18), but A's belief is mistaken. B's negative answer causes A to *revise* her belief. Belief revision is a more complex process than information update, but it is an indispensable ingredient of communication!

But even this is not yet the full story. Let us continue with the afternoon episode:

A: Are you a Mosaic candidate?
B: Yes, we talked this morning.

Obviously *A forgot* what had happened that morning. She is once again uncertain whether P or not P. Such limitations to our rationality happen frequently in real life, some people have a good memory, others do not, witness the bounded rationality that game theorists often emphasize. Memory limitations are one key aspect of this: players in a game need not even remember all previous moves.

The final question-answer pair brings the story to a climax, revealing one more game-theoretic aspect to communication:

A: Sorry. Do you have a concrete idea for your proposal due on April 5?
B: No, I have no idea.

This is a genuine question, as A does not know and expects the answer from B. But there is much more to its meaning. Behind every ordinary question, there is a game-theoretic 'meta-question'. Why does A ask this? Both A and B are in the same competitive "application game", and A hopes to obtain more knowledge about her opponent, so that A's actions afterwards may depend on B's answer, but B might not be truthful in his answer either, this might be his best strategy. Similar things often occur in games, where players try to achieve certain goals through interaction. Stable and successful social behavior arises when the various strategies are in equilibrium.

Research problem How can we design a logic of communication that includes all aspects described here, and allows us to analyze a broad range of forms of human behavior? It needs to combine the dynamic processes of information update, and belief revision when encountering new evidence.

But it also needs to take various sources of bounded rationality seriously, e.g. memory limitations - much more than the usual idealized logical theories with omniscient agents. And it needs to explore various subtle features of the strategic interaction between social agents. In my thesis, I would like to explore these questions, in particular, introduce bounded rationality into logical theory.

1.2 Background and Goals of Work

As it happens, there are already several building blocks for the joint approach that I need, in the form of logical systems that are currently investigated in the research community at the interface of logic, computer science, and game theory.

Update logic To describe the concept of information update, exhibited in the Mosaic dialogue, dynamic update logic (cf. van Benthem 1996; Veltman 1996; Gerbrandy 1999; Baltag et al., 1998) is the best current system, which describes how to update a static information model to a new one when some informative event takes place. The following definition is the heart of update logic:

$$(s,\ a) \sim_i (t,\ b) \quad \text{iff} \quad \text{both } s \sim_i t \text{ and } a \sim_i b.$$

This means the uncertainty of original states and original action yields the new uncertainty. In a logical reasoning system this update rule validates the following key *reduction axiom* relating knowledge to action:

$$\langle \boldsymbol{A}, a\rangle\langle j\rangle\varphi \leftrightarrow (PRE_a \wedge \bigvee\{\langle j\rangle\langle \boldsymbol{A}, b\rangle\varphi : a \sim_j b \text{ for some } b \text{ in } \boldsymbol{A}\})$$

It means that after an action a (in action model $\boldsymbol{A}$) agent j knows φ *if and only if* agent j knows that after action b which she cannot distinguish from a, φ will hold. Such principle is of importance in that it allows us to relate our knowledge after an action takes place to our knowledge beforehand, which makes a crucial role in planning of AI.

There are two basic components in update logic:

(a) an *epistemic model* $\boldsymbol{M}$ of all relevant possible worlds with agents' uncertainty relations indicated,

(b) an *action model* $\boldsymbol{A}$ of all relevant actions, again with agents' uncertainty relations between them.

Action has restrictions – one can only consider the action which possibly happens. These are encoded by

(c) *preconditions* PRE_a for actions a.

The knowledge of preconditions are supposed to be common knowledge among agents. The updated epistemic model is computed as follows:

$$\boldsymbol{M} \times \boldsymbol{A} = \{(s,\ a) \mid s \in \boldsymbol{M},\ a \in \boldsymbol{A}\ \&(\boldsymbol{M},\ s) \models PRE_a\}$$ [1]

To relate the abstract mathematical model in update logic to what happens in real life, we have the following basic assumptions:

(a) A *social situation* s involving the intuitive concepts of knowledge and common knowledge corresponds to a *mathematical model* S(multi-agent Kripke model).

(b) An *operation* taking situation s to situation $o(s)$ corresponds to a mathematical model $O(S)$(an *update*) in the following way: if s corresponds to S in (a), then $o(s)$ corresponds to $O(S)$.

(c) The update does not change the truth of the *fact*.

Belief revision Belief revision occurs when new information contradicts previous beliefs, and no simple update is possible: the whole structure of one's beliefs has to be rearranged. Belief revision theory (C.Alchourron-P.Gärdenfors-D.Makinson 1985, P.Gardenfors 1988) describes belief update for agents in AI, law, and in general philosophical settings. Concerning mathematical representation, we have the following basic assumptions:

(a) An agent's epistemic state can be represented by a *belief set*, that it, a set of formulas from classical propositional logic.

(b) A revision operator $\circ$ brings a belief set A and a formula φ to a new belief set $A \circ \varphi$.[2]

[1] The *actual world* of the new model is the pair consisting of the actual world in $\boldsymbol{M}$ and the actual action in $\boldsymbol{A}$.

[2] Lindström and Rabinowicz treated it as a relation between theories (belief sets) rather than a function on theories, see Lindström and Rabinowicz 1997.

The fundamental contribution of AGM is the following well-known set of rationality postulates:

(R1) $A \circ \varphi$ is a belief set

(R2) $\varphi \in A \circ \varphi$

(R3) $A \circ \varphi \subseteq Cl(A \cup \{\varphi\})$[3]

(R4) If $\neg\varphi \notin A$, then $Cl(A \cup \{\varphi\} \subseteq A \circ \varphi$

(R5) $A \circ \varphi = Cl(false)$ iff $\vdash \neg\varphi$

(R6) If $\vdash \varphi \Leftrightarrow \psi$ then $A \circ \varphi = A \circ \psi$

(R7) $A \circ (\varphi \wedge \psi) \subseteq Cl(A \circ \varphi \cup \{\psi\})$

(R8) If $\neg\psi \notin A \circ \varphi$ then $Cl(A \circ \varphi \cup \{\psi\}) \subseteq A \circ (\varphi \wedge \psi)$.

Intuitively, R1 and R2 mean that a belief set should include φ after revision by φ. R3 and R4 express that if the new belief is consistent with the belief set, then the revision should not discard any of the old beliefs and should not add any new belief except those implied by the combination of the old beliefs with the new belief. R5 says that agents can incorporate any consistent belief, and R6 states that the syntactic form of the new belief does not affect the revision process. The last two postulates say that if ψ is consistent with $A\circ\varphi$ then $A\circ(\varphi\wedge\psi)$ is $A\circ\varphi\circ\psi$.

AGM postulates that belief can change in the following three ways:

- *Expansion*: Expanding a belief set K by a sentence A together with the logical consequences, obtaining $K + A$.
- *Contraction*: A sentence in K is retracted without adding any new facts. Meanwhile, some other sentences from K must be given up, obtaining $K - A$.

[3]Here Cl denotes the deductive closure.

- *Revision*: A new sentence that is inconsistent with a belief set K is added, but in order to make the resulting belief set consistent, some old sentences in K are deleted, obtaining $K * A$[4].

Bounded rationality Most logical literature assumes that players' knowledge and abilities are perfect. But limited powers of observation and processing are the reality. Modern game theory incorporates this in various ways: cf. the imperfect information games, sequential equilibrium, and finite-automaton strategies in Osborne & Rubinstein, 1994. Van Benthem 2001 shows how imperfect information games with memory-bounded players can be analyzed in an ordinary dynamic-epistemic logic with added uncertainty links.

So far, all these systems have their limitations. E.g., AGM belief revision theory deals with *belief* revision, but it leaves out two crucial aspects present in epistemic update logics: knowledge and ignorance, and multi-agent interaction. On the other hand, update logic doesn't deal with the true dynamics of belief change. A merge seems to be called for.

Recent work: Plausibility logic combing update with revision Attempts to merge the above blocks have been carried out by Aucher 2003, van Ditmarsch & Labuschagne 2003 and van Ditmarsch 2004. In particular, Aucher presents a logical system combining update logic with belief revision, therefore we can characterize the update with not only knowledge but also belief involved. Technically, a new belief operator B_j^k is introduced into the update logic and expresses belief up to degree k. With plausibility assigned to states and actions, when belief is updated, one can record changes in plausibility. Here is the key formula:

$$\kappa'_j(w,a) = Cut_{Max}(\kappa_j(w) + \kappa^*_j(a) - \kappa^w_j(\varphi)) \quad (\kappa)$$

Using this formula, one can calculate the plausibility of a new state (w,a) from that of the previous state and action. For precise definitions of this formula and related notions, we refer to Section 2.1 and 2.2.

[4]K. Sergerberg treated these three ways as three actions in dynamic doxastic logic, introducing three operators accordingly: $[+\varphi]\chi, [-\varphi]\chi$ and $[*\varphi]\chi$, the intuitive interpretations are 'after expanding the set of beliefs by φ, it is the case χ', 'after contracting the set of beliefs by φ, it is the case χ, and 'after revising the set of beliefs by φ, it is the case χ' respectively. For details, see K. Sergerberg 1995.

Goals of current work There are several goals this thesis attempts to obtain:

- to understand plausibility logic, its notions of importance, its logical properties, etc.
- to propose variations for updating belief plausibility that allow for diversity of agents,
- to understand the dramatic difference between agents with 'the best memory' and agents with 'the worst memory' in games in terms of their update behavior,
- to explore the intermediate cases, how k-memory agents update their information, where k can go up arbitrarily, as well as societies of different agents and their interaction.

Summarizing, our aim is to develop plausibility logic, incorporating bounded rationality into it, to better understand the variety of human behavior as illustrated in the Mosaic dialogue and in many other circumstances, for instance, in games.

1.3 A Guide to the Thesis

Besides the introduction, the thesis consists of three chapters and three Appendixes. A brief overview of each of these parts follows.

Chapter 2 This chapter begins with a review of the basics of plausibility logic, its language and semantics, the full logical system PL, including its static part and its dynamic part. Then we focus in particular on the static part $\mathsf{PL_S}$, presenting an improved completeness proof and extending it in several ways. Special attention will be given to a new variation of plausibility updating rule, which allows for diversity of agents.

- **Static aspects**

- ***Theorem proof*** Several theorems will be derived in the static plausibility logical system $\mathsf{PL_S}$. The following theorem:

$$\vdash B_j^k(B_j^k\varphi \to \varphi) \textit{ for all } k \in \mathbb{N},$$

will contribute considerably to simplifying the completeness proof, which we will also present in detail. Cf. Section 2.2.

- ***Extending*** $\mathsf{PL_S}$ ***with common knowledge*** We first extend $\mathsf{PL_S}$ with the common knowledge operator C_G in the classical way, obtaining a new system $\mathsf{PL_S^C}$ which becomes more expressive. It is then possible to deal with settings like the Mosaic dialogue. For example, a logical formula like $[P!]C_{\{A,B\}}P$ expresses that after B's answer, P is common knowledge between A and B. Meanwhile, we will present the completeness proof for the system $\mathsf{PL_S^C}$ in the line of propositional dynamic logic. Cf. Section 2.3.

- ***Adding common belief*** Another significant notion - common belief up to a degree (denoted by D^k_G) is also added in two different ways. One way is to define common belief up to a degree in pure belief version: $D^k_G\varphi$ is true, if everyone in G believes up to degree k that φ, everyone in G believes up to a degree k that everyone in G believes up to degree k that φ, etc. We get the system $\mathsf{PL_S^{CD}}$. Another way is to define it with common knowledge and belief operator together: φ is a common belief up to degree k among the group G if everyone in G believes φ up to degree k, which is common knowledge in G: We obtain the system $\mathsf{PL_S^{CD'}}$. The completeness proof for these two new systems are also presented. Cf. Section 2.4.

- **Dynamic aspects**

- ***New plausibility updating rule*** Besides these technical variations on static plausibility logic, a more important move is that we propose a new parameterized formula for plausibility update, which allows for a much greater variety of forms of behavior:

$$\kappa'_j(w,a) = \frac{1}{\boldsymbol{\lambda} + \boldsymbol{\mu}}(\boldsymbol{\lambda}\kappa_j(w) + \boldsymbol{\mu}\kappa^*_j(a)).$$

Where λ and μ are the weight that an agent j gives to the state w and to the action a respectively. Thanks to the parameters, we get different variations of this rule, hereby modelling the following five different types of agents: *highly radical agents* who just take the plausibility of the previous action as their new plausibility, *radical agents* whose new plausibility is close to that of the previous action, *middle of the road agents* who take the average of the plausibilities of the previous state and action, *highly conservative agents* who take the plausibility of the previous state as the new plausibility, *conservative agents* whose new plausibility is close to that of the previous state. In this way, we can precisely illustrate how different types of agents revise their belief plausibility when they encounter new information. Cf. Section 2.5.

Chapter 3 This chapter written with van Benthem is devoted to an exploration of two extreme types of game-theoretic agents: those with Perfect Recall of everything that happened in a game, and Memory-free agents who observe only the last-played action. We occupy ourselves with their behaviors in terms of update logic.

- ***Axioms for Perfect Recall and Memory-free agents*** We begin with dynamic epistemic language, presenting two axioms for these two types of agents: Perfect Recall agents satisfy:

  $$K_i[a]p \to [a]K_i p.$$

  Intuitively, this means Perfect Recall agents know their own moves and also remember their past uncertainties as they were at each stage. As the opposite of Perfect Recall, we propose a new axiom for Memory-free agents:

  $$\langle a\rangle p \to U[a]\langle i\rangle p.$$

  This says that the agent can only know things after an action which are true wherever the action has been performed. Cf. Section 3.2.

- ***Characterizing Perfect Recall agents*** Standard update logic presupposes prefect agents, in a precise sense, this is just what the reduction axiom characterizes:

  $$\langle \boldsymbol{A},a\rangle\langle i\rangle\varphi\leftrightarrow(PRE_a\wedge\bigvee\{\langle i\rangle\langle \boldsymbol{A},b\rangle\varphi{:}a{\sim_i}b \textit{ for some } b \textit{ in } \boldsymbol{A}\}).$$

  Another approach to describe the Perfect Recall agents is via *structural conditions* on the relations of tree $\mathcal{E}$ of event sequences, such as *PR*, *UNL*, and *BIS-INV*, where different types of agents i correspond to different types of uncertainty relation $\sim_i$. We will prove a characterization theorem for the update of Perfect Recall agents:

  The abstract tree $\mathcal{E}$ satisfies PR, UNL, and BIS-INV iff $\mathcal{E}$ is isomorphic to a particular tree model Tree($\boldsymbol{M}$, $\boldsymbol{A}$), where *Tree*($\boldsymbol{M}$, $\boldsymbol{A}$) starts from an initial state model $\boldsymbol{M}$ and an action model $\boldsymbol{A}$, repeating product updates forever. Cf. Section 3.3.

- ***Characterizing Memory-free agents*** It turns out to be possible to similarly characterize the update of Memory-free agents in above two senses. We present the following new reduction axiom for Memory-free agents:

  $$\langle a\rangle\langle i\rangle\varphi \leftrightarrow (PRE_a \ \&\ E \bigvee_{b\sim_i a} \langle b\rangle\varphi).$$

  It is also possible to characterize such agents structurally just as Perfect Recall agents, but with different structure conditions PR^- and UNL^+. We will also prove the following result:

  > *An equivalence relation $\sim_i$ on $\mathcal{E}$ is Memory-free iff the two conditions PR^- and UNL^+ are satisfied.*

  The behavior of Memory-free agents is closely related to finite automata; another excursion to the automata theory bring us an exciting fact:

  > *Memory-free agents are exactly those whose uncertainty relation is generated by a rigid finite-state automaton.*

  Finally, we concentrate on the different level of knowledge that these two different agents may obtain and propose a possible way of making such differences more explicit, i.e. enriching the language with a converse action operator. Cf. Section 3.4.

- ***Discussions*** We also discuss other possible modifications of the product update that create different types of agents, exploring the idea that players with different abilities necessarily live together in social settings and how they interact with each other. This resulting epistemic and process diversity raises many new types of problems, such as taking advantage of knowing each other's limitations, and learning other people's 'types'.

Chapter 4 The thesis ends with a discussion of six possible directions in which this work can be extended.

Appendix A Something simpler occurs here! As we know, there are not only the knowledge operator but also the new belief operator B_j^k in plausibility logic. The belief operator B_j^k is certainly useful in expressing the firmness or degree of the belief. But incidentally, we will show that the meaning of $B_j^k\varphi$ can be expressed by a propositional constant p_j^k together with the knowledge operator:

$$B_j^k\varphi := K_j(p_j^k \rightarrow \varphi),$$

where p_j^k means intuitively 'agent j assigns the world where she stands the degree of belief at most k'. This yields a new atomic system $\mathsf{PL}_{\mathsf{S}}^{-}$. The completeness of this new system will also be presented in detail.

Appendix B We begin with a motivating example to show that if we change our perspective on the product update, i.e. we do update as: $\boldsymbol{M}\times\boldsymbol{A}_1\times\boldsymbol{A}_2\ldots$ where actions models can be different, instead of the perspective in Chapter 3: $\boldsymbol{M}\times\boldsymbol{A}\times\boldsymbol{A}\ldots$ where action models are uniform. Then the update definition for Memory-free agents in Chapter 3 fails since some worlds will be gone forever according to the precondition restriction, but they are needed in the later update stages to get uncertainty for bounded memory agents. We first present Synder's inclusive proposal; the basic definition for the update of 0-memory agents is:

Definition B.2

(2a) $\boldsymbol{M}\times\boldsymbol{A}=\{(s,a) : s \in \boldsymbol{M} \textit{ and } a \in \boldsymbol{A}\}$

(2b) $(s,a) \sim_i (t,b)$ *iff* $(\boldsymbol{M}, s \models PRE_a$ *iff* $\boldsymbol{M}, t \models PRE_b)$ *and* $a \sim_i b$

We then present another proposal: the copy action proposal. The key definition for the update of 0-memory agents is:

Definition B.3

(3a) $\boldsymbol{M}\times\boldsymbol{A}=\{(s,a) : s \in \boldsymbol{M} \textit{ and } a \in \boldsymbol{A} \textit{ and } s \models PRE_a\}$

(3b) *For* $a, b \neq C!$, $(s,a) \sim_i (t,b)$ *iff* $a \sim_i b$

Both of these two proposals work well, they manage to keep the worlds that would have gone, but in a different manner. The similarity and difference of these intuitions are further explored as well. New questions are raised. Extension of such results to the k-memory agents case is briefly discussed.

Appendix C For bounded memory agents, what to forget is as important as what to remember. We start with a simple observation, i.e. it is presupposed that bounded memory agents forget the *earliest* information when new information comes in, but this is not *always* the case. Intuitively, there are many other possibilities: the agent may choose to forget the information that she has not used so *frequently* , or she may choose to forget the information that she thinks will not be used in the near *future*, etc. More diversity exists! Incidently, this has been investigated extensively in computer science. We then focus on a concrete example from computer science and reinterpret it in terms of update logic. We find that even the behavior of agents with the same memory capacity can be very different. The following 'replacement policy' embodies exactly the various behaviors of bounded agents in computer science:

- First In First Out (FIFO): Replace the 'oldest' data in the memory, i.e. the data which was loaded before all the others.
- Least Recently Used (LRU): Replace the data which has not been referenced to since all the others have been referenced to.
- Optimal (OPT): Replace the data that will not be used for the longest period of time.

We discuss in detail their difference from various points of view, and conclude that several desirable properties such as agents' preference, consideration of cost, should be taken into account when we are constructing an update logic for bounded memory agents.

Chapter 2

Plausibility Logic

In this chapter, we will briefly first review the basics of plausibility logic, its language, semantics and the full logical system PL[1]. Secondly, we will come to its logical variations, present an improved completeness proof, extend $\mathsf{PL_S}$ with the notion of common knowledge and common belief. Finally, we will propose a new plausibility update rule to show the diverse strategies of different kinds of agents.

2.1 Review

Ordinal ranking approach The most significant idea in plausibility logic is that a κ-ranking is introduced into the update logic system, which makes it possible to express belief up to a degree. The κ-ranking is widely called an ordinal ranking, which was presented by Spohn 1988. It is a function κ from a given set W of possible worlds into the class of ordinals such that the worlds with the smallest ordinals are the most plausible. Some possible worlds are assigned the smallest ordinal 0. In other words, κ represents a plausibility grading of the possible worlds. The plausibility ranking of possible worlds can be extended to a ranking of propositions, we define

$$\kappa(A) = min\{\kappa_j(w) : w \in A\}.$$

Intuitively, the κ-ranking assigns a degree of *surprise* to each world in W, where 0 means unsurprising and a higher number denotes greater surprise,

[1]We will take $\mathsf{PL_S}$ to denote the static part of the system PL. $\mathcal{L}$ and $\mathcal{L}_S$ express its full language and static language, respectively.

less firmness of belief.

Definition 2.1 *A **belief epistemic model** $M = (W, \{\sim_j : j \in G\}, \{\kappa_j : j \in G\}, V)$,*[2] *is a tuple where:*

1. *W is a non-empty set of possible worlds called the states of model.*[3]
2. *G is a finite set of agents.*[4]
3. *$\sim_j$ is an equivalence relation defined on W for each agent j.*
4. *κ_j is an operator, ranging from 0 to Max, defined on all states.*[5]
5. *V is a valuation.*

Definition 2.2 *A **belief epistemic action model** Σ is a tuple $(\Sigma, \sim_j, \kappa^*_j, PRE)$ such that:*

1. *Σ is a non-empty set of simple actions.*
2. *$\sim_j$ is an equivalence relation defined on Σ for each agent j.*
3. *κ^*_j is an operator, ranging from 0 to Max, defined on all actions.*[6]
4. *PRE is a function from the set of actions to the formulas of $\mathcal{L}_s$.*

Definition 2.3 *Given a belief epistemic model M and a belief epistemic model Σ, we define their **product update** to be the the epistemic action model*

$$M \otimes \Sigma = (W \otimes \Sigma, \sim'_j, \kappa'_j, V')$$

given by the following:

---

[2]This is slightly different from Aucher's definition since the actual world is not in the model explicitly. We see the actual world as one of the possible worlds (this is also widely accepted). Furthermore, we have not found the advantage of incorporating it in the model yet. In the same way, we do not include the actual action in the action model neither.

[3]We will not distinguish states from worlds in this context.

[4]We will take i or j to denote arbitrary agent in G. $j \in G$ is often omitted if it is clear in the context.

[5]This is intended to describe an agent's plausibility preference among her indistinguishable worlds. Max is an arbitrary fixed natural number different from 0.

[6]This is intended to describe an agent's plausibility preference among her indistinguishable actions.

