Representing Plausible Beliefs about States, Actions, and Processes

by

Toryn Qwyllyn Klassen

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Representing Plausible Beliefs about States, Actions, and Processes

Toryn Qwyllyn Klassen
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
2021

This thesis deals with the topic of modelling an agent's beliefs about a dynamic world in a way that allows for changes in beliefs, including retracting of beliefs, based on the agent's observations. We work within the field of knowledge representation, and represent the beliefs of the agent using a logical theory. In particular, we are concerned with representing what initial conditions the agent considers (im)plausible, what effects the agent thinks actions (im)plausibly have, and what processes in the environment the agent thinks have (im)plausibly occurred or will occur.

Our approach uses the situation calculus, a standard knowledge representation framework for modelling action and change. Furthermore, we build on an existing framework in the situation calculus for modelling changing beliefs, where beliefs are determined using a plausibility ordering on situations. This supports modelling changing beliefs, since when the most plausible options are refuted by observations, the agent can fall back to the next most plausible options. Our concern is with how to specify this plausibility ordering using a logical theory. We propose to define the ordering by counting certain properties of situations, indicated by distinguished predicates, which we call "abnormality" predicates. This is inspired by how minimization of abnormalities has been used in circumscription, an approach to default reasoning.

We show how beliefs about plausible and implausible action effects can be represented by having the axioms describing effects refer to abnormalities. Furthermore, we extend the account of belief to allow for beliefs about ongoing exogenous processes, described by a program (written in ConGolog, a standard programming language for use with the

situation calculus). We show how having these programs refer to abnormalities allows for representing plausible and implausible environment behavior. Finally, we present a formal definition of "knowing how" to achieve goals, in terms of belief, which allows for the agent to change its beliefs about what it knows how to do.

Acknowledgements

I would like to thank my supervisors, Sheila McIlraith and Hector Levesque, without whom this thesis would not exist. I also thank the members of my supervisory committee, Graeme Hirst and Gerhard Lakemeyer, and former member David Olson, for their guidance. Additionally, I thank Fangzhen Lin for serving as my external examiner, and Michael Grüninger for serving on my examination committee.

Furthermore, I acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC), the Province of Ontario, the University of Toronto, and the Vector Institute.

Contents

Chapter 1

Introduction

1.1 Overview

People have beliefs about their environment. Some of these beliefs are about the environment's state, for example, that there is a coffee cup on the table. Other beliefs are about how actions can change the environment, for example, that picking up the cup will remove it from the table. Another sort of beliefs is about the events that are unfolding, for example, that a coworker will take the cup if it's left on the table. All these sorts of beliefs inform people's interaction with the world and their ability to achieve goals. Furthermore, observations can reveal beliefs to be mistaken or outdated, and (ideally) those beliefs get changed.

This thesis deals with the topic of modelling beliefs about a dynamic world in a way that allows for changes based on observations made by an agent. We work within the tradition of *knowledge representation*, where the beliefs of the agent are described using a logical theory. In particular, we are concerned with representing

1. what initial conditions the agent considers (im)plausible,
2. what effects the agent thinks actions (im)plausibly have,
3. and what processes in the environment the agent thinks have (im)plausibly occurred or will occur.

Representing plausibility supports modelling changing beliefs, since when the most plausible options are refuted by observations, the agent can fall back to the next most plausible options. To illustrate, imagine an agent that believes the coffee cup is on the table. The agent then moves its arm and hand in a certain way and believes that it has picked up the cup. However, after sensing that its hand is empty, the agent has to

revise its beliefs. The agent considers various possibilities – the cup was never on the table, its grip failed, or someone else took the cup first – and comes to believe what it considers the most plausible option, that it failed to pick up the cup. However, after additionally sensing that the cup is not on the table, that option is ruled out, and the agent compares the plausibility of the remaining two options. For someone to have taken the cup is considered more plausible, so the agent comes to believes that.

What this thesis aims to do is describe how to formally specify an agent so that such changes of belief come out as logical consequences. Being able to represent the agent's uncertainty, and give it the ability to retract its beliefs, arguably goes some way towards addressing the criticism that logic-based approaches are "brittle" (see, e.g., Domingos and Lowd, 2019). To the extent that the specification of the agent is designed by people, using plausibility reduces the burden on the designers to get the beliefs exactly right.

We conduct our work within one of the standard knowledge representation approaches for dealing with action and change, the *situation calculus* (McCarthy, 1963; McCarthy and Hayes, 1969; Reiter, 2001). In Reiter's version of the situation calculus, which is a language in second-order logic, situations represent histories of actions. From a situation, there are various possible successor situations, each corresponding to a choice of action to perform. Therefore, the set of situations is organized into a tree or forest (in the graph theory sense), depending on whether there is one or more initial situations. Properties that can vary from situation to situation (e.g., due to changes caused by actions, or because they vary between initial situations) are represented using *fluents*, predicates that take a situation argument.

An environment can be described in the situation calculus with a set of axioms, an *action theory*, which is typically handcrafted by a human axiomatizer. Action theories traditionally contain axioms describing the initial state, the preconditions of actions (when they are possible to execute), and how each fluent is changed by actions. Sometimes, an action theory as a whole is taken to represent the knowledge of an agent, but other times (and in this thesis) the theory is meant to describe reality and there may be additional axioms explicitly describing what is known or believed by the agent. (Multiple agents can also be considered, but we will not be doing so in this thesis.)

We now turn to discussing the modelling of belief. The standard way of describing beliefs or knowledge in logic, following Hintikka (1962), is in terms of *possible worlds*. An *accessibility relation* relates one world to another if in the first world the agent considers that the second world may be the actual one. What is known or believed by an agent in a particular world is defined as what is true in all accessible possible worlds. Belief and knowledge can be described in *modal* logics that introduce special operators for

these modalities. Alternatively, an accessibility relation can be encoded in classical first-order (or second-order) logic, as was done by Scherl and Levesque (2003). For them, the "possible worlds" were situations in the situation calculus.

Note that we are concerned with *categorical* beliefs – propositions are either believed or not. This can be contrasted with probabilistic representations of uncertainty, where the agent assigns probabilities to propositions instead of simply believing them or not. Probabilities have become the dominant way of describing uncertainty in many areas of AI, so while our work will not be based on using them, we say a bit more about them here.

Probabilities have the virtue of, in many cases, being straight-forward to estimate from frequencies in data. On the other hand, relevant data may not always be available, and it may be difficult for humans to come up with explicit probabilities reflecting their own degrees of belief. Another issue is that probabilities are not easily integrated with categorical beliefs. For example, if a proposition P was defined to be believed whenever its subjective probability was above some threshold t, then unless t is exactly equal to 1 (or 0), beliefs would not be closed under conjunction. That is, P could be believed and Q could be believed without the conjunction $P \wedge Q$ being believed. That goes against the standard logical account of belief, and even some logical versions of belief that are limited in an attempt to make computing beliefs tractable (e.g., Liu et al., 2004).

One response would be to just give up on the traditional notion of belief in favor of a probabilistic account. After all, the idea that beliefs should be closed under conjunction leads to the "paradox of the prefix" (Makinson, 1965), which involves a writer who believes each of the sentences written in a book, yet believes that the book contains errors. The idea that human beliefs should be described using probabilities has been taken up by some researchers in cognitive science (e.g., Goodman et al., 2014). What extent and role probabilities should play in describing human reasoning remains a topic of debate (Marcus and Davis, 2013; Johnson-Laird et al., 2015).

In this thesis, we will continue to use the traditional notion of categorical belief. Further philosophical discussion of the relation between probabilities and categorical beliefs can be found in the literature (e.g., Spohn, 1988; Hunter, 1996; Kaplan, 2005). For technical discussions of probability and various alternative representations of uncertainty, we refer the reader to Halpern (2003).

As previously mentioned, we want for it to be possible for beliefs to be changed by observations. The *belief revision* literature, following Alchourrón et al. (1985), has been concerned with modelling changes of categorical beliefs given new information. (Much of this literature is concerned with a somewhat more general and abstract setting than ours,

where beliefs are revised to accommodate arbitrary sentences.) In order to specify how beliefs can change and be retracted over time, some further structure beyond the possible worlds model is needed. An ordering on possible worlds can be used to govern how beliefs get revised, by having the agent's new beliefs determined by the top worlds (according to the ordering) that are consistent with the new information (Grove, 1988). The ordering can be thought of as indicating which worlds are more *plausible* than others. A revision of beliefs can be accomplished by just having the agent's observations cause worlds in which the observation is not true to become inaccessible (Friedman and Halpern, 1999a,b; Shapiro et al., 2011).[1]

In this thesis, we build on the framework of Shapiro et al. (2011), whose model of belief in the situation calculus extends Scherl and Levesque's (2003) by incorporating plausibility and so allows for belief revision. Our concern is with how to specify the plausibility ordering using a logical theory. The plausibility ordering provides the basis of how the agent initially calculates what it believes, and its beliefs after actions. What we propose is to define the ordering by counting certain properties of situations, indicated by distinguished fluents. We call these fluents "abnormality" fluents, after McCarthy (1986), though to be clear, what we are measuring is subjective plausibility to the agent, which may differ from normality.[2] The most plausible situations are those with where the abnormality fluents are minimized. By referring to abnormality fluents in appropriate ways in an action theory, an axiomatizer can specify the plausibility of various things. Importantly, in writing such a theory, the axiomatizer does not just specify what the agent believes, but also less plausible alternatives that the agent considers possible.

Thesis statement Measuring the plausibility of a situation by counting the number of abnormalities contained within it allows for a perspicuous way of representing revisable beliefs about various aspects of a dynamic environment, including its state, the effects and preconditions of actions, and the behavior of environment processes.

We should note that the way we "count" abnormalities is a little more complicated than we've explained so far, as we allow for abnormalities to have associated priority levels (also an idea from McCarthy (1986)). This can make it easier to axiomatize domains. For example, if a flying saucer were to arrive and abduct a person, should that correspond to the existence of one abnormality or two (or many)? The important thing may be that those events are much less plausible than everyday things, and so should be associated with (at least one) high priority abnormality.

[1] There are also more complicated approaches, where the ordering itself gets changed by revisions (e.g., Darwiche and Pearl, 1997).

[2] The relationship between normality and plausibility is discussed by Boutilier (1994).

This thesis is mostly at the level of a specification of how an agent should reason. However to support future work in automating the reasoning process, in Chapter 4 we have some results about a form of *regression*, a popular reasoning procedure for the situation calculus. The situation calculus has been used in implemented robots (e.g., Burgard et al., 1999) and software agents (e.g., McIlraith et al., 2001). The approach in this thesis could potentially find a role in such applications.

1.2 Contributions

Below we give an overview of the technical content of the thesis. We divide the contributions into three parts, corresponding to chapters 3–5, respectively.

1.2.1 Specifying plausibility levels (Chapter 3)

Chapter 3 describes our way of assigning plausibility levels to initial situations in the situation calculus. Shapiro et al. (2011) had described how the changes in an agent's beliefs over time could be modeled in the situation calculus by associating plausibility values with situations. However, Shapiro et al. did not provide any very convenient mechanism for specifying the assignment of plausibility values to situations.

We extend their framework by measuring plausibility by counting the number of "abnormal" atomic formulas true in a situation – more plausible situations have fewer abnormalities, roughly speaking (we also allow for abnormalities to have priorities). How plausible arbitrary formulas seem to the agent can be controlled by setting the epistemic accessibility relation so that the only accessible situations are those where there is some relation between abnormal atoms and other formulas.

Note that outside the context of beliefs, the idea of minimizing abnormalities has a long history in *circumscription* (McCarthy, 1980, 1986), a technique for *default reasoning*. Default reasoning involves making assumptions ("by default") so as to draw more conclusions than those that are entailed (in classical logic). Circumscription can define a form of entailment where, instead of considering what's true in all models of the premises (as in classical entailment), what's considered is what's true in the most "normal" models. For circumscription, the minimization of abnormalities was traditionally considered in terms of set containment, but there also is a variant where (as in our work) abnormalities are counted, *cardinality-based circumscription* (Liberatore and Schaerf, 1995, 1997; Sharma and Colomb, 1997; Moinard, 2000).

We show how this approach for specifying plausibility levels avoids some issues with

the rival method from Schwering and Lakemeyer (2014). Our approach allows for features that independently contribute to the plausibility of a situation to be easily described (avoiding the so-called "drowning problem"). Also, our approach allows for a (countably) infinite number of plausibility levels to be described. We also prove a result on how cardinality-based circumscription generalizes a form of *lexicographic entailment* (Benferhat et al., 1993; Lehmann, 1995), another default reasoning technique.

Finally, we consider a couple ways to extend our work in different directions. First, we consider allowing what's abnormal to change over time. This provides a simple way of representing the plausibility of exogenous events, though it also leads to some counterintuitive results. We will return to modelling exogenous events in Chapter 5. In the second extension, we propose action theories which allow the agent to have incorrect knowledge about the effects of actions. Our main result for them is that they mostly follow the postulates for belief revision proposed by Alchourrón et al. (1985) (the "AGM postulates").

Summary of contributions in Chapter 3

- We propose counting abnormalities as a way of defining plausibility levels within the framework of Shapiro et al. (2011), and formalize this using second-order logic.
- We show how this approach avoids some issues with the rival method for specifying plausibility levels from Schwering and Lakemeyer (2014).
  - Our approach allows for features that independently contribute to the plausibility of a situation to be easily described.
  - Our approach allows for a (countably) infinite number of plausibility levels to be described.
- We prove a result on how cardinality-based circumscription generalizes a form of lexicographic entailment, another default reasoning technique.
- We show how changing abnormalities can be used to assign plausibilities to the occurrence (or non-occurrence) of exogenous actions.
- We consider action theories which allow the agent to have incorrect knowledge about the effects of actions, and have a proof of how closely they follow the AGM postulates for belief revision.

1.2.2 Changing beliefs about domain dynamics (Chapter 4)

Chapter 4 applies the approach of specifying plausibility levels with abnormalities to describing the behavior of actions – their effects, preconditions, and the sensing information they provide to the agent. In particular, we focus on how theories can be written so as to control how general of conclusions an agent should draw from observations. To illustrate, we will propose a formal setting in which at one point an agent can believe (a formalization of)

> If I (try to) pick up anything, I will be holding it. (1.1)

and then, after sensing its failure to pick up a cup, believe

> If I pick up anything, I will be holding it – with the exception of that one cup that one time. (1.2)

After a second time failing to pick up the cup, the agent can conclude

> If I pick up anything, I will be holding it, unless it's that cup. (1.3)

Finally, after trying to pick up another object also doesn't result in it being held, the agent can conclude that

> If I pick up anything, I will be holding it as long as it's not slippery (and those two objects were slippery). (1.4)

We suggest a format for writing action theories so as to easily specify how much the agent should change its beliefs, and so are able to formalize the example above (in §4.4). In particular, we suggest patterns (using abnormalities) to follow when writing the axioms describing action effects, so that the agent can, as the result of unexpected observations, make the sorts of generalization seen in the example: that there was a one-time exception to the expected action effect, that the action behaves differently with respect to particular objects, or that the action behaves differently with respect to particular classes of objects.

More generally, we show that when axioms describing domain dynamics are written to refer to abnormalities, in some cases the agent will believe "normalized" axioms that don't refer to abnormalities. We show how our framework also allows for changing beliefs about the precondition axioms that specify when actions are possible to execute, and the sensing axioms that describe how sensors work. This means that, for example, the agent can compare the results from two sensors to conclude that one is broken.

Finally, we provide a result about how (potentially changed) beliefs about action effects can be incorporated into *regression* (Reiter, 2001, §4.5), a formula-rewriting procedure that can simplify theorem-proving. This suggests potential computational applications of our work.

Summary of contributions in Chapter 4

- We prove that (in some cases) when the axioms describing domain dynamics are written to refer to abnormalities, the agent will believe "normalized" axioms that don't refer to abnormalities.
- We propose patterns to follow when writing axioms about actions effects, in order to control how general of conclusions the agent draws about the behavior of actions from unexpected observations.
- We also show how our theories can be used to model changing beliefs about
  - the results of sensing, and
  - the preconditions of actions.
- We describe how to apply regression with our theories, including how to use beliefs about action effects within the regression procedure, and prove its correctness.

1.2.3 Environment processes and knowing-how (Chapter 5)

People's beliefs about the events occurring around them inform their understanding of the current state of the world, what has happened in the past, and what will happen in the future, as illustrated by the following everyday examples:

- A person goes to a meeting and expects to see the other invitees.
- Night is expected to follow day.
- A customer at a restaurant expects to be served what they ordered.

These sorts of beliefs are also important for people's ability to accomplish goals (e.g., after placing an order, the restaurant customer believes that to obtain food, it will now be sufficient to wait). In this chapter, we view such beliefs as coming about because it is also believed that certain exogenous processes (that is, processes that are external to the agent) are taking place.

We propose a logical account of the beliefs of an agent in the presence of ongoing exogenous processes. We also give a formalization of *knowing how* to achieve goals in such a setting, defining knowing how in terms of belief. This allows for changes of the agent's beliefs about what it knows how to do.

We continue to define belief as truth in the most plausible accessible situations, but use an accessibility relation which is defined using a program that represents knowledge about ongoing processes. The idea is that accessible situations must be ones that could have been reached by following the program. The program is written in the ConGolog programming language (De Giacomo et al., 2000), a standard language to use with the situation calculus. The actions that constitute a run of the program may include actions by the agent itself or by other entities, and are not necessarily observable to the agent. Note that Kelly and Pearce (2015) had suggested an accessibility relation like this as future work.