1. *$W \otimes \Sigma = \{(w,a) \in W \times \Sigma : M, w \models PRE_a\}$.*
2. *We define $\sim'_j$ such that $(w,a) \sim'_j (v,b)$ iff $w \sim_j v$ and $a \sim_j b$.*
3. *$\kappa'_j(w,a) = Cut_{Max}(\kappa_j(w) + \kappa^*_j(a) - \kappa^w_j(\varphi))$,*

   where $\varphi = PRE_a$, $\kappa^w_j(\varphi) = min\{\kappa_j(v) : v \in V(\varphi) \text{ and } v \sim_j w\}$

   $$Cut_{Max}(x) = \begin{cases} x & \text{if } 0 \leq x \leq Max \\ Max & \text{if } x > Max. \end{cases}$$

   Cut_{Max} is a technical device to ensure that the new κ-value fits in the range of the κ scale of the new belief epistemic model, i.e. fits in the set $\{0, \ldots, Max\}$.
4. *V' equals to original valuation on the worlds.*

Definition 2.4 *Let a finite set of proposition variables Φ and a finite set of agents G be given. The full **language** $\mathcal{L}$ is given by the rule*

- *Sentences $\varphi := \top \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid K_j\varphi \mid B^k_j\varphi \mid [\pi]\varphi$.*
- *Programs $\pi := \sigma_i\psi_1 \ldots \psi_n \mid \pi + \rho \mid \pi.\rho$.*

where $p \in \Phi$ and $j \in G$, $k \in \mathbb{N}$.

There are two kinds of syntactic objects: sentences and programs. Programs of the form $\sigma_i\psi_1 \ldots \psi_n$ are simple programs or actions.[7] Note that they may not be "atomic" because the sentences ψ_j may themselves contain actions. $+$ and $.$ are the disjoint union operation and composition operation on programs respectively. $K_j\varphi$ is read "agent j knows φ", $B^k_j\varphi$ is read "agent j believes φ up to degree k", $[\pi]\varphi$ is read "after program or action π is performed, φ holds".

For a better understanding of the definition of the semantics for a simple action $\sigma_i\psi_1 \ldots \psi_n$, we will give a general definition of a signature-based belief epistemic action model, starting with the definition of action signature:

Definition 2.5 *An **action signature** is a structure*

[7] We will not distinguish actions from programs in this context. $\sigma_i\psi_1 \ldots \psi_n$ is also often written as σ_i, ψ.

$$\mathbf{\Sigma} = (\Sigma, \sim_j, \kappa_j^*, (\sigma_1, \sigma_2, \ldots, \sigma_n))$$

where $\sigma_1, \sigma_2, \ldots, \sigma_n$ *is an enumeration of* Σ *in a list without repetitions, we call the elements of* Σ *simple actions.*

Definition 2.6 *Let* $\mathbf{\Sigma}$ *be an action signature, let* $\Gamma \subseteq \Sigma$ *and* $[\![\psi_1]\!], \ldots, [\![\psi_n]\!]$ *be a list of epistemic propositions. We obtain a* ***signature-based belief epistemic action model*** $(\mathbf{\Sigma}, \Gamma)([\![\psi_1]\!], \ldots, [\![\psi_n]\!])$ *in the following way:*

- *The set of actions is* Σ*, and the accessibility relations are those given by the action signature.*
- *For* $j = 1, \ldots, n, PRE_{\sigma_j} = [\![\psi_j]\!]$.
- *The set of distinguished actions is* Γ.

In the special case that Σ is the singleton set $\{\sigma_i\}$, we write the resulting signature-based program model as $(\Sigma, \sigma_i)([\![\psi_1]\!], \ldots, [\![\psi_n]\!])$.

Thanks to the κ-ranking, we are able to assign a number to states and actions, thereby distinguishing their different plausibilities. For instance, $\kappa_j(v) > \kappa_j(w)$ means that agent j believes that world w is more plausible than world v.[8] This also yields a precise definition of belief up to a degree, which we will see in the following definition.

Definition 2.7 *Let* $M = (W, \sim_j, \kappa_j, V)$ *be a belief epistemic model, the* ***semantics of sentences*** *are defined as follows (we omit the classical ones):*

- $M, w \models K_j\varphi$ *iff for all* v *s.t.* $w \sim_j v$, $M, v \models \varphi$.
- $M, w \models B_j^k\varphi$ *iff for all* v *s.t.* $w \sim_j v$ *and* $\kappa_j(v) \leq k, M, v \models \varphi$.
- $M, w \models [\pi]\varphi$ *iff for all* v *s.t.* $w[\![\pi]\!]_M v, M([\![\pi]\!]), v \models \varphi$.[9]

The ***semantics of programs*** *are given by*:

- $[\![\sigma_i\psi_1 \ldots \psi_n]\!] = (\Sigma, \sigma_i)([\![\psi_1]\!], \ldots, [\![\psi_n]\!])$.

[8] The smaller the κ-value is, the more plausible the world is.

[9] $M([\![\pi]\!])$ denotes the updated model, $[\![\pi]\!]_M$ expresses the relation between the original model and the updated model.

- $[\![\pi.\rho]\!] = [\![\pi]\!].[\![\rho]\!]$.
- $[\![\pi + \rho]\!] = [\![\pi]\!] + [\![\rho]\!]$.

Axiomatic system PL We first present the static part $\mathsf{PL_S}$ of the axiomatic system PL:

1. All propositional tautologies.
2. $B_j^m(\varphi \to \psi) \to (B_j^m\varphi \to B_j^m\psi)$ $\quad$ B_j^m-*distribution*
3. $K_j(\varphi \to \psi) \to (K_j\varphi \to K_j\psi)$ $\quad$ K_j-*distribution*
4. $K_j\varphi \to \varphi$
5. $B_j^m\varphi \to K_jB_j^m\varphi$ $\quad$ for all $m \in \mathbb{N}$
6. $\neg B_j^m\varphi \to K_j\neg B_j^m\varphi$ $\quad$ for all $m \in \mathbb{N}$
7. $B_j^m\varphi \to B_j^{m'}\varphi$ $\quad$ for all $m \geq m'$
8. $K_j\varphi \leftrightarrow B_j^m\varphi$ $\quad$ for all $m \geq Max$

9. From $\vdash \varphi$ and $\vdash \varphi \to \psi$ infer $\vdash \psi$
10. From $\vdash \varphi$ infer $\vdash B_j^m\varphi$ $\quad$ B_j^m-*generalization*
11. From $\vdash \varphi$ infer $\vdash K_j\varphi$ $\quad$ K_j-*generalization*

In the static system $\mathsf{PL_S}$, besides all propositional tautologies, the important axioms are Axioms 4–8, which are about the properties of knowledge and belief. Axiom 5 is on *positive* introspection, "if one believes something, then one knows that one believes it". Axiom 6 is about *negative* introspection, "if one does not believe something, then one knows that one does not believe it". Axiom 7 states that a stronger belief implies a weaker one.

Adding the following dynamic axioms and rules to the above static system $\mathsf{PL_S}$, we get the full system PL:

12. $[\pi](\varphi \to \psi) \to ([\pi]\varphi \to [\pi]\psi)$ $\quad$ $[\pi]$-*distribution*
13. $[\sigma_i, \psi]p \leftrightarrow (\psi_i \to p)$

14. $[\sigma_i, \psi]\neg\chi \leftrightarrow (\psi_i \to \neg[\sigma_i, \psi]\chi)$

15. $[\sigma_i, \psi]\varphi \wedge \chi \leftrightarrow ([\sigma_i, \psi]\varphi \wedge [\sigma_i, \psi]\chi)$

16. $[\sigma_i, \psi]K_j\varphi \leftrightarrow (\psi_i \to \bigwedge\{K_j[\sigma_k, \psi]\varphi : \sigma_k \sim_j \sigma_i\})$

17. $[\sigma_i, \psi]B_j^m\varphi \leftrightarrow (\psi_i \to \bigwedge\{B_j^{l-1}\neg\psi_k \wedge \neg B_j^l\neg\psi_k \to B_j^{m+l-\kappa_j^*(\sigma_k)}$ $[\sigma_k, \psi]\varphi : \sigma_k \sim_j \sigma_i$, and $l \in \{0, \ldots, Max\}\})$ where $m < Max$

18. $[\pi][\rho]\varphi \leftrightarrow [\pi.\rho]\varphi$

19. $[\pi + \rho]\varphi \leftrightarrow [\pi]\varphi \wedge [\rho]\varphi$

20. Form $\vdash \varphi$ infer $\vdash [\pi]\varphi$ $[\pi]$-*generalization*

In the dynamic part, besides the axioms and rules from dynamic logic, Axioms 13–17 are so-called *reduction axioms*, of which Axiom 16 and Axiom 17 are crucial, because they express the interaction between knowledge and action, belief and action, respectively. As we emphasized before, such axioms make it possible to relate agents' knowledge or belief after the execution of some action with their knowledge or belief beforehand.

For the axiomatic system above, Aucher has given a completeness proof. In the next section we will present an improved completeness proof, which turns out to be much simpler. We will make use of one theorem of $\mathsf{PL_S}$. The detailed proof of this theorem will also be provided. Given the reduction axioms, it is easy to see that $\mathsf{PL_S}$ is equivalent to the full system PL. So we only need to consider the completeness for $\mathsf{PL_S}$.

2.2 Simplified Completeness Proof

2.2.1 Formal Derivation in $\mathsf{PL_S}$

In order to get more familiar with derivations in $\mathsf{PL_S}$, we will prove several theorems in $\mathsf{PL_S}$, the last one will contribute in the new completeness proof. We omit the definition of derivation, which is classical.

Theorem 2.2.1 $K_j\varphi \to B_j^k\varphi$ for all $k \in \mathbb{N}$

1. $B_j^{Max}\varphi \to B_j^k\varphi$ for $k \leq Max$ (Axiom 7)
2. $K_j\varphi \to B_j^{Max}\varphi$ (Axiom 8, $k = Max$)
3. $K_j\varphi \to B_j^k\varphi$ for $k \leq Max$ (1, 2, PC[10])
4. $K_j\varphi \to B_j^k\varphi$ for $k > Max$ (Axiom 8, PC)
5. $K_j\varphi \to B_j^k\varphi$ for all k (3, 4, PC)

The above theorem is easy to understand, because if an agent knows something then she believes it up to any degree. Now we look at the following theorem, which means if an agent believes something up to degree k if and only if she believes up to degree k that she believes up to degree k that something holds.

Theorem 2.2.2 $\vdash B_j^k B_j^k\varphi \leftrightarrow B_j^k\varphi$ *for all* $k \in \mathbb{N}$

1. $\neg K_j \neg B_j^k\varphi \to B_j^k\varphi$ (Axiom 6, PC)
2. $\langle K_j\rangle B_j^k\varphi \to B_j^k\varphi$ (1, ML)
3. $B_j^k\varphi \wedge \langle B_j^k\rangle\top \to \langle B_j^k\rangle(\varphi \wedge \top)$ (ML)
4. $B_j^k\varphi \wedge \langle B_j^k\rangle\top \to \langle B_j^k\rangle\varphi$ (3, PC)
5. $B_j^k\varphi \to \langle B_j^k\rangle\varphi$ (4, PC)
6. $\langle B_j^k\rangle\varphi \to \langle K_j\rangle\varphi$ (Theorem 2.2.1, ML)
7. $B_j^k\varphi \to \langle K_j\rangle\varphi$ (5, 6, PC)
8. $B_j^k B_j^k\varphi \to \langle K_j\rangle B_j^k\varphi$ (7, PC)
9. $B_j^k B_j^k\varphi \to B_j^k\varphi$ (2, 8, PC)
10. $K_j B_j^k\varphi \to B_j^k B_j^k\varphi$ (Theorem 2.2.1, PC)
11. $B_j^k\varphi \to B_j^k B_j^k\varphi$ (Axiom 5, 10, PC)
12. $B_j^k B_j^k\varphi \leftrightarrow B_j^k\varphi$ (9, 11, PC) ■

[10]Here, PC denotes ‘Propositional Logic’, and ML denotes ‘Modal Logic’

As we already know that in the system $\mathsf{PL_S}$, $B_j^k\varphi \to \varphi$ is not its axiom since our belief up to a degree may be not true. But we will prove the following Theorem 2.2.3 in $\mathsf{PL_S}$, it means that the agent believes up to degree k that what she believes up to degree k is true.

Theorem 2.2.3 $\vdash B_j^k(B_j^k\varphi \to \varphi)$ *for all* $k \in \mathbb{N}$

1. $\neg B_j^k\varphi \vee B_j^k\varphi$ (PC)
2. $K_j\neg B_j^k\varphi \vee K_jB_j^k\varphi$ (1, axiom 5, PC)
3. $B_j^k\neg B_j^k\varphi \vee B_j^kB_j^k\varphi$ (2, Theorem 2.2.1, PC)
4. $B_j^k\neg B_j^k\varphi \vee B_j^k\varphi$ (3, Theorem 2.2.2, PC)
5. $\neg B_j^k\varphi \to \neg B_j^k\varphi \vee \varphi$ (PC)
6. $B_j^k\neg B_j^k\varphi \to B_j^k(\neg B_j^k\varphi \vee \varphi)$ (5, B_j^k-*generalization*)
7. $\varphi \to \neg B_j^k\varphi \vee \varphi$ (PC)
8. $B_j^k\varphi \to B_j^k(\neg B_j^k\varphi \vee \varphi)$ (7, B_j^k-*generalization*)
9. $(B_j^k\neg B_j^k\varphi \vee B_j^k\varphi) \to B_j^k(\neg B_j^k\varphi \vee \varphi)$ (6, 8, PC)
10. $B_j^k(\neg B_j^k\varphi \vee \varphi)$ (4, 9, PC)
11. $B_j^k(B_j^k\varphi \to \varphi)$ (10, PC) ■

This theorem can be thought of as a weaker formulation of $B_j^k\varphi \to \varphi$. From the point view of correspondence theory, this means the worlds that agents can see are reflexive. We will see how this theorem contributes considerably to the new completeness proof in the next subsection.

2.2.2 Completeness Theorem

Lemma 2.2.1 *Let* Γ *be a maximal* $\mathsf{PL_S}$*-consistent set of formulas . Then it holds for all* φ, ψ *that:*

- *exactly one of* φ *and* $\neg\varphi$ *is in* Γ.

- $\varphi \wedge \psi \in \Gamma$ *iff* $\varphi \in \Gamma$ *and* $\psi \in \Gamma$.
- *if* $\varphi \in \Gamma$ *and* $\varphi \to \psi \in \Gamma$, *then* $\psi \in \Gamma$.
- *if* $\vdash_{\mathsf{PL_S}} \varphi$, *then* $\varphi \in \Gamma$.

Proof It is a classical proof that we do not give. ■

Lemma 2.2.2 *Every* $\mathsf{PL_S}$*-consistent set of formula* Γ *can be extended to a maximal* $\mathsf{PL_S}$*-consistent set.*

Proof This is also a classical proof. ■

Completeness Theorem *The system* $\mathsf{PL_S}$ *is strongly complete with respect to its canonical model.*

Proof To prove the completeness of $\mathsf{PL_S}$, it suffices to show that

> Every $\mathsf{PL_S}$-consistent set Γ of formulas is satisfiable on some belief epistemic model. (*)

To get this, we define the canonical model as follows:

$M^c = (W^c, \sim_j, \kappa_j, V)$

- $W^c = \{w_V : V$ maximal $\mathsf{PL_S}$-consistent set$\}$
- $\sim_j = \{(w_V, w_W) : V/K_j \subseteq W\}$ where $V/K_j = \{\varphi : K_j\varphi \in V\}$
- $\kappa_j(w_W)= \min\{k : W/B_j^k \subseteq W\}$
- $w_W \in V(p)$ iff $p \in W$

Now we need to show that

$$\varphi \in V \Leftrightarrow M^c, w_V \models \varphi.$$

By induction on the structure of formula φ. We only consider the belief case

($\Rightarrow$) Assume $B_j^k\varphi \in V$ (**), to show $M^c, w_V \models B_j^k\varphi$, i.e. to show that

For every w_W, if $w_V \sim_j w_W, \kappa_j(w_W) \leq k$ then $M^c, w_W \vDash \varphi$.

Assume furthermore that for every $w_W, w_V \sim_j w_W$ and $\kappa_j(w_W) \leq k$. Since $B_j^k\varphi \to K_j B_j^k\varphi$ is axiom, $B_j^k\varphi \to K_j B_j^k\varphi \in V$, together with (**) we get $K_j B_j^k\varphi \in V$, (V is an MCS). Since $w_V \sim_j w_W$, by definition of $\sim_j$, we have $B_j^k\varphi \in W$. Since $\kappa_j(w_W) \leq k$, by the definition of $\kappa_j(w_W)$, from $B_j^k\varphi \in W$ we can get $\varphi \in W$. According to the inductive hypothesis, $M^c, w_W \vDash \varphi$.

($\Leftarrow$) Assume $B_j^k\varphi \notin V$. To show $M^c, w_V \nvDash B_j^k\varphi$. Let set Σ be $\{\neg\varphi\} \cup \{B_j^k\varphi \to \varphi : \varphi \in V\} \cup \{\varphi : K_j\varphi \in V\}$. We claim that Σ is consistent. For suppose not, then there are $\neg\varphi$, $B_j^k\beta_1 \to \beta_1, \dots B_j^k\beta_s \to \beta_s, \alpha_1, \dots, \alpha_r$ in Σ, s.t.

$$\vdash (B_j^k\beta_1 \to \beta_1) \wedge \dots \wedge (B_j^k\beta_s \to \beta_s) \wedge (\alpha_1 \wedge \dots \wedge \alpha_r) \to \varphi$$
$$\vdash B_j^k((B_j^k\beta_1 \to \beta_1) \wedge \dots \wedge (B_j^k\beta_s \to \beta_s) \wedge (\alpha_1 \wedge \dots \wedge \alpha_r)) \to B_j^k\varphi$$
$$\vdash B_j^k(B_j^k\beta_1 \to \beta_1) \wedge \dots \wedge B_j^k(B_j^k\beta_s \to \beta_s) \wedge (B_j^k\alpha_1 \wedge \dots \wedge B_j^k\alpha_r) \to B_j^k\varphi$$

By Theorem 2.2.3, we can drop the first part of the formula above and obtain

$$\vdash B_j^k\alpha_1 \wedge \dots \wedge B_j^k\alpha_r \to B_j^k\varphi$$

Since for any α_i, $1 \leq i \leq k$ we have $K_j\alpha_i \in V$, since $K_j\alpha_i \to B_j^k\alpha_i \in V$ (axiom), we get $B_j^k\alpha_i \in V$, $B_j^k\alpha_1 \wedge \dots \wedge B_j^k\alpha_r \in V$, so $B_j^k\varphi \in V$. But this is impossible because $\neg B_j^k\varphi \in V$. We conclude that Σ is consistent. By Lemma 2.2.2, there is an MCS W extending Σ and $\neg\varphi \in W$, so $\varphi \notin W$. According to the inductive hypothesis, $M^c, w_W \nvDash \varphi$. ■

In this section, we have given derivations of three theorems in the system $\mathsf{PL_S}$ and a simpler completeness proof for the system $\mathsf{PL_S}$. In the next two sections we will extend the system $\mathsf{PL_S}$ with notions of *common knowledge* and *common belief* step by step. We first add the common knowledge operator to system $\mathsf{PL_S}$, which will make the system more expressive.

2.3 Incorporating Common Knowledge

We will give an extended axiomatic system with common knowledge and present the completeness proof in the line of propositional dynamic logic.

2.3.1 Adding Common Knowledge

We augment the language $\mathcal{L}_S$ with a new operator C_G. The resulting language is denoted by $\mathcal{L}_S^C$. $C_G\varphi$ is read "it is common knowledge among the group G that φ". We often use $E_G\varphi$ to express "everyone in the group G knows that φ", i.e. $\bigwedge_{j\in G} K_j\varphi$. Putting it formally, we have the definition below:

Definition 2.3.1 *Let M be a belief epistemic model, we have*

$$M, w \models E_G\varphi \text{ iff } M, w \models K_j\varphi \text{ for all } j \in G.$$

Intuitively, $C_G\varphi$ is true, if everyone in G knows φ, everyone in G knows that everyone in G knows φ, etc. We first look at a simple example as below:

Example $G = \{1, 2, 3\}$, φ is common knowledge among the group G. Then we have:

$K_1\varphi$, $K_2\varphi$, $K_3\varphi$, $K_1K_2\varphi$, $K_1K_3\varphi$, $K_1K_2K_3\varphi$, etc.

Now some useful notations: Let $E_G^0\varphi$ be an abbreviation for φ, and $E_G^{n+1}\varphi$ for $E_GE_G^n\varphi$ and in particular, $E_G^1\varphi$ for $E_G\varphi$. We have the definition below:

Definition 2.3.2 *Let M be a belief epistemic model, then*

$$M, w \models C_G\varphi \text{ iff } M, w \models E_G^n\varphi \text{ for } n = 1, 2, \dots$$

We go ahead and give the following definition, which turns out to be very useful in the completeness proof, discussions and many applications.

Definition 2.3.3 *v is G-**reachable from w in m steps** ($m \geq 1$) if there exists a sequence of states, $w_0, w_1, \dots, w_m$ s.t. $w_0 = w$ and $w_m = v$ and for all $0 \leq i \leq m-1$, $(w_i, w_{i+1}) \in K_j$ for some $j \in G$. If for some $m \geq 1$, v is G-reachable from w in m steps, then we say v is G-**reachable from w** .*

Lemma 2.3.4 *Let M be a belief epistemic model,*

(1) $M, w \models E_G^n\varphi$ *iff* $M, v \models \varphi$ for all v *that are G-reachable from w in n steps.*

(2) $M, w \models C_G\varphi$ *iff* $M, v \models \varphi$ *for all* v *that are* G*-reachable from* w.

Proof Induction on n, it is straightforward to get (1), (2) follows from (1).

Concerning the axiomatics, we add new *axioms* and a *rule* to the system $\mathsf{PL_S}$, obtaining a new system $\mathsf{PL_S^C}$:

$C1$. $E_G\varphi \leftrightarrow \bigwedge_{j\in G} K_j\varphi$

$C2$. $C_G\varphi \leftrightarrow E_G(\varphi \wedge C_G\varphi)$ (*Fixed-point Axiom*)

$RC1$. From $\vdash \varphi \rightarrow E_G(\psi \wedge \varphi)$ infer $\vdash \varphi \rightarrow C_G\psi$ (*Induction Rule*)

2.3.2 Completeness of $\mathsf{PL_S^C}$

In order to prove the completeness of the system $\mathsf{PL_S^C}$, we follow Fagin, Halpern, Moses & Vardi 1995. The method is in line of propositional dynamic logic . The completeness proof is non-trivial, since the language with transitive closure of the accessibility relation '$\rightarrow_j$' is not compact, infinite maximal consistent sets of formulas may be not satisfiable. So we only take maximal consistent set of formulas of some parts of the language. Now we first define such parts as 'closure':

Definition 2.3.5 *The* ***closure*** *of* φ *is the minimal set* $\Phi \subseteq \mathcal{L}_S^C$ *s.t.*

- $\varphi \in \Phi$.
- *if* $\psi \in \Phi$ *and* χ *is a subformula of* ψ, *then* $\chi \in \Phi$.
- *if* $\psi \in \Phi$ *and* ψ *itself is not a negation, then* $\neg\psi \in \Phi$.
- *if* $C_G\psi \in \Phi$, *then* $K_j(\psi \wedge C_G\psi) \in \Phi$ *for all* $j \in G$.