ConGolog programs can be non-deterministic, giving one way to represent uncertainty about the various things that are happening concurrently in the environment. Furthermore, by having the ConGolog program refer to abnormalities within its branching conditions, we can have that the agent considers some execution traces more plausible than others, and the agent will be able to revise its beliefs about what's going on. We prove that in some cases, the agent will believe that a "normalized" program that doesn't refer to abnormalities is running (analogously to how in Chapter 4 we prove that the agent may believe normalized dynamics axioms).

The example about ordering in a restaurant that we gave above recalls early work in AI on *scripts* (Schank and Abelson, 1975), which are representations of knowledge about what typically happens in common situations (ordering food in a restaurant is the best-known example). This sort of knowledge is not naturally represented in a traditional action theory in the situation calculus, which is focused on describing what changes are possible, not on what changes will happen most plausibly over extended intervals.

Returning to the topic of goals, our formal definition of "knowing how" generalizes a definition by Lespérance et al. (2000) to take exogenous processes into account. In later work, Lespérance et al. (2008) had considered knowing-how in the context of an exogenous process, and we borrow some aspects of their approach. However, they did not model false beliefs or plausibility, and so could not, for example, formalize an agent revising its beliefs about what it knows how to do. We also formalize a version of knowing-how which describes goals that can be achieved with sequential plans.

Summary of contributions in Chapter 5

- We present an approach to modeling defeasible belief in the situation calculus where the accessible situations over time are constrained to be reachable by following a ConGolog program.
- We prove that under some conditions, if the ConGolog program that's running refers to abnormalities, the agent will believe that a simpler "normalized" program that doesn't refer to abnormalities is running.
- We introduce a definition of knowing-how in terms of belief, that takes into account both how beliefs may be false and the running of exogenous processes.
  - We prove that this definition generalizes Lespérance et al.'s (2000), among other properties.
  - We also formalize a version of knowing-how which describes goals that can be achieved with sequential plans.
  - Our approach supports revision of beliefs about knowing-how.

1.3 Structure of the thesis

In Chapter 2, we provide technical background on the situation calculus, formal models of belief, and belief revision. Further related work that is relevant to particular later chapters is included in them. The three main technical components of the thesis, which were described in the previous section, are split among Chapters 3, 4, and 5. Chapter 3 describes how counting abnormality predicates can be used to establish a plausibility ordering on situations, and the advantages of this approach. Chapter 4 applies that technique to describing the plausibility of different domain dynamics. Chapter 5 considers exogenous processes and what the agent can be said to "know how" to do. Note that Chapters 4 and 5 are mostly independent of each other (and so it is not necessary to read Chapter 4 before Chapter 5). Finally, in the conclusion (Chapter 6) we suggest possible future work.

Chapter 2

Background

2.1 Introduction

This chapter provides background on formal models of action, knowledge and belief, and belief revision. We will assume familiarity with the basics of first- and second-order logic (see for example the textbook by Enderton (2001)). Note that except where otherwise specified, we are assuming a single-agent setting, so any beliefs are those of that agent. We could also think of all actions as being performed by that agent, though that makes less difference. Note that in §3.5.1 and throughout Chapter 5 we will explicitly distinguish between endogenous actions (by the agent) and exogenous actions. A couple further topics relevant to this thesis will be introduced in later chapters – non-monotonic reasoning in Chapter 3, and "knowing how" in Chapter 5.

In §2.2 we discuss modelling action and change, focusing on the situation calculus (§2.2.2). As mentioned in Chapter 1, the situation calculus is a language in second-order logic, in which the behavior of actions is described using an action theory, and such theories are typically handcrafted by a human axiomatizer. We describe the vocabulary of the situation calculus and the form that action theories take in some detail. We also cover calculating entailments of action theories. Finally, we discuss a programming language designed for use with the situation calculus, which we will apply in Chapter 5.

We then review how the knowledge and beliefs of agents have been formally modelled (including in the situation calculus) in §2.3. Finally, in §2.4 we review the *revision* of beliefs, and again consider how that has been modelled in the situation calculus. The approach to modelling belief in terms of plausibility due to Shapiro et al. (2011) that is described in §2.4.2 will be the starting point for the later chapters of this thesis.

2.2 Formal theories of action and change

There have been a large number of formalisms for reasoning about action and change proposed. We give a brief overview of a sample of them in §2.2.1 before describing in detail in §2.2.2 the one that we will be using in this thesis, the *situation calculus*.

2.2.1 Overview

Here we consider some commonly used formalisms for reasoning about action and change.

Situation calculus The *situation calculus* (McCarthy, 1963; McCarthy and Hayes, 1969; Reiter, 2001), which we will discuss in much more detail in §2.2.2, is one of the oldest logical formalisms for describing action and change. The situation calculus describes change in the world with *fluents*, predicates that take a situation argument. A situation was originally conceptualized as "the complete state of affairs at some instant of time" (McCarthy, 1963), but in Reiter's version of the situation calculus (which will be used by this thesis), situations are histories of actions. Some of the historical development of the situation calculus is described by Lin (2008). There also are versions of the situation calculus using modal logic (Lakemeyer, 2010; Lakemeyer and Levesque, 2011), in which there are modal operators corresponding to actions, and situations are part of the semantics but not represented by terms in the language.

Fluent calculus The fluent calculus (Thielscher, 1998, 1999) is similar to Reiter's version of the situation calculus, but additionally has objects representing *states* of the world. The language includes a function mapping situations (sequences of actions) to states. The behavior of actions can be described with "state update axioms" that say how the states in consecutive situations differ, which is argued to have computational advantages compared to situation calculus action theories.

Event calculus The event calculus (Kowalski and Sergot, 1986) is an alternative to the situation calculus. Unlike the situation calculus, in the event calculus time is modelled in a linear way, i.e., there is a single timeline, instead of a branching tree with multiple possible futures (corresponding to different action choices).

Action languages Various propositional "action languages" like $\mathcal{A}$ have been proposed for describing transition systems (see Gelfond and Lifschitz, 1998). In $\mathcal{A}$, the behavior of actions is described using rules of the form A **causes** L **if** F where A is an action name,

L is a literal, and F is a conjunction of literals. There are extensions of $\mathcal{A}$ that allow for some other types of rules, e.g., to describe indirect effects of actions.

Temporal logics Another way of describing change over time is with a modal temporal logic, with operators for temporal relations (e.g., that something holds forever). There are various temporal logics, including linear temporal logic (LTL) (Pnueli, 1977) and computation tree logic (CTL) (Clarke et al., 1986), which model linear and branching time, respectively. The behavior of actions can be described using temporal logics, as described by, e.g., Calvanese et al. (2002).

We now turn to further exploring the situation calculus.

2.2.2 The situation calculus

We follow the version of the situation calculus proposed by Reiter (2001). The situation calculus is a language for describing actions and change, with semantics given by (multi-sorted) second-order logic. The sorts are situations, actions, and objects. For convenience, we let the natural numbers be a subsort of objects, and will suppose that arithmetic operations have the standard interpretation.[1]

2.2.2.1 Notation

We now describe some notational conventions. Each of the classes of symbols we describe below may also appear with decorations (e.g., subscripts). We will use s as a variable of type situation; a and b as variables of type action; i and j as numeric variables; and x, y, and z as variables for objects. Predicate symbols start with an uppercase letter, and function/constant symbols with a lowercase letter. We will use uppercase Roman letters like P and Q for second-order predicate variables (and sometimes as metalogical symbols for predicates).

We use lowercase Greek letters like ϕ and τ as metalogical symbols for formulas and terms, and uppercase Greek letters like Γ and Δ for sets of formulas. For a finite set of formulas Γ, their conjunction can be written as $\bigwedge \Gamma$. We may abbreviate a (possibly empty) sequence of terms $\tau_1, \ldots, \tau_k$ using vector notation as $\vec{\tau}$. A *ground* term does not refer to any variables.

Quantifiers We also adopt these conventions regarding quantifiers:

[1] This is just for convenience, since the natural numbers and arithmetic operations on them can be characterized with axioms in second-order logic (Zach, 2020, §7.7).

- Following Reiter (2001), we will sometimes use the notation of putting a dot after a quantifier, as in $\forall x.\ \phi$, to indicate that that quantifier has the widest possible scope.

- We will sometimes leave outer universal quantifiers on sentences implicit, e.g., using $\phi(x)$ to stand for $\forall x.\ \phi(x)$, though not for second-order quantifiers.

- We use $\forall\phi$ to denote the universal closure of a formula ϕ, i.e., the sentence $\forall\vec{x}.\ \phi$, where $\vec{x}$ is the sequence of all free variables in ϕ.

Model theory

We will typically use $\mathfrak{I}$ as a symbol for an *interpretation*, which is a pair $\langle\mathcal{D},\mathcal{I}\rangle$ where

- $\mathcal{D}$ is the domain (a set of entities, which can be partitioned into the different sorts – situations, actions, and objects), and

- $\mathcal{I}$ is the mapping which assigns predicate symbols to subsets of the domain and function symbols to functions on the domain.

(Note that this use of "domain" is distinct from how the word is sometimes used informally for an environment, e.g. a microworld in which blocks can be picked up.)

We will use μ for a variable assignment, which maps first-order variables to objects in the domain, second-order predicate variables to subsets of the domain, and second-order function variables to functions on the domain. We use $\mathfrak{I},\mu \models \phi$ to indicate that the formula ϕ (possibly including free variables) is satisfied by the interpretation $\mathfrak{I}$ and variable assignment μ. (Note that if ϕ has no free variables, its satisfaction does not depend on the variable assignment.) An interpretation is said to be a *model* of a sentence (or set of sentences) if it makes that sentence (every sentence in the set) true.

The $\models$ symbol is also used for *entailment*. A set of sentences Γ entails a sentence ϕ, written $\Gamma \models \phi$, if every model of Γ is also a model of ϕ. The notion of entailment is central to using the situation calculus, in which the entailments of action theories are of interest. (For example, an action theory might entail that a particular goal can be achieved by performing a certain sequence of actions.) This can be contrasted with other uses of logic, like in the problem of *model checking*, in which what's investigated is what a specific interpretation satisfies (see, e.g., Grohe, 2001). In this thesis, most of our uses of interpretations will be in service of proving things about entailments.

Sometimes, we will want to talk about objects in domains, for which we will use as metalogical symbols the same symbols we use for variables of the appropriate sort, but

decorated with a circumflex or "hat" (ˆ). For example, we may use $\hat{s}$ to stand for a situation object, or $\hat{a}$ to stand for an action object. Note that (unlike some other writers) we are not in general assuming any relationship exists between a particular logical variable like s and the hatted metalogical symbol $\hat{s}$. In particular, unless specifically mentioned we are *not* assuming that $\hat{s}$ is the situation denoted by s (with respect to a given variable assignment).

Where no confusion can arise, we may refer to terms of a given sort by the name of what they denote, e.g., we may call a situation term like S_0 a situation. At other times (in proofs involving interpretations) we will need to distinguish between situation terms and situation objects, and similarly for other sorts.

2.2.2.2 The language of the situation calculus

In the situation calculus, properties that can change (e.g., whether an object is being held) are modelled using *fluents*[2], predicates (or functions) whose last argument is a situation. For example, $\mathsf{Holding}(x, s)$ could represent the property of the agent holding x in situation s. We may informally express $\mathsf{Holding}(x, s)$ by saying that $\mathsf{Holding}(x)$ is true in s. We will assume that there are only finitely many fluent symbols.

Changes are brought about by actions. We'll assume that there are finitely many action function symbols, that is, symbols like pick and drop (where $\mathsf{pick}(x)$ is the action of picking up x, and $\mathsf{drop}(x)$ is the action of dropping x). In the situation calculus, situations represent histories of actions performed starting from an initial situation. Time is modelled as a branching structure: from a situation s, for any action a, $\mathsf{do}(a, s)$ is the future situation that results from performing a in s. We use the abbreviation $\mathsf{do}([a_1, \ldots, a_k], s)$ for $\mathsf{do}(a_k, \mathsf{do}([a_1, \ldots, a_{k-1}], s))$, i.e., for the successive application of actions $a_1, \ldots, a_k$ starting from s (note that $\mathsf{do}([\,], s)$ is just s itself).

The constant S_0 denotes the actual initial situation – the root of the situation tree (note that it is an exception to the convention that constants be lowercase). In some versions of the situation calculus, it is the only initial situation. Others additionally have alternative initial situations (so situations are organized as a forest rather than a single tree), and later chapters of this thesis will use such a version.

The special predicate $\mathsf{Poss}(a, s)$ is used to mean that the action a is possible to execute in situation s. Note that situations whose histories include actions that were not possible to execute still exist; Reiter (2001, p. 53) called them "ghost" situations. In contrast, situations in which all the actions performed were possible (at the time they

[2]McCarthy (1963) explained the choice of terminology by saying that "The term was used by Newton for a physical quantity that depends on time".

were executed) are called *legal* or *executable*.

The special binary predicate $s \sqsubset s'$ means that s' is the situation resulting from applying one or more actions in s. Note that $s \sqsubseteq s'$ can be defined as an abbreviation for $s \sqsubset s' \vee s = s'$. Legality can be defined using it:

Definition 2.2.1 (Legal)**.**

$$\mathsf{Legal}(s) \stackrel{\text{def}}{=} \forall a, s^*.\ (\mathsf{do}(a, s^*) \sqsubseteq s) \supset \mathsf{Poss}(a, s^*)$$

Some papers using the situation calculus have featured an ordering relation on situations that is like $\sqsubset$ but requires that the actions executed be possible. We can define that as an abbreviation:

Definition 2.2.2 ($s < s'$)**.**

$$s < s' \stackrel{\text{def}}{=} s \sqsubset s' \wedge \forall s^*, a.\ (s \sqsubset \mathsf{do}(a, s^*) \sqsubseteq s') \supset \mathsf{Poss}(a, s^*)$$

Some versions of the situation calculus include further special symbols, for example to represent the results of sensors (we will return to that when we discuss modelling beliefs in the situation calculus). We can use the abbreviation $\mathsf{Init}(s)$, defined below, to say that s is an initial situation.

Definition 2.2.3 (Init)**.**

$$\mathsf{Init}(s) \stackrel{\text{def}}{=} \neg\exists a, s'.\ s = \mathsf{do}(a, s')$$

Shapiro (2005) used a $\mathsf{root}(s)$ function, whose value was the initial situation preceding s, which we will also sometimes find useful.

Finally, we will sometimes make use of the special situation term "*now*". Intuitively it acts as a placeholder, to be syntactically substituted with another situation term that denotes the current situation (later we'll see how what's "current" is determined in different cases). Given a formula ϕ referring to *now*, we will write $\phi[s]$ for the formula that is like ϕ but substitutes s for *now*. Furthermore, sometimes we may follow the convention of writing formulas in which every situation argument is *now* in a "situation-suppressed" way by omitting the situation arguments, e.g., writing $F(\vec{x})$ for $F(\vec{x}, now)$. We will see *now* used within *ConGolog programs* (§2.2.2.5) and beliefs (§2.3.2).

2.2.2.3 Action theories

While we have informally described the meaning of elements of the language, like situation terms and the $\sqsubset$ symbol, it's important to remember that since we are just using standard second-order logic, we need *axioms* to give meaning to them. Furthermore, when dealing with any particular domain, e.g., the classic "blocks world" where an agent can pick up objects, we need axioms describing that environment so that we can see what is entailed (e.g., whether a particular sequence of actions will construct a tower). The standard way of axiomatizing domains in the situation calculus is by using some variation of *basic action theories* (Reiter, 2001).

Basic action theories

A basic action theory consists of the following sets of axioms:

- domain-independent foundational axioms, that describe the structure of the tree of situations (in basic action theories, there is only one initial situation);
- initial state axioms, which describe S_0;
- successor state axioms (SSAs), specifying for each fluent how its value in a non-initial situation depends on the previous situation;
- precondition axioms that describe when actions are possible to execute,
- and unique names axioms for actions.

We will describe each of these types of axioms in turn.

Foundational axioms The four standard foundational axioms, given by (Reiter, 2001, p. 50), are

$$\mathsf{do}(a_1, s_1) = \mathsf{do}(a_2, s_2) \supset [a_1 = a_2 \wedge s_1 = s_2] \tag{2.1}$$

$$\forall P.\ \big(P(\mathsf{S}_0) \wedge [\forall a, s.\ P(s) \supset P(\mathsf{do}(a, s))]\big) \supset \forall s.\ P(s) \tag{2.2}$$

$$\neg(s \sqsubset \mathsf{S}_0) \tag{2.3}$$

$$s \sqsubset \mathsf{do}(a, s') \ \equiv\ (s \sqsubset s' \vee s = s') \tag{2.4}$$

The first foundational axiom, Equation 2.1, specifies how any situations with different action histories are distinct. Equation 2.2 is a second-order axiom (note that P is a second-order variable, for a predicate that takes a situation argument) sometimes called

the "induction axiom", and is the only second-order axiom used in basic action theories. It says that there are no situations other than those that are the result of doing zero or more actions in S_0 (more literally, it says that any set P of situations that includes S_0 and its successors includes all situations). The last two foundational axioms just describe how the $\sqsubset$ relation works. It can be shown that in any model of the foundational axioms, the situation objects can be organized as the nodes in a tree, where the denotation of S_0 is the root, and the edges are actions (Reiter, 2001, p. 51).