Note that the closure of any formula φ yields a finite set of formulas Φ.

Definition 2.3.6 *Let* Φ *be a closure. A finite set of formulas* $\Gamma \subseteq \Phi$ *is* ***maximal consistent*** *in* Φ *iff:*

- Γ *is consistent.*
- *There is no* $\Delta \subseteq \Phi$ *such that* $\Gamma \subset \Delta$ *and* Δ *is consistent.*

Lemma 2.3.7 *Let Φ be a closure of a formula φ. If $\Gamma \subseteq \Phi$ is consistent, then there is a set $\Gamma' \subseteq \Phi$ such that $\Gamma \subseteq \Gamma'$ and Γ' is maximal consistent in Φ.*

Proof The proof is classical.

Definition 2.3.8 *Given a formula φ, the model $M^\varphi = (W^\varphi, \sim_j^\varphi, V^\varphi)$ is given by*

- $W^\varphi = \{w_V : V$ *is a maximal consistent set in* $\Phi\}$
- $\sim_j^\varphi = \{(w_V, w_W) : V/K_j \subseteq W\}$ *where* $V/K_j = \{\varphi : K_j\varphi \in V\}$
- $w_W \in V^\varphi(p)$ *iff* $p \in W$

Truth Lemma *If $V \in W^\varphi$, then for all $\psi \in \Phi$, it holds that $\psi \in V$ iff $M^\varphi, w_V \models \psi$.*

Proof By induction on the structure of formula ψ, we only consider the common knowledge case, i.e. we will prove

$$C_G\psi \in V \text{ iff } M^\varphi, w_V \models C_G\psi.$$

($\Rightarrow$) Assume $C_G\psi \in V$, we will prove a stronger claim: if w_W is G-reachable from w_V in k steps, then $\psi \in W$ and $C_G\psi \in W$.
By induction on k.
Base case: $k = 1$. Since $V \in W^\varphi$, i.e. V is a MCS. By Axiom $C2$, it follows $E_G(\psi \wedge C_G\psi)$. Then if w_W is G-reachable from w_V in one step, we have $(\psi \wedge C_G\psi) \in W$. Since W is a MCS, it follows that $\psi \in W$ and $C_G\psi \in W$.
Inductive step: Assume that the conclusion holds for k, we prove the case for $k+1$.
If w_W is G-reachable from w_V in $k+1$ step, then there exists W' such that $w_{W'}$ is G-reachable from w_V in k steps and w_W is G-reachable from $w_{W'}$ in one step. By the induction hypothesis, then $\psi \in W$ and $C_G\psi \in W$. Hence ψ and $C_G\psi$ are in W. So we know that $\psi \in W$ for all w_W that are G-reachable from w_V. By the main induction hypothesis, $M^\varphi, w_V \models \psi$ for all w_W that are G-reachable from w_V. So, $M^\varphi, w_V \models C_G\psi$.

($\Leftarrow$) Assume $M^\varphi, w_V \models C_G\psi$. We can describe each world w_W of M^φ by the conjunction of the formulas in W. φ_W is taken to denote such conjunction, which is a formula in $\mathcal{L}_S^C$, since W is a finite set. Let $\mathcal{W}$ =$\{W$ is a maximal consistent set in $\Phi : M^\varphi, w_W \models C_G\psi\}$. Define $\varphi_{\mathcal{W}}$ to be $\bigvee_{w\in\mathcal{W}} \varphi_W$. That is, this disjunction describes all of the states where $C_G\psi$ holds. Since the set $\mathcal{W}$ is finite, it follows that $\varphi_{\mathcal{W}}$ is a formula in $\mathcal{L}_S^C$. Now we have to prove that

$$\vdash \varphi_{\mathcal{W}} \to E_G(\psi \wedge \varphi_{\mathcal{W}}). \quad (1)$$

First it is easy to prove that

$$\vdash \varphi_W \to K_j\psi. \quad (2)$$

We define $\bar{\mathcal{W}} = \{W$ is a maximal consistent set in $\Phi : M^\varphi, w_W \not\models C_G\psi\}$. If $W \in \mathcal{W}$, and $W' \in \bar{\mathcal{W}}$, then $\vdash \varphi_W \to K_j\neg\varphi_{W'}$ (3). From (2) and (3) we get

$$\vdash \varphi_W \to K_j(\psi \wedge (\textstyle\bigwedge_{W'\in\bar{\mathcal{W}}} \neg\varphi_{W'})).$$

It can be shown that $\vdash \varphi_{\mathcal{W}} \leftrightarrow (\bigwedge_{W'\in\bar{\mathcal{W}}} \neg\varphi_{W'})$, so we get (1). By the Induction Rule, we get

$$\vdash \varphi_{\mathcal{W}} \to C_G\psi.$$

Since $V \in \mathcal{W}$, we have $\vdash \varphi_V \to \varphi_{\mathcal{W}}$, so

$$\vdash \varphi_V \to C_G\psi.$$

It follows that $C_G\psi \in V$. Otherwise, $\neg C_G\psi \in V$, then V is not $\mathcal{L}_S^C$-consistent, contradiction. ■

Completeness *If* $\models \varphi$, *then* $\vdash \varphi$.

Proof Suppose it is not the case that $\vdash \varphi$. Then $\{\neg\varphi\}$ is consistent and there is an *MCS* V in the closure φ of $\neg\varphi$ s.t. $\neg\varphi \in V$. By the Truth Lemma, we get M^c, $w_V \models \neg\varphi$, so it is not the case that M^φ, $w_V \models \varphi$. ■

We have seen how to incorporate the notion of common knowledge into the system $\mathsf{PL_S}$ in this section, another equally important notion is common belief, we will consider how to add common belief operator to the system in the next section.

2.4 Incorporating Common Belief

In this section, we will introduce common belief operator in two different ways, which yields two different axiomatic systems, the completeness of these systems will be given.

2.4.1 Common Belief in Pure Belief Version

A new operator D_G^k is added to the language $\mathcal{L}_S^C$, we obtain the resulting language $\mathcal{L}_S^{CD}$. $D_G^k\varphi$ denotes "it is common belief up to degree k among the group G that φ". Similarly, $F_G^k\varphi$ expresses "everyone in group G believes up to degree k that φ". i.e. $\bigwedge_{j\in G} B_j^k\varphi$. We have the following formal definition:

Definition 2.4.1 *Let M be a belief epistemic model, we have*

$$M, w \models F_G^k\varphi \textit{ iff } M, w \models B_j^k\varphi \text{ for all } j \in G.$$

According to an intuition similar to that in common knowledge, $D_G^k\varphi$ is true, if everyone in G believes up to degree k that φ, everyone in G believes up to degree k that that everyone in G believes up to degree k that φ, etc. Similar abbreviations as in the preceding section are applicable. Let $F_G^{0,k}\varphi$ be an abbreviation for φ, and $F_G^{n+1,k}\varphi$ for $F_G^kF_G^{n,k}\varphi$, and in particular, $F_G^{1,k}\varphi$ for $F_G^k\varphi$. And we have the following definition:

Definition 2.4.2 *Let M be a belief epistemic model, then*

$$M, w \models D_G^k\varphi \textit{ iff } M, w \models F_G^{n,k}\varphi \textit{ for } n = 1, 2, \ldots$$

We also give the following useful definition:

Definition 2.4.3 *v is G-B-**reachable from w in m steps**[11] ($m \geq 1$) if there exists a sequence of states, $w_0, w_1, \ldots, w_m$ s.t. $w_0 = w$ and $w_m = v$ and for all $0 \leq i \leq m-1$, $(w_i, w_{i+1}) \in B_j^k$ for some $j \in G$. If for some $k \geq 1$, v is G-B-reachable from w in m steps, then we say v is G-B-**reachable from w**.*

Lemma 2.4.4 *Let M be a belief epistemic model,*

[11] B in *G-B-reachable* for 'belief'.

(1) $M, w \models F_G^{n,k}\varphi$ *iff* $M, v \models \varphi$ *for all* v *that are* G*-reachable from* w *in* n *steps.*

(2) $M, w \models D_G^k\varphi$ *iff* $M, v \models \varphi$ *for all* v *that are* G*-*B*-reachable from* w*.*

Proof Induction on n, it is straightforward to get (1), (2) follows from (1).

We now turn to a variation of the example in the previous section:

Example $G = \{1, 2, 3\}$, φ is common belief up to degree k among the group G. Then we have:

$B_1^k\varphi$, $B_2^k\varphi$, $B_3^k\varphi$, $B_1^kB_2^k\varphi$, $B_1^kB_3^k\varphi$, $B_1^kB_2^kB_3^k\varphi$, etc.

Note that the plausibility of common belief is the same as that of the individual belief. Now we discuss the axiomatic system with common belief. The following new *axioms* and *rule* are proposed to be added to the system $\mathsf{PL}_{\mathsf{S}}^{\mathsf{C}}$, we obtain the system $\mathsf{PL}_{\mathsf{S}}^{\mathsf{CD}}$:

$D1$. $F_G^k\varphi \leftrightarrow \bigwedge_{j\in G} B_j^k\varphi$

$D2$. $D_G^k\varphi \leftrightarrow F_G^k(\varphi \wedge D_G^k\varphi)$ (*Fixed-point Axiom*)

$RD1$. From $\vdash \varphi \rightarrow F_G^k(\psi \wedge \varphi)$ infer $\vdash \varphi \rightarrow D_G^k\psi$ (*Induction Rule*)

2.4.2 Common Belief defined with Common knowledge

Recall how we have just defined the notion of common belief up to degree k. We state it again below to compare it with our new proposed definition:

> $D_G^k\varphi$ *is true, if everyone in* G *believes up to degree* k *that* φ*, everyone in* G ***believes up to degree*** k *that that everyone in* G *believes up to degree* k *that* φ*, etc.*

Note that we define common belief up to a degree k in a very similar way that we define common knowledge, another point is that we only refer to the belief operator. In this sense, we may say that the above definition is in a *pure belief version*. We now propose alternative definition for common belief, i.e. we define it with a knowledge operator and a belief operator:

> *$D^k_G\varphi$ is true, if everyone in G believes up to degree k that φ, everyone in G **knows** that everyone in G believes up to degree k that φ, etc.*

In fact, we are talking about common belief up to degree k in term of belief up to a degree and *common knowledge*. To make this clear, we look at the above example again:

Example $G = \{1, 2, 3\}$, φ is common belief up to degree k among the group G. This time we have the different formulas:

$F^k_G\varphi$ [12], $K_1F^k_G\varphi$, $K_2F^k_G\varphi$, $K_3F^k_G\varphi$, $K_1K_2F^k_G\varphi$, etc.

This means φ is a common belief up to degree k among the group G if everyone in G believes φ up to degree k, which is common knowledge in G. So we have

$$D^k_G\varphi =: C_GF^k_G\varphi.$$

Defining the common belief operator with a common knowledge and belief operator is different from what we first proposed, where the common belief operator D^k_G is a completely new operator, since the common belief operator is in this variant only an abbreviation for $C_GF^k_G$. Concerning the axiomatic system, we will add the following axioms to the system $\mathsf{PL}^{\mathsf{C}}_{\mathsf{S}}$, obtaining $\mathsf{PL}^{\mathsf{CD}'}_{\mathsf{S}}$:

D1'. $F^k_G\varphi \leftrightarrow \bigwedge_{j\in G} B^k_j\varphi$

D2'. $D^k_G\varphi \leftrightarrow E_G(\varphi \wedge D^k_G\varphi)$ (*Fixed-point Axiom*)

RD1'. From $\vdash \varphi \to F^k_G(\psi \wedge \varphi)$ infer $\vdash \varphi \to D^k_G\psi$ (*Induction Rule*)

Completeness The completeness proofs of these new systems with common belief turn out to be easy to get. For the system $\mathsf{PL}^{\mathsf{CD}}_{\mathsf{S}}$, the completeness proof is very similar to the case of common knowledge. If we look at the completeness proof for common knowledge, we only refer to the transitive closure of the accessible relation, which exactly is what we have according to the fixed point axiom for common belief. We are not going to repeat it here. Another way to get its completeness proof is induced by Aucher's work on the relation between BMS-model and

[12] It abbreviates $(B^k_1\varphi \wedge B^k_2\varphi \wedge B^k_3\varphi)$.

belief epistemic model. The main idea is: first redefine the BMS-model into a belief epistemic model, i.e. BMS belief epistemic model. Then it is equivalent to our defined belief epistemic model, since we already have the completeness of BMS update logic system, then it follows the completeness of system $\mathsf{PL}_{\mathsf{S}}^{\mathsf{CD}}$. For details, we refer to Aucher 2003.[13] For the system $\mathsf{PL}_{\mathsf{S}}^{\mathsf{CD}'}$, its completeness proof is even straightforward, since D_G^k is only an abbreviation for $C_G F_G^k$, the completeness of $\mathsf{PL}_{\mathsf{S}}^{\mathsf{CD}'}$ follows directly from that of the system $\mathsf{PL}_{\mathsf{S}}^{\mathsf{C}}$.

In the just above sections, we have done some technical work to the system PL_{S}, presenting an improved completeness proof, extending it with the notion of common knowledge and common belief. Most of such work is related to the static aspect of the system. We have not considered the dynamic aspect, the core of the plausibility logic: How does an agent change her plausibility of belief in the update process? Aucher presented a rule for this kind of calculation, the new plausibility comes from those of the previous state and action. In the next section, we will propose a new updating rule which allows for different types of agents.

2.5 Plausibility Updating

Plausibility Updating Rule Let w be the original state, a the action happening in the state, their plausibilities are $\kappa_j(w)$ and $\kappa_j^*(a)$ respectively. To compute the plausibility of a new state $\kappa'_j(w,a)$ in an updated model $\boldsymbol{M} \times \boldsymbol{A}$, Aucher gave the following rule:

$$\kappa'_j(w,a) = Cut_{Max}(\kappa_j(w) + \kappa_j^*(a) - \kappa_j^w(\varphi)) \quad (\kappa)$$

where $\varphi = PRE_a$, $\kappa_j^w(\varphi) = min\{\kappa_j(v) : v \in V(\varphi) \text{ and } v \sim_j w\}$ and as we know, Cut_{Max} is a technical device ensuring that the new κ-value fits in the range of the κ scale of the new belief epistemic model (see definition in section 2.1.). With the (κ) rule, one can calculate the plausibility of a new state (w,a) from those of the previous state and action.

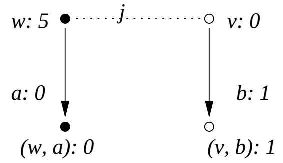
2.5.1 Motivating Observations

We now look at several scenarios and calculate the κ-value according to the above (κ) rule.

[13]He did not introduce the notion of common belief there, but it is not hard to do.

Scenario 1 There are two worlds w and v that agent j cannot distinguish, two actions a, b, with w and a the actual world and action respectively. Let $\kappa_j(w)$=5, $\kappa_j(v) = 0$, $\kappa_j^*(a) = 0$ and $\kappa_j^*(b) = 1$. $PRE_a = \{w\}$ and $PRE_b = \{v\}$.[14]

With the above rule, it is easy to get: $\kappa_j^{'}(w, a) = 0$, $\kappa_j^{'}(v, b) = 1$.



Since $\kappa_j^w(PRE_a)$ is the smallest κ-value in $\boldsymbol{M}$ among all worlds, in this case, the precondition of the action only has one world, and we have:

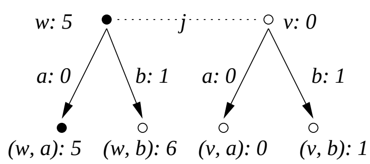
$$\kappa_j^w(PRE_a) = \kappa_j(w) \text{ and } \kappa_j^v(PRE_b) = \kappa_j(v)$$

So, the (κ) rule collapses, we are actually calculating with the following rule

$$\kappa_j'(w, a) = \kappa_j^*(a).$$

In other words, in this case the agent just took the plausibility of the previous action as that of the new state: in this way, she revised her belief.

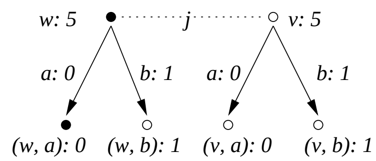
Scenario 2 With the same conditions as that in scenario 1, except that we set $PRE_a = PRE_b = \{w, v\}$. We have the following results described in the picture:



[14]For simplicity, we take the worlds where the action may happen as its precondition instead of some formula in $\mathcal{L}$.

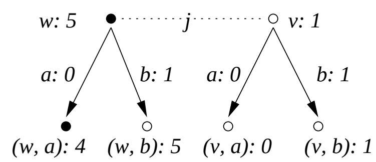
In Scenario 2, we take $PRE_a = PRE_b = \{w, v\}$. The intuition is: When an agent dwells in indistinguishable worlds and considers possible actions, she may think those actions probably happen in all her indistinguishable worlds because her knowledge is the same everywhere in the indistinguishable worlds. The following Scenarios are also based on this intuition. We take both w and v as possible worlds in which actions a and b may take place, and only change the κ value assigned to the states. The next scenario is a special case, i.e. the agent gives the indistinguishable worlds the same plausibility. Then we have the following scenario.

Scenario 3 The same as scenario 2, but with $\kappa(v)$=5. Then the results are pictured as follows:



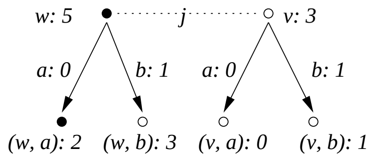
This scenario is an extreme case, but the results are intuitively correct. If the agent assigns the same plausibility to her indistinguishable worlds, then the plausibility changes in the same way. This is shown exactly in the above picture.

Scenario 4 The same as scenario 3, but with $\kappa(v)$=1. Then the results are shown as follows:



To compare how the different value of state v affects the results, we change Scenario 4 into:

Scenario 5 The same as Scenario 4, but with $\kappa(v)$=3. Then the results are shown as follows:



Analyzing Scenario 2, 4 and 5, we have the following interesting phenomena:

- Given different plausibilities to indistinguishable states, for the more plausible state, i.e. the state v the above scenarios, the agent in each case has the *same* revision strategy: She just takes the plausibility of the previous action as that of the new state. This is similar to what happens in Scenario 1.

- For the less plausible state, i.e. the state w in the above scenarios, the plausibility of the new state is produced in different ways:

  - If there is a most plausible state among the indistinguishable states, i.e.its κ value is 0, as shown in Scenario 2, then for the less plausible state, w, the plausibility of the new state is just the *sum* of that of the previous state and action because $\kappa_j^v(PRE_a) = \kappa_j^v(PRE_b) = 0$. For instance, in Scenario 2, given the plausibility of the state ($\kappa_j(w)$=5), after the action with plausibility 0(1), the agent assigns the new state the plausibility 5(6).
  - If there is no most plausible state, as in the Scenarios 4 and 5, for the less plausible state, w, its new plausibility value is affected by the value of the more plausible state, that is, when the agent assigns a plausibility to the new state, the precondition of the action plays a role in her estimation.

Furthermore, we have observed that there are several assumptions for the (κ) rule that we will discuss now:

Firstly, as we observed, in the above revision process, the agent always accepts the incoming information from an action, never rejects it. This is what happens in AGM belief revision, the new information has priority. In order to get the new information, agents even discard the old information when some inconsistency occurs. So we can say that in fact the (κ) rule is processing the revision in the sense of AGM. As we have seen, in some cases the agent even takes the plausibility of the previous action as that of the new state. However, there is one possibility: If the agent does not want to accept the incoming information or if she likes to accept it to some extent (not completely), how will she react then? How does she change her belief plausibilities? We often encounter such possibilities in real life.

Secondly, the (κ) rule assumes that different agents give the same weight to the previous state and action. This is also the case in AGM belief revision, or in classical belief revision theory: It was assumed that all the agents react similarly when they are in the same situation. However, it is quite possible that agents may behave differently, in particular, they may assign different *weights* to the previous state and action: Some agents take the previous state seriously, some attach importance to the previous action, this kind of variety makes human behaviors diverse and more interesting. How to describe such differences then?

Thirdly, the (κ) rule assumes that given the plausibility of the states and actions, when the agent considers the plausibility of the action, she will consider the preconditions of the actions. Does this fit with our intuition? For instance, in the above Scenario 5, when the agent is in the world w (its plausibility value is 5), when she thinks of two actions which she cannot distinguish (giving them two plausibility values, 0 and 1.), what she will think is just that if action with plausibility 0 happens at w, what is the new plausibility after such a update then? The same question for the action with plausibility 1. It is not clear to me how to incorporate the role of the precondition into this consideration.

So we may say that Aucher's (κ) rule is processing the revision in the sense of AGM belief revision, that is, the agents give priority to all incoming information, they behave in a similar manner, and they give equal weight to the previous state and action. In the next section, we will present a new plausibility updating rule based on our discussion above, trying to grasp all these desirable intuitions, i.e. the various attitudes of the agents when they encounter incoming information, and the different weight they give to the previous state and action.

2.5.2 Variations: Diversity of Agents

In this section we will first propose a new plausibility updating rule . Then we analyze the different cases induced by this rule because of the parameter variations, to illustrate the diversity of agents and how they revise their belief plausibility. Finally, we will explore the possibility of introducing such diversity in plausibility logic.

New plausibility updating rule *Let the weight that an agent j gives to the state w be λ and the weight to the action a be μ. If $\kappa_j(w) \geq \kappa_j^*(a)$, then the plausibility of the new state (w, a) is calculated by the following rule :*

$$\kappa_j^{'}(w,a) = \kappa_j(w) - \frac{\boldsymbol{\mu}}{\boldsymbol{\lambda}+\boldsymbol{\mu}}((\kappa_j(w) - \kappa_j^*(a)) \quad (\natural_1).$$

Or expressed differently:

$$\kappa_j^{'}(w,a) = \kappa_j^*(a) + \frac{\boldsymbol{\lambda}}{\boldsymbol{\lambda}+\boldsymbol{\mu}}((\kappa_j(w) - \kappa_j^*(a)) \quad (\natural_2).$$

We will see that these two variations yield the same results, only the difference in λ and μ is essential.

By duality, if $\kappa_j^*(a) \geq \kappa_j(w)$, then we can get the following rules:

$$\kappa_j^{'}(w,a) = \kappa_j^*(a) - \frac{\boldsymbol{\lambda}}{\boldsymbol{\lambda}+\boldsymbol{\mu}}((\kappa_j^*(a) - \kappa_j(w)) \quad (\natural_3).$$

Or expressed differently:

$$\kappa_j^{'}(w,a) = \kappa_j(w) + \frac{\boldsymbol{\mu}}{\boldsymbol{\lambda}+\boldsymbol{\mu}}((\kappa_j^*(a) - \kappa_j(w)) \quad (\natural_4).$$

In contrast to the (κ) rule, we drop the κ-value for the precondition; we also discard the *Cut* function for reasons which we will explain later in this section, and we add two parameters: λ and μ, to indicate the different weights that agents give to the states and actions. By the different values of these weights, we can distinguish different types of agents, some agents are eager to change their original belief once some action happens, others stick to their original belief and do not care about what has happened. The above rule looks like the barycenter calculation formula,[15] but not exactly,

[15] The barycenter is the center of mass of two or more bodies which are orbiting each other, and is the point around which both of them orbit. The distance from the center of a body to the barycenter in a simple two-body case can be calculated as follows:

$$r_b = r_2 \frac{m_2}{m_1+m_2}$$

where r_b is the distance from body 1 to the barycenter, r_2 is the distance between the two bodies, m_1 is the mass of body 1, and m_2 is the mass of body 2.
Actually our formula is not exactly the barycenter, but the inspiration comes from the barycenter formula whose aim is to get the distance value from one object to the barycenter. But we want to get the final κ value from the value of the state and action.

since our aim here is to get the plausibility for the (w, a) from that of w and a.

Note that these fours rules above can be simplified as

$$\kappa_j^{'}(w,a) = \frac{1}{\boldsymbol{\lambda}+\boldsymbol{\mu}}(\boldsymbol{\lambda}\kappa_j(w) + \boldsymbol{\mu}\kappa_j^*(a)) \quad (\natural).$$

But we prefer the slightly heavy ones in the following discussion in order to show how the new plausibility is calculated from that of the previous states and actions, in particular, what exact role the weight makes in such calculations. However, our discussions in other context may refer to the simplified version.

Thanks to the parameters, we will witness the distinctive behavior of different types of agents. Let us look at the following cases:

Case 1: μ=0
Highly conservative agents Recall that μ is the weight the agent gives to the action, μ=0 means the agent does not consider the effect of the action. Putting it formally, the ($\natural_2$) rule turns into

$$\kappa_j^{'}(w,a) = \kappa_j(w).$$

This means that the agent just takes the plausibility that she gives to the previous state as that of the new state. She does not consider the plausibility of the action at all. We may think that the agent throws away the incoming information, insists on what she had already. In this sense, we say the agent is highly conservative. And this also means that belief revision is not successful for the highly conservative agent since she ignores the new information after all.

Case 2: $\lambda > \mu$
Conservative agents: $\lambda > \mu$ means agents gives greater weight to the state than that to the action. To get a clear idea, we only look at the following concrete example: Let λ=8, μ=2, $\kappa_j(w)$=5 and $\kappa_j^*(a)$=1.

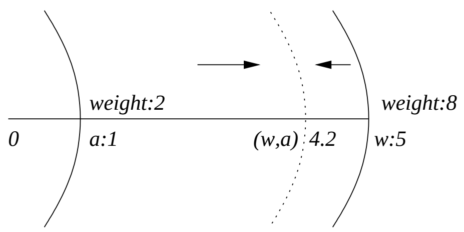
By ($\natural_1$), we have the following calculation:

$$\kappa_j^{'}(w,a) = 5 - \frac{2}{8+2}(5-1) = 5 - 0.8 = 4.2$$

By ($\natural_2$), we get the same value as follows:

$$\kappa_j^{'}(w,a) = 1 + \frac{8}{8+2}(5-1) = 1 + 3.2 = 4.2$$

From the assumption, we know that the agent gives a much greater weight to the state ($\lambda = 8$) than that to the action ($\mu = 2$), the result ($\kappa_j^{'}(w,a) = 4.2$) is much closer to the plausibility of the state ($\kappa_j(w) = 5$) than that to the plausibility of action ($\kappa_j^*(a) = 1$): the distance of the former is 0.8, but the latter is 3.2. This means the plausibility of the action has some, but only a limited effect on the agent's plausibility of the new state. In this sense, we say these agents are conservative. To make this more perspicuous, we give the picture below:



Plausibility Space

Where the *arc line* at the left (right) side denotes the plausibility of the action a (the state w), the *dotted arc line* denotes the plausibility of new state (w,a).

From the above pictures, we can easily see *why* the result we get by ($\natural_1$) is same as that by ($\natural_2$). To reach the dotted arc line, we can start from the arc line at the right side, which is what ($\natural_1$) describes. Similarly, we can also reach it from the arc line at the left side, which is what ($\natural_2$) says.

Case 3: λ=0

Highly radical agents As we know, λ is the weight that an agent gives to the previous state, λ=0 means that the agent does not consider the effect from the state. Formally, the ($\natural_1$) rule becomes

$$\kappa_j^{'}(w,a) = \kappa_j^*(a).$$

This implies that the agent just takes the plausibility that she gives to the action as the plausibility of new state. The plausibility of the previous state does not make any difference at all, in this sense, we say agent is highly radical. Our calculation with (κ) rule in the scenario 1, the new plausibility generated from the more plausible state, are of this kind: in such cases, agents are highly radical. It is easy to see, belief revision is successful in this case, the agent accepts the incoming information completely.

Case 4: $\lambda < \mu$
Radical agents: Again, we consider a variation of the above example: Let $\lambda = 2, \mu = 8$, $\kappa_j(w) = 5, \kappa^*_j(a) = 1$.

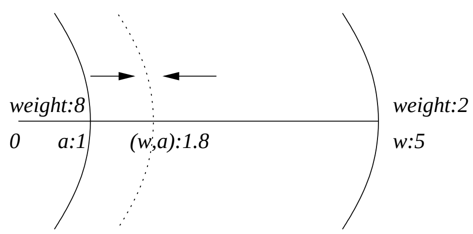
By ($\natural_1$),

$$\kappa_j^{'}(w, a) = 5 - \frac{8}{8+2}(5-1) = 5 - 3.2 = 1.8$$

By ($\natural_2$),

$$\kappa_j^{'}(w, a) = 1 + \frac{2}{8+2}(5-1) = 1 + 0.8 = 1.8$$

This time, the agent gives a much greater weight to the action ($\mu = 8$) than that to the state ($\lambda = 2$), the result ($\kappa_j^{'}(w, a) = 1.8$) is much closer to the plausibility of the action ($\kappa^*_j(a) = 2$) than that to the plausibility of state ($\kappa_j(w) = 5$): the distance of the former is 0.8, but the latter is 3.2. Again, this means the plausibility of the previous state has some, but a very limited effect on the agent's plausibility of the new state. In this sense, we say the agents are more radical. Here is the picture again:



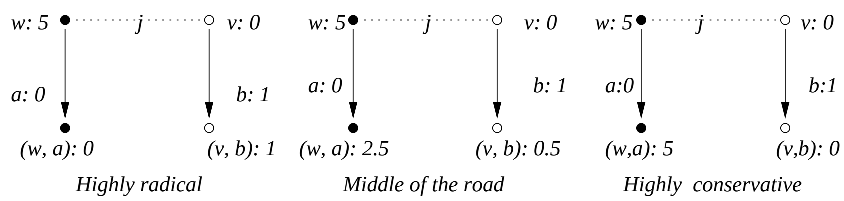
Plausibility Space

Case 5: $\lambda = \mu$
Middle of the road agents Besides conservative and radical agents, there are some agents, who give the previous state and action the same weight, they believe plausibility of states and action should play an equally important role in determining the plausibility of the new state. In other

words, in this case we take $\lambda = \mu$, obtaining the following rule:

$$\kappa_j^{'}(w,a) = \frac{1}{2}((\kappa_j(w) + \kappa_j^*(a)).$$

Note: Considering the κ-ranking function, the only problem is that we have to extend its range into non-ordinals, since quotients such as $\frac{\boldsymbol{\mu}}{\boldsymbol{\lambda}+\boldsymbol{\mu}}$ produce non-ordinals. Considering our qualitative approach, this minor change may not be so fundamental in this context. Now we look back at Scenario 1 again with our new updating rule. We only compare the following three cases: i.e. highly radical agents, Middle of the road agents, highly conservative agents as shown in the following picture:



As we have seen, the case of highly radical agents is exactly the same as what we got by (κ) rule in the Scenario 1. From this comparison, it is very clear how different agents update their belief plausibility. In the following text, we will give some further facts that we observed and discuss their intuitive consequences.

Fact 1 *Let $\kappa_j(w)$, $\kappa_j^*(a)$ and $\kappa_j^{'}(w,a)$ express the plausibility value of the previous state, the action after and the new state, as usual. Then the following results hold*:

If $\kappa_j(w) \leq \kappa_j^*(a)$, then $\kappa_j(w) \leq \kappa_j^{'}(w,a) \leq \kappa_j^*(a)$.

If $\kappa_j^*(a) \leq \kappa_j(w)$, then $\kappa_j^*(a) \leq \kappa_j^{'}(w,a) \leq \kappa_j(w)$.

We only look at the first result. For the part $\kappa_j(w) \leq \kappa_j^{'}(w,a)$, according to the rule ($\natural_3$), if $\kappa_j(w) \leq \kappa_j^*(a)$, $\kappa_j^*(a) - \kappa_j(w)$ is a non-negative number. Balanced with $\frac{\boldsymbol{\lambda}}{\boldsymbol{\lambda}+\boldsymbol{\mu}}$, it is still non-negative; subtracting it from $\kappa_j^*(a)$ makes the final value, $\kappa_j^{'}(w,a)$, less than $\kappa_j^*(a)$. Similarly, the rule ($\natural_4$) will give us the reason that the final value, $\kappa_j^{'}(w,a)$, is greater than $\kappa_j(w)$, i.e.

$\kappa_j^{'}(w,a) \leq \kappa_j^*(a)$. Actually it is much easier to understand the above results with our given pictures.

Now we are going to give another explanation to the above Fact from a new perspective: We assume that the plausibility of the action is somehow related to the plausibility of previous state. There are two kinds of cases as described below:

An action is ***negative*** *with respect to the previous state if it provides information undermining the belief in the previous state. An action is* ***positive*** *to the previous state if it provides information verifying belief in it.*

Put it in another way, if an action is negative, we take $\kappa_j(w) \leq \kappa_j^*(a)$, and if an action is positive, we take $\kappa_j^*(a) \leq \kappa_j(w)$. Intuitively speaking, in the former case, the plausibility of the new state should be greater than that of the previous state. In the latter case, the plausibility of the new state should be less than that of the previous state. This is exactly what we get in the above Fact.

From the above Fact, another Fact follows.

Fact 2 *Let $\kappa_j(w)$, $\kappa_j^*(a)$ and $\kappa_j^{'}(w,a)$ express the plausibility value of the previous state, the action after and the new state, as usual. Then the following result holds*:

$$\kappa_j^{'}(w,a) \leq \kappa_j^*(a) \quad \text{or} \quad \kappa_j^{'}(w,a) \leq \kappa_j(w).$$

In other words, the plausibility of the new state is bounded by the plausibility of either the previous state or the previous action. We need not do the rescaling work, this helps explain why we do not use the *Cut* function as in the (κ) rule.

Introducing diverse agents in plausibility logic We have seen five types of agents and how they update their belief plausibility. We know that when agents encounter new information, their behavior may be quite different. Plausibility logic has not addressed such differences yet. How to incorporate these into plausibility logic? The first thing we can do is to replace the subscript j of the belief operator with different types of agents, we can take the abbreviations hc_j, hr_j, r_j, c_j, m_j to express the highly conservative agent j, the highly radical agent j, the radical agent j, the conservative agent j, the middle of the road agent j. Then $B^k_{hc_j}\varphi$ expresses "the highly conservative agent j believes φ up to degree k". $B^k_{c_j}\varphi$ expresses "the conservative agent j believes φ up to degree k". In this way, we are able to express the properties of some specific agents. In fact, we divide the group G of agents into different subgroups, each representing one kind of agent. We can construct a logical system for each type of agent. This makes it possible to study the properties of specific kinds of agent, to compare their differences. Further research may even cover the interaction between such different types of agents. Meanwhile, it may be easier to obtain reduction axioms for each type of agent. For instance, for the highly conservative agents, the reduction axiom seems to be:

$$[\sigma_i, \psi]B^m_{hc_j}\varphi \leftrightarrow \bigwedge\{B^m_{hc_j}[\sigma_k, \psi]\varphi : \sigma_k \sim_{hc_j} \sigma_i\}$$

We will leave these topics for future investigation.

2.6 Conclusion

In this chapter, our work is closely related to the logical system PL. We first reviewed the basics of plausibility logic, the language and the semantics, its full system, including its static part and its dynamic part. Then we focused on the static plausibility logic $\mathsf{PL_S}$, and derived three theorems in it, one of which results in an improved completeness proof. After that, we extended the static plausibility logic with the notion of common knowledge, and with the notion of common belief up to a degree in two different ways. The completeness proofs for these extended system are also given. Concerning dynamic changes of plausibility, based on our analysis of Aucher's plausibility update rule, we proposed a new parameterized plausibility update rule. Thanks to the parameters, we can obtain different rules for the following five types of agents: Highly radical agents, Radical agents, Middle of the road agents, Highly conservative agents, Conservative agents. We can see clearly how these different agents update their belief plausibility in different ways.

However, this is just one perspective of seeing diversity in agents. As we have seen in the Mosaic dialogue in the introduction, agents often forget things, they may have different memory capacities. This is really another perspective that we will explore in the next chapter: Some agents have perfect memory, so-called Perfect Recall agents, they remember everything that happened, others have bounded memory and they just remember what happened in the last k-steps. In particular, there are agents who only observe the last-played action, so-called Memory-free agents. We will concentrate on these two extreme cases: Perfect Recall agents and Memory-free agents, and investigate how they update their information given their divergent memory capacities, what is the precise difference between them. We will also explore the idea that players with different memory abilities necessarily live together in social settings.

Chapter 3

Diversity of Logical Agents in Games

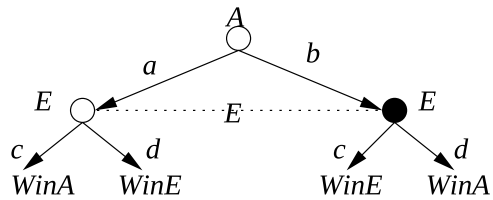
3.1 Varieties of Imperfection

Logical agents are usually taken to be epistemically perfect. But in reality, imperfections are inevitable. Even the most logical reasoners may have limited powers of observation of relevant events, generating uncertainty as time proceeds. In addition, agents can have processing bounds on their knowledge states, say, because of finite memory capacities. This note is a brief exploration of how different types of agents can be described in logical terms, and even co-exist inside the same logical system. Our motivating interest in undertaking this study concerns games with imperfect information, but our only technical results so far concern the introduction of imperfect agents into current logics for information update and belief revision. For a more complete discussion of the issues, refer to van Benthem 2001 and other chapters of this thesis.

3.2 Imperfect Information Games and Dynamic-epistemic Logic

Dynamic-epistemic language Games in extensive form are trees (S, $\{R_a\}_{a\in A}$), consisting of nodes for successive states of play, with players' moves represented as binary transition relations between nodes. Imperfect information is encoded by equivalence relations $\sim_i$ between nodes that model uncertainties for player i. Nodes in these structures are naturally described in a combined *modal-epistemic language*. An action modality $[a]\varphi$ is true at a node x when φ holds after every successful execution of move a at x, and a knowledge modality $K_i\varphi$ is true at x when φ holds at every node $y \sim_i x$. As usual, we write $\langle a\rangle$, $\langle i\rangle$ for the existential duals of these modalities. Such a language can describe many common scenarios.

Example Not knowing one's winning move
In the following two-step game tree, player ***E*** does not know the initial move that was played by ***A***:



The modal formula $[a]\langle d\rangle Win_E \wedge [b]\langle c\rangle Win_E$ expresses the fact that ***E*** has a winning strategy in this game, and at the root, she knows both conjuncts. After ***A*** plays move b from the root, however, in the black intermediate node, ***E*** knows merely 'de dicto' that playing either c or d is a winning move, as is expressed by the joint modal-epistemic formula $K_E(\langle c\rangle Win_E \vee \langle d\rangle Win_E)$. But she does not know 'de re' of any specific move that it guarantees a win: $\neg K_E\langle c\rangle Win_E \wedge \neg K_E\langle d\rangle Win_E$ also holds. In contrast, given the absence of dotted lines for ***A***, whatever is true at any stage of this game is known to ***A***. In particular, at the black intermediate node, ***A*** does know that c is a winning move for ***E***. ■

Remark Temporal Language
For some purposes, it is also useful to have *converse* relations $a^{\cup}$ for moves a, looking back up into the tree. In particular, these help describe play so far by mentioning the moves that have been played, while they also allow us to look back and say what could have happened if play had gone differently. Both are very natural things to say about the course of a game. This is a simple temporal logic variant of the basic modal-epistemic language.

Strategies, plans, and programs A modal-epistemic language describes players' moves and what they know about their step-by-step effects. Explicit information about agents' global behaviour can be formulated in a *dynamic-*

epistemic language, which adds complex program expressions. A *strategy* for player i is a function from i's turns x in the game to possible moves at x, while we might think of a *plan* as any relation constraining these choices, though not always to a unique one. Such binary relations and functions can be described using (i) single moves a, (ii) tests (φ)? on the truth of some formula φ, combined using operations of (iii) union $\cup$, relational composition ;, and iteration *. In particular, these operations define the usual program constructs *IF THEN ELSE* and *WHILE DO.* As for test conditions, in this setting, it only makes sense to use φ which an agent *knows to be true or false.* Without loss of generality, we can assume that such conditions have the epistemic form $K_i\varphi$. The resulting programs are called 'knowledge programs' in Fagin, Halpern, Moses & Vardi 1995. Van Benthem 2001 proves that in finite imperfect information games, the following two notions from logic and game theory coincide:

(a) strategies that are defined by knowledge programs,

(b) *uniform strategies*, where players choose the same move at every two nodes which they cannot distinguish.

Valid laws of reasoning about agents and plans Universally valid principles of our language include the minimal modal or dynamic logic, plus the epistemic logic matching the uncertainty relations – in our case, multi-S5. Logics like this were used in Moore 1985 to study planning agents in AI. Of course, here we are most interested in players' changing knowledge as a game proceeds. The language allows us to make these issues precise. For instance, if a player is certain now that after some move φ is the case, then after that move, is she certain that φ is the case? In other words, does the following formula hold under all circumstances?

$$K_i[a]p \rightarrow [a]K_ip$$

The answer is negative for most of us. I know that I am boring after drinking – but it does not follow (unfortunately) that after drinking, I know that I am boring. The interchange axiom is only plausible for actions without 'epistemic side-effects'. And the converse implication can be refuted similarly. In general, dynamic-epistemic logic has no significant interaction axioms at all for knowledge and action. If such axioms hold, this must be due to special features of the situation.

Example Games versus general dynamic-epistemic models
Imperfect information games themselves do satisfy a special axiom. The tree structure is common knowledge, and players cannot be uncertain about it. This is expressed by the following axiom – where M is the union of all available moves m in the game, and $m^{\cup}$ is the converse relation of m:

$$\langle i\rangle p \rightarrow \langle (M \cup M^{\cup})^{*}\rangle p \qquad \#$$

The effect of # can be stated as a modal frame correspondence. Epistemically accessible worlds are reachable from the root via sequences of moves:

Fact # is true on a frame iff, *for all* s, t, *if* $s \sim_i t$, *then* $s(M \cup M^{\cup})^{*} t$.

Using this condition, every general model for a modal-epistemic language can be unraveled to a tree of finite action sequences in the usual modal fashion, with uncertainties $\sim_i$ between X, Y just in case $last(X) \sim_i last(Y)$. The map from sequences X to worlds $last(X)$ is then a bisimulation for the whole combined language. ■

Without this constraint, we get 'misty games' (cf. Hötte 2003), where players need not know what their moves are or what sort of opponent they are dealing with. This broader setting is quite realistic for planning problems. We return to it at the end of this paper.

Axioms for perfect agents In the same correspondence style, the above knowledge-action interchange law really describes a special type of agent. To see this, we first observe that

Fact $K_i[a]p \rightarrow [a]K_i p$ corresponds to the relational frame condition that *for all* s, t, u, *if* $sR_a t$ & $t \sim_i u$, *then there is a* v *with* $s \sim_i v$ & $vR_a u$.

This condition says that new uncertainties for an agent are always grounded in earlier ones. The equivalence can be proved, e.g., by appealing to the Sahlqvist form of this axiom. Incidentally, this and further observations about the import of axioms may be easier to understand using the equivalent existential versions, here: $\langle a\rangle\langle i\rangle p \rightarrow \langle i\rangle\langle a\rangle p$.

Precisely this relational condition was identified in van Benthem 2001 as a natural version of players having *Perfect Recall* in the game-theoretic sense: They know their own moves and also remember their past uncertainties as they were at each stage. The actual analysis is slightly more complex in the case of games. First, consider nodes where it is the player's turn: then $K_i[a]p$ implies $[a]K_ip$ for the same action a. Perfect Recall does not exclude, however, that moves by one player may be indistinguishable for others, and hence at another player's turn, $K_i[a]p$ implies merely that $[b]K_ip$ for some indistinguishable action b. But there are more versions of perfect recall in game theory. Some allow players uncertainty about the number of moves played by their opponents. Bonanno 2004 has an account of such variants in essentially our correspondence style, now including a temporal operator into the language.

Remark A similar analysis works for the converse dynamic-epistemic axiom $[a]K_ip \rightarrow K_i[a]p$, whose frame truth demands a converse frame condition of 'No Learning', stating essentially that current uncertainty links remain under identical actions (cf. Fagin, Halpern, Moses & Vardi 1995). We will encounter this principle in a modified form in Section 3.2.

Agents with Perfect Recall also show special behavior with respect to their knowledge about complex plans, including their own strategies.

Fact Agents with Perfect Recall validate all dynamic-epistemic formulas of the form $K_i[\sigma]p \rightarrow [\sigma]K_ip$, where σ is a knowledge program.