To describe the other, domain-specific, components of action theories, we first introduce the notion of *uniform* formulas. Intuitively, a formula φ is *uniform* in a situation term σ if φ describes only the situation σ.

Definition 2.2.4 (uniform formula (Reiter, 2001, Definition 4.4.1)). A formula φ is uniform in a situation term σ if φ

- does not mention Poss or $\sqsubset$,
- does not quantify over situations,
- does not mention equality on situations,
- and σ is the last argument to any fluent mentioned by φ.

Other special predicates that we introduce later, like $\mathsf{B}(s', s)$ and $\mathsf{SF}(a, s)$, we will also not allow in uniform formulas.

Initial state axioms The initial state axioms are uniform in S_0. They describe the initial state of affairs, though not necessarily completely. For example, there might be a constant c such that the initial state axioms neither entail $\mathsf{Holding}(\mathsf{c}, \mathsf{S}_0)$ nor $\neg\mathsf{Holding}(\mathsf{c}, \mathsf{S}_0)$. Indeed, the set of initial state axioms can be empty.

Successor state axioms An SSA for a relational fluent F is a sentence of the form

$$F(\vec{x}, \mathsf{do}(a, s)) \equiv \phi_F(\vec{x}, a, s)$$

where ϕ_F is a formula uniform in s whose free variables are among $\vec{x}$, a, and s. The SSA describes how the value of F in a non-initial situation is determined by the action that just happened and the last situation. For example, the relational fluent $\mathsf{Holding}(x, s)$ might have the SSA

$$\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv a = \mathsf{pick}(x) \vee (a \neq \mathsf{drop}(x) \wedge \mathsf{Holding}(x, s)), \tag{2.5}$$

saying that x is held if it was just picked up or if it was already held and not just dropped. Similarly, an SSA for a functional fluent f is a sentence of the form

$$f(\vec{x}, \mathsf{do}(a, s)) = y \;\equiv\; \phi_f(\vec{x}, y, a, s)$$

where ϕ_f is a formula uniform in s whose free variables are among $\vec{x}, y, a$, and s. In some cases, a functional fluent's SSA can be written in a simplified form; e.g., if f never changes, then we could write $f(\vec{x}, \mathsf{do}(a, s)) = f(\vec{x}, s)$ instead of $f(\vec{x}, \mathsf{do}(a, s)) = y \;\equiv\; f(\vec{x}, s) = y$.

Precondition axioms A precondition axiom is a sentence of the form

$$\mathsf{Poss}(\alpha(\vec{x}), s) \;\equiv\; \phi_\alpha(\vec{x}, s)$$

where α is an action function symbol (i.e., $\alpha(\vec{x})$ is a term of type action) and $\phi_\alpha(\vec{x}, s)$ is a formula uniform in s whose free variables are among $\vec{x}$ and s. For example, to model the limited carrying capacity of a robot the axiomatizer might want for it only to be possible to pick up an object if nothing is currently being held:

$$\mathsf{Poss}(\mathsf{pick}(x), s) \;\equiv\; \forall y.\, \neg \mathsf{Holding}(y, s).$$

Unique names axioms for actions The set of these axioms includes, for any two distinct action function symbols α_1 and α_2,

$$\alpha_1(\vec{x}) \neq \alpha_2(\vec{y})$$

and, for any action function symbol α_1,

$$[\alpha_1(\vec{x}) = \alpha_1(\vec{y})] \supset [\vec{x} = \vec{y}].$$

The purpose of the unique names axioms is so that, for example, we can write an action theory using the SSA in Equation 2.5 without worrying about there being models of the theory where $\mathsf{pick}(x)$ and $\mathsf{drop}(x)$ denote the same action (note that in such models, objects that are held would always remain held.)

To wrap up, the formal definition of a basic action theory follows:

Definition 2.2.5 (basic action theory (BAT) (Reiter, 2001, Definition 4.4.5)). A *basic action theory (BAT)* is a set of axioms $\Sigma = \Sigma_{\mathsf{found}} \cup \Sigma_{\mathsf{ssa}} \cup \Sigma_{\mathsf{pre}} \cup \Sigma_0 \cup \Sigma_{\mathsf{una}}$ where

- Σ_{found} is the set of the four foundational axioms;

- Σ_{ssa} is the set of successor state axioms, one for each fluent;
- Σ_{pre} is the set of precondition axioms, one for each action function symbol;
- Σ_0 is the set of initial state axioms;
- and Σ_{una} is the set of unique names axioms for actions.

Furthermore, if there are functional fluents, Σ must obey the consistency property from (Reiter, 2001, p. 60).

Theories with multiple initial situations

Basic action theories have only one initial situation. Having multiple initial situations will be useful when formalizing belief (which we will discuss in §2.3.2), as they can represent ways the agent believes the world could be. Multiple initial situations require some changes to the foundational axioms, as described by Levesque et al. (1998, §7). The second-order induction axiom (Equation 2.1) has to be replaced by one that quantifies over all initial situations:

$$\forall P.\ \big([\forall s.\ \mathsf{Init}(s) \supset P(s)] \wedge [\forall a, s.\ P(s) \supset P(\mathsf{do}(a, s))]\big) \supset \forall s.\ P(s). \tag{2.6}$$

Furthermore, we need this new axiom (similar to Equation 2.3):

$$\mathsf{Init}(s') \supset \neg(s \sqsubset s'). \tag{2.7}$$

In any model of the revised foundational axioms, the situation objects are organized in a forest (i.e., a collection of trees, each rooted at a different initial situation).

Perhaps less obviously, we may also want a foundational axiom describing what initial situations exist. Levesque et al. (1998, §7) suggested having an initial situation for every possible combination of fluent values, and gave a second order axiom for that. A version is given below. Suppose that the relational fluents of the language are $F_1, \ldots, F_n$ and the functional fluents are $f_1, \ldots, f_m$. Then we have the following axiom, where $P_1, \ldots, P_n$ are second-order predicate variables, and $p_1, \ldots, p_m$ are second-order function variables.

$$\forall P_1, \ldots, P_n, p_1, \ldots, p_m\ \exists s.\ \mathsf{Init}(s) \wedge \left[\bigwedge_{i=1}^{n} \forall \vec{x}.\ F_i(\vec{x}, s) \equiv P_i(\vec{x})\right] \wedge \left[\bigwedge_{i=1}^{m} \forall \vec{x}.\ f_i(\vec{x}, s) = p_i(\vec{x})\right] \tag{2.8}$$

Lakemeyer and Levesque (1998) suggested a foundational axiom requiring the existence of even more initial situations, where actions behave differently. Some authors, like Shapiro

(2005), allow for initial state axioms to describe situations other than S_0.

The $\mathsf{root}(s)$ function that we mentioned previously can be useful when dealing with multiple initial situations. To define that we would need another foundational axiom, for example,

$$(\mathsf{root}(s) = s^*) \equiv \mathsf{Init}(s^*) \wedge s^* \sqsubseteq s. \tag{2.9}$$

Shapiro (2005, Axiom 2.2.6) gives an alternative (recursive) axiom for root.

2.2.2.4 Calculating entailments of action theories

The situation calculus is a language in second-order logic – and most parts of action theories are first-order – so general-purpose theorem-proving techniques (for example, those found in the first-order theorem prover Vampire (Kovács and Voronkov, 2013)) can in principle be applied to reason about the logical consequences of action theories. However, researchers investigating the situation calculus have apparently not viewed such as being practical enough, though this is rarely explicitly stated (but see Brachman and Levesque, 2004, pp. 310–311).

Before we discuss situation-calculus-specific reasoning mechanisms, let us first note that there are some metalogical results showing that for a broad class of sentences, whether a BAT entails a member of this class can be determined without using the second-order induction axiom. Consider the following definition:

Definition 2.2.6 ($\exists s$ sentence (Pirri and Reiter, 1999, Definition 5.2)). A sentence ϕ is said to be an $\exists s$ sentence iff it has a *prenex normal form* with no universal quantifiers over situations, i.e., it can be equivalently written as

$$\vec{\xi_i}(\exists s_1)\vec{\xi_2}(\exists s_2)\cdots(\exists s_k)\vec{\xi_k}\psi$$

for some $k \geq 0$, where each $\vec{\xi_i}$ is a sequence of zero or more quantifiers that are not over situations, and ψ contains no quantifiers.

We should note that any first-order formula can be rewritten into an equivalent formula in prenex normal form, through some rewriting rules that move quantifiers around (Enderton, 2001, p. 160). The significance of $\exists s$ sentences is their role in the following proposition, in which we refer to the second-order induction axiom (Equation 2.2) as "induction".

Proposition 2.2.1 (Pirri and Reiter, 1999, Theorem 4(1) and 4(3)). Suppose that Σ is a BAT and ϕ is a first-order $\exists s$ sentence. Then

- $\Sigma \models \phi$ if and only if $\Sigma \setminus \{\text{induction}\} \models \phi$.
- If ϕ does not mention the symbol $\sqsubset$ nor compare situations for equality, then $\Sigma \models \phi$ if and only if $\Sigma_{\mathsf{ssa}} \cup \Sigma_{\mathsf{pre}} \cup \Sigma_0 \cup \Sigma_{\mathsf{una}} \models \phi$.

So we see that for $\exists s$ sentences, no second-order reasoning is needed to determine if they are entailed by a BAT.

As previously mentioned, there has been work on situation-calculus-specific reasoning mechanisms (e.g., Kelly and Pearce, 2010; Yehia et al., 2012; Ewin et al., 2015). One commonly considered reasoning task is the *projection problem*, the problem of determining whether a formula describing the situation resulting from performing some actions in S_0 is entailed by an action theory (Reiter, 2001, §4.6.2). One well-known technique for projection is *progression* (Lin and Reiter, 1997; Vassos and Levesque, 2013), which involves iteratively updating the part of an action theory describing S_0 to instead describe the resulting situation after performing an action. Perhaps the most popular technique for projection in the situation calculus is *regression*, and we devote the rest of this section to it (we will be using regression in Chapter 4).

Regression is a formula-rewriting procedure that can in some cases simplify theorem-proving. Certain formulas, called *regressable* formulas, can be rewritten into formulas that do not refer to any situations other than S_0, which may make them easier to prove.

Definition 2.2.7 (regressable (Reiter, 2001, Definition 4.5.1)). A first-order formula ϕ is regressable if all of the following hold:

1. for each term of sort situation mentioned by ϕ, the term has the syntactic form $\mathsf{do}(\vec{\alpha}, \mathsf{S}_0)$
2. for each atom of the form $\mathsf{Poss}(\alpha, \sigma)$ mentioned by ϕ, α has the syntactic form $\alpha'(\vec{t})$ for some action function symbol α'
3. ϕ does not quantify over situations
4. ϕ does not refer to $\sqsubset$, nor compare situations for equality

A slightly broader definition of regressable was used by Pirri and Reiter (1999), but this suffices for our purposes. One easy observation to make (which oddly does not seem to have been explicitly stated in the literature) is the following:

Observation 2.2.1 (the regressable sentences are a subset of the $\exists s$ sentences). Since regressable sentences do not contain situation variables, they have prenex normal forms which do not have quantified situation variables at all, and so are $\exists s$ sentences.

It follows that Proposition 2.2.1 applies to regressable sentences, yielding the following corollary.

Corollary 2.2.1. Suppose Σ is a BAT and ϕ is a regressable sentence. Then

$$\Sigma \models \phi \text{ if and only if } \Sigma \setminus \{\text{induction}\} \models \phi.$$

Furthermore, since the definition of "regressable" that we're using does not allow references to $\sqsubset$ or comparing situations for equality,

$$\Sigma \models \phi \text{ if and only if } \Sigma_{\mathsf{ssa}} \cup \Sigma_{\mathsf{pre}} \cup \Sigma_0 \cup \Sigma_{\mathsf{una}} \models \phi.$$

So the regressable sentences are a class that are simpler to prove, in the sense that not all axioms from a BAT are needed to entail them. Pirri and Reiter (1999, Definition 4.3) defined a regression operator $\mathcal{R}[\phi]$ which rewrote a regressable formula ϕ into one that was uniform in S_0. The main component of regression rewriting is repeatedly replacing subformulas of the form $F(\vec{\tau}, \mathsf{do}(\alpha, \sigma))$, where F is a fluent, with $\phi_F(\vec{\tau}, \alpha, \sigma)$, where ϕ_F is from the RHS of the SSA for F. (Functional fluents are handled in a similar but more complex way.) This ultimately removes all references to situations other than S_0.

The central theoretical result about regression is the following proposition, which has been called the "regression theorem":

Proposition 2.2.2 (Pirri and Reiter, 1999, Theorem 3(2)). Let Σ be a BAT and ϕ a regressable sentence. Then $\Sigma \models \phi$ iff $\Sigma_0 \cup \Sigma_{\mathsf{una}} \models \mathcal{R}[\phi]$.

Intuitively, the regression procedure exhausts the extent to which SSAs and precondition axioms are needed for proving the entailment of a regressable sentence. (Of course, in the general case the remaining entailment problem is still undecidable.) Regression "forms the basis for many planning procedures and for automated reasoning in the situation calculus" (Reiter, 2001, p. 61).

Remark 2.2.1. Regression is commonly discussed in the literature in a way which could be read as suggesting (incorrectly) that the significance of regression is in removing the need for the second-order induction axiom. For example, Fritz (2009, p. 15) claims that "Although any situation calculus action theory is second-order, many reasoning tasks can be reduced to first-order theorem proving by using regression". Even Reiter (2001, p. 66) writes that the regression theorem

> [...] reduces the evaluation of regressable sentences to a first-order theorem-proving task *in the initial theory* [...] together with unique names axioms for

> actions. [...] In particular, none of the foundational axioms [...] are required, and this means especially that the second-order induction axiom is not required.

That statement leaves out the fact that the induction axiom would not have been required to prove the original regressable sentence either, as was pointed out in Corollary 2.2.1. The work done to eliminate the need for induction is done by the definition of *regressable*, not the procedure of *regression*.

2.2.2.5 ConGolog

In the situation calculus as we've described it so far, all the actions are treated as being primitives, as opposed to being composed of other actions. There is not in general, for example, an action corresponding to the execution of two other actions in sequence. To describe complex arrangements of actions, a programming language can be used. ConGolog is a programming language, designed for use with the situation calculus, introduced by De Giacomo et al. (2000). It extends the original Golog (Levesque et al., 1997) with support for concurrent processes. We will be using ConGolog in Chapter 5.

In ConGolog, programs are represented as another type of object (in addition to situations, actions, and objects), which allows them to be quantified over (this is used in axiomatizing how they behave). ConGolog programs can refer to (encodings of) formulas and terms. These expressions make use of the situation term "*now*" (that we introduced on page 16) to refer to the current situation. To illustrate, a program can include a conditional statement

$$\textbf{if } \phi \textbf{ then } \delta_1 \textbf{ else } \delta_2 \textbf{ endIf}$$

which, if executed in a situation s, will result in the program δ_1 being executed if $\phi[s]$ is true, and the program δ_2 being executed otherwise.

Below we list some of the constructs of ConGolog, and what executions they produce. Note that any execution ultimately consists of a (possibly empty) sequence of primitive actions being performed. In these constructions, ϕ corresponds to a formula, α to an action term, and δ_1 and δ_2 are arbitrary ConGolog programs.

nil	nothing happens
α	the primitive action $\alpha[s]$ gets executed, where s is the current situation
$\phi?$	the process blocks until reaching a situation s where $\phi[s]$ is true
$\delta_1;\ \delta_2$	δ_1 and δ_2 are executed in sequence
$\delta_1 \mid \delta_2$	either δ_1 or δ_2 is executed (non-deterministically)

$\pi x.\ \delta(x)$	$\delta(x)$ is executed with non-deterministic choice of x
δ^*	δ is executed 0 or more times (non-deterministically)
if ϕ **then** δ_1 **else** δ_2 **endIf**	conditional branching
while ϕ **do** δ **endWhile**	while loop
$\delta_1 \parallel \delta_2$	concurrent execution of δ_1 and δ_2
$\delta_1 \rangle\!\rangle \delta_2$	concurrent execution, with higher priority for δ_1

(ConGolog also includes procedures, but for the examples we'll see in this thesis it will suffice to treat procedures as abbreviations.)

Concurrency is just the interleaving of steps from each of the involved processes, and in prioritized concurrency, the higher priority process takes a step whenever there is a next step it can take in the current situation. If one concurrent process reaches a $\phi?$ instruction in a situation where $\phi[s]$ is not true, then that process is blocked – there is no step it can take. A process is also blocked if the next primitive action it would execute is not possible to execute in the current situation.

The semantics of ConGolog are given with two predicates, $\mathsf{Trans}(\delta, s, \delta', s')$ and $\mathsf{Final}(\delta, s)$. The first of these, $\mathsf{Trans}(\delta, s, \delta', s')$, says that it's possible to take a step in executing δ from situation s and end up in situation s', with the part of the program remaining to be executed being δ'. $\mathsf{Final}(\delta, s)$ says that δ can legally terminate in situation s. These predicates are characterized using axioms like the following:

$$\begin{aligned}
\mathsf{Trans}(\mathsf{nil}, s, \delta', s') &\equiv \mathsf{False} \\
\mathsf{Trans}(\alpha, s, \delta', s') &\equiv \mathsf{Poss}(\alpha[s]) \wedge (\delta' = \mathsf{nil}) \wedge (s' = \mathsf{do}(\alpha[s], s)) \\
\mathsf{Trans}([\delta_1; \delta_2], s, \delta', s') &\equiv (\mathsf{Final}(\delta_1, s) \wedge \mathsf{Trans}(\delta_2, s, \delta', s')) \vee \\
&\quad \exists \delta''.\ \mathsf{Trans}(\delta_1, s, \delta'', s') \wedge \delta' = [\delta''; \delta_2] \\
\mathsf{Trans}((\delta_1 \rangle\!\rangle \delta_2), s, \delta', s') &\equiv \exists \delta_1'.\ \mathsf{Trans}(\delta_1, s, \delta_1', s') \wedge \delta' = (\delta_1' \rangle\!\rangle \delta_2) \vee \\
&\quad \exists \delta_2'.\ \mathsf{Trans}(\delta_2, s, \delta_2', s') \wedge \delta' = (\delta_1 \rangle\!\rangle \delta_2') \wedge \neg \exists \delta_1'.\ \mathsf{Trans}(\delta_1, s, \delta_1', s')
\end{aligned}$$

See the original paper (De Giacomo et al., 2000) for the complete list. Also, the representation of programs as terms requires many axioms (and several other sort of objects); again, see (De Giacomo et al., 2000) for details.

Finally, a predicate Trans^* is defined as the reflexive transitive closure of Trans. So $\mathsf{Trans}^*(\delta, s, \delta', s')$ means that situation s' can be reached by following 0 or more steps of the program δ from s, with the program δ' left over to still run.

Note that different programs may be equivalent in the sense of having the same

possible transitions. For example, it's a consequence of the ConGolog axioms that

$$\mathsf{Trans}^*(\textbf{if } \mathsf{True} \textbf{ then } \delta \textbf{ endIf}, s, \delta', s') \equiv \mathsf{Trans}^*(\delta, s, \delta', s').$$

This allows us to simplify programs in some cases, which we make use of in Chapter 5.

Finally, aside from the constructs described above, we allow our language to contain further terms that also denote programs, e.g., bobsFavoriteProgram might be a constant for Bob's favorite program. To illustrate, that might be used in an expression like the following, which is a statement of equality between two terms of the program sort (for this example, suppose that $\mathsf{OwnedByBob}(x, s)$ is a fluent and $\mathsf{transfer2Bob}(x)$ is an action).

$$\mathsf{bobsFavoriteProgram} = \textbf{while } \exists x.\ \neg\mathsf{OwnedByBob}(x) \textbf{ do } \pi x.\ \mathsf{transfer2Bob}(x) \textbf{ endWhile}$$

We will call terms like the one on the RHS of this equality (but not the one on the LHS) *literal program terms*.

Definition 2.2.8 (literal program term). We will say that a program term is a *literal program term* if its syntactic form is built up entirely from the ConGolog constructs (primitive actions, sequences, if-then-else, loops, etc.).

2.3 Formal models of knowledge and belief

We first consider how knowledge and belief have been formally modelled in general, before going into detail on a way to model them in the situation calculus.

2.3.1 Overview

A standard way of representing knowledge and beliefs using logic has been the possible worlds model, where what is believed is defined to be what's true in a set of "accessible" possible worlds (Hintikka, 1962). This is typically expressed using epistemic *modal* logics, which borrow their semantics from modal logics of necessity (Kripke, 1963).

Modal logic extends propositional (or higher order) logic using *modal operators*, which act on sentences. For example, logics of necessity typically include a $\Box$ operator for necessity, so that $\Box\phi$ means that it's necessarily the case that ϕ. In epistemic logics, there typically is a $\mathbf{K}$ operator for knowledge (or $\mathbf{B}$ for belief), so that $\mathbf{K}\phi$ means that ϕ is known.

Semantics for propositional modal logic can be given by a Kripke structure $\mathfrak{M}$, which includes a set W (whose elements are called "worlds"), a valuation function v mapping

each $w \in W$ to a truth assignment, and an accessibility relation $R \subseteq W \times W$. A sentence that doesn't refer to knowledge has its truth in w determined by the assignment $v(w)$. The truth conditions of knowledge at a world w (given $\mathfrak{M}$) are defined by

$$\mathfrak{M}, w \models \mathbf{K}\phi \qquad \text{if} \qquad \mathfrak{M}, w' \models \phi \text{ for every } w' \text{ such that } R(w, w')$$

In the field of epistemology, a major topic is how knowledge should be defined (for example, as *justified true belief*) (Ichikawa and Steup, 2018). However, many works in logic do not require more properties of knowledge (compared to belief) other than that what is known must be true (Fagin et al., 1995). Note that if the accessibility relation R is reflexive, i.e., every possible world is accessible from itself, then what is known/believed will necessarily be true. Furthermore, AI researchers often do not distinguish between knowledge and belief. Segerberg (1999, footnote 2) said that "The distinction between knowledge and belief is difficult to draw, and more often than not today's modal logicians, especially in the computer science camp, seem uninterested in trying to draw one."

It's common for logical accounts of knowledge and belief to feature introspection, which comes in two varieties:

Positive introspection If φ is known, then it's known that φ is known.

Negative introspection If φ is not known, then it's known that φ is not known.

The properties correspond to simple mathematical conditions on the accessibility relation. Positive introspection arises from the relation being transitive, and negative introspection from the relation being Euclidean.[3] These properties, in particular negative introspection for knowledge, are somewhat controversial (see, e.g., Halpern et al., 2009).

Another matter that has raised some philosophical controversy is how quantification should interact with beliefs, in particular the subject of "quantifying-in" (Quine, 1956; Kaplan, 1968). Quantifying into knowledge allows for a way of distinguishing between *de dicto* and *de re* knowledge. This distinction is exemplified with the difference between $\mathbf{K}(\exists x\ \mathsf{Spy}(x))$ (the agent knows that there is a spy) and $\exists x\ \mathbf{K}(\mathsf{Spy}(x))$ (the agent knows who some spy is). See (Hobbs, 1985, §4) for a discussion of whether this really captures the meaning of "knowing who".

Using quantification in modal logic also presents the choice of whether the domain of quantification should be the same in all worlds. To illustrate a consequence of that, consider the following formula schemas, versions of which were first considered by Barcan (1946), after whom they are named.

[3] A binary relation R is Euclidean if, whenever $R(x, y)$ and $R(x, z)$, then $R(y, z)$.

Barcan formula: If for every x, $\phi(x)$ is known, then $\forall x\ \phi(x)$ is known, i.e.,

$$(\forall x\ \mathbf{K}\phi(x)) \supset \mathbf{K}(\forall x\ \phi(x)).$$

converse Barcan formula: If $\forall x\ \phi(x)$ is known, then for every x, $\phi(x)$ is known, i.e.,

$$\mathbf{K}(\forall x\ \phi(x)) \supset \forall x\ \mathbf{K}\phi(x).$$

All instances of the Barcan formula and converse Barcan formula are satisfied in a structure where the domain of quantification is the same in all worlds, but they may not be true if the domain varies. See (Fitting, 1999) for further discussion.

It's straight-forward to model the knowledge of multiple agents, by having multiple accessibility relations. In a multi-agent settings, *common knowledge* can also be considered. A proposition ϕ is common knowledge to a group if everyone in the group knows ϕ, knows that everyone in the group knows ϕ, knows that everyone in the group knows that everyone in the group knows ϕ, and so on (see, e.g., Fagin et al., 1995).

The possible worlds account has disadvantages. According to it, an agent will know all the logical consequences of its knowledge (for example, if φ is true at all accessible worlds and so is $\varphi \supset \psi$, then so will be ψ). This has been called the "problem of logical omniscience" (Stalnaker, 1991). It means that we can't accurately represent the limited knowledge of people, or of physically realizable artificial agents – at least if we expect the agents to be able to compute all their knowledge. There have been a variety of more restricted forms of knowledge and belief suggested (e.g., Hintikka, 1975; Levesque, 1984; Fagin and Halpern, 1988; Elgot-Drapkin and Perlis, 1990; Halpern et al., 1994; Liu et al., 2004; Lakemeyer and Levesque, 2014; Klassen et al., 2015; Klassen, 2015; Lakemeyer and Levesque, 2019; Solaki et al., 2019).

Finally, note that so far we've just been talking about knowing (or believing) *that* something is true, or knowing the identity of an object. Another form of knowledge is knowing *how* to do things. We will discuss that in Chapter 5.

2.3.2 In the situation calculus

Much as traditional modal logics of knowledge use an accessibility relation over possible worlds, in the situation calculus knowledge can be defined in terms of an accessibility relation over situations, as was shown by Moore (1980). Here we describe the approach to that that comes from Scherl and Levesque (2003). They used action theories with multiple initial situations (to serve as epistemic alternatives), and a predicate which we

will call $\mathsf{B}(s', s)$ to mean that s' is accessible from s. Note that the order of the arguments to B was chosen to be consistent with fluents, and is the opposite from how accessibility relations in modal logic are typically described.

A knowledge operator $\mathbf{Know}(\phi, s)$ ("ϕ is known in s") is defined as an abbreviation,

$$\mathbf{Know}(\phi, s) \stackrel{\text{def}}{=} \forall s'.\ \mathsf{B}(s', s) \supset \phi[s'], \tag{2.10}$$

where $\phi[s]$ stands for the formula that is like ϕ but substitutes s for the special "indexical" situation term now (as we previously discussed with respect to ConGolog). (We assume the variable introduced by the expansion of **Know**, here written as s', does not appear as a free variable in ϕ). Again, we may suppress now arguments, e.g., writing $\mathbf{Know}(F(x, now), s)$ as $\mathbf{Know}(F(x), s)$. Note that unlike with ConGolog programs, there is no encoding of formulas as terms involved in defining knowledge.

While Scherl and Levesque required what was known to be true, we will sometimes use **Know** in cases where B isn't reflexive. Note also that (of course) not all true things have to be known, since there may be accessible situations where those things are false.

Remark 2.3.1. To illustrate the importance of using now within the **Know** operator, note for example that for any fluent F, it follows from the definition of **Know** that

$$\models \forall \vec{x}.\ F(\vec{x}, \mathsf{S}_0) \supset \mathbf{Know}(F(\vec{x}, \mathsf{S}_0), \mathsf{S}_0).$$

It's only by having the known formula depend on now that we can get interesting knowledge (the agent in S_0 doesn't have to know whether $F(\vec{x}, now)$ is true, because they don't know that they're in S_0). So fluents now serve a second purpose: we may want to describe properties like F as fluents (i.e., taking a situation argument) even if they can't change, just so that the agent can fail to know whether they're true.

Remark 2.3.2. The domain of objects does not depend on the situation. A consequence of that is that Scherl and Levesque's account results in the Barcan and converse Barcan formulas holding in all situations.

To allow the agent to learn about its environment, Scherl and Levesque allowed actions to provide sensing information. To represent this, a predicate $\mathsf{SF}(a, s)$ can be used.[4] Intuitively, executing an action a in s produces a binary sensing result, and $\mathsf{SF}(a, s)$ is true iff that result is positive. The SF predicate can be described in an action theory with

[4]The SF predicate was introduced by Levesque (1996). Scherl and Levesque used a function instead of a predicate and so did not restrict sensing results to be binary.

an additional set of axioms (beyond those from Definition 2.2.5), *sensing axioms*, which are sentences of the form

$$\mathsf{SF}(\alpha(\vec{x}), s) \equiv \phi_\alpha(\vec{x}, s)$$

where α is an action function symbol and $\phi_\alpha(\vec{x}, s)$ is a formula uniform in s whose free variables are among $\vec{x}$ and s. Note that a sensing axiom is like a precondition axiom except for referring on the LHS to the SF predicate instead of Poss.

To illustrate, the sensing axiom

$$\mathsf{SF}(\mathsf{sense}, s) \;\equiv\; \exists x.\, \mathsf{Holding}(x, s)$$

says that the sense action senses whether anything is currently being held. If for a particular action function symbol α we don't want it to provide any sensing information, we can just set the sensing result to always be True, i.e., write $\mathsf{SF}(\alpha(\vec{x}), s) \equiv \mathsf{True}$. We will sometimes refer to actions which do provide sensing information as *sensing actions*.

Scherl and Levesque gave this SSA-like axiom for B:

$$\begin{aligned}\mathsf{B}(s'', \mathsf{do}(a, s)) \;\equiv\; \big[\exists s'.\, &\mathsf{B}(s', s) \wedge (s'' = \mathsf{do}(a, s')) \wedge \\ &\mathsf{Poss}(a, s') \wedge (\mathsf{SF}(a, s') \equiv \mathsf{SF}(a, s))\big]\end{aligned} \tag{2.11}$$

That is, for a situation to be accessible after performing an action a, that situation must be the result of doing a in some other situation that was previously accessible, a must have been possible to execute, and the sensing result of a must reflect the true value. We will call Equation 2.11 the SSA for B even though, strictly speaking, it doesn't match the definition of an SSA since the RHS is not a uniform formula.

Note that this SSA means that the accessibility relation will be such that the agent always knows exactly which actions have occurred (assuming that only initial situations are initially accessible). More complicated accessibility relations which don't require that have also been proposed (see e.g., Shapiro and Pagnucco, 2004; Kelly and Pearce, 2015), and will be considered later in this thesis.

Scherl and Levesque (2003, Theorem 6) showed that if any of various restrictions – reflexiveness, Euclideanness, symmetry, or transitivity – is imposed on the initial accessibility relation (i.e., on what situations are accessible from initial situations), that restriction will continue to hold after any number of possible actions (i.e., possible as specified using Poss). Furthermore, Scherl and Levesque showed how the procedure of regression (§2.2.2.4) can be extended to also regress formulas using the **Know** abbrevia-

tion.

In the approach of Scherl and Levesque (2003), the SSAs, preconditions axioms, and sensing axioms in the theory apply to all situations, so the agent always knows them. An alternative approach, suggested by Lakemeyer and Levesque (1998) (and also followed by other papers like (Schwering and Lakemeyer, 2014, 2015)), allows actions to behave differently in different situations. The following definition is useful for describing their work:

Definition 2.3.1 (relativized axiom). Let $\phi(s)$ be such that $\forall s.\ \phi(s)$ is an SSA, precondition axiom, or sensing axiom. Then the corresponding *axiom relativized to* σ, where σ is situation term, is the formula $\forall s.\ (\sigma \sqsubseteq s) \supset \phi(s)$.

Intuitively, relativized axioms only constrain the behaviors of actions on the (sub)tree rooted at σ. Let us also introduce some notation, which we will find use for later:

Definition 2.3.2 (Γ:σ). Let Γ be a set of SSAs, precondition axioms, and/or sensing axioms. Given a situation term σ, Γ relativized to σ, written Γ:σ, is the set of corresponding axioms relativized to σ.

For example, if Σ_{ssa} is a set of SSAs, then Σ_{ssa}:S_0 is the set of corresponding relativized axioms that only constrain the behavior of actions on the tree rooted at S_0. Lakemeyer and Levesque suggested including SSAs, precondition axioms, and sensing axioms relativized to S_0 in the action theory. Furthermore, they suggested having the agent believe (potentially different) sets of SSAs, precondition axioms, and sensing axioms relativized to *now*. This allowed for incorrect beliefs about dynamics to be represented in a simple way. They also had a more complicated axiom (their Axiom F8) for describing what initial situations exist, so as to have initial situations from which actions behave in arbitrary ways (so that the agent could consider those ways possible).

2.4 Belief revision

We have already considered change in knowledge, as Scherl and Levesque (2003) allowed for an agent to gain information by sensing. However, in their approach there was no way for the agent to retract conclusions. Since Scherl and Levesque assumed knowledge was true, there didn't need to be. However, when an agent has false beliefs, then it's desirable to be able to correct them. How beliefs should be revised is the topic of the field of belief revision. A survey of the field was made by Peppas (2008). We will give a brief overview, before describing how belief revision has been modelled in the situation calculus.

2.4.1 Overview

Much of the traditional work in belief revision does not actually use epistemic logic, but rather implicitly represents the beliefs of an agent as a set K of propositional formulas, closed under logical consequence. The question is how that set of formulas should be modified to incorporate new information (possibly inconsistent with what was originally in the set). That is, what constraints should there be on $K * \phi$, the revision of K by a (propositional) formula ϕ?

Alchourrón, Gärdenfors, and Makinson (1985) proposed a set of postulates that a rational belief revision operator $*$ should follow. These have been called the AGM postulates, after the initials of the authors. The postulates have not been universally accepted, but are very influential. We list them below (with names from (van Ditmarsch, 2005)). Note that $K + \phi$, called the *expansion* of K by ϕ, is just the closure under logical consequence of $K \cup \{\phi\}$.