The proof is a straightforward induction on programs. For knowledge tests $(K_i\varphi)?$, we have $K_i[(K_i\varphi)?]p \leftrightarrow K_i(K_i\varphi \rightarrow p)$ in dynamic logic, and then $K_i(K_i\varphi \rightarrow p) \leftrightarrow (K_i\varphi \rightarrow K_ip)$ in epistemic S5, and $(K_i\varphi \rightarrow K_ip) \leftrightarrow [(K_i\varphi)?]K_ip$ in dynamic logic. For the program operations of choice and composition, the inductive steps are obvious, and program iteration may be dealt with as repeated composition. ■

This simple observation implies that an agent with Perfect Recall who knows what a plan will achieve will also know about these effects halfway through, when some part of his strategy has been played and only some remains. Again, this is not true for all types of agent. This is only one of many delicate issues that can be raised about players' knowledge of their strategies. Indeed, a knowledge statement about *objects*, like 'knowing one's strategy', has aspects that cannot be expressed in our formalism at all. We leave this for further elaboration elsewhere.

Axioms for imperfect agents But there are other types of agents! At the opposite of Perfect Recall, there are agents with bounded memory, who can only remember a fixed number of previous events. Such players with 'bounded rationality' are modelled in game theory by restricting them to strategies that can be implemented by some finite automaton (cf. Osborne & Rubinstein 1994). Van Benthem 2001 considers the most drastic form of memory restriction, to just the last event observed. These *Memory-free* agents will be our guiding example of epistemic limitations in this paper.

In modal-epistemic terms, Memory-free agents satisfy a Memory Axiom:

$$\langle a \rangle p \rightarrow U[a]\langle i \rangle p \qquad MF$$

This involves extending our language with a *universal modality* $U\varphi$ stating that φ holds in all worlds. The technical meaning of *MF* is as follows.

Claim The axiom *MF* corresponds to the structural frame condition that, *if* $sR_a t$ & $uR_a v$, *then* $v \sim_i t$.

Thus, nodes where the same action has been performed are indistinguishable. Reformulated in terms of knowledge, the axiom becomes $\langle a \rangle K_i p \rightarrow U[a]p$. This says that the agent can only know things after an action which are true wherever the action has been performed. Therefore, Memory-free agents know very little indeed! We will study their behavior further in Section 3.4. For now, we return to perfection.

3.3 Update for Perfect Agents

Imperfect information trees merely provide a static record of what uncertainties players are supposed to have at various stages of a game. And then we have to think of some plausible scenario which might have produced these uncertainties. One general mechanism of this kind is provided by *update logics* for actions with epistemic import. We briefly recall the basics (cf. Baltag, Moss & Solecki 1998).

Product update A general update step has two components:

(a) an *epistemic model* $\boldsymbol{M}$ of all relevant possible worlds with agents' uncertainty relations indicated,

(b) an *action model* $\boldsymbol{A}$ of all relevant actions, again with agents' uncertainty relations between them.

Action models can have any pattern of uncertainty relations, just as in epistemic models. This reflects agents' limited powers of observation. E.g., in a card game, $\boldsymbol{M}$ might be the initial situation after the cards have been dealt, while $\boldsymbol{A}$ contains all legal moves that players have. Some actions are public and transparent to everyone, like throwing a card on the table. Others, like drawing a new card from the stock, are only transparent to the player who draws, while others cannot distinguish actions of drawing different cards. But there is still one more element. Not every action needs to be possible at each world. E.g., I can only draw the Ace of Hearts if it is still in the stock on the table. Such restrictions are encoded by

(c) *preconditions* PRE_a for actions a,

which are supposed to be common knowledge among agents. In the simplest case, these are formulated in the pure epistemic language describing facts and agents' (mutual) information about them. Now, the next epistemic model $\boldsymbol{M}\times\boldsymbol{A}$ is computed as follows:

The domain is $\{(s,\ a) \mid s$ a world in $\boldsymbol{M}$, a an action in $\boldsymbol{A}$, $(\boldsymbol{M},\ s) \models PRE_a\}$.
The new uncertainties satisfy $(s,\ a) \sim_i (t,\ b)$ iff both $s \sim_i t$ and $a \sim_i b$.
A world $(s,\ a)$ satisfies a propositional atom p iff s already did in $\boldsymbol{M}$.

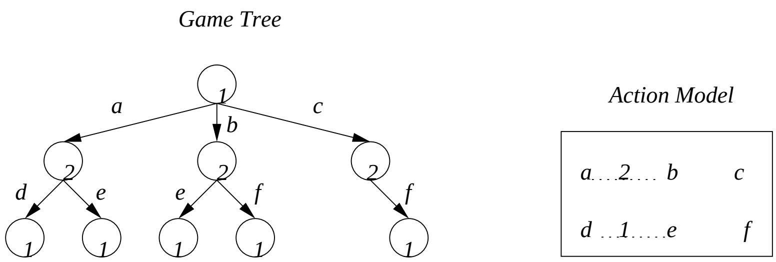
In particular, the *actual world* of the new model is the pair consisting of the actual world in $\boldsymbol{M}$ and the actual action in $\boldsymbol{A}$. The product rule says that uncertainty among new states can only come from existing uncertainty via indistinguishable actions. This simple mechanism covers surprisingly many forms of epistemic update. Baltag, Moss & Solecki 1998, van Benthem 2003, and many other recent publications provide introductions to update logics and the many open questions one can ask about them.

The same perspective may now be applied to imperfect information games, where successive levels correspond to successive repetitions of the sequence

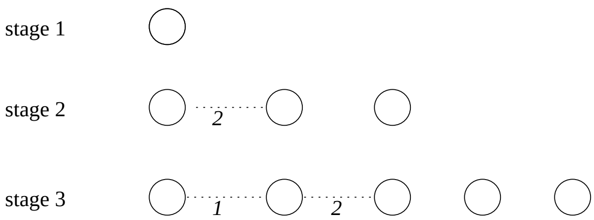
$$\boldsymbol{M},\ \boldsymbol{M}\times\boldsymbol{A},\ (\boldsymbol{M}\times\boldsymbol{A})\times\boldsymbol{A},\ ...$$

The result is an obvious tree-like model *Tree*($\boldsymbol{M}$, $\boldsymbol{A}$), which may be infinite.

Example Propagating uncertainty along a game
The following illustration is from van Benthem 2001. Suppose we are given a game tree with admissible moves (preconditions will be clear immediately). Let the moves come with epistemic uncertainties encoded in an action model:



Then the imperfect information game can be computed with levels as follows:



■

Now enrich the modal-epistemic language with a dynamic operator

$$\boldsymbol{M},\ s \models \langle \boldsymbol{A},\ a\rangle\varphi \quad \text{iff} \quad (\boldsymbol{M},\ s) \times (\boldsymbol{A},\ a) \models \varphi$$

Then valid principles express how knowledge is related before and after an action. In particular, we have this key *reduction axiom*:

$$\langle \boldsymbol{A},\ a\rangle\langle \mathrm{i}\rangle\varphi \leftrightarrow (PRE_a \wedge \bigvee\{\langle \mathrm{i}\rangle\langle \boldsymbol{A},\ b\rangle\varphi\text{: } a \sim_i b \textit{ for some } b \textit{ in } \boldsymbol{A}\})$$

Such laws simplify reasoning about action and planning: We can reduce epistemic properties of later stages to epistemic information about the current stage. From left to right, this axiom is the earlier Perfect Recall, but now with a twist compared with earlier formulations. If an agent cannot distinguish certain actions from the actual one, then those may show up in his epistemic alternatives. The opposite direction from right to left is the No Learning principle. But it does not say that agents can never learn, only that no learning is possible for them among indistinguishable situations by using actions that they cannot distinguish.

The preceding logical observations show that product update is geared toward special agents, viz. those with Perfect Recall. The fact that the reduction axiom is valid shows that perfect memory must have been built into the very definition. And it is easy to see how. The two clauses in defining the new relation $(s,\ a) \sim_i (t,\ b)$ give equal weight to

(a) $s \sim_i t$: past states representing the 'memory component',

(b) $a \sim_i b$: options for the newly observed event.

Changes in this mechanism will produce other 'product agents' by assigning different weights to these two factors (see Section 3.5). But first, we determine the essence of product update from the general perspective of Section 3.2. The following result improves a theorem in van Benthem 2001.

Abstract characterization of product update Consider a tree-like structure $\mathcal{E}$ with possible events (or actions) and uncertainty relations among its nodes, which can also verify atomic propositions p, q, ... The only contrast with a real tree is that we allow a bottom level with multiple roots. Nodes X, Y, ... are at the same time finite sequences of events, and the symbol $^\frown$ expresses concatenation of events. Intuitively, we think of such a tree structure $\mathcal{E}$ as the possible evolutions of some process – for instance, a game. A particular case is the above model $Tree(\boldsymbol{M},\ \boldsymbol{A})$ starting from an initial epistemic model $\boldsymbol{M}$ and an action model $\boldsymbol{A}$, and repeating product updates forever. Now, the preceding discussion shows that the following two principles are valid in $Tree(\boldsymbol{M},\ \boldsymbol{A})$, which can be stated as general properties of a tree $\mathcal{E}$. They represent Perfect Recall and 'Uniform No Learning', respectively:

PR If $X^\frown(a) \sim_i Y$, then $\exists b\, \exists Z$: $Y = Z^\frown(b)$ & $X \sim_i Z$.

UNL If $X^\frown(a) \sim_i Y^\frown(b)$, then $\forall U, V$: if $U \sim_i V$, then $U^\frown(a) \sim_i V^\frown(b)$, provided that $U^\frown(a)$, $V^\frown(b)$ both occur in the tree $\mathcal{E}$.

Moreover, the special nature of the preconditions in product update, as definable conditions inside the current epistemic model, validates one more abstract constraint on the tree $\mathcal{E}$:

BIS-INV The set $\{X \mid X^\frown(a) \in \mathcal{E}\}$ of nodes where action a can be performed is closed *under purely epistemic bisimulations* of nodes.

Now we have all we need to prove a converse representation result.

Theorem For any tree $\mathcal{E}$, the following are equivalent:

(a) $\mathcal{E} \cong$ *Tree*$(\boldsymbol{M}, \boldsymbol{A})$ for some $\boldsymbol{M}$, $\boldsymbol{A}$

(b) $\mathcal{E}$ satisfies *PR*, *UNL*, *BIS-INV*

Proof From (a) to (b) is the above observation. Now, from (b) to (a). Define an epistemic model $\boldsymbol{M}$ as the set of initial points in $\mathcal{E}$ and copy the relations $\sim_i$ from $\mathcal{E}$. The action model $\boldsymbol{A}$ contains all possible actions occurring in the tree, where we set

$$a \sim_i b \quad \text{iff} \quad \exists X\, \exists Y\colon X^\frown(a) \sim_i Y^\frown(b)$$

We also need to know that the *preconditions* PRE_a for actions a are as required. For this, we use the well-known fact that in any epistemic model, any set of worlds that is closed under epistemic bisimulations must have a definition in the epistemic language – though admittedly, one allowing infinite conjunctions and disjunctions. The abstract setting of our result allows no further finitization of this definability.

Now, the obvious identity map F sends nodes X of $\mathcal{E}$ to corresponding states in the model *Tree* $(\boldsymbol{M}, \boldsymbol{A})$. First, we observe the following fact about $\mathcal{E}$ itself:

Lemma If $X \sim_i Y$, then $length(X) = length(Y)$.

Proof If X, Y are initial points in $\mathcal{E}$, both their lengths are 0. Otherwise, let X have length $n+1$. By *PR*, X's initial segment of length n stands in the relation $\sim_i$ to a proper initial segment of Y whose length is that of Y minus 1. Repeating this observation peels off both sequences to initial points after the same number of steps.

Claim $X \sim_i Y$ holds in $\mathcal{E}$ iff $F(X) \sim_i F(Y)$ holds in *Tree*$(\boldsymbol{M}, \boldsymbol{A})$.

The proof is by induction on the common length of the two sequences X, Y. The case of initial points is clear by the definition of $\boldsymbol{M}$. As for the inductive steps, consider first the direction $\Rightarrow$. If $U^\frown(a) \sim_i V$, then by *PR*, $\exists b\ \exists Z$: $V = Z^\frown(b)$ & $U \sim_i Z$. By the inductive hypothesis, we have $F(U) \sim_i F(Z)$. We also have $a \sim_i b$ by the definition of $\boldsymbol{A}$. Moreover, given that the sequences $U^\frown(a)$, $Z^\frown(b)$ both belong to $\mathcal{E}$, their preconditions as listed in $\boldsymbol{A}$ are satisfied. Therefore, in *Tree*$(\boldsymbol{M}, \boldsymbol{A})$, by the definition of product update, $(F(U), a) \sim_i (F(Z), b)$, i.e. $F(U^\frown(a)) \sim_i F(Z^\frown(b))$.

As for the direction $\Leftarrow$, suppose that in *Tree*$(\boldsymbol{M}, \boldsymbol{A})$ we have $(F(U), a) \sim_i (F(Z), b)$. Then by the definition of product update, $F(U) \sim_i F(Z)$ and $a \sim_i b$. By the inductive hypothesis, from $F(U) \sim_i F(Z)$ we get $U \sim_i Z$ in $\mathcal{E}$(*). Also, by the given definition of $a \sim_i b$ in the action model $\boldsymbol{A}$, we have $\exists X\ \exists Y$: $X^\frown(a) \sim_i Y^\frown(b)$(**). Taking (*) and (**) together, by *UNL* we get $U^\frown(a) \sim_i Z^\frown(b)$, provided that $U^\frown(a)$, $V^\frown(b) \in \mathcal{E}$. But this is so since the preconditions PRE_a, PRE_b of the actions a, b were satisfied at $F(U)$, $F(Z)$. This means these epistemic formulas must also have been true at U, V – so, given what PRE_a, PRE_b defined, $U^\frown(a)$, $V^\frown(b)$ exist in the tree $\mathcal{E}$.■

This result is only one of a kind, and its assumptions may be overly restrictive. In many game scenarios, preconditions for actions are not purely epistemic, but rather depend on what happens over time. E.g., a game may have initial factual announcements – like the Father's saying that at least one child is dirty in the puzzle of the Muddy Children. These are not repeated, even though their preconditions still hold at later stages. Describing this requires preconditions PRE_a for actions a that refer to the temporal structure of the tree $\mathcal{E}$, and then the above invariance for purely epistemic bisimulations would fail. Another strong assumption is our use of a single action model $\boldsymbol{A}$ that gets repeated all the time in levels $\boldsymbol{M}$, $(\boldsymbol{M}\times\boldsymbol{A})$, $(\boldsymbol{M}\times\boldsymbol{A})\times\boldsymbol{A}$, ... to produce the structure *Tree*$(\boldsymbol{M}, \boldsymbol{A})$. A more local perspective would allow different action models $\boldsymbol{A}_1, \boldsymbol{A}_2$, ... in stepping from one tree level to another. And an even more finely-grained view arises if single moves in a game themselves can be complex action models. In the rest of this paper, for convenience, we stick to the single-model view.

3.4 Update Logic for Bounded Agents

Limitations on information processing The information-processing capacity of agents may be bounded in various ways. One of these is 'external': Agents may have restricted powers of observation. This kind of restriction is built into the definition of action models, with uncertainties for agents – and the product update mechanism of Section 3.2 reflects this. Another type of restriction is 'internal': Agents may have bounded memory. Agents with Perfect Recall had limited powers of observation but perfect memory. At the opposite extreme we find Memory-free agents which can only observe the last event, without maintaining any record of what went on before. In this section, we explore this extreme case.

Characterizing types of agent In the preceding, agents with Perfect Recall have been described in various ways. Our general setting was the tree $\mathcal{E}$ of event sequences, where different types of agents i correspond to different types of uncertainty relation $\sim_i$. One approach was via *structural conditions* on such relations, such as *PR*, *UNL*, and *BIS-INV* in the above characterization theorem. Essentially, these three constraints say that

$$X \sim_i Y \quad \text{iff} \quad \textit{length(X)} = \textit{length(Y)} \text{ and } \textit{X(s)} \sim_i \textit{Y(s)} \text{ for all positions } s$$

Next, these conditions also validated corresponding *axioms in the dynamic-epistemic language* that govern typical reasoning about the relevant type of agent. But thirdly, we also think of agents as a sort of *processing mechanism*. Intuitively, an agent with Perfect Recall is a push-down store automaton maintaining a stack of all past events and continually adding new observations. Such a processing mechanism was provided by our representation theorem, viz. epistemic product update.

Bounded memory Another broad class of agents arises by assuming bounded memory up to some fixed finite number k of positions. In general trees $\mathcal{E}$, this makes two event sequences X, Y $\sim_i$-equivalent for such agents

i iff their last k positions are $\sim_i$-equivalent. In this section we only consider the most extreme case of this, viz. *Memory-free agents* i:

$$X \sim_i Y \quad \text{iff} \quad \mathit{last}(X) \sim_i \mathit{last}(Y) \text{ or } X = Y = \textit{the empty sequence} \quad \$$$

Agents of this sort only respond to the last-observed event. In particular, their uncertainty relations can now cross between different levels of a game tree: They need not know how many moves have been played. Perhaps contrary to appearances, such limited agents can be quite useful. Examples are *Tit-for-Tat* players in the iterated Prisoner's Dilemma which merely repeat their opponents' last move (Axelrod 1984), or *Copy-Cat* players in game semantics for linear logic which can win 'parallel disjunctions' of games $G \vee G^d$ (Abramsky 1996). Incidentally, these are players with a hard-wired *strategy*: a point that we will discuss below. It is easy to characterize such agents in terms similar to what we did with Perfect Recall.

Fact An equivalence relation $\sim_i$ on $\mathcal{E}$ is Memory-free in the sense of \$ if and only if the following two conditions are satisfied:

PR^- *If* $X^\frown(a) \sim_i Y$, *then* $\exists b \sim_i a\ \exists Z$: $Y = Z^\frown(b)$.

UNL^+ *If* $X^\frown(a) \sim_i Y^\frown(b)$, *then* $\forall U, V$: $U^\frown(a) \sim_i V^\frown(b)$, provided that $U^\frown(a)$, $V^\frown(b)$ both occur in the tree $\mathcal{E}$.

Proof If an agent i is Memory-free, its relation $\sim_i$ evidently satisfies PR^- and UNL^+. Conversely, suppose that these conditions hold. If $X \sim_i Y$, then either X, Y are both the empty sequence, and we are done, or, say, $X = Z(a)$. Then by PR^-, $Y = U(b)$ for some $b \sim_i a$, and so $\mathit{last}(X) \sim_i \mathit{last}(Y)$. Conversely, the reflexivity of $\sim_i$ plus UNL^+ imply that, if the right-hand side of the equivalence \$ holds, then $X \sim_i Y$. ■

It is also easy to give a characteristic modal-epistemic axiom for this case. First, set

$$a \sim_i b \quad \text{iff} \quad \exists X\, \exists Y : X^\frown(a) \sim_i Y^\frown(b)$$

Fact The following equivalence is valid for Memory-free agents:

$$\langle a\rangle\langle i\rangle\varphi \leftrightarrow (PRE_a \ \&\ E \bigvee_{b \sim_i a} \langle b\rangle\varphi)$$

Here $E\varphi$ is an additional *existential modality* saying that φ holds in at least one node. This axiom looks at first glance like the Perfect Recall axiom of Section 3.3, but note that there is no epistemic modality $\langle i \rangle$ on the right-hand side of the equivalence. Also, this new axiom implies axiom *MF* from Section 3.2, assuming that basic actions are partial functions.

Remark Reduction axioms for an existential modality
Once the static description language gets extended, to restore the harmony of an update logic, one should also extend the dynamic update reduction axioms with a clause for the new operator. E.g., returning to Section 3.3, the following reduction axiom is valid for standard product update:

$$\langle \boldsymbol{A},\ a\rangle E\varphi \leftrightarrow (PRE_a \wedge E \bigvee \langle \boldsymbol{A},\ b\rangle\varphi \textit{ for some } b \textit{ in } \boldsymbol{A})$$

The process mechanism: finite automata The processor of Memory-free agents is a very simple *finite automaton* creating their correct $\sim_i$ links:

> States of the automaton: all equivalence classes $X^{\sim_i}$
> Transitions for actions a: $X^{\sim_i}$ goes to $(X^\cap(a))^{\sim_i}$

There are only finitely many states since we had only finitely many actions in the tree $\mathcal{E}$. The transitions are well-defined, since by the No Learning assumption UNL^+, if $X \sim_i Y$, then $X^\cap(a) \sim_i Y^\cap(a)$. The automaton starts in the equivalence class of the empty event sequence. Repeating transitions, it is then easy to see that

> When the automaton is given the successive members of an event sequence X as input, it ends in state $X^{\sim_i}$

In particular, $X \sim_i Y$ iff the automaton ends in the same state on both of these event sequences. Moreover, the combination of the conditions UNL^+ and PR^- on Memory-free agents tells us something about the special type of automaton that suffices:

> All transitions a end in the same state (as $X^\cap(a) \sim_i Y^\cap(a)$ for all $X,\ Y$), and by PR^-, no transition ends in the initial state.

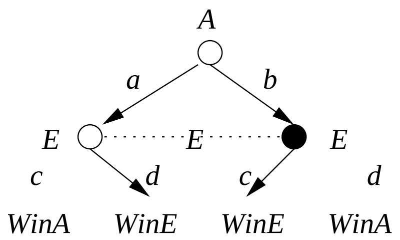
Let us call such automata *rigid*. They only have states for the last-observed event, and such states will even coincide when the events are not epistemically distinguishable for the agent.

Fact Memory-free agents are exactly those whose uncertainty relation is generated by a rigid finite-state automaton.

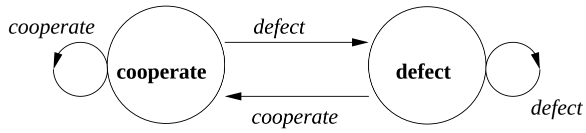
Of course, more complex finite automata can have more differentiated responses to observed events a, up to some fixed finite number of cases.

Remark Automata theory
Connections with automata theory, in particular the Nerode representation of finite automata recognizing regular sets of event sequences, are found in van Benthem & ten Cate 2003. Also, the above framework can be extended with more general preconditions for game actions referring to time, by generalizing to the action/test automata used for propositional dynamic logic in Harel, Kozen & Tiuryn 2000.

Strategies and automata The preceding automata for bounded agents are reaction devices to incoming observations. But it is also tempting to think of automata as generators of behaviour – in particular, as specific *strategies*. The latter view is more in line with the usual treatment of our motivating examples, like *Tit-for-Tat* or *Copy-Cat*. A strategy for player i in a game is a function assigning moves to turns for i, these moves are responses to *other players'* actions. This is easily visualized in game trees $\mathcal{E}$. E.g., player $\boldsymbol{E}$'s winning strategy in the game of Section 3.2 looks as follows:



But the reflection in finite automata will be a little different then, as players do not respond to a last action if played by themselves (these are 'non-events' for the purpose of a strategy). Thus, the usual automaton for *Tit-for-Tat* encodes actions by the agent itself as *states*, while actions by the opponent are the true observed events:

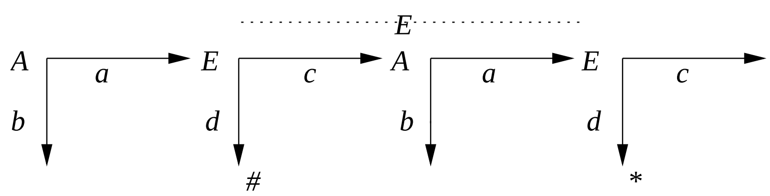


We do not undertake an integration of the two sorts of finite automata here. Either way, the simplicity of such automata for agents and their strategies may also be seen by considering the special syntactic form of Memory-free strategies as simple knowledge programs in the dynamic-epistemic language.