(AGM$*$1)	$K * \phi$ is deductively closed	type
(AGM$*$2)	$\phi \in K * \phi$	success
(AGM$*$3)	$K * \phi \subseteq K + \phi$	upper bound
(AGM$*$4)	If $\neg\phi \notin K$, then $K + \phi \subseteq K * \phi$	lower bound
(AGM$*$5)	$K * \phi$ is inconsistent iff $\models \neg\phi$	triviality
(AGM$*$6)	If $\models \phi \equiv \psi$, then $K * \phi = K * \psi$	extensionality
(AGM$*$7)	$K * (\phi \wedge \psi) \subseteq (K * \phi) + \psi$	iteration upper bound
(AGM$*$8)	If $\neg\psi \notin K * \phi$, then $(K * \phi) + \psi \subseteq K * (\phi \wedge \psi)$.	iteration lower bound

The first postulate just ensures that $K * \phi$ has the right type, that of a deductively closed theory (like K). Postulate (AGM$*$2) says that revision is successful, in that the formula revised by is believed. Postulates (AGM$*$3) and (AGM$*$4) relate revising by ϕ to expanding by ϕ. The triviality postulate requires the agent to incorporate the new information in a consistent way, if there's any way to do so. The extensionality postulate, (AGM$*$6), says that the results of revising by equivalent formulas should be the same. The last two postulates relate revising by a conjunction $\phi \wedge \psi$ to first revising by ϕ and then expanding by ψ.

Note that the postulates are not sufficient to specify a unique revision function $*$. Various revision functions have been proposed in the literature. As Peppas and Williams (2018) note, many do not satisfy all the postulates.

Grove (1988) showed that any AGM revision operator corresponds to a "system of spheres", essentially an ordering on worlds (technically, a preorder, since distinct worlds can be equally ranked). After revision by φ, beliefs are determined by the best worlds (according to the ordering) in which φ is true. We can think of the ordering as representing plausibility to the agent.

The AGM postulates are intended to describe changes of belief resulting from gaining information in a setting where the world itself does not change. A different set of postulates, the KM postulates, have been proposed to describe belief change in cases where the world changes (Katsuno and Mendelzon, 1991). Those are called cases of *belief update* rather than *belief revision*.

Belief revision has also been considered in modal logics (e.g., Segerberg, 1995; van Ditmarsch, 2005). One relevant work to this thesis is that of Friedman and Halpern (1999a). They considered belief change over time in a modal temporal logic, and modelled both revision and update by having a prior plausibility measure on worlds (a generalization of a system of spheres), and *conditioning* that on observations. Their framework is very general, but under some conditions, conditioning basically just involves discarding worlds that are inconsistent with observations.

Other approaches to iterated (i.e., repeated) belief revision involve changing the plausibility ordering when a revision is made. For example, revising by ϕ could correspond to making all worlds in which ϕ is true more plausible than any world in which ϕ is false. A large number of ways of modifying the plausibility ordering for belief revision are catalogued by Rott (2009). Axioms for iterated revision have also been proposed (e.g., Darwiche and Pearl, 1997).

Belief revision has also been considered within the fluent calculus (Jin and Thielscher, 2004), event calculus (Tsampanaki et al., 2019), and the situation calculus. We will now consider the situation calculus in more detail.

2.4.2 In the situation calculus

In this section we describe the approach of (Shapiro, 2005; Shapiro et al., 2011) to iterated belief revision (and update) within the situation calculus.

The approach builds on the work of Scherl and Levesque and uses the B and SF predicates we previously described in §2.3.2. In order to allow for beliefs to be retracted (which Scherl and Levesque did not), Shapiro et al. defined belief as truth in the *most plausible* accessible situations rather than in all accessible situations. With this approach, sensing can cause an agent to lose a belief by making inaccessible all the situations that

were previously the most plausible accessible ones. Sensing still worked the same way as in (Scherl and Levesque, 2003); situations incompatible with sensing results became inaccessible. Therefore the approach to belief revision is similar to Friedman and Halpern's (1999a). (Schwering et al. (2017) gave an approach to belief revision in the situation calculus where the plausibility ordering was modified instead.)

Shapiro et al. used a function pl to assign plausibility levels (natural numbers) to situations, where lower numbers indicate higher plausibility. Their SSA for pl specifies that the function never changes:

$$\mathsf{pl}(\mathsf{do}(a, s)) = \mathsf{pl}(s).$$

Belief was defined in terms of plausibility and accessibility. We'll find the following abbreviation convenient:

Definition 2.4.1. $s \leq_{\mathsf{pl}} s' \stackrel{\text{def}}{=} \mathsf{pl}(s) \leq \mathsf{pl}(s')$

That is, $s \leq_{\mathsf{pl}} s'$ if s is at least as plausible as s' (note the order). Shapiro et al. (2011) defined $\mathsf{MPB}(s', s)$ to mean that that s' is one of the most plausible situations accessible from s.

Definition 2.4.2 (MPB)**.**

$$\mathsf{MPB}(s', s) \stackrel{\text{def}}{=} \mathsf{B}(s', s) \wedge \forall s''.\, \mathsf{B}(s'', s) \supset s' \leq_{\mathsf{pl}} s''$$

They used MPB in defining a belief operator $\mathbf{Bel}$:

$$\mathbf{Bel}(\phi, s) \stackrel{\text{def}}{=} \forall s'.\, \mathsf{MPB}(s', s) \supset \phi[s']$$

So $\mathbf{Bel}(\phi, s)$ is true if ϕ is true in the most plausible accessible situations from s. This can be contrasted with $\mathbf{Know}(\phi, s)$ from Scherl and Levesque, which was defined to be true if ϕ is true in *all* the situations accessible from s. Note that belief is still closed under logical consequence, since it's still defined in terms of what's true in a set of situations.

They showed that their approach mostly satisfies the AGM postulates for belief revision. It also satisfies some of the KM postulates for belief update, and some of the DP postulates for iterated belief revision (Darwiche and Pearl, 1997).

Here, we will explain how Shapiro et al. related their approach to the AGM postulates, as this will be relevant in Chapter 3. In their approach, revisions are brought about by certain actions (since all change is the result of actions in the situation calculus). Now, the first thing that we have to do is define a language for the beliefs that the postulates

will apply to. The AGM postulates would not be expected to apply to beliefs about the past, for example. To see why, note that (AGM∗3) and (AGM∗4) require that revising by a sentence that is already believed should produce no change in beliefs at all. However, performing any action (including a revision action) will cause the agent to believe that that action has been performed.

Definition 2.4.3 ($\mathcal{L}_{\mathsf{now}}$). The language $\mathcal{L}_{\mathsf{now}}$ is the set of sentences uniform in *now* that do not refer to any functional fluents.

Recall that uniform formulas can't refer to the B predicate or quantify over situations, so $\mathcal{L}_{\mathsf{now}}$ cannot refer to beliefs. Shapiro (2005, p. 72) assumed that $\mathcal{L}_{\mathsf{now}}$ was propositional and finite, but only needed that for proving one of the KM postulates (Katsuno and Mendelzon, 1991), which we aren't concerned with here.

Now that we have a language, we can define belief states and expansions. Note that, following Shapiro, the definitions are made in terms of a given model $\mathfrak{I}$ of the action theory Σ. The reason for this is that Σ itself may not provide enough information to determine exactly what the agent believes in a given situation.

Definition 2.4.4 (Shapiro, 2005, Definition 3.4.22). The belief state in a ground situation term σ (w.r.t. $\mathfrak{I}$) is denoted by $K(\sigma)$ and defined to be

$$K(\sigma) \stackrel{\text{def}}{=} \{\psi \in \mathcal{L}_{\mathsf{now}} : \mathfrak{I} \models \mathbf{Bel}(\psi, \sigma)\}$$

That is, the belief state in σ (w.r.t. $\mathfrak{I}$) is just the set of sentences (in $\mathcal{L}_{\mathsf{now}}$) that the agent believes at the situation denoted by σ in the model $\mathfrak{I}$.

Definition 2.4.5 (Shapiro, 2005, Definition 3.4.23). The expansion of a ground situation term σ by ϕ (w.r.t. $\mathfrak{I}$) is denoted by $\sigma + \phi$ and is defined as

$$\sigma + \phi \stackrel{\text{def}}{=} \{\psi \in \mathcal{L}_{\mathsf{now}} : \mathfrak{I} \models \mathbf{Bel}(\phi \supset \psi, \sigma)\}$$

So the expansion of σ by ϕ is another belief state (set of sentences), including the sentences the agent (in the situation denoted by σ) believes are implied by ϕ.

Now we define "revision actions". As this definition does not depend on what the agent believes, it is made in terms of the action theory Σ rather than a particular model of Σ, unlike the last two definitions.

Definition 2.4.6 (Shapiro, 2005, Definition 3.4.10). Given a sentence ϕ uniform in *now*, a ground action term α is a revision action for ϕ, with respect to an action theory

Σ, if the following holds:

$$\Sigma \models \forall s.\ \mathsf{Poss}(\alpha, s) \wedge [\mathsf{SF}(\alpha, s) \equiv \phi[s]] \wedge \left[\bigwedge_{F \text{ a fluent}} \forall \vec{x}.\ F(\vec{x}, s) \equiv F(\vec{x}, \mathsf{do}(\alpha, s))\right]$$

That is, α is a revision action for ϕ if in every situation, the action α

- is possible,
- senses whether ϕ is true,
- and doesn't change the value of any fluent.

Using revision actions, revision can then be defined. Note that since revision actions are sensing actions, it's only possible to revise by true formulas (i.e., performing a revision action for ϕ will only cause the agent to believe ϕ if ϕ is true).

Definition 2.4.7 (Shapiro, 2005, Definition 3.4.24). Suppose that α is a revision action for ϕ and σ is a ground situation term. The revision of σ by ϕ (in terms of α, and w.r.t. $\mathfrak{I}$) is denoted by $\sigma * \phi$ and is defined as

$$\sigma * \phi = \begin{cases} \mathsf{do}(\alpha, \sigma) & \text{if } \mathfrak{I} \models \phi[\sigma] \\ \text{undefined} & \text{otherwise} \end{cases}$$

So the revision of σ by ϕ is a situation term, the result of doing a revision action for ϕ. Note that there's an asymmetry between how revision and expansion are defined: $\sigma + \phi$ is a belief state, whereas $\sigma * \phi$ is a situation term. Therefore, the K function needs to be applied to get the belief state after revision, $K(\sigma * \phi)$.

All this notation requires the AGM postulates to look a bit different. We quote the translation by Shapiro (2005, pp. 74–75) into this notation below:

(AGM$*$1) $K(\sigma * \phi)$ is deductively closed	type
(AGM$*$2) $\phi \in K(\sigma * \phi)$	success
(AGM$*$3) $K(\sigma * \phi) \subseteq \sigma + \phi$	upper bound
(AGM$*$4) If $\neg\phi \notin K(\sigma)$, then $\sigma + \phi \subseteq K(\sigma * \phi)$	lower bound
(AGM$*$5) $K(\sigma * \phi) = \mathcal{L}_{\mathsf{now}}$ iff $\models \neg\phi$	triviality
(AGM$*$6) If $\models \phi \equiv \psi$, then $K(\sigma * \phi) = K(\sigma * \psi)$	extensionality
(AGM$*$7) $K(\sigma * \phi \wedge \psi) \subseteq (\sigma * \phi) + \psi$	iteration upper bound

(AGM∗8) If $\neg\phi \notin K(\sigma * \phi)$, then $(\sigma * \phi) + \psi \subseteq K(\sigma * \phi \wedge \psi)$ iteration lower bound

Shapiro (2005) showed that all the postulates other than (AGM∗5) were satisfied, when revision was defined. The reason (AGM∗5) is not satisfied is that the agent's beliefs may become inconsistent after revising by ϕ, if there were not previously any accessible situations where ϕ was true (furthermore, if the agent's beliefs are inconsistent they will remain so after any revision). Later, Shapiro et al. (2011, p. 178) showed that under the assumption that the accessibility relation is reflexive, (AGM∗5) will be satisfied (because revision is defined only for true formulas).

2.5 Conclusion

In this chapter, we have reviewed logical formalizations of action and change, knowledge and belief, and belief revision. In particular, we have focused on modeling those things in the situation calculus, which we will be using throughout the rest of this thesis.

Chapter 3

Specifying plausibility levels

3.1 Introduction

In this chapter,[1] we present a framework supporting

1. iterated belief change (including retraction of beliefs) and
2. the modeling of action and change, in the context of
3. a simple qualitative specification of what the agent considers plausible.

To do so, we build on the work of Shapiro et al. (2011), who created a framework for modeling iterated belief change in the situation calculus, as we described in §2.2.2. Their approach already has properties (1) and (2); to achieve (3), we incorporate a way of specifying levels of plausibility.

Recall the relevance of plausibility to Shapiro et al.'s frameworks: A central idea behind their approach to belief change is that the agent's beliefs are determined by truth in all the *most plausible* accessible situations, and it is the accessibility relation, not the plausibility levels, that changes over time. However, the initial plausibility levels still have to be described somehow, which has been viewed as difficult. Writing initial state axioms to explicitly assign plausibility levels can be inconvenient. As Schwering and Lakemeyer (2014) (and even Shapiro et al. themselves) point out, the actual numbers used for plausibility levels are not very important. We may also note that writing explicit numbers in an action theory may make it harder to modify. To avoid using plausibility levels at all, Demolombe and Parra (2006) even created an alternative approach to belief revision that instead had sensing actions modify "imaginary" situations that were accessible to agents.

[1] This chapter is based in part on a paper that appeared at KR 2018 (Klassen et al., 2018).

We propose to specify plausibility levels by counting the extensions of distinguished "abnormality" fluents. This approach is based on *cardinality-based circumscription* (CBC) (Liberatore and Schaerf, 1995, 1997; Sharma and Colomb, 1997; Moinard, 2000), a technique for *non-monotonic* reasoning. We provide background on non-monotonic reasoning in §3.2 before describing the details of our approach in §3.3. Counting abnormalities will be the basis for plausibility and belief throughout all the rest of this thesis. In §3.3.4 we introduce *immutable abnormality action theories* (IAATs) that are used in this chapter and the next, and which make the assumption that abnormality fluents don't change over time (corresponding to how Shapiro et al. had fixed plausibility values).

In §3.4 we compare our approach to specifying plausibility levels against potential alternatives. Shapiro et al. had suggested constraining plausibility levels by describing conditional beliefs. Schwering and Lakemeyer (2014) built on that idea by automatically deriving plausibility levels from a set of conditionals. This derivation is essentially the same as the one used by System Z (Pearl, 1990), a system for non-monotonic reasoning, in ranking models based on conditionals. As we will show, Schwering and Lakemeyer's approach inherits some limitations of System Z, which our approach does not share. We then explain why we aren't basing our work on traditional (not cardinality-based) circumscription (McCarthy, 1980, 1986; Lifschitz, 1994). Finally, we provide further evidence for the utility of cardinality-based circumscription by proving that it is more general than another non-monotonic system, lexicographic entailment.

In §3.5 we suggest two ways of enriching the action theories that we use. First, it is natural to consider allowing (non-sensing) actions to change the extensions of abnormality fluents. This turns out to provide a simple way of representing the plausibility of exogenous events, which is more general than a previous extension of the framework of Shapiro et al. to exogenous events that was proposed by Shapiro and Pagnucco (2004). However, we also show that theories with changing abnormalities can exhibit some unusual behavior with respect to beliefs about the past. Second, in §3.5.2 we present another form of action theory in which separate axioms are used to describe the agent's beliefs about the environment's dynamics and the actual dynamics (as in Lakemeyer and Levesque, 1998). We show that the AGM postulates mostly hold for these theories as well.

Finally, before concluding we discuss some further related work in §3.6.

3.2 Background on non-monotonic reasoning

As previously mentioned, our approach to specifying plausibility levels will be based on cardinality-based circumscription, a form of non-monotonic reasoning. The alterna-

tive approaches we compare against in §3.4 will also be based on various forms of non-monotonic inference. Therefore, in this section we provide a brief background on what non-monotonic reasoning is, and give traditional (not cardinality-based) circumscription as an example.

In classical logic, if a set of sentences Γ entails ϕ,

$$\Gamma \models \phi,$$

then the union of Γ with any other set of sentences Δ will also entail ϕ:

$$\Gamma \cup \Delta \models \phi.$$

That is, adding more premises cannot reduce the set of conclusions that can be drawn. This property is called *monotonicity*. Logics, entailment operators, or reasoning procedures that do not have that property are therefore *non-monotonic*.

Non-monotonic reasoning shows up in many commonsense inferences, like drawing "default" conclusions such as assuming that a bird can fly until given evidence otherwise. There have been a wide variety of approaches to non-monotonic reasoning studied within the field of knowledge representation, including default logic (Reiter, 1980), autoepistemic logics (Moore, 1985; Levesque, 1990), and various conditional logics (more on these later).

For this section we will just present circumscription (McCarthy, 1980, 1986; Lifschitz, 1994), one of the most widely-studied forms of non-monotonic inference. There are a number of variants. Here, to give the flavor we present the simple version from Brachman and Levesque (2004, §11.3).

We suppose that there are number of distinguished predicate symbols, $\mathsf{Ab}_1, \ldots, \mathsf{Ab}_n$, which we will call *abnormality* predicates (the term "abnormality" relates to the idea that in default reasoning, people assume that things are normal). We are going to define a form of entailment which, instead of considering all models, considers only the least abnormal models.

Definition 3.2.1 ($\leq_{\mathsf{circ}}$). Given interpretations $\mathfrak{I}_1 = \langle \mathcal{D}, \mathcal{I}_1 \rangle$ and $\mathfrak{I}_2 = \langle \mathcal{D}, \mathcal{I}_2 \rangle$ with the same domain $\mathcal{D}$,

$$\mathfrak{I}_1 \leq_{\mathsf{circ}} \mathfrak{I}_2 \text{ iff for every } i, \text{ it is the case that } \mathcal{I}_1[\mathsf{Ab}_i] \subseteq \mathcal{I}_2[\mathsf{Ab}_i]$$

That is, $\mathfrak{I}_1 \leq_{\mathsf{circ}} \mathfrak{I}_2$ if the extension of each abnormality predicate in $\mathfrak{I}_1$ is a subset of

the extension of that predicate in $\mathfrak{I}_2$. We can then define

$$\mathfrak{I}_1 <_{\mathsf{circ}} \mathfrak{I}_2 \text{ iff } \mathfrak{I}_1 \leq_{\mathsf{circ}} \mathfrak{I}_2 \text{ and not } \mathfrak{I}_2 \leq_{\mathsf{circ}} \mathfrak{I}_1$$

The $<_{\mathsf{circ}}$ relation is then used in defining an entailment operator that only considers the least abnormal models.

Definition 3.2.2 ($\models_{\mathsf{circ}}$). For Γ a set of formulas and φ a formula,

$$\Gamma \models_{\mathsf{circ}} \varphi$$

if for every model $\mathfrak{I}$ of Γ, either $\mathfrak{I} \models \varphi$ or there is another model $\mathfrak{I}'$ of Γ such that $\mathfrak{I}' <_{\mathsf{circ}} \mathfrak{I}$.

This form of entailment amounts to considering what's true in the least abnormal models of Γ, assuming there are no infinite descending chains of less abnormal models of Γ. We can use $\models_{\mathsf{circ}}$ for default reasoning, like the classic example of inferring that a bird flies (included for instance in (Brachman and Levesque, 2004, §11.3)).

Example 3.2.1.
We have that

$$\{\forall x.\ (\mathsf{Bird}(x) \wedge \neg \mathsf{Ab}(x)) \supset \mathsf{Fly}(x), \mathsf{Bird}(\mathsf{tweety})\} \models_{\mathsf{circ}} \mathsf{Fly}(\mathsf{tweety}).$$

That is, if Tweety is a bird, and a bird x flies unless $\mathsf{Ab}(x)$ is true, then Tweety is assumed to fly. This is because in the minimal models, Ab is minimized so $\mathsf{Ab}(\mathsf{tweety})$ is false.

Note that if we were to add $\mathsf{Ab}(\mathsf{tweety})$ to the left-hand-side of the entailment in Example 3.2.1, then we would no longer get the right-hand-side, which shows that $\models_{\mathsf{circ}}$ is indeed a non-monotonic entailment operator.

In more general forms of circumscription, abnormalities can be given priority levels, so that it's preferable to minimize one abnormality predicate rather than another (if the choice has to be made). Also, some predicates can be kept *fixed* during minimization. Being able to keep some predicates fixed has uses, e.g., to prevent minimizing the set of abnormally non-flying birds from minimizing the set of penguins, but it also introduces complications (Brachman and Levesque, 2004, §11.3.3). We should also note that circumscription can be described using second-order logic, in that, given a sentence α, it's possible to define a second order sentence that classically entails φ just in case $\{\alpha\} \models_{\mathsf{circ}} \varphi$.

Note that many interpretations, even ones sharing a domain, will be incomparable by $\leq_{\mathsf{circ}}$. Incomparable interpretations are however treated by $\models_{\mathsf{circ}}$ as though they were equally abnormal. This motivates the following definition:

Definition 3.2.3 ($\lesssim_{\mathsf{circ}}$). Given interpretations $\mathfrak{I}_1$ and $\mathfrak{I}_2$ with the same domain, we define $\mathfrak{I}_1 \lesssim_{\mathsf{circ}} \mathfrak{I}_2$ if either $\mathfrak{I}_1 \leq_{\mathsf{circ}} \mathfrak{I}_2$, or the interpretations are incomparable by $\leq_{\mathsf{circ}}$ (i.e., neither $\mathfrak{I}_1 \leq_{\mathsf{circ}} \mathfrak{I}_2$ nor $\mathfrak{I}_2 \leq_{\mathsf{circ}} \mathfrak{I}_1$).

So $\mathfrak{I}_1 \lesssim_{\mathsf{circ}} \mathfrak{I}_2$ can be read as saying that $\mathfrak{I}_1$ is at least as normal as $\mathfrak{I}_2$. We will shortly be using "abnormalities" to describe implausibility, and we would like to have the at-least-as-plausible-as relation be transitive (Grove (1988) showed that any AGM revision operator corresponds to a transitive plausibility relation). Unfortunately, $\lesssim_{\mathsf{circ}}$ is not transitive, as the following example shows.

Suppose that Ab_1, Ab_2, and Ab_3 are the only abnormality predicates in the language, all with the same priority and all 0-ary, and that interpretations $\mathfrak{I}_1$, $\mathfrak{I}_2$, and $\mathfrak{I}_3$ have the same domain and are such that

$$\mathfrak{I}_1 \models \mathsf{Ab}_1 \wedge \neg\mathsf{Ab}_2 \wedge \neg\mathsf{Ab}_3$$
$$\mathfrak{I}_2 \models \neg\mathsf{Ab}_1 \wedge \mathsf{Ab}_2 \wedge \neg\mathsf{Ab}_3$$
$$\mathfrak{I}_3 \models \mathsf{Ab}_1 \wedge \neg\mathsf{Ab}_2 \wedge \mathsf{Ab}_3$$

Then we have that $\mathfrak{I}_1 <_{\mathsf{circ}} \mathfrak{I}_3$ but it can be seen that any other pair of these three interpretations is not comparable using $\leq_{\mathsf{circ}}$. Therefore, we have $\mathfrak{I}_3 \lesssim_{\mathsf{circ}} \mathfrak{I}_2$ and $\mathfrak{I}_2 \lesssim_{\mathsf{circ}} \mathfrak{I}_1$, but not $\mathfrak{I}_3 \lesssim_{\mathsf{circ}} \mathfrak{I}_1$.

For a plausibility ordering to not be transitive can produce some undesirable behavior, as we will revisit in §3.4.3. The form of cardinality-based circumscription we will present will involve a transitive plausibility relation.

3.3 Defining plausibility and belief with abnormalities

In this section we develop our alternative for specifying plausibility levels. As we've said, it involves counting abnormalities, an idea from cardinality-based circumscription. Therefore, we will first describe CBC (§3.3.1) and show how it can be expressed in second-order logic (§3.3.2). We then show how we can use that second-order formulation as the basis for determining plausibility levels in the situation calculus (§3.3.3), and introduce the action theories that we'll be using in this chapter (and the next) in §3.3.4.

3.3.1 Cardinality-based circumscription (CBC)

Cardinality-based circumscription is a variant of circumscription that has not been commonly used, but has appeared a few times in the literature. Sharma and Colomb (1997) used CBC for diagnosis. Liberatore and Schaerf (1995, 1997) defined CBC in a propositional setting, and showed that it was closely related to certain belief revision operators. Moinard (2000) proved a number of properties of propositional CBC.

Here we present a simple but first-order form of prioritized CBC, where prioritized "abnormality" predicates are minimized and no predicates are kept fixed.[2] We will be using the abnormality predicates as a way of measuring plausibility (which may lead one to want to write slightly different theories than if they were really measuring *normality*, though we will not discuss this distinction further).

Suppose that we have a finite set of abnormality predicates $\mathsf{Ab}_1, \mathsf{Ab}_2, \ldots \mathsf{Ab}_n$, each with an associated priority (intuitively, a higher priority abnormality is a sign of greater implausibility). Let us say that there are k distinct priority levels, and that $\vec{A}^i$ is the list of abnormality predicates of the ith highest priority.

Definition 3.3.1 (abnormality vector). To any interpretation $\mathfrak{I} = \langle \mathcal{D}, \mathcal{I} \rangle$, with domain $\mathcal{D}$ and interpretation mapping $\mathcal{I}$, we can assign a k-ary *abnormality vector* $\vec{c}(\mathfrak{I})$ where each entry is either a natural number or ∞, and whose ith entry is the sum of the cardinalities of the extensions of the priority i abnormality predicates, i.e.,

$$\vec{c}(\mathfrak{I})_i = \sum_{\mathsf{Ab} \in \vec{A}^i} |\mathcal{I}[\mathsf{Ab}]|.$$

Note that we do not distinguish between different infinite cardinalities (i.e., there is only one ∞), and that for a 0-ary predicate, the cardinality of its extension will either be 0 or 1 (depending on whether the interpretation makes it false or true).

Example 3.3.1.
Suppose that Ab_1 is a unary predicate with the highest priority, the binary predicate Ab_2 and 0-ary predicate Ab_3 have lower priority (the same as each other), and Ab_4 is a unary predicate that has the lowest priority. So $\vec{A}^1 = \langle \mathsf{Ab}_1 \rangle$, $\vec{A}^2 = \langle \mathsf{Ab}_2, \mathsf{Ab}_3 \rangle$, and $\vec{A}^3 = \langle \mathsf{Ab}_4 \rangle$. Then consider an interpretation $\mathfrak{I} = \langle \mathcal{D}, \mathcal{I} \rangle$ where

$$\mathcal{D} = \mathbb{N} \qquad \text{(the set of natural numbers)}$$

[2] It's similar to that used by Klassen et al. (2017).

and

$$\begin{aligned}
&\mathcal{I}[\mathsf{Ab}_1] = \{42, 64\} \\
&\mathcal{I}[\mathsf{Ab}_2] = \{\langle 1, 2\rangle, \langle 3, 4\rangle, \langle 7, 0\rangle\} \\
&\mathcal{I}[\mathsf{Ab}_3] = \{\langle\rangle\} && \text{(that is, } \mathsf{Ab}_3 \text{ is true in the interpretation)} \\
&\mathcal{I}[\mathsf{Ab}_4] = \{n : n > 5\}
\end{aligned}$$

Then $\vec{c}(\mathfrak{I}) = \langle 2, 4, \infty\rangle$.

Abnormality vectors can be ordered in a lexicographic way, i.e., we define an ordering $<$ on abnormality vectors as follows:

Definition 3.3.2. Given interpretations $\mathfrak{I}_1$ and $\mathfrak{I}_2$, we define $\vec{c}(\mathfrak{I}_1) < \vec{c}(\mathfrak{I}_2)$ if there is some i so that $c(\mathfrak{I}_1)_i < c(\mathfrak{I}_2)_i$ and so that for all $j < i$, we have $c(\mathfrak{I}_1)_j \leq c(\mathfrak{I}_2)_j$.

That is, lesser abnormality vectors are ones that count a smaller number of abnormalities, giving higher priority to the higher priority abnormalities (one higher priority abnormality outweighs any number of lower priority abnormalities). We can then define (as usual for circumscription) a form of entailment in which only the minimal models are considered, where minimality now means having a minimal abnormality vector (note that since the abnormality vectors are well-ordered, there are never infinite descending chains of models).

Definition 3.3.3 ($\models_{\mathsf{card}}$). For Δ a set of sentences and β a sentence, we write $\Delta \models_{\mathsf{card}} \beta$ if for every interpretation $\mathfrak{I}$ such that $\mathfrak{I} \models \Delta$, either $\mathfrak{I} \models \beta$ or there is another interpretation $\mathfrak{I}'$ such that $\vec{c}(\mathfrak{I}') < \vec{c}(\mathfrak{I})$ and $\mathfrak{I}' \models \Delta$.

To give an example, Example 3.2.1 about Tweety flying is simple enough that CBC behaves like traditional circumscription on it:

$$\{\forall x.\ (\mathsf{Bird}(x) \wedge \neg\mathsf{Ab}(x)) \supset \mathsf{Fly}(x), \mathsf{Bird}(\mathsf{tweety})\} \models_{\mathsf{card}} \mathsf{Fly}(\mathsf{tweety}).$$

This is because in the minimal models, the cardinality of the extension of Ab is minimized (and so has cardinality 0 in this case).

On the other hand, it's not hard to find examples on which $\models_{\mathsf{card}}$ and $\models_{\mathsf{circ}}$ differ. For example, if Ab_1, Ab_2, and Ab_3 all have the same priority, we have

$$((\mathsf{Ab}_1 \wedge \mathsf{Ab}_3) \vee \mathsf{Ab}_2) \models_{\mathsf{card}} \mathsf{Ab}_2$$

since models in which only Ab_2 is true have fewer abnormalities than models in which $\mathsf{Ab}_1 \wedge \mathsf{Ab}_3$ is true. However,

$$((\mathsf{Ab}_1 \wedge \mathsf{Ab}_3) \vee \mathsf{Ab}_2) \not\models_{\mathsf{circ}} \mathsf{Ab}_2$$

since $\{\mathsf{Ab}_2\}$ is not a subset of $\{\mathsf{Ab}_1, \mathsf{Ab}_3\}$ (there are subset-minimal models in which both Ab_1 and Ab_3 are true).

Finally, note that the ordering on interpretations induced by the ordering of their abnormality vectors is transitive, unlike the $\lesssim_{\mathsf{circ}}$ relation.

3.3.2 Expressing CBC in second-order logic

As for regular circumscription, it's also possible to describe CBC using formulas of second-order logic. This was shown for some forms of CBC by Sharma and Colomb (1997, §4.1.1), and we can do the same for ours, based on their approach. This machinery will be useful when we turn to incorporating counting abnormalities into the situation calculus. The main thing to take away from this section will be the ordering $\preccurlyeq^{\infty}_{\mathsf{card}}$ in Definition 3.3.6, which can be used to compare the cardinality of the extensions of predicates, taking priority levels into account. In order to define that, though, we first define a couple simpler orderings, which we call $\leq_{\mathsf{card}}$ and $\leq^{\infty}_{\mathsf{card}}$. Neither of these relations consider priority levels, and the one with the simplest definition, $\leq_{\mathsf{card}}$, does not compare infinite cardinalities in the way that we want.

Suppose that $\vec{P} = \langle P_1, \ldots, P_m \rangle$ and $\vec{Q} = \langle Q_1, \ldots, Q_m \rangle$ are lists of predicates. Note that the cardinality of any set is at most the cardinality of another iff there is an injective function from the first to the second. Furthermore, the sum of the cardinalities of two sets is equal to the cardinality of their disjoint union. Therefore, it can be seen that the sum of the cardinalities of the extensions of $P_1, \ldots, P_m$ is at most the sum of the cardinalities of the extensions of $Q_1, \ldots, Q_m$ iff there is an injective function from the disjoint union of the extensions of $P_1, \ldots, P_m$ to the disjoint union of the extensions of $Q_1, \ldots, Q_m$. We can express this property in second order logic, and do so in the following definition (note that Sharma and Colomb (1997, Definition 4.4) did so for the case where $m = 1$).

Definition 3.3.4. We will use the abbreviation $\vec{P} \leq_{\mathsf{card}} \vec{Q}$ to stand for the second-order sentence

$$\exists \{F_{ij} : 1 \leq i, j \leq k\}.\ \mathsf{INJECTIVE}(F_{ij} : 1 \leq i, j \leq k) \wedge \bigwedge_i \left[\forall \vec{x}_i.\ P_i(\vec{x}_i) \supset \bigvee_j \exists \vec{y}_j \left(F_{ij}(\vec{x}_i, \vec{y}_j) \wedge Q_j(\vec{y}_j) \right) \right]$$

where $\mathsf{INJECTIVE}(F_{ij} : 1 \leq i, j \leq k)$ is an abbreviation for

$$\bigwedge_{i,j} \forall \vec{x}_i, \vec{x}'_i, \vec{y}_j \; [F_{ij}(\vec{x}_i, \vec{y}_j) \wedge F_{ij}(\vec{x}'_i, \vec{y}_j) \supset \vec{x}_i = \vec{x}'_i] \wedge$$
$$\bigwedge_{i,j,k:i\neq k} \forall \vec{x}_i, \vec{x}_k, \vec{y}_j \; \neg[F_{ij}(\vec{x}_i, \vec{y}_j) \wedge F_{kj}(\vec{x}_k, \vec{y}_j)].$$

We can read $\vec{P} \leq_{\mathsf{card}} \vec{Q}$ as saying that the sum of the cardinalities of the extensions of $P_1, \ldots, P_m$ is at most the sum of the cardinalities of the extensions of $Q_1, \ldots, Q_m$. However, while the $\leq_{\mathsf{card}}$ relation compares predicates by cardinality, it's in a way that is a bit more fine-grained than what we want, since it discriminates between differing infinite cardinalities – unlike the abnormality vectors we defined earlier. To match those, we want to define a relation that is like $\leq_{\mathsf{card}}$ except for treating all infinities as being equal.

To do so, let us first define that $\mathsf{INF}(P)$, where P is a predicate symbol, abbreviates the second-order sentence

$$\exists R.\; \forall \vec{x}, \vec{y}, \vec{z} \; [R(\vec{x}, \vec{y}) \wedge R(\vec{y}, \vec{z}) \supset R(\vec{x}, \vec{z})] \wedge \forall \vec{x} \; [\neg R(\vec{x}, \vec{x}) \wedge \exists \vec{y} \; P(\vec{y}) \wedge R(\vec{x}, \vec{y})],$$

saying that there is a transitive, irreflexive, serial relation on the extension of P. This is true iff P has an infinite extension. Note that the number of entries in each of $\vec{x}, \vec{y}$, and $\vec{z}$ in the expansion of $\mathsf{INF}(P)$ matches the arity of P. Finally, we can define a relation $\leq^{\infty}_{\mathsf{card}}$ that is like $\leq_{\mathsf{card}}$ except for treating all infinities as being equal.