This concludes our discussion of Memory-free agents per se. To highlight them even more, we add a few contrasts with agents with Perfect Recall.

Differences in what agents know Memory-free agents i know less than agents with Perfect Recall. The reason is that their equivalence classes for $\sim_i$ tend to be larger. E.g., *Tit-for-Tat* only knows she is in two of the four possible matrix squares (*cooperate, cooperate*) or (*defect, defect*). But amongst many other failures, she does not know the accumulated score at the current stage. It is also tempting to say that Memory-free agents can only run very simplistic strategies. But this is not quite right, since any knowledge program makes sense for all agents. The point is just that certain knowledge conditions will evaluate differently for both. E.g., a Perfect Recall agent may be able to act on conditions like "action a has occurred twice so far", which a Memory-free agent can never execute, since she can never know that the condition holds. Thus the difference is rather in the number of non-equivalent available uniform strategies and the successful behaviour guaranteed by these.

Example How Memory-free agents may suffer
Consider the following game tree for an agent ***A*** with perfect information, and a Memory-free agent ***E*** who only observes the last move.



Suppose that outcome # is a bad thing, and $*$ a good thing for ***E***. Then the desirable strategy "play d only after you have seen two a's" is unavailable to ***E*** – while it is available to a player with Perfect Recall. ■

Another difference between Perfect Recall agents and Memory-free agents has to do with what they know about their *strategies*. We saw that an agent with Perfect Recall for atomic actions also satisfies the key implication

$$K_i[\sigma]p \rightarrow [\sigma]K_ip, \text{ when } \sigma \text{ is any complex knowledge program.}$$

By contrast, the *MF* Memory Axiom

$$\langle a\rangle p \rightarrow U[a]\langle i\rangle p$$

does not 'lift' to arbitrary knowledge programs instead of the single action a. To see this, it suffices to look at the case of a choice program $a \cup b$. Our eventual reduction version

$$\langle a\rangle\langle i\rangle\varphi \leftrightarrow (PRE_a \;\&\; E \bigvee_{b\sim_i a} \langle b\rangle\varphi)$$

is a bit harder to generalize at all, because we would first have to analyze what it means to be indistinguishable from a complex action.

Memory and time A good way of making differences between agents more explicit is the introduction of a richer language. So far, we have mostly looked at a purely epistemic language for preconditions and an epistemic language with forward action modalities for describing updates or general moves through a game tree. With such a language, some of the intuitive distinctions that we want to make between different agents cannot be expressed. E.g., suppose that there is just one initial world s and one action, the identity *Id*, which always succeeds:

$$s \qquad (s, Id) \qquad ((s, Id), Id) \qquad ...$$

Thus, each horizontal level contains just one world. In this model, the uncertainty lines for Perfect Recall agents and Memory-free agents are different. The latter see all worlds ending in *Id* as indistinguishable, whereas product update for the former makes all worlds different. Nevertheless, agents know exactly the same purely epistemic statements in each world. The technical reason is that all states are epistemically bisimilar, and composing the uncertainty lines for a player with bisimulation links makes no difference to what she knows. But intuitively, the Perfect Recall player should know how many actions have occurred, since her uncertainties did not cross levels. Now, if we want to let agents know explicit statements about where they are in the game, we can add the backward-looking *converse action modalities* mentioned in Section 3.2. Then an agent knows, e.g., that two moves have been played if it knows that two consecutive converse actions are possible, but not three. Thus, a temporal dynamic-epistemic language is more true to what we would want to say intuitively about players and their differences. Moreover, this language can also express more complex preconditions for actions, resulting in the definability of a much broader range of strategies (cf. Rodenhauser 2001).

Remark Backward-looking update
A backward-looking temporal language also enriches update logic. Our reduction axioms so far were forward-looking analysis of *preconditions*, reducing what agents know after an action has taken place to what they knew before. What about converse reduction axioms of the form, say:

$$\langle a^{\cup}\rangle\langle i\rangle\varphi \leftrightarrow (PRE_{a^{\cup}} \;\&\; E \bigvee_{b^{\cup}\sim_i a^{\cup}} \langle b^{\cup}\rangle\varphi)?$$

These are related to *postconditions* for actions a: The strongest that we can say when a was performed in a world satisfying φ is that $\langle a^{\cup}\rangle\varphi$ must hold. Such postconditions are known to be impossible to define, even for simple public announcements, in the open-ended total universe of all epistemic models. But things are more controlled in our trees $\mathcal{E}$ which fix the previous history for any current world. In that case, we can convert at least earlier full commutativity axioms like the interchange of $\langle a\rangle\langle i\rangle$ and $\langle i\rangle\langle a\rangle$ to backward-looking versions.

A final caveat This discussion has been somewhat impressionistic. In particular, it is easy to *over-interpret* our formal models in terms of 'knowledge talk'. At any given state, the bare fact is that an agent i has the set of all its $\sim_i$ alternatives. Depending on how *we* describe that set, we attribute various forms of knowledge to the agent. But most of these are just correlations – like when we say that *Tit-for-Tat* knows that it is in a 'cooperative' state. Such a description need not correspond to any *representational attitude* inside the agent. This mismatch is a limitation of epistemic logic in general, and over-interpretation occurs just as well for agents with Perfect

Recall. These are triggered by possibly complex 'horizontal' knowledge conditions $K\varphi$ referring to the current tree level in structures like $\mathcal{E}$ or $Tree(\boldsymbol{M}, \boldsymbol{A})$. But we, as outside observers, may identify these as equivalent to simple assertions about the past of the process, such as "action a has occurred twice". And even when we use the above richer temporal language, this still need not imply matching richer representations inside the agent.

3.5 Spectra of Agents: Modulating Product Update

Toward a spectrum of options Perfect Recall agents and Memory-free agents are two extremes with room in the middle. Using the automata of Section 3.4, one might define update for progressively better informed k-bit agents having k memory cells, creating much great diversity. By contrast, agents with Perfect Recall seemed the natural children of product update. But even here there is room for alternative stipulations! The following type of agent is closely related to the Memory-free ones discussed before.

Forgetful updaters As we saw in Section 3.3, product update for new uncertainties mixed a memory factor (viz. uncertainty between old states) and an observation factor (viz. uncertainty between actions). Agents might weigh these differently. A Memory-free agent, by necessity, gives weight 0 to the past. If updating agents only remember their last action, how do they update their information? Here is a simple new definition. We drop the memory factor when defining product models $\boldsymbol{M}\times\boldsymbol{A}$, and set:

$$(x, a) \sim_i (y, b) \quad \text{iff} \quad a \sim_i b\ !$$

Thus, new uncertainty comes only from uncertainty about observed actions. Just as before, this leads to a valid *reduction axiom*:

Fact The following equivalence is valid with forgetful update:

$$\langle \boldsymbol{A},\ a\rangle\langle i\rangle\varphi \leftrightarrow (PRE_a \wedge E \bigvee\langle \boldsymbol{A},\ b\rangle\varphi\text{: } a \sim_i b \textit{ for some } b \textit{ in } \boldsymbol{A})$$

As before, to restore the harmony of the complete system, we also need a reduction axiom for the new modality E, which turns out to be

$$\langle \boldsymbol{A},\ a\rangle E\varphi \leftrightarrow (PRE_a \wedge E \bigvee \langle A,\ b\rangle\varphi \textit{ for some } b \textit{ in } \boldsymbol{A})$$

And it is also possible to give an abstract characterization of forgetful updaters by modifying the main theorem of Section 3.3.

In the original version of this paper, it was suggested that forgetful updaters are precisely the Memory-free agents of Section 3.4. But as was pointed out by Josh Snyder (personal communication), this seems wrong. Consider the following scenario. A forgetful updater is uncertain between world s with p and world t with $\neg p$. There are two possible actions:

a with precondition: $p \wedge \neg Kp$
b with precondition: $Kp \vee (\neg p \wedge \neg K\neg p)$

Let the actual actions be a, b in that order. Then the successive product updates for forgetful updaters are

(i) from $\{s,\ t\}$ to $\{(s,\ a),\ (t,\ b)\}$, without an uncertainty link, so the agent knows that p in the actual world $(s,\ a)$, whereas he knows that $\neg p$ in the unrelated world $(t,\ b)$

(ii) from $\{(s,\ a),\ (t,\ b)\}$ to $\{((s,\ a),\ b)\}$, since neither a nor b can be performed in $(t,\ b)$.

But in that final model, the agent still knows that p, even though a Memory-free agent would not know p because she would be uncertain between $((s,\ a),\ b)$ and $(t,\ b)$. Snyder 2004 has a solution for this by modifying product update so as to keep all worlds around, whether or not preconditions of actions are satisfied, while redefining uncertainty relations in some appropriate fashion. Another option might be the addition of suitable 'copy actions' that keep earlier sequences alive at later levels.

The upshot of this discussion is that forgetful updaters are not the same as our earlier Memory-free agents, although they are close. In the remainder of this section, we mention some other modulations on product update that create different types of agents.

Probabilistic modulations Letting agents give different weights to memory and observation in computing a new information state is an idea from a well-known tradition preceding modern update logics, viz. inductive logic and Bayesian statistics. Different agents or 'inductive methods' differ in the weight they put on experience versus observation. To implement this perspective in update logics, we need a *probabilistic* version of product update, as defined in van Benthem 2003.

Belief revision and plausibility update But staying closer to our qualitative setting, we can also give another natural example of diversity with a numerical flavour. In the theory of *belief revision*, it has long been recognized that agents may obey different rules, more conservative or more radical, when incorporating new information. Such rules are different options for computing new states on the basis of incoming evidence. Such diversity will even arise for agents with epistemic Perfect Recall, as we will now show.

In general, information update is a different mechanism from belief revision, but the two viewpoints can be merged. Aucher 2003 adds a function κ to epistemic models $\boldsymbol{M}$ and action models $\boldsymbol{A}$ which assigns *plausibility values* to states and actions. Here $\kappa_i(v) > \kappa_i(w)$ means that agent i believes that world w is more plausible than world v. This allows us to define degrees of belief in a proposition as truth in all worlds up to a certain plausibility:

$$\boldsymbol{M},\, s \models B_i^k\varphi \text{ iff } \boldsymbol{M},\, t \models \varphi \text{ for all worlds } t \sim_i s \text{ with } \kappa(t) \leq k.$$

Incidentally, we can also define $B_i^k\varphi$ as $K_i(p_i^k \to \varphi)$, provided we add suitable propositional constants p_i^k to the language (cf. Appendix A).

Next, plausibilities of actions indicate what an agent believes about what most likely took place. Computing the plausibility of a new state (w, a) in a product model $\boldsymbol{M}\times\boldsymbol{A}$ requires some intuitive rule. Aucher himself proposes an 'addition formula' for κ-values, subtracting a 'correction factor':

$$\kappa_j'(w,a) = Cut_M(\kappa_j(w) + \kappa_j^*(a) - \kappa_j^w(PRE_a))$$

Here Cut is a technical 'rescaling' device, and the correction $\kappa_j^w(PRE_a)$ is the smallest κ-value in $\boldsymbol{M}$ among all worlds $v \sim_i w$ satisfying PRE_a.

A continuum of revision rules In our current perspective, we see this stipulation not as the unique update rule for plausibility but as a choice for a particular type of agent. Aucher's formula makes an agent 'eager' in the following sense: The factor for the last-observed action weighs just as heavily as that for the previous state, even though the latter might encode a long history of earlier beliefs. But we can easily create further diversity by changing the above formula into one with parameters λ and μ:

$$\kappa_j^{'}(w, a) = \frac{1}{\boldsymbol{\lambda} + \boldsymbol{\mu}}(\boldsymbol{\lambda}\kappa_j(w) + \boldsymbol{\mu}\kappa_j^*(a))$$

By changing values of λ and μ, we can distinguish many different types of agents. For details, cf. Section 2.5.

Remark Belief revision by bounded agents
It is also possible to use ideas from Section 3.3, and consider belief revising agents with bounded memory. For a more extensive study of belief revision by agents with bounded resources, cf. the dissertation Wasserman 2000.

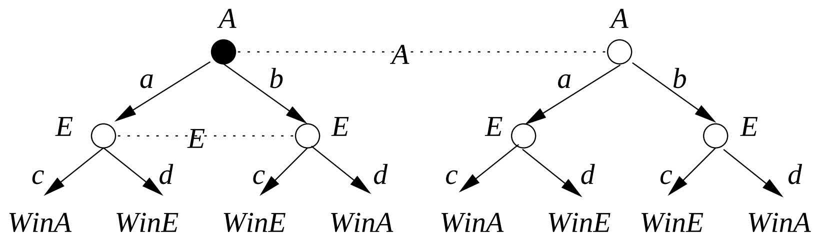
Coming to terms with belief revision, in addition to information update, is natural – also from our motivating viewpoint of games. After all, players of a game surely do not just update on the basis of observed past moves. They also revise their expectations about future actions of opponents. Further examples of this will arise in our final sections.

3.6 Mixing Different Types of Agents

So far, we have looked at agent types separately. But agents live in groups, whose members may have different types. Turing machines might communicate with finite automata, and humans occasionally meet Turing machines, like their computers, or finite automata, like very stupid people. What makes groups of agents most interesting is that they *interact*. In this setting, a host of new questions arises – of which we discuss just a few.

Uncertainty and exploitation Do different types of agents know each other's type? There is an issue of definition first. What does it *mean* to know the type of another agent? One could think of this, e.g., as knowing that the agent satisfies all axioms for its type, as formulated in Sections 3.2, 3.3 and 3.4. But then, in imperfect information games, or the more general trees $\mathcal{E}$ studied above, the types of all agents are *common knowledge*, because these axioms hold everywhere in the tree. Introducing ignorance of types requires more complex structures in the sense of Hötte 2003. Suppose that agent ***A*** does not know if his opponent is a Memory-free agent or not. Then we need disjoint unions of game trees with uncertainty links between them. Indeed, this extension already arises when we assume that some agent i does not know the precise uncertainties of its opponent between i's actions.

Example Ignorance of the opponent type
The following situation is a simple variant of the example in Section 3.2.



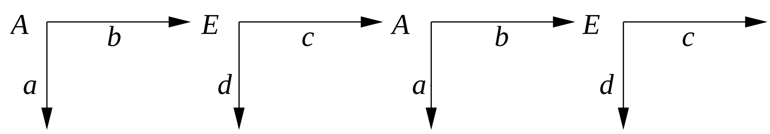
At the start of the game, agent ***A*** does not know whether ***E*** has limited powers of observation or not. In particular, note that the earlier axiom $\langle \boldsymbol{A} \rangle p \rightarrow \langle (M \cup M^{\cup})^{*} \rangle p$ for imperfect information games fails here. The 'second root' toward the right is an epistemic alternative for ***A***, but it is not reachable by any sequence of moves. ■

Can an agent take advantage of knowing another agent's type? Of course. It would be tedious to give overly formal examples of this, since we all know this phenomenon in practice. Suppose that I know that after returning a serve of mine, you always step toward the middle of the court. Then passing you all the time along the outer line is a simple winning strategy. A more dramatic scenario of this sort occurs in the recent movie "Memento" about a man who has lost his long-term memory and has fallen into the hands of unscrupulous cops and women. But *must* a Memory-free agent do badly against a more sophisticated epistemic agent? That depends on the setting. E.g., Memory-free *Tit-for-Tat* managed to win against much more sophisticated computer programs (Axelrod 1984). But even this does not do justice to the complexity of interaction!

Learning and revision over time In practice, we need not know the types of other agents and may have to *learn* them. Such learning mechanisms are themselves a further source of interesting epistemic diversity, as is pointed out in Hendricks 2003. In general, there is no guarantee at all that a learning method will reveal the type of an opponent. Evidently, observing a finite number of moves can never tell us for certain whether we are playing against an agent with Perfect Recall or against a finite-state automaton with a large finite memory beyond the current number of rounds played so far. But there is a weaker sense of learning that may be more relevant here. We may enter a game with certain hypotheses about the agents that we are playing against. And such hypotheses can be updated by observations that we make as time goes by. E.g., I can *refute* the hypothesis that you are a Memory-free agent by observing different responses to the same move of mine at different stages of the game. Or, I can have the justified hypothesis that you are Memory-free, and one observed response to a move of mine then reveals a part of your fixed strategy.

Two kinds of update? Intuitively, the game situations just described go beyond the information and plausibility update of Sections 3, 4, 5. But to arrive at a more definite verdict, one has to separate concerns. The above questions involve many general issues about update that arise even without diversity of agents. For instance, learning about one's opponent's type is akin to the well-known question of learning one's opponent's strategy. Types may be viewed as sets of strategies, so learning the type amounts to some useful intermediate reduction in the strategic form of the game. We illustrate a few issues here in a concrete scenario.

Example Finding out about types and strategies
Consider the following game of perfect information. Suppose that ***A*** knows that ***E*** is Memory-free: What does it take him then to find out which particular strategy ***E*** is running?



This scenario illustrates the danger in discussing these matters. For, if ***A*** *knows* that ***E*** is Memory-free, the latter fact is true, and hence, at her second turn, ***E*** can never play d, since she has already played c in response to b in order to get there at all. So, we can only sensibly talk about *beliefs* here. In the simplest case, these can be modelled as *subsets of all runs of the game from now on*

viz. those future runs which the agent takes to be most likely. Thus, ***A***'s belief would rule out the 'non-homogeneous run' for ***E*** in this game, even though further observation might refute the belief, forcing ***A*** to revise. Now, *belief revision* means that, as the game is played and moves are observed, this set of most plausible runs gets modified. E.g., suppose that ***E*** in fact plays d at her first turn. Then the hypothesis that she was Memory-free seems vindicated, and we also know part of her strategy. But this is again too hasty. We have not tested any global assertion about her strategy, precisely because the game is over, and we have no means of observing what ***E*** would have done at her second turn. ■

Thus, we must be sensitive to distinctions like 'predicting what *will* happen' versus 'predicting what *would* happen' in some stronger counterfactual sense. Hypotheses about one's opponents' type are of the latter sort, and they may be harder to test. The representation of alternative scenarios and suitable update mechanisms over these need not be the same in both cases. In particular, we might need two kinds of update. One is the local computation of players' uncertainties at nodes of the game concerning facts and other players' information, as described by the earlier product update and plausibility update. The other is the changing of longer-term expectations about strategies over time by observing the course of the game.

Remark Local versus global update?
Despite the appealing distinction made just now, uncertainty about the future can sometimes be 'folded back' into local update. Consider any game of perfect information. Uncertainties about the strategy played by one's opponent may be represented in a new *imperfect information* game, whose initial state consists of all possible strategy profiles with appropriate uncertainty lines for players between these. Update on such a structure occurs as consecutive moves are played in the game, which can be seen as a form of public announcement ruling out certain profiles from the diagram. Likewise, belief revision becomes plausibility update on strategy profiles. For details, see van Benthem 2004a, 2004b. ■

Update can get even more subtle than this with learning global types.

Consider the earlier example where $\boldsymbol{A}$ did not know if $\boldsymbol{E}$ had perfect information or not. How can $\boldsymbol{A}$ find out? If only moves are observed, we would have to say that having just a single uncertainty line for $\boldsymbol{A}$ between the real root and the 'pseudo-root' makes no sense. For, after move a is played, $\boldsymbol{A}$ has learnt nothing that would now enlighten him, so there should be an uncertainty line at the mid-level as well. But in another sense, $\boldsymbol{A}$ *has* learnt something! He now knows that $\boldsymbol{E}$ is uncertain, so he is in the game on the left. To make sense of this second scenario, we have to assume that introspection into $\boldsymbol{A}$'s epistemic state also counts as an update signal.

We leave matters here. What we hope to have shown is that diversity of agents raises some interesting issues, while sharpening our intuitions about the required mix of update and revision in games. In particular, instead of theorizing about abstract revision mechanisms, a hierarchy of agent types suggests very concrete switching scenarios as our beliefs about a type get contradicted by events in the course of the game.

Merging update logic and temporal logic To make sense of the issues in this section, we need to introduce a richer framework than our dynamic-epistemic logic so far. We now need to maintain global hypotheses about behaviour of agents in future courses of the game, which can be updated as time proceeds. This temporal intuition reflects computational practice, as well as philosophical studies of agency and planning (cf. Belnap et al. 2001). It is also much like questions in standard game theory about predicting the future behaviour of one's opponents: 'rational', or less so. Technically, we think the best extension for this broader sort of update would be *branching temporal models* with a suitable language referring to behaviour over time (cf. Fagin, Halpern, Moses & Vardi 1995, Parikh & Ramanujam 2003). The above tree structures $\mathcal{E}$ can easily support such a richer language. Van Benthem 2004a has a few speculations on update in such a temporal setting, but we leave the matter for future investigation.

3.7 Conclusion

The point of this paper is quite modest. We think that diversity of agents is a fact of life, and moreover, that it is interesting from a logical point of view. Indeed, one could even apply it to other logical core tasks, like inference by more clever and more stupid agents. Technically, we have shown that it is easy to describe different kinds of epistemic agents in update logics. Several interesting questions arise now. One is the further mathematical study of special patterns in arbitrary imperfect information games, viewed as trees of actions with epistemic uncertainties. In particular, our representation results may have more sophisticated versions for other kinds of behaviour. One could see this as the fine-structure of general models for dynamic-epistemic logic. Also, we would like a richer temporal perspective, where belief changes as expectations about the future are revised. This calls for a merge of temporal logic, update logic, and belief revision. Finally, we think that interaction of diverse agents is a topic with many logical repercussions, of which we have merely scratched the surface.

Acknowledgement We thank Josh Snyder for his penetrating comments, and for raising some exciting follow-up issues about our framework that we must leave for other occasions.

Chapter 4

Future Work

We already summarized the main contributions of this thesis in the introduction. Here we conclude with a discussion of several possible directions in which this work can be extended. Some of them have been addressed to some extent in the preceding:

- ***Plausibility logic with diverse agents*** We witnessed five different types of agents in Chapter 2, their different policies to update the belief plausibility when encountering incoming information. How to incorporate such diversity into current plausibility logic? Plausibility logic has not distinguished different agents within its system, there are no axioms for special sorts of agents. Given the diversity introduced, what will the logical system look like? As we know, the most important axioms in dynamic part are reduction axioms. For instance, because of the different update policies of the belief plausibility, the reduction axiom on the belief operator for one sort of agents must be different from that for another. A reduction axiom for the high conservative agents has been proposed at the end of Chapter 2. What are the reduction axioms for others?