Definition 3.3.5. We define $\vec{P} \leq^{\infty}_{\mathsf{card}} \vec{Q}$ as the sentence

$$(\vec{P} \leq_{\mathsf{card}} \vec{Q}) \vee \bigvee_{i,j} \big(\mathsf{INF}(P_i) \wedge \mathsf{INF}(Q_j)\big).$$

We also define $\vec{P} <^{\infty}_{\mathsf{card}} \vec{Q}$ as $\neg(\vec{Q} \leq^{\infty}_{\mathsf{card}} \vec{P})$.

Finally, we want to define a relation that treats some predicates as higher priority than others. Suppose that we partition the elements of $\vec{P}$ among $\vec{P}^1, \ldots, \vec{P}^k$ (where $k \leq m$), so that $\vec{P}^1$ contains the highest priority predicates from $\vec{P}$, $\vec{P}^2$ contains the second highest priority predicates, and so on. Then we define the prioritized relation $\prec^{\infty}_{\mathsf{card}}$ as follows:

Definition 3.3.6. Let $\vec{P}^1, \ldots, \vec{P}^k \prec^{\infty}_{\mathsf{card}} \vec{Q}^1, \ldots, \vec{Q}^k$ abbreviate

$$\bigvee_i \Big(\vec{P}^i <^{\infty}_{\mathsf{card}} \vec{Q}^i \wedge \Big(\bigwedge_{j<i} \vec{P}^j \leq^{\infty}_{\mathsf{card}} \vec{Q}^j\Big)\Big).$$

We can then also define $\vec{P}^1, \ldots, \vec{P}^k \preccurlyeq^{\infty}_{\mathsf{card}} \vec{Q}^1, \ldots, \vec{Q}^k$ as $\neg(\vec{Q}^1, \ldots, \vec{Q}^k \prec^{\infty}_{\mathsf{card}} \vec{P}^1, \ldots, \vec{P}^k)$.

Finally, given a sentence α, it is possible to use $\prec^{\infty}_{\mathsf{card}}$ to define a second-order sentence that entails β just in case $\alpha \models_{\mathsf{card}} \beta$ (similarly to for traditional circumscription). However, we will not need that in this thesis.

3.3.3 Determining the plausibility of situations

We now return to discussing the situation calculus. In order to compare the plausibility levels of situations, we propose to introduce *abnormality fluents*. Each abnormality fluent keeps the same value over time, as specified by SSAs of the form

$$\mathsf{Ab}_i(\vec{x}, \mathsf{do}(a, s)) \equiv \mathsf{Ab}_i(\vec{x}, s) \tag{3.1}$$

for each i. Later on (in §3.5.1) we will explore relaxing this condition, but for now we are following the approach of Shapiro et al., where plausibility levels do not change.

There are priorities associated with the abnormality fluents. Let us use the notation $\vec{A}^i[s]$ to refer to the list of priority i abnormality fluents, with their situation terms fixed to s. We can now redefine the relation $\leq_{\mathsf{pl}}$ (from Definition 2.4.1) to describe when one situation is at least as plausible as another.

Definition 3.3.7 (redefining $\leq_{\mathsf{pl}}$ (from Definition 2.4.1)). We redefine $s \leq_{\mathsf{pl}} s'$ as an abbreviation for a second-order formula:

$$s \leq_{\mathsf{pl}} s' \stackrel{\text{def}}{=} \vec{A}^1[s], \ldots, \vec{A}^k[s] \preccurlyeq^{\infty}_{\mathsf{card}} \vec{A}^1[s'], \ldots, \vec{A}^k[s']$$

Where before the plausibility of s' and s'' was compared by comparing $\mathsf{pl}(s')$ and $\mathsf{pl}(s'')$, now we check in which situation more abnormal fluents hold (taking into account priority). All the rest of the machinery of Shapiro et al. will still work as originally intended. The only role of the plausibility values was to define a total preorder on situations (Shapiro et al., 2011, p. 169 footnote), which we now get by comparing abnormalities.

Remark 3.3.1. If we wanted to continue using Shapiro et al.'s (2011) plausibility function pl, we could relate it to abnormalities by including a second-order axiom like this in our action theories:

$$[\mathsf{Init}(s) \wedge \mathsf{Init}(s')] \supset [(\mathsf{pl}(s) \leq \mathsf{pl}(s')) \equiv (s \leq_{\mathsf{pl}} s')]$$

Note however that this could require that pl's range not be the natural numbers, because there is not in general a way to assign natural numbers to situations that will give the

same ordering as that derived from counting abnormalities (consider what number would have to be assigned to a situation with infinitely many abnormalities).

Since abnormality fluents will define the plausibility of situations in a fixed, domain-independent way, to specify what an agent considers plausible for a particular domain it's necessary to use the accessibility relation to associate abnormality fluents and regular ones. To illustrate, suppose that the axiomatizer wants to have the agent think that birds most plausibly fly. To get that it would suffice to have the accessibility relation set so that in the accessible situations with the fewest abnormalities, each bird flies.

One way to describe the accessibility relation is with formulas describing beliefs. For example, the formula $\mathbf{Bel}\big(\forall x.\ (\mathsf{Bird}(x) \wedge \neg\mathsf{Ab}(x)) \supset \mathsf{Fly}(x), \mathsf{S}_0\big)$ says that in all the most plausible situations accessible from S_0, non-abnormal birds fly. However, this is not sufficient to specify that it's most plausible that each bird flies. For instance, we could have an action theory Σ (similar to a BAT, but including the axioms from Equations 2.6 and 2.7 allowing for multiple initial situations), which does not refer to belief, such that

$$\Sigma \cup \Big\{\mathbf{Bel}\Big(\big[\forall x.\ (\mathsf{Bird}(x) \wedge \neg\mathsf{Ab}(x)) \supset \mathsf{Fly}(x)\big] \wedge \mathsf{Bird}(\mathsf{tweety}), \mathsf{S}_0\Big)\Big\} \not\models \mathbf{Bel}(\mathsf{Fly}(\mathsf{tweety}), \mathsf{S}_0).$$

That is, believing that non-abnormal birds fly and Tweety is a bird does not necessarily mean that it is believed that Tweety flies. This is because $\models$ is classical (second-order) entailment, which is monotonic, and $\neg\mathbf{Bel}(\neg\mathsf{Ab}(\mathsf{tweety}), \mathsf{S}_0)$ can be consistent with what's on the left-hand-side.

We can resolve this, while staying with classical entailment, by addressing two issues:

1. The accessibility relation is underconstrained – we've failed to say, for instance, that it's *not* a condition for a situation to be accessible that $\mathsf{Ab}(\mathsf{tweety})$ is true there.

2. The set of initial situations is also underconstrained – we've failed to say that there even *exist* situations (accessible or not) in which $\mathsf{Ab}(\mathsf{tweety})$ *isn't* true.

To address the first issue, we will fully specify the initial accessibility relation, using *only-knowing* (Levesque, 1990; Lakemeyer and Levesque, 1998). To define an only-knowing operator **OKnow**, we first define an expression $\mathsf{SameHist}(s', s)$ that is true when s and s' have the same action histories from possibly different initial situations.

Definition 3.3.8 (SameHist). We define $\mathsf{SameHist}(s, s')$ as the following abbreviation (from Lakemeyer and Levesque (1998)):

$$\mathsf{SameHist}(s, s') \stackrel{\text{def}}{=} \forall P.\ [\ldots \supset P(s, s')],$$

where P is a second-order variable and the ellipsis abbreviates the conjunction of the following:

$$\forall s_1, s_2.\ (\mathsf{Init}(s_1) \wedge \mathsf{Init}(s_2)) \supset P(s_1, s_2)$$
$$\forall a, s_2, s_2.\ P(s_1, s_2) \supset P(\mathsf{do}(a, s_1), \mathsf{do}(a, s_2))$$

Now, we can get to defining the only-knowing operator.

Definition 3.3.9 (OKnow).

$$\mathbf{OKnow}(\phi, s) \stackrel{\text{def}}{=} \forall s'.\ \mathsf{B}(s', s) \equiv (\phi[s'] \wedge \mathsf{SameHist}(s', s))$$

That is, ϕ is all that is known if the accessible ones are exactly those in which ϕ is true – and which have the same action history, since the agent is always aware of the actions that have occurred. Plausibility is not involved here: what's only-known is known with certainty. (Note that for the rest of this thesis, we are only going to be concerned with what is only-known in S_0, for which purpose it would suffice if $\mathsf{SameHist}(s', s)$ were defined as $\mathsf{Init}(s')$.)

Remark 3.3.2. Only-knowing was originally introduced for formalizing a form of non-monotonic reasoning that arises when the only-known formula refers to beliefs (Levesque, 1990). We will not be considering any instances of that in this thesis.

To address the second issue, similarly to Levesque et al. (1998, p. 173) we can include among the foundational axioms in the action theory a second-order axiom that specifies there are initial situations with all combinations of fluent values (our Equation 2.8 on page 20). So finally, by including that extra foundational axiom in the action theory Σ, we have that if all that is known is that non-abnormal birds fly and Tweety is a bird, then it will be believed that Tweety flies:

$$\Sigma \cup \Big\{\mathbf{OKnow}\Big(\big[\forall x.\ (\mathsf{Bird}(x) \wedge \neg\mathsf{Ab}(x)) \supset \mathsf{Fly}(x)\big] \wedge \mathsf{Bird}(\mathsf{tweety}), \mathsf{S}_0\Big)\Big\} \models \mathbf{Bel}(\mathsf{Fly}, \mathsf{S}_0).$$

In general, we can specify what an agent considers plausible by having it only-know a knowledge base that relates regular fluents to abnormality ones. The action theories that we will be considering next will actually themselves include an axiom to specify what is only-known in S_0.

3.3.4 Immutable abnormality action theories (IAATs)

Definition 3.3.10 (IAAT). An *immutable abnormality action theory (IAAT)* is a set of axioms

$$\Sigma_{\mathsf{found}} \cup \Sigma_{\mathsf{ssa}} \cup \Sigma_{\mathsf{pre}} \cup \Sigma_{\mathsf{sense}} \cup \Sigma_0 \cup \Sigma_{\mathsf{una}} \cup \{\mathbf{OKnow}(\bigwedge \Sigma_{\mathsf{KB}},\ \mathsf{S}_0)\}$$

where

- Σ_{found} is the set of foundational axioms, including Equations 2.1, 2.3, 2.4, 2.6, 2.7, an axiom asserting the existence of initial situations with all combinations of fluent values (Equation 2.8), and Equation 2.9 (for $\mathsf{root}(s)$);
- Σ_{ssa} is a set of successor state axioms, including Equation 2.11 for B, axioms for each abnormality fluent in the form of Equation 3.1, and axioms for every other fluent;
- Σ_{pre} is a set of precondition axioms, one for each action function symbol;
- Σ_{sense} is a set of sensing axioms, one for each action function symbol;
- Σ_0 is a set of initial state axioms, which are uniform in S_0;
- Σ_{una} is a set of unique names axioms for actions;
- and Σ_{KB} is a set of axioms (uniform in *now*) describing what the agent initially knows.

We require Σ to obey the consistency property for functional fluents from (Reiter, 2001, p. 60). Finally, for later use (in Chapter 4) we'll find it convenient to have a functional fluent, $\mathsf{history}(s)$, which stores a representation of the sequence of actions that have occurred in s. To define the $\mathsf{history}$ fluent, we assume Σ_0 contains an axiomatization of lists, specifying how concatenation works, and that $\cdot$ is a function symbol for concatenation. We require that Σ_{ssa} contain the following SSA:

$$\mathsf{history}(\mathsf{do}(a, s)) = \mathsf{history}(s) \cdot a.$$

Σ_0 should contain $\mathsf{history}(\mathsf{S}_0) = \langle\rangle$ and Σ_{KB} should contain $\mathsf{history}(\mathit{now}) = \langle\rangle$, where $\langle\rangle$ denotes the empty list.

The main difference between IAATs and the theories of Shapiro et al. is in how the initial plausibility levels are specified, and so IAATs have many similar properties. In particular it can be seen that IAATs satisfy the AGM postulates to the same extent.

Proposition 3.3.1. Let Σ be an IAAT. For any model $\mathfrak{I}$ of Σ and any ground situation term σ, all the AGM postulates other than (AGM$*$5) are satisfied when revision is defined.

Proof. The proof is essentially the same as the ones in (Shapiro, 2005, §3.4.6) and (Shapiro et al., 2011, Appendix A), except that the pl function is not used in determining plausibility. □

The following example illustrates how an IAAT can be used to model plausible beliefs in a dynamic setting.

Example 3.3.2.
Consider a domain with a light. There are two actions, the sensing action $\mathsf{senseLit}$ that senses whether the light is on (Lit), and the action flipUp, which flips the light switch up (Up) and also turns the light on (Lit) if it is not burnt out (Burnt). The agent knows that initially the light is on iff the switch is up and the light isn't burnt out (and the environment dynamics ensure this relationship continues to hold at all times). In the real initial situation, the switch is up but the light is burnt out. The agent initially considers that it would be implausible for the switch to be down and even more implausible for the light to be burnt out. In formalizing all this below, we make use of two abnormality predicates, $\mathsf{Ab}_1(s)$ and $\mathsf{Ab}_2(s)$, where $\mathsf{Ab}_2(s)$ has higher priority. Ab_1 will be associated with the switch being up and Ab_2 with the light being burnt out. The IAAT is described below:

$$\mathbf{OKnow}([\neg\mathsf{Ab}_1 \supset \mathsf{Up}] \wedge [\neg\mathsf{Ab}_2 \supset \neg\mathsf{Burnt}] \wedge [(\mathsf{Up} \wedge \neg\mathsf{Burnt}) \equiv \mathsf{Lit}], \mathsf{S}_0)$$
$$\neg\mathsf{Lit}(\mathsf{S}_0) \wedge \mathsf{Up}(\mathsf{S}_0) \wedge \mathsf{Burnt}(\mathsf{S}_0)$$
$$\mathsf{Burnt}(\mathsf{do}(a, s)) \equiv \mathsf{Burnt}(s)$$
$$\mathsf{Up}(\mathsf{do}(a, s)) \equiv a = \mathsf{flipUp} \vee \mathsf{Up}(s)$$
$$\mathsf{Lit}(\mathsf{do}(a, s)) \equiv (a = \mathsf{flipUp} \wedge \neg\mathsf{Burnt}(s)) \vee \mathsf{Lit}(s)$$
$$\mathsf{SF}(\mathsf{senseLit}) \equiv \mathsf{Lit}(s)$$
$$\mathsf{SF}(\mathsf{flipUp}) \equiv \mathsf{True}$$

The agent will at first believe the light is on. After sensing that it isn't, the agent will then believe (incorrectly) that the switch is down. After also performing the flipUp action and sensing again, the agent will finally realize that the light is burnt out. This is formalized by the proposition below.

Proposition 3.3.2. Let Σ be the IAAT described above. Then Σ entails each of the following:

1. $\mathbf{Bel}(\mathsf{Lit} \wedge \mathsf{Up} \wedge \neg\mathsf{Burnt}, \mathsf{S}_0)$

2. $\mathbf{Bel}(\neg\mathsf{Lit} \wedge \neg\mathsf{Up} \wedge \neg\mathsf{Burnt}, \mathsf{do}(\mathsf{senseLit}, \mathsf{S}_0))$

3. $\mathbf{Bel}(\neg\mathsf{Lit} \wedge \mathsf{Up} \wedge \mathsf{Burnt}, \mathsf{do}([\mathsf{senseLit}, \mathsf{flipUp}, \mathsf{senseLit}], \mathsf{S}_0))$

Proof. We consider each of the three points.

1. There are accessible situations from S_0 where both Ab_1 and Ab_2 are false; that is, it can be shown that

   $$\Sigma \models \exists s'.\ \mathsf{B}(s', \mathsf{S}_0) \wedge \neg\mathsf{Ab}_1(s') \wedge \neg\mathsf{Ab}_2(s')$$

   Therefore, those are the most plausible accessible situations from S_0, i.e., we get

   $$\Sigma \models \forall s'.\ \mathsf{MPB}(s', \mathsf{S}_0) \supset [\neg\mathsf{Ab}_1(s') \wedge \neg\mathsf{Ab}_2(s')]$$

   So by the definition of **Bel**, we get

   $$\Sigma \models \mathbf{Bel}(\neg\mathsf{Ab}_1 \wedge \neg\mathsf{Ab}_2, \mathsf{S}_0).$$

   We also get that the formula that Σ specifies is initially only-known is believed (since whatever is true in all accessible situations must be true in all the most plausible accessible situations):

   $$\Sigma \models \mathbf{Bel}([\neg\mathsf{Ab}_1 \supset \mathsf{Up}] \wedge [\neg\mathsf{Ab}_2 \supset \neg\mathsf{Burnt}] \wedge [(\mathsf{Up} \wedge \neg\mathsf{Burnt}) \equiv \mathsf{Lit}], \mathsf{S}_0)$$

   From that and the previous entailment we get the desired result, since beliefs are closed under logical consequence.