- ***Extending the dynamic part of plausibility logic*** In Chapter 2 we mainly extended static plausibility logic. The common knowledge operator, common belief up to a degree have been added. However, is it also possible to extend dynamic plausibility logic? Precisely speaking, we need new reduction axioms for the operator of common knowledge and common belief respectively. This does not seem so straightforward. If we also take the diversity of agents into account, the situation will become even more complicated.

- ***Interaction between different agents*** So far, the update mechanism for Perfect Recall agents and k-memory agents have become clear to us. However, these agents do not exist in isolation, they may coexist in the same system and interact with each other. How to describe the interaction between them? If they know each other's type, how do they make use of such knowledge? If they do not know each other's type, how can they get this knowledge through interaction? They can certainly make hypotheses about each other, then revise their belief as time goes by. How to characterize such a process by plausibility logic? Is plausibility logic sufficiently sophisticated to treat such phenomena?

- ***Adding time to plausibility logic*** As just illustrated, agents may revise their hypotheses as time passes. In other words, agents maintain global hypotheses about the behavior of other agents in the future, which can be updated as time proceeds. To make sense of the issues playing a role here, we need to introduce a richer framework than our dynamic-epistemic logic so far provides. Adding time to plausibility logic seems promising. Van Benthem 2004a has a few speculations on update in such a temporal setting.

- ***Other sources of diversity?*** In this thesis, our concerns about bounded rationality have focused on the diversity of agents in 'belief revision policies', 'memory bounds' and 'selection mechanisms' of bounded agents in computer science, but are there other sources of diversity of agents which also are relevant to our logic?

- ***Application to game analysis*** We now have a plausibility logic which contains many important epistemic notions: knowledge, belief up to a degree, common knowledge, common belief up to a degree. Moreover, we also know how knowledge is updated, how belief is revised and about the interaction between knowledge or belief and action. On the other hand, we have various kinds of agents, they have different memory capacity, different strategies to revise their belief plausibility. Games provide us a free playground to test our ideas. Applying our results to analyzing games seems very interesting. For instance, van Benthem 2001 shows how imperfect information games with memory-bounded players can be analyzed in an ordinary dynamic-epistemic logic with added uncertainty links. Such work will help us to better understand real intelligent human behavior, and may also help us to improve the logical system itself.

Appendix A

Atomic Plausibility Logic

As we know, in contrast to update logic which only has the knowledge operator, the system $\mathsf{PL_S}$ had a new belief operator B_j^k, by which it is possible to express the firmness or degrees of the belief. Incidentally, we will show in this chapter that the meaning of $B_j^k\varphi$ can be expressed by a propositional constant together with the knowledge operator. This yields a new atomic system, the completeness of this new system will also be given.

A.1 Language and Semantics

Definition A.1.1 *Let a finite set of proposition variables Φ and a finite set of agents G be given. The **language** $\mathcal{L}_S^-$ is given by the rule*

$$\varphi := \top \mid p \mid p_j^k \mid \neg\varphi \mid \varphi \wedge \psi \mid K_j\varphi,$$

where $p \in \Phi$, $j \in G$, and $k \in \mathbb{N}$.

We introduce a new atomic proposition constant p_j^k, which means 'agent j assigns the world where she stands the degree of belief at most k'. Putting it formally, we have the definition as below:

Definition A.1.2 $\quad M, w \vDash p_j^k \quad$ *iff* $\quad \kappa_j(w) \leq k.$

Now, we can define the belief operator with this constant and the knowledge operator as:

Definition A.1.3 $B_j^k\varphi := K_j(p_j^k \to \varphi)$.

Then we obtain a new system PL_S^- which consists of the following axioms and derivation rules:

1. All propositional tautologies.
3. $K_j(\varphi \to \psi) \to (K_j\varphi \to K_j\psi)$ *K_j-distribution*
4. $K_j\varphi \to \varphi$
7. $p_j^m \to p_j^{m'}$ for all $m \leq m'$
8. p_j^m $m \geq Max$

9. From $\vdash \varphi$ and $\vdash \varphi \to \psi$ infer $\vdash \psi$
11. From $\vdash \varphi$ infer $\vdash K_j\varphi$ *K_j-generalization*

As we can see, there are no axioms or rules for the belief operator B_j^k in the system PL_S^-. In this sense, we may say that we *deconstruct* the system PL_S. In the next section, we will give the completeness proof for the new atomic system PL_S^-.

A.2 Completeness Theorem

Theorem A.2.1 *The system PL_S^- is complete with respect to its canonical model.*

Proof To prove the completeness of PL_S^-, it suffices to show that

> Every PL_S^--consistent set Γ of formulas is satisfiable in some epistemic model.(*)

To get this, we define the canonical model as follows:

$M^c = (W^c, \sim_j, \kappa_j, V)$

- $W^c = \{w_W\colon W$ maximal PL_S^--consistent set$\}$

- $\sim_j = \{(w_W, w_V)\colon W/K_j \subseteq V\}$ where $W/K_j = \{\varphi\colon K_j\varphi \in W\}$
- $\kappa_j(w_W) = \min\{k\colon p_j^k \in W\}$
- $w_W \in V(p)$ iff $p \in W$

Now we need to show that

$$\varphi \in V \Leftrightarrow M^c, w_V \models \varphi.$$

By induction on the structure of formula φ. We only consider the case of the constant p_j^k:

($\Rightarrow$) Assume $p_j^k \in V$. We have $\kappa_j(w_V) \leq k$. Then by Definition A.1.2, we get $M^c, w_V \models p_j^k$.

($\Leftarrow$) Assume $M^c, w_V \models p_j^k$. We know $p_j^{\kappa(w_V)} \in V$ and $\kappa_j(w_V) \leq k$. By axiom 7, $p_j^{\kappa(w_V)} \to p_j^k$. So, we get $p_j^k \in V$. ■

Next, we will prove that each theorem of the system $\mathsf{PL_S}$ is a theorem of the system $\mathsf{PL_S^-}$. To get this result, we first define an embedding function t from $\mathcal{L}_S$ to $\mathcal{L}_S^-$ as below:

$$t(p) = p$$
$$t(\neg\varphi) = \neg t(\varphi)$$
$$t(\varphi \vee \psi) = t(\varphi) \vee t(\psi)$$
$$t(K_j\varphi) = K_j t(\varphi)$$
$$t(B_j^k\varphi) = K_j(p_j^k \to t(\varphi)).$$

Lemma A.2.2 *Let M^* be the model of the system $\mathsf{PL_S^-}$. Then we have: $M^*, w \models t(\varphi)$ iff $M, w \models \varphi$.*

Proof Induction on the structure of formula φ. We only consider the case $B_j^k\varphi$.
$M^*, w \models t(B_j^k\varphi)$
iff $M^*, w \models K_j(p_j^k \to t(\varphi))$
iff $\forall v(w \sim_j v \Rightarrow (M^*, v \models p_j^k \Rightarrow M^*, v \models t(\varphi)))$
iff $\forall v(w \sim_j v \Rightarrow (\kappa_j(v) \leq k \Rightarrow M, v \models \varphi))$ (By inductive hypothesis)

iff $M, w \models B_j^k \varphi$. ■

Theorem A.2.3 $\vdash_{\mathsf{PL_S}} \varphi$ *iff* $\vdash_{\mathsf{PL_S^-}} t(\varphi)$.

Proof ($\Rightarrow$) is easy to prove by induction on the length of the proof.
($\Leftarrow$) By contraposition. Suppose it is not the case that $\vdash_{\mathsf{PL_S}} \varphi$. Then, by completeness, for some model M, world w, $M, w \models \neg\varphi$. By Lemma A.2.2, $M^*, w \models \neg t(\varphi)$. So, it is not the case that $\vdash_{\mathsf{PL_S^-}} t(\varphi)$. ■

Remark Given the atomic plausibility logic $\mathsf{PL_S^-}$, we can extend that also with the common knowledge operator. The way we add the common knowledge operator will be completely similar to what we did with static plausibility logic $\mathsf{PL_S}$ in Chapter 2. Furthermore, the completeness proof is also the same. For common belief, thanks to the propositional constant p_j^k, we now define common belief with a knowledge operator together with p_j^k instead of a belief operator, the completeness proof turns out more trivial than before.

Appendix B

Update for Forgetful Agents

In this chapter we begin with a motivating example, then present two different proposals of update for forgetful agents: the inclusive proposal and the copy action proposal, and end with a comparison between them.

B.1 Preliminaries and Motivating Example

We start with several preliminaries as follows:

- ***Agents with n-memory*** In this chapter, forgetful agents are also viewed as 0-memory agents. By 'an n-memory agent', we mean that agent remembers only the n actions before the most recent action. A 0-memory agent, then, knows only what she learned from the most recent action; an agent with memory of length 1 knows only what she learned from the most recent action and the one before it.

- ***Shift perspective of product update*** As addressed at the end of Section 3.3, there are various possible perspectives of product update! We now turn to one of them, i.e. we do product update as: $\boldsymbol{M} \times \boldsymbol{A}_1 \times \boldsymbol{A}_2 \ldots$ where actions models can be different. Recall that in order to prove the characterization theorem, we had to give a more complex perspective of product update in Section 3.3 to make it fit the form: $\boldsymbol{M} \times \boldsymbol{A} \times \boldsymbol{A} \ldots$ where the action models are uniform.

- ***Update for Memory-free agents*** For comparison and further discussion, we give the definition of update for Memory-free agents again:

Definition B.1

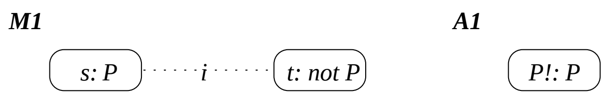
(1*a*) $\boldsymbol{M} \times \boldsymbol{A} = \{(s, a) : s \in \boldsymbol{M} \text{ and } a \in \boldsymbol{A} \text{ and } \boldsymbol{M}, s \models PRE_a\}$.

(1*b*) $(x, a) \sim_i (y, b)$ *iff* $a \sim_i b$.

(1*c*) $\boldsymbol{M} \times \boldsymbol{A},\ (s, a) \models p$ *iff* $\boldsymbol{M}, s \models p$.

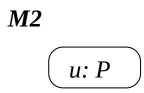
The item (1*b*) means that new uncertainty comes only from uncertainty about observed actions.

Motivating example In the following, we will take Snyder's example as our basic example. The example consists of an agent i uncertain about P at first; then P is announced, an action Id takes place afterwards. We will look at how the agent i updates her knowledge encountering these actions. The original state model $\boldsymbol{M}_1$ and the first action model $\boldsymbol{A}_1$ are pictured below:



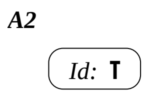
there are two possible worlds in the state model $\boldsymbol{M}_1$, where the agent i is uncertain about P. The dotted line expresses i's uncertainty, the valuations of the propositions are represented in the state model: P holds at s, not-P holds at t. There is only one action in the action model $\boldsymbol{A}_1$: P is announced, we also put the precondition, P after the colon in the action model.

To take the product update according to Definition B.1, we have the following new state model $\boldsymbol{M}_2$, abbreviating $(s, P!)$ as u:

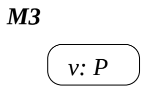


Since P does not hold at the world t, the action $P!$ only possibly happens at the world s, so we get only one world u in the new model. Intuitively, after the announcement of P, agent i should now know that P, and obviously the agent i *does* know P in $\boldsymbol{M}_2$.

Now the Id action happens; its action model is shown below:



There is only one action in $\boldsymbol{A}_2$: Id, which takes place everywhere. By Definition B.1, we get the new state model $\boldsymbol{M}_3$, abbreviating (u,Id) as v:



Intuitively, the action Id being performed, the forgetful agent i should no longer know whether P, because she has 0 memory, she already forgot what had happened one step ago, and she should be uncertain again whether P. But by the above illustration, the agent i *knows* P. This is counter-intuitive!

New definition called for As we have seen, if we take Definition B.1 as that of the update for forgetful agents, we get a counterintuitive conclusion.[1] In the above example, to get uncertainty after the Id action, we need another world which comes from t, so that not-P can hold at that world. However, as shown above, the world t was gone forever after the first update since it does not satisfy the precondition of the first action $P!$. So, the action model $\boldsymbol{M}_2$ can be only executed at the state u and it is impossible for the world t to come back – this is what we do not want when modelling agents with bounded memory. Because of the limited memory capacity, bounded memory agents may be uncertain about something afterwards even if they are certain about it beforehand. This is reality! To get this correct, we will present two different proposals in the following two sections.

B.2 Inclusive Proposal

Snyder presents the following new definition of update for 0-memory agents:

Definition B.2

(2a) $\boldsymbol{M} \times \boldsymbol{A} = \{(s,a) : s \in \boldsymbol{M} \textit{ and } a \in \boldsymbol{A}\}$.

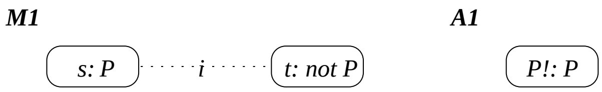
(2b) $(s,a) \sim_i (t,b)$ *iff* $(\boldsymbol{M}, s \models PRE_a$ *iff* $\boldsymbol{M}, t \models PRE_b)$ *and* $a \sim_i b$.

In contrast to (1a) of Definition B.1, (2a) leaves the precondition restriction out, this simply makes it possible to keep all the worlds around. (2b) now

[1] Recall that we now take a new perspective on product update, i.e. the action models can be different. If we stick to the perspective in Section 3.3, i.e. the action models are uniform, Definition B.1 works.

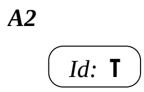
has a clause ($\boldsymbol{M},\ s \models PRE_a$ iff $\boldsymbol{M},\ t \models PRE_b$) giving restrictions to the states. This clause now plays a very crucial role to define the uncertainty relation in the new models. The item about the fact update is omitted. It would be the same as ($1c$). We will also ignore it in the later discussions. In order to understand this new definition, we now look at the example again:

The original state model and action model stay as before:

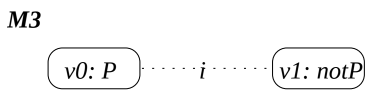


According to Definition B.2, we obtain a different new state model $\boldsymbol{M}_2$ this time, abbreviating $(s, P!)$ as u_0 and $(t, P!)$ as u_1:

There are two worlds in $\boldsymbol{M}_2$, one is the *possible* world, u_0, in the sense that the precondition of the action holds in it, another is the *impossible* world, u_1, in the sense that the precondition of the action does not hold in it. This is different from what we obtained from Definition B.1. Furthermore, by ($2b$), there is no uncertainty relation between these two worlds. So the agent i knows that P in $\boldsymbol{M}_2$. Now the same action Id happens. Again we picture the action model below:



By definition B.2, we get a new state model $\boldsymbol{M}_3$, abbreviating (u_0, Id) as v_0 and (u_1, Id) as v_1:



By ($2b$) of the definition, we get the uncertainty relation as shown above between the two states. That means, the agent i is uncertain whether P. This conclusion is what we expect for a 0-memory agent.

Snyder 2004 has an extensive discussion on how to extend the result for 0-memory agents to k-memory agents. This is not hard to do it in technical regards, but it indeed give rise to several complex situations that we should take into account. We are not going to review this here.

B.3 Copy Action Proposal

To make the update fit the behavior of 0-memory agents, an alternative proposal is to simply introduce a copy action to bring those states that would have vanished back into the update process. We propose the following definition:

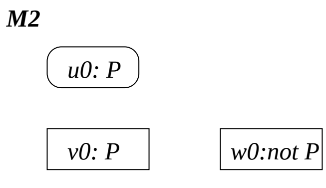
Definition B.3

(3a) $\boldsymbol{M}\times\boldsymbol{A}=\{(s,a) : s \in \boldsymbol{M}$ *and* $a \in \boldsymbol{A}$ *and* $s \models PRE_a\}$.

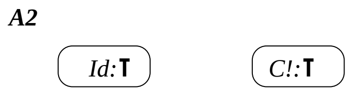
(3b) *For* $a, b \neq C!$, $(s,a) \sim_i (t,b)$ *iff* $a \sim_i b$.

In contrast to Definition B.1, we let the item (1a) unchanged. In order to keep states presence, a copy action, $C!$, is introduced and moreover we set $PRE_{C!} = \top$, i.e. it can take place in any state. To see how the copy action works, let us look at the above example, but now with additional copy action in the action model:

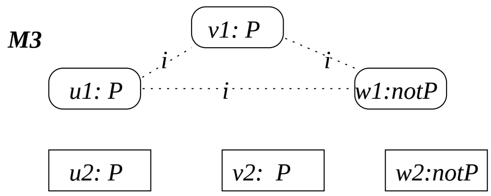
By Definition B.3, with the abbreviations: $u_0 \hookleftarrow (s,P!)$; $v_0 \hookleftarrow (s,C!)$; $w_0 \hookleftarrow (t,C!)$, we obtain the new state model $\boldsymbol{M}_2$ below:



The agent i then knows P. We take the rectangular box to express the states arising from the copy action. We add the copy action in the same way to the Id action, obtaining the new action model $\boldsymbol{A}_2$ as shown below:



According to Definition B.3 again, with the abbreviations: $u_1 \hookleftarrow (u_0, Id)$; $v_1 \hookleftarrow (v_0, Id)$; $w_1 \hookleftarrow (w_0, Id)$; $u_2 \hookleftarrow (u, C!)$; $v_2 \hookleftarrow (v_0, C!)$; $w_2 \hookleftarrow (w_0, C!)$, we get the new model $\boldsymbol{M}_3$, pictured below:



We also get the uncertainty relation by item $(2b)$. That means the agent i is uncertain whether P. Again this is what we expect for 0-memory agents.

Copy action restricted As we have seen, copy action plays a considerable role in bring back those worlds that would have vanished. However, note that not only the worlds that do not satisfy the preconditions of actions return, but those that *do* satisfy the preconditions also return, because we set the copy action to take place in any state. In fact, it is not necessary to bring the latter worlds back since they are already there because of the actions that really happened. That means, the copy action has done *much more* than needed, and so there are many more states in the state models. Fortunately, it turns out to be possible to restrict the copy action to reduce the number of the redundant states in the state models, i.e. we restrict copy action $Cr!$ by setting its precondition to $PRE_{Cr!} = \top r$, namely, the restricted copy action *only* happens in those states that do not satisfy the preconditions of the non-copy actions. We give the following definition:

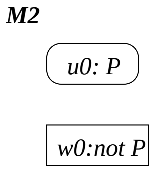
Definition B.4

(4a) *For* $a \neq Cr!$*,* $\boldsymbol{M}\times\boldsymbol{A}=\{(s,a) : s \in \boldsymbol{M}$ *and* $a \in \boldsymbol{A}$ *and* $s \models PRE_a\}$
$\cup \{(t, Cr!) : t \in \boldsymbol{M}$ *and* $t \not\models PRE_a\}$.

(4b) *For* $a, b \neq Cr!$*,* $(s,a) \sim_i (t,b)$ *iff* $a \sim_i b$.

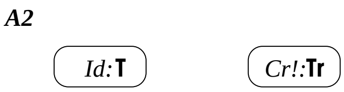
This definition seems a little complex, but the intuition behind it is easy to understand: Item $(4a)$ just says that we let the copy action *only* copy those states that do not satisfy the preconditions of the non-copy actions instead of copying all the states as defined in B.3. We will see the number of copies reduced considerably in this way. Let us now revisit the above example. The state model $\boldsymbol{M}_1$ and the action model $\boldsymbol{A}_1$ are the same as before except with restricted copy action:

According to Definition B.4, with the abbreviations: $u_0 \hookleftarrow (s, P!)$; $w_0 \hookleftarrow (t, Cr!)$, we obtain $\boldsymbol{M}_2$:

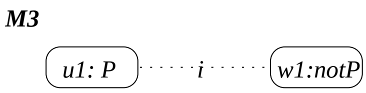


Here the restricted copy action only happens in the state t, since the precondition of the action $P!$, P, does not hold at t. As before, the agent i knows P.

The action model $\boldsymbol{A}_2$ is given as follows:



By Definition B.4, with the abbreviations: $u_1 \hookleftarrow (u_0, Id)$; $w_1 \hookleftarrow (w_0, Id)$, we get the new model $\boldsymbol{M}_3$ pictured below:



Note that the restricted copy action here does nothing because the action Id happens in all the states. Obviously, we obtain the same results we got when applying Definition B.3: The agent i is uncertain whether P. But now the number of the worlds in the state model has been reduced. Nevertheless, our further discussion will still refer to Definition B.3 instead of B.4.

Extension to k-memory agents We now give the main idea how to extend the copy action proposal for 0-memory agents to the case of k-memory agents. We start with the case that the agent has 1-memory, the update definition is given by:

Definition B.5

(5a) $\boldsymbol{M}\times\boldsymbol{A}_{-1}\times\boldsymbol{A}$
$=\{(s, a_{-1}, a)\text{: } (s, a_{-1})\in\boldsymbol{M}\times\boldsymbol{A}_{-1} \text{ and } a\in\boldsymbol{A} \text{ and } (s, a_{-1}) \models PRE_a\}$

(5b) *For* $a_{-1}, a, b_{-1}, b \neq C!$,
$(s, a_{-1}, a) \sim_i (t, b_{-1}, b)$ *iff* $a_{-1} \sim_i b_{-1}$ *and* $a \sim_i b$

It is not difficult in the same way to generalize this result to the k-memory case. Here we only give the definition for the uncertainty relation:

For any non-copy action $a_{-k}, \ldots, a_{-1}, a, b_{-k}, \ldots, b_{-1}, b$,
$(s, a_{-k}, \ldots, a_{-1}, a) \sim_i (t, b_{-k}, \ldots, b_{-1}, b)$
iff $(a_{-k} \sim_i b_{-k})$
$\ldots$
and $(a_{-1} \sim_i b_{-1})$
and $(a \sim_i b)$.

This ends our discussion of the two proposals. In the next section we compare them.

B.4 Comparison

Two paths to the same goal As illustrated in section B.1, Definition B.1 fails to characterize the update for forgetful agents because some worlds that do not satisfy the preconditions of actions disappear forever, this is not what we want. Our goal is to keep those worlds around, this is achieved by both of the above two different proposals, either leaving the precondition restriction out, as we did in the inclusive proposal, or introducing a copy action to bring these worlds back, as we did in the copy action proposal.

From the technical point of view, whether to keep the *precondition restriction* or not is the major difference between these two paths. This simple choice comes from even deeper different intuitions:

- The inclusive proposal eliminates the precondition restriction, witness (1*b*) of Definition B.2. This simply produces all of the worlds for the updated model: not only the possible ones but also the impossible worlds. The intuition is that one *should not* remove the worlds because of the precondition restriction. For bounded memory agents, all the possible worlds make sense, including impossible worlds. Such impossible worlds may be useful to the agents at later stages. So, it is wise to keep them around from the very beginning.

- The copy action proposal keeps the precondition restriction but brings the worlds back by the copy action. The reason for keeping the precondition restriction is that it helps one see exactly how the update proceeds, i.e. which states we can get after some action happens. This is what an agent expects, even if she is memory bounded. This is similar to the case for Perfect Recall agents. But it pays the price that some worlds will disappear because of the precondition restriction. In order to bring them back, the copy action is added. So the copy action here can be viewed as an ancillary tool, which remedies that fact that worlds disappear because of the precondition restriction.

Two questions As we have seen, the above two proposals work very well. However, it seems that some further explanation needs to be made:

- For the inclusive proposal, as we have seen, it produces not only possible worlds but impossible worlds in the updated model. For instance, applying Definition B.2 to the example, we get possible world u_0 and impossible world u_1. They are put together in the new model. One is inclined to think that they are of the same kind. How can we see the difference between them? How to interpret such impossible worlds? On the other hand, though the agents are memory bounded, they still hope to distinguish the possible worlds from the impossible ones at a specific stage, so it is intuitively desirable that the agent can notice such differences between two kinds of worlds: some satisfy the precondition, others not.

- In contrast to the inclusive proposal, for the copy action proposal we can easily distinguish the impossible world from the possible one in the updated model, since the impossible worlds arising by copy actions. But there is another question we must clarify. As we know, the original action model does not include the copy action. We simply add a copy action to the action model and moreover require that it can happen in any state. Furthermore, the copy action is also added to every action model later on. It is totally different from other actions really occurring in the action model. This seems rather artificial, perhaps some further explanation is needed to be made from other perspectives.

We will leave these and perhaps other questions for future research.

In this chapter, we have concentrated on the two proposals for the update of 0-memory agents, investigated their different techniques and intuitions, and raised different questions for them. Bounded memory agents are new to us, even their update mechanism is new to us, probably there are many properties we have not thought of or covered yet. In the next chapter, we will turn to reality to explore how bounded memory agents work in computer science, which may give us some inspiration to improve our proposals of update for bounded memory agents.

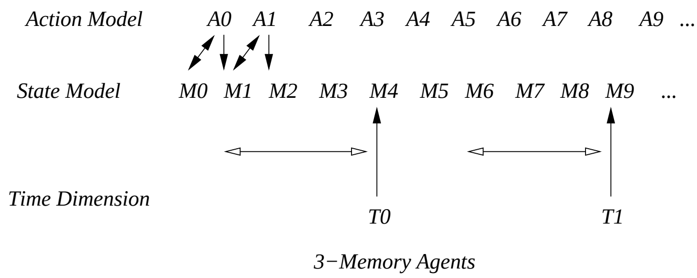
Appendix C

What to Forget for Bounded Agents? Even Diverse...

In this chapter, we will begin with a simple observation, i.e. the update for bounded memory agents we obtained presupposes that the agents forget the earliest information when new information comes in, but this is not always the case. We then describe some possible ways out that the bounded memory agents may take, for instance, they may forget information that has not been used so frequently. Incidently, this is also extensively studied in computer science. We then review the research in computer science and analyze a real example to understand the various behaviors of the bounded memory agents. Finally we conclude that the previous proposals of update for bounded memory agents are not adequate to characterize those properties that bounded memory agents have, a new source of diversity needs to be considered.

C.1 Always the Earliest?

In our preceding discussion about how the bounded agents update their information, there is an implicit assumption: the agent always forgets the earliest information when new information comes in. In other words, given limited memory, the agent does not choose which information should be remembered, which one can be forgotten. This means the agent does not use her own preference regarding the information in her memory. Suppose there is a time dimension, then the earliest information is replaced by the newest one, pictured as follows:



The agent has 3-memory. At time point $T0$, the information in her memory is described by the line with two empty arrowheads at each end, which traces the state models from $M1$ to $M3$. Similarly, at time point $T1$, her information concerns the state models from $M6$ to $M8$.

But do bounded agents *always* forget the earliest information? In the above picture, it can easily be imagined that the 3-memory agent may choose to forget $M8$ rather than $M5$ at time point $T1$, that is she chooses to remember the information about model $M5$, $M6$ and $M7$ instead of $M6$, $M7$ and $M8$. It seems reasonable that the agent has preferences about the information in her memory: She may think some information to be more important, and some to be trivial. When she must forget something, she may *choose* something trivial to forget and remember something of the most importance. This is a quite natural idea. A new question arises: What kind of preference may bounded agents have about the information in their memory? Putting in another way, which information is to be forgotten?

Intuitively again, the simplest case is that the agents choose to forget the information they have used for the *longest* period to forget. This is exactly what we did observe above. Nevertheless, there are many other possibilities: the agent may choose to forget the information that she has not used so *frequently*, or she may choose to forget the information that she thinks will not be used in the near *future*, etc. Incidently, this has been investigated extensively in computer science. We will encounter more precise discussions in the following, where we think of the device of limited memory in computer science as bounded memory agents. A brief review will appear in the next section.

C.2 Bounded Agents in Computer Science

Memory has been a very important notion in computer science. It often refers to some device consisting of a large array of bytes. To execute a process, normally, the *whole* program and data of this process must be in the physical memory. No matter how large the size of the physical memory has become by the technical progress, it usually has a limited size. If the space that one process needs is much larger than the memory can provide, how to execute the process? Can we only load some parts of a process into physical memory to execute it? This is the goal of ongoing so-called 'virtual memory' research. Virtual memory allows the execution of processes that can not be completely in the physical memory. So it is possible that an extremely large virtual memory is available for programmers when only a smaller physical memory is available.

What would happen if the process refers to a data (or a page) that was not brought into the memory? For instance, the memory of a 3-memory agent is full of data, and new data not in her memory is coming in for a new process execution, how does she update her data incorporating new information? Intuitively, because of the limited memory, the agent should first make sure that there is enough room to store the incoming data. She would choose which information should be discarded first, then absorb the new information. This process is called the *dynamic swapping process* where the operating system controls the swapping of data in and out of physical memory as they are required by the active processes. It decides *which* data is to be replaced by the request one. This is often referred to as 'replacement policy'. These policies embody exactly the various behaviors of the bounded agents. In the literature on this issue, many different replacement policies occur. Here we only list the main three approaches we are going to explore:[1]

- First In First Out (FIFO): Replace the 'oldest' data in the memory, i.e. the data loaded before all the others.
- Least Recently Used (LRU): Replace the data which has not been referenced since all the others have been referenced.

[1] There are many others we are not going to explain now, e.g.

- Last In First Out (LIFO): Replace the data most recently loaded into the memory.
- Least Frequently Used (LFU): Replace the data used least often of the data currently in the memory.

- Optimal (OPT): Replace the data that will not be used for the longest period of time. Obviously, this algorithm requires *future knowledge* of the replacement actions, which is not usually available. Thus, this policy is usually used for comparison studies.

The above three approaches correspond exactly to the possible directions that we have thought the agent may take intuitively. To better understand how these policies work, we turn to a specific example in the next section.

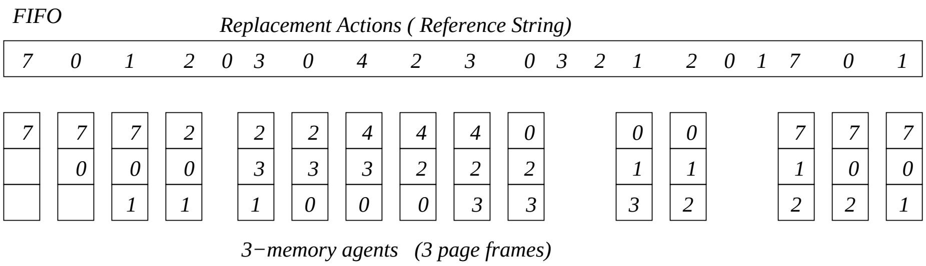
C.3 Various Update Behaviors

In this section, we will first focus on a concrete example from A. Silberschatz, P.B. Galvin & G.Gagne 2003, understanding the various behaviors of bounded memory agents, then further compare them from different points of view. To interpret the example in a new way, we first set the following 'term transference':

- The page frame is viewed as the memory space of the bounded agent, e.g. 3 page frames is 3-memory agent. The reference string is thought of as a series of actions, we will call them *replacement actions*.
- Replacement action has a special *property*: it brings in the new information and meanwhile replaces the old data in the memory. The number in the following example is also taken to express the data or information it brings in.
- In particular, the *precondition* of replacement action holds only at one preferred state which actually is the state to be replaced.

We will see the different replacement policies provide different rules to determine which state is the preferred one for the replacement action.

We now start with applying the policy FIFO to the example:



This is an information updating process of a 3-memory agent i when encountering incoming replacement actions. The first three actions are easy to understand, because the agent has three empty memory spaces and she can easily accept the first three incoming data, i.e. (7, 0, 1). After that, we obtain

K_i7, K_i0, and K_i1.

Then action 2 comes in, according to the rule given by FIFO, 7 was brought in first, so the agent selects 7 as the victim to replace. That is, we have

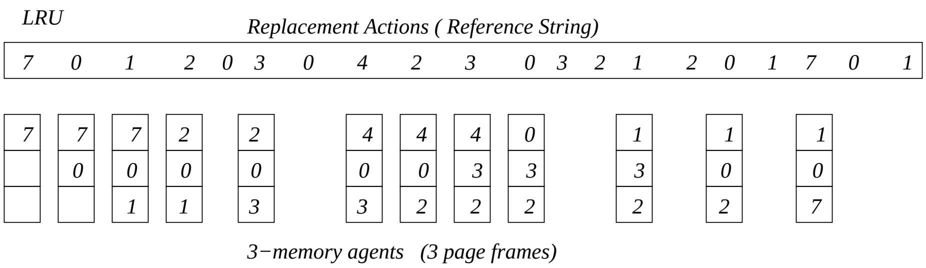
$PRE(2) = 7$.

That means action 2 only happens to the state 7. After this replacement action, we get

K_i2, and moreover $\neg K_i7$.

because the agent only has 3-memory, the incoming data 2 drives 7 out of her memory. Similarly, since 0 is already in memory, i.e. K_i0, so the agent need not update her information when 0 comes in, it will not add new information at all. The process continues as shown in the above picture.

Let us look at the behavior of the policy LRU in the above same example:



We only look at what happens when it comes to the action 4, where we already have

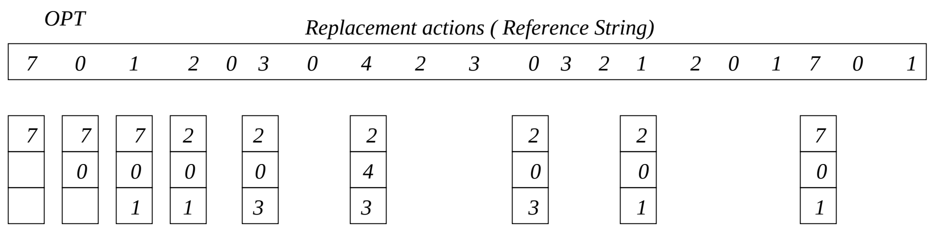
K_i2, K_i0, and K_i3.

In particular, among the three data in memory, 2 was used least recently according to the rule of LRU, i.e.

$$PRE(4) = 2.$$

so the agent chooses 2 as the victim to replace, which is different from what we saw in FIFO. According to LRU, the data to be replaced is the one which has not been referenced for the longest time.

Finally we turn to the behavior of the OPT policy to the above example. The same replacement actions are shown below:



3−memory agents (3 page frames)

We look at what happens when it comes to the action 2, we have

$$K_i7,\ K_i0, \text{ and } K_i1.$$

According to the rule of OPT, because 7 will not be used until the 18th replacement action, whereas 0 will be used at the 5th action and 1 at the 14th. That is,

$$PRE(2) = 7.$$

The action 7 chooses 2 to replace. Similarly, the action 3 replaces 1, as 1 will be the last of the three pages in memory to be referenced again.

Comparing the behaviors In the following, by 'page fault' we refer to an interrupt that occurs when a program requests data not currently in the physical memory. Given the above analysis, we now conclude the comparison in several aspects:

- **Page fault** As shown above, in all three policies the first three actions cause faults that fill the three empty memory space. For the whole process, different policies have different number of page faults:

OPT has only 9 page faults, while FIFO has 15 page faults and LRU has 12 page faults. This is easy to see from the above pictures since we only pictured the cases where page fault occurs. For a bounded memory agent, it is easy to understand that reducing the number of page faults means saving costs. Obviously, OPT has a better behavior than FIFO and LRU in this aspect.

- **Belady's Anomaly** How to reduce the number of page faults then? Intuitively, the easiest way is to add more memory space. We expect this strategy would make the number of page faults being reduced. However, this does not hold for every policy, especially not for FIFO though it is simple and also easy to understand. This unexpected result is the well-known Belady's Anomaly, it was first demonstrated by Belady in 1970.[2] But policies like LRU and OPT do not suffer from such anomalies, i.e. adding more memory space would reduce the number of page faults.

- **Implementation** Concerning the implementation issues, we have a problem with OPT: It is difficult to implement. As we have seen, the major reason is that it requires future knowledge of the replacement actions, this usually cannot be obtained. However, FIFO and LUR are easier to implement, the operating system can keep track of when each data comes in by recording the time of incoming information or by maintaining a stack of information.

- **Selection mechanism** The biggest difference between these three policies is that they have a different selection mechanism of the victim to replace, i.e. the precondition of the replacement actions varies from one option to another. FIFO selects the victim according to the preference in *time*: the replacement action happens to the earliest one. The *frequency* of the page being used is the basis on which LRU selects its victim: the replacement action happens to the one least recently used. OPT selects the one that *will not* be used for the longest time.

[2] To illustrate the problem, we take the following reference string:

3, 2, 1, 0, 3, 2, 4, 3, 2, 1, 0, 4, 2, 3, 2, 1, 0, 4.

The replacement procedure is omitted here, since it is easy to obtain by FIFO rule. It turns out that the number of faults for the 4-memory case is 15, which is greater than that the number of faults for 3-memory case, 14. That means adding a fourth memory space results in an increase in the page-fault rate.

Note that the precondition here is a little bit different from what we have in update logic, because the same action may have different precondition at different stages. E.g. in the above example for FIFO, we know that $PRE(2) = 7$ for the first action 2, but $PRE(2) = 3$ for the second action 2. It seems that the precondition here is a sort of *principle* rather than some specific formula or fact.

C.4 New Source of Diversity

In this section, we will conclude with several desirable properties of update for bounded memory agents, which calls for considerations:

First, diversity exists even in bounded agents with the same memory capacity! We are now familiar with the diversity of agents with different memory ability in Chapter 3: Some agents have Perfect Recall, others are Memory-free. We have witnessed how they update their information in a different manner. In the example just above, we have seen how 3-memory agents update their knowledge by three different policies. The main question we have considered is: which data is the victim to be forgotten or replaced? FIFO selects the earliest one, LRU selects the one least recently used, OPT selects the one that will not be used for the longest time. This means even for bounded agents with same memory, diversity still exists. We have seen the different strategies that 3-memory agents have taken in the above example. They may have different *preference*. Such kind of preference is significant and should be incorporated into the very definition of update for the bounded memory agents. Classical update logic has not done this, our new proposals in the previous chapter have not done so either. Further investigation and new technical improvement are needed!

Secondly, when comparing the different behaviors of the bounded agents in the preceding, the number of page-faults has been our focus. In the above example, FIFO has 9 page faults, while FIFO has 15 page faults and LRU has 12 page faults. Such a perspective is new to us. For the Perfect Recall agents who always remember everything that happens all the time, it is not necessary to think of the cost problem at all given their infinite memory ability. But circumstances change for bounded memory agents. We now do have to consider how to reduce the number of page-faults to save costs. To get a proper update mechanism for a bounded memory agent, such realities and related issues should definitely be taken into account.

The last point is that in the above example, when the replacement action happens, the old data in the memory is replaced by the incoming information. Actually it does not disappear at all, it goes into the so-called *backing store*. So once some information is called for again, we can fetch it from the backing store. Namely, we put the information replaced into the backing store for the future reference. The backing store seems like a stack that the physical memory bring with it. For bounded memory agents, it is necessary to have such a store. Recall that the same goal of the inclusive proposal and the copy action proposal in the previous section is to keep the worlds that would have disappeared around for future reference, it seems that at this point those two proposals have the same spirit as what occurs in computer science.

Summarizing, we have seen more diversity of the bounded memory agents here in computer science. To get a proper update logic for bounded agents, their preference, the cost, perhaps other things, should be incorporated into the very definition. We will leave this topic for future investigation.

Bibliography

[1] S. Abramsky: Semantics of Interaction: an Introduction to Game Semantics, in P. Dybjer & A. Pitts, eds., *Proceedings 1996 CLiCS Summer School*, Cambridge University Press, Cambridge, 1-31.

[2] C. Alchourrn, P. Gardenfors & David Makinson : On the logic of theory change: Partial meet contraction and revision functions, *Journal of Symbolic Logic* 50: pp. 510-530, 1985.

[3] G. Aucher: *A Combined System of Update Logic and Belief Revision*, Master of Logic Thesis, ILLC University of Amsterdam, 2003.

[4] R. Axelrod: *The Evolution of Cooperation*, Basic Books, New York, 1984.

[5] A. Baltag, L. Moss & S. Solecki: The Logic of Public Announcements, Common Knowledge and Private Suspicions, *Proceedings TARK* 1998, 43-56, Morgan Kaufmann, Los Altos, 1998.

[6] N. Belnap, M. Perloff & M. Xu: *Facing the Future*, Oxford University Press, Oxford, 2001.

[7] J. van Benthem: *Exploring Logical Dynamics*, CSLI Publications, Stanford,1996.

[8] J. van Benthem: Games in Dynamic-Epistemic Logic, *Bulletin of Economic Research* 53:4, 219-248 (Proceedings LOFT-4, Torino), 2001a.

[9] J. van Benthem: Extensive Games as Process Models, *Journal of Logic, Language and Information* 11: 289-313, 2001b.

[10] J. van Benthem: *Logic in Games*, Lecture Notes, ILLC Amsterdam & Philosophy Stanford, 1999-2003.

[11] J. van Benthem: One is a Lonely Number: on the Logic of Communication, Tech Report PP-2002-27, ILLC Amsterdam.

[12] J. van Benthem: Conditional Probability Meets Update Logic, *Journal of Logic, Language and Information* 12: 409-421, 2003.

[13] J. van Benthem & B. ten Cate: Automata and Update Agents in Event Trees, working paper, Department of Philosophy, Stanford University, 2003.

[14] J. van Benthem: A Mini-Guide to Logic in Action, *Philosophical Researches*, Supp: 21-30, Beijing, 2004a.

[15] J. van Benthem: Local versus Global Update in Games, working paper, Department of Philosophy, Stanford University, 2004b.

[16] P. Blackburn, M. de Rijke & Y. Venema: *Modal Logic*, Cambridge University Press, Cambridge 2001.

[17] G. Bonanno: Memory and Perfect Recall in Extensive Games, *Games and Economic Behavior* 47: 237-256, 2004.

[18] H. van Ditmarsch and Labuschagne, 2003, Dynamic Doxastic Logic for Defeasible Belief Revision, work in progress.

[19] H. van Ditmarrsch, 2004, Dynamic Belief Revision, Lecture Slides at University of Liverpool.

[20] R. Fagin, J. Halpern, Y. Moses & M. Vardi: *Reasoning about Knowledge*, The MIT Press, Cambridge (Mass.) 1995.

[21] P. Gardenfors: *Knowledge in Flux: Modeling the Dynamics of Epistemic States*, Bradford Books, MIT Press, Cambridge, Mass, 1988.

[22] J. Gerbrandy: *Bisimulation on Planet Kripke*, Ph.D dissertation, ILLC Amsterdam, 1999.

[23] D. Harel, D. Kozen & J. Tiuryn: *Dynamic Logic*, The MIT Press, 2000.

[24] V. Hendricks: Active Agents, *Journal of Logic, Language and Information* 12: 469-495, 2003.

[25] T. Hötte: *A Model for Epistemic Games*, Master of Logic Thesis, ILLC University of Amsterdam, 2003.

[26] S. Lindström & W. Rabinowicz: Extending Dynamic Doxastic Logic: Accomodating iterated beliefs and Ramsey Conditionals within DDL, L. Lindahl, P.Needham, and R.Sliwinski (Eds) *Foor Good Measure*, Uppsala Philosophy Studies 46, Uppsala University, Department of Philossophy, Uppsala, Sweden 1997.

[27] J.-J.Ch. Meyer & W. van der Hoek: *Epistemic Logic for AI and Computer Science*, Cambridge Tracts in Theoretical Computer Science 41, 1995.

[28] R. Moore: A Formal Theory of Knowledge and Action, J. R. Hobbs & R. C. Moore (Eds.), *Formal Theories of the Common Sense World*, Ablex Publishing, Norwood, NJ, pp. 319-358, 1985.

[29] M. Osborne & A. Rubinstein: *A Course in Game Theory*, The MIT Press, Cambridge (Mass.) 1994.

[30] R. Parikh & R. Ramanujam: A Knowledge-Based Semantics of Messages, *Journal of Logic, Language and Information* 12: 453-467, 2003.

[31] R. Parikh & R. Väänänen: Finte Information Logic, *http://www.sci.brooklyn.cuny.edu/cis/parikh/parikh-pubs.html*, 2003.

[32] B. Rodenhauser: *Updating Epistemic Uncertainty: an essay in the logic of information change*, Master of Logic Thesis, ILLC University of Amsterdam, 2001.

[33] K. Sergerberg: Belief Revision From the Point of View of Doxastic Logic, Bull,. of the IGPL, Vol.3 No.4,pp. 535-553 1995.

[34] A. Silberschatz, P.B. Galvin & G.Gagne: *Operating System Concepts*, John Wiley & Sons, Inc. New York, NY, USA 2003.

[35] J. Snyder: Product Update for Agents with Bounded Memory, manuscript, Department of Philosophy, Stanford University, 2004.

[36] W. Spohn. Ordinal conditional functiuns: A dynamic theory of epistemic states. In W. L. Harper and B. Skyrms (Eds), *Causation in Decision, Belief Change, and Statistics*, reidel, Dordrecht, vol.2, pp. 105-134, 1988.

[37] F. Veltman: Defaults in Update Semantics, *Journal of Philosophical Logic* 25: 221 - 261,1996.

[38] R. Wassermann: *Resource Bounded Belief Revision*, Ph.D dissertation, ILLC University of Amsterdam, 2000.

Index