2. After the sensing action $\mathsf{senseLit}$ is performed, the agent learns that Lit was initially false (and must still be false, since the sensing action didn't change that). So we have

   $$\Sigma \models \mathbf{Bel}(\neg\mathsf{Lit}, \mathsf{do}(\mathsf{senseLit}, \mathsf{S}_0))$$

   Since no world-altering actions have been performed, we still have that the agent believes the only-known formula from the theory:

   $$\Sigma \models \mathbf{Bel}([\neg\mathsf{Ab}_1 \supset \mathsf{Up}] \wedge [\neg\mathsf{Ab}_2 \supset \neg\mathsf{Burnt}] \wedge [(\mathsf{Up} \wedge \neg\mathsf{Burnt}) \equiv \mathsf{Lit}], \mathsf{do}(\mathsf{senseLit}, \mathsf{S}_0))$$

Therefore, the agent can conclude that either Up is false (in which case Ab_1 must be true), or Burnt is true (in which case Ab_2 must be true). This means that there are no situations accessible from $\mathsf{do}(\mathsf{senseLit}, \mathsf{S}_0)$ where both abnormalities are false:

$$\Sigma \models \neg\exists s'.\ \mathsf{B}(s', \mathsf{do}(\mathsf{senseLit}, \mathsf{S}_0)) \wedge \neg\mathsf{Ab}_1(s') \wedge \neg\mathsf{Ab}_2(s')$$

However, it can be seen that there are accessible situations where only one of them is true. The more plausible of those are the ones where Ab_1 is true, since it has lower priority (so it matters less that it's true). Therefore, we have

$$\Sigma \models \mathbf{Bel}(\mathsf{Ab}_1 \wedge \neg\mathsf{Ab}_2, \mathsf{do}(\mathsf{senseLit}, \mathsf{S}_0))$$

The result then follows.

3. In the situation considered here, the agent flipped the switch up (flipUp) before sensing again (and finding that the light is still not on). We no longer need to consider plausibility. Since the agent knows the SSA of Up, at this point, every accessible situation has Up true, and because of the sensing action just performed, in every accessible situation Lit is false. Using the SSA for Lit the agent can conclude that Burnt must have been true (and is still true, since the SSA for Burnt says that doesn't change). □

This example did not illustrate it, but recall that a single higher priority abnormality is more important than any number of lower priority abnormalities. This can be useful for modelling domains with some extremely implausible events (e.g., alien abductions) but sometimes we may want to, for example, model scenarios where evidence accumulates and eventually grows strong enough for the agent to accept some implausible proposition. For that, a different approach may be more convenient. We could associate numeric *weights* to abnormalities to determine how much they contribute to the implausibility of a situation. The difference between weights and priorities is that, unlike with priorities, enough low weight abnormalities will outweigh a high weight abnormality. We could introduce weights without changing the formalism by introducing the shorthand

$$\mathsf{Ab}_i^k(\vec{x}, s) \stackrel{\text{def}}{=} \bigwedge_{j=1}^{k} \mathsf{Ab}_i(j, \vec{x}, s).$$

Intuitively, Ab_i^k behaves as an abnormality fluent with weight k should; for it to be true is counted as k abnormalities. Note that Example 3.3.2 would have worked the same if we had just given Ab_2 a higher weight than Ab_1, rather than a higher priority. We will make more use of weights in Chapter 4.

3.4 Comparisons

CBC has been seldom used in the literature. Below, we provide support for why CBC is an appropriate choice for specifying plausibility levels by considering some alternatives. First, we consider a technique Shapiro et al. had proposed for constraining the plausibility levels by encoding *conditional* beliefs. We then consider *only-believing*, a more sophisticated technique proposed by Schwering and Lakemeyer (2014) that also was based on conditional beliefs. Afterwards, we explain why we could not have used regular circumscription in the way we have used CBC. Finally, we show how CBC is more general than another technique that might be considered, lexicographic entailment.

3.4.1 Using conditional beliefs

Shapiro et al. (2011, p. 177) suggested that "To facilitate the specification of the initial belief state of the agent" a conditional belief operator can be used. Intuitively, a conditional belief in ψ given ϕ, which we will write as $\mathbf{Bel}(\phi \Rightarrow \psi, s)$, means that in the most plausible accessible situations from s where ϕ is true, ψ is also true. This can be defined as an abbreviation using $\leq_{\mathsf{pl}}$.

$$\mathbf{Bel}(\phi \Rightarrow \psi, s) \stackrel{\text{def}}{=} \forall s'.\ \left[\mathsf{B}(s', s) \wedge \phi[s'] \wedge \forall s''.\ (\mathsf{B}(s'', s) \wedge \phi[s'']) \supset s' \leq_{\mathsf{pl}} s''\right] \supset \psi[s'].$$

Note that $\Rightarrow$ is not the material conditional (which we write as $\supset$), but is more similar to the counterfactual conditional from Lewis (1973).

Shapiro et al. use this for one example, where they have an action theory including sentences specifying several conditional beliefs and negations of conditional beliefs. The theory does not entail a unique assignment of plausibility values to situations, but does establish enough of an ordering to get the relevant results for the example.

Schwering and Lakemeyer (2014) criticized this approach for requiring the use of negated conditional beliefs and not uniquely determining the plausibility levels. They presented an approach which also is based on conditionals but avoids those problems, which we therefore turn to.

3.4.2 Only-believing

A proposal to address the problem of specifying plausibility levels can be found in the logic $\mathcal{ESB}$ (Schwering and Lakemeyer, 2014; Schwering et al., 2017). In $\mathcal{ESB}$, which is a modal version of situation calculus without any explicit situation terms, initial plausibility levels can be determined from a set of conditionals that are "only-believed".

In the semantics of $\mathcal{ESB}$, an *epistemic state* is a sequence $e = (e_1, e_2, e_3, \dots)$ of sets of *worlds*, where $e_j \subseteq e_{j+1}$ (and the sequence converges, in that for some N, $e_n = e_N$ when $n > N$). For our purposes, it will suffice to understand a world as providing truth values for first-order sentences (we will not go over how actions are handled in $\mathcal{ESB}$). The idea is that the entries $e_1, e_2, e_3, \dots$ in an epistemic state e correspond to plausibility levels, with less plausible worlds only being in higher-numbered entries (so the epistemic state can also be thought of as an ordering on worlds). An epistemic state e satisfies the sentence $\mathbf{B}(\phi \Rightarrow \psi)$ if ψ is true at all the most plausible worlds where ϕ is true (it's the modal version of $\mathbf{Bel}(\phi \Rightarrow \psi, s)$). Belief in ϕ can be defined with $\mathbf{B}(\mathsf{True} \Rightarrow \phi)$.

Now, let us explain *only-believing*. Suppose that $\Gamma = \{\phi_1 \Rightarrow \psi_1, \dots, \phi_m \Rightarrow \psi_m\}$, where each ϕ_i and ψ_i is an objective formula (not containing any belief or knowledge operators). Let Γ' be the set $\{\phi_1 \supset \psi_1, \dots, \phi_m \supset \psi_m\}$ that is like Γ but with the conditional symbols replaced by material conditionals. The semantics of only-believing is as follows: an epistemic state $e = (e_1, e_2, e_3, \dots)$ satisfies the sentence $\mathbf{O}(\Gamma)$ ("Γ is all that is believed") iff e_1 satisfies all the material conditionals in Γ' and e_{j+1} satisfies the subset of those material conditionals whose antecedents are not true in any world in e_j. That is,

$$e_1 = \{w : w \models \bigwedge \Gamma'\}$$

and for each $j \geq 1$ we have that

$$e_{j+1} = \{w : w \models \bigwedge\{(\phi_i \supset \psi_i) \in \Gamma' : \forall w' \in e_j, w' \not\models \phi_i\}\}.$$

The ordering on worlds given by this epistemic state is essentially that which System Z (Pearl, 1990) would have derived from the conditionals, as described by Schwering (2016, §4.7).

A feature of only-believing is that any conditional only-believed is also believed, which is convenient if the axiomatizer wants to ensure that a conditional is believed. In contrast, if in our approach a knowledge base includes a sentence like

$$(\phi_i \wedge \neg \mathsf{Ab}_i) \supset \psi_i$$

that doesn't guarantee that the most plausible worlds in which ϕ_i is true will have ψ_i true, because that depends on the rest of the knowledge base (which could, for example, also include $(\phi_i \wedge \neg \mathsf{Ab}_j) \supset \neg\psi_i$).

However, the similarity of only-believing to System Z means that some limitations

are inherited, as Schwering et al. (2017, p. 75) note. In particular, the conditionals that are only-believed are not treated as being fully "independent" of each other. Adapting an example from Pearl (1990, §3) gives

$$\mathbf{O}(\mathsf{Penguin} \Rightarrow \mathsf{Bird}, \mathsf{Bird} \Rightarrow \mathsf{Fly}, \mathsf{Penguin} \Rightarrow \neg\mathsf{Fly}, \mathsf{Bird} \Rightarrow \mathsf{Beak}) \models \neg\mathbf{B}(\mathsf{Penguin} \Rightarrow \mathsf{Beak}).$$

An intuitive reading of what's believed is that a penguin most plausibly is a bird ($\mathsf{Penguin} \Rightarrow \mathsf{Bird}$), a bird most plausibly flies ($\mathsf{Bird} \Rightarrow \mathsf{Fly}$), a penguin most plausibly doesn't fly ($\mathsf{Penguin} \Rightarrow \neg\mathsf{Fly}$), and a bird most plausibly has a beak ($\mathsf{Bird} \Rightarrow \mathsf{Beak}$). With these beliefs, the agent unfortunately does not believe that a penguin most plausibly has a beak ($\mathsf{Penguin} \Rightarrow \mathsf{Beak}$). This has been called the "drowning problem", and what is lacking from System Z and other systems with this problem has been called "strong independence" (Strasser and Antonelli, 2016).

To give perhaps the simplest example that shows the problem, the epistemic state corresponding to $\mathbf{O}(\mathsf{True} \Rightarrow \mathsf{P}, \mathsf{True} \Rightarrow \mathsf{Q})$ – that is, to only-believing that P is most plausibly true and that Q is most plausibly true – has only two distinct entries:

$$e_1 = \{w : w \models \mathsf{P} \wedge \mathsf{Q}\}$$
$$e_2 = e_3 = e_4 = \ldots \text{ is the set of all worlds}$$

If the agent with this epistemic state were to learn that P were false, on revising their beliefs they would also lose their belief in Q (since they would discard all worlds from e_1). Intuitively, we would like to have that P and Q are features that independently contribute to the plausibility of a world.

The following example illustrates that IAATs can easily represent independent beliefs (avoiding the drowning problem).

Example 3.4.1.
In the domain for this problem, there are two fluents, $\mathsf{P}(s)$ and $\mathsf{Q}(s)$, whose values never change, and two sensing actions, senseP and senseQ, which respectively sense the values of P and Q. We'll also make use of two abnormality fluents, $\mathsf{Ab}_1(s)$ and $\mathsf{Ab}_2(s)$, of the same priority. In S_0, the actual initial situation, P and Q are false. However, the agent does not know this. Instead, its knowledge base says that P is true (unless there is an abnormality) and Q is true (unless there is a different abnormality). For our action theory,

we can axiomatize this description as follows:

$$\mathsf{P}(\mathsf{do}(a,s)) \equiv \mathsf{P}(s) \qquad \mathsf{Q}(\mathsf{do}(a,s)) \equiv \mathsf{Q}(s)$$
$$\mathsf{SF}(\mathsf{senseP},s) \equiv \mathsf{P}(s) \qquad \mathsf{SF}(\mathsf{senseQ},s) \equiv \mathsf{Q}(s)$$
$$\neg\mathsf{P}(\mathsf{S}_0) \wedge \neg\mathsf{Q}(\mathsf{S}_0)$$
$$\mathbf{OKnow}((\neg\mathsf{Ab}_1 \supset \mathsf{P}) \wedge (\neg\mathsf{Ab}_2 \supset \mathsf{Q}), \mathsf{S}_0)$$

Initially, the accessible situations are exactly those initial situations where $(\neg\mathsf{Ab}_1 \supset \mathsf{P}) \wedge (\neg\mathsf{Ab}_2 \supset \mathsf{Q})$ is true. Because belief is defined as what is true in the accessible situations with the fewest abnormalities, the agent initially (mistakenly) believes $\mathsf{P} \wedge \mathsf{Q}$. If it performs the sensing action senseP, it will come to correctly believe that P is false (but retain its belief that Q is true). If it then also performs senseQ, it will correctly believe that both P and Q are false. The proposition below formalizes these claims.

Proposition 3.4.1. Let Σ be the IAAT described above. Then Σ entails each of the following:

1. $\mathbf{Bel}(\mathsf{P} \wedge \mathsf{Q}, \mathsf{S}_0)$
2. $\mathbf{Bel}(\neg\mathsf{P} \wedge \mathsf{Q}, \mathsf{do}(\mathsf{senseP}, \mathsf{S}_0))$
3. $\mathbf{Bel}(\neg\mathsf{P} \wedge \neg\mathsf{Q}, \mathsf{do}([\mathsf{senseP}, \mathsf{senseQ}], \mathsf{S}_0))$

Proof. This follows straight-forwardly from minimizing abnormalities. The agent always assumes that as few of $\{\mathsf{Ab}_1(now), \mathsf{Ab}_2(now)\}$ are true as its observations allow. □

Aside from the lack of strong independence, another issue with only-believing is that despite being used in a first-order logic, it works essentially the same as the propositional System Z. The epistemic state induced by only-believing a finite number m of conditionals will only have a finite number of distinct entries – at most $m + 1$ (Schwering, 2016, Theorem 4.5.3). However, it's easy to come up with examples for which it's desirable to distinguish between a number of plausibility levels that does not have a clear bound. For example, for every n, an agent might think that a conspiracy involving n people is more plausible than one with $n + 1$ people. Again, using our approach we can easily formalize that, as we show below.

Example 3.4.2.
This example will show the benefits of being able to define an unbounded number of plausibility levels.

Consider a language with the unary relational fluent $\mathsf{Conspirator}$, where the intended meaning of $\mathsf{Conspirator}(x)$ is that x is part of a conspiracy. There is one (sensing) action, $\mathsf{reveal}(x)$, which reveals to the agent whether $\mathsf{Conspirator}(x)$ is true. Who is a conspirator never changes, and in the actual initial situation S_0, everyone is a conspirator. However, the agent thinks that situations with fewer conspirators are more plausible:

$$\mathsf{SF}(\mathsf{reveal}(x), s) \equiv \mathsf{Conspirator}(x, s)$$
$$\mathsf{Conspirator}(x, \mathsf{do}(a, s)) \equiv \mathsf{Conspirator}(x, s)$$
$$\mathsf{Conspirator}(x, \mathsf{S}_0)$$
$$\mathbf{OKnow}\big(\forall x.\ \neg\mathsf{Ab}(x, \mathit{now}) \supset \neg\mathsf{Conspirator}(x, \mathit{now}), \mathsf{S}_0\big)$$

The following proposition says that the agent always believes that the only conspirators are those that have been revealed.

Proposition 3.4.2. Let Σ be the IAAT described above, and let $\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3, \ldots$ be constant symbols. Then for any k,

$$\Sigma \models \mathbf{Bel}\Big(\forall x.\ \mathsf{Conspirator}(x, \mathit{now}) \equiv \Big[\bigvee_{i=1}^{k} x = \mathsf{c}_i\Big],\ \mathsf{do}([\mathsf{reveal}(\mathsf{c}_1), \ldots, \mathsf{reveal}(\mathsf{c}_k)], \mathsf{S}_0)\Big)$$

Proof. After the actions $\mathsf{reveal}(\mathsf{c}_1), \ldots, \mathsf{reveal}(\mathsf{c}_k)$, the agent has learned that $\mathsf{Ab}(\mathsf{c}_1, \mathit{now})$, $\ldots, \mathsf{Ab}(\mathsf{c}_k, \mathit{now})$ must be true, but can still assume that no other object is abnormal (and so no other object is a conspirator). □

So we see that our approach has a couple advantages over only-believing. We can easily represent independent beliefs, and infinitely many plausibility levels.

3.4.3 Subset-based circumscription

The original, and by far the most commonly considered, form of circumscription involves comparing sets, not by cardinality, but by set inclusion (see §3.2). We will call this "subset-based circumscription" or SBC to distinguish it from CBC.

In contrast to SBC, CBC requires the axiomatizer to make the stronger commitment that any set of $n + 1$ abnormalities is less plausible than any set of n abnormalities (if all are at the same priority level), regardless of set inclusion. Furthermore, cardinality-based minimization can behave counterintuitively when infinitely many abnormalities are believed, as the following example shows.

Proposition 3.4.3. Suppose that Σ is an IAAT including

$$\mathbf{OKnow}(\forall i.\ (\exists j.\ i = 2 \times j) \supset \mathsf{Ab}(i, now), \mathsf{S}_0),$$

that is, the agent thinks that all even numbers are abnormal. Then

$$\Sigma \models \forall i.\ (\exists j.\ i = 2 \times j + 1) \supset \neg\mathbf{Bel}(\mathsf{Ab}(i, now), \mathsf{S}_0) \wedge \neg\mathbf{Bel}(\neg\mathsf{Ab}(i, now), \mathsf{S}_0).$$

That is, for each odd number, the agent neither believes that number is abnormal, nor that that number is not abnormal.

Proof. All the most plausible situations accessible from S_0 have infinitely many abnormalities in them (because every even number must be abnormal). The cardinality of the set of even numbers is equal to the cardinality of the union of the set of even numbers and any subset of the odd numbers. Therefore, for any odd number i, there are most plausible accessible situations from S_0 where $\mathsf{Ab}(i)$ is true and ones where $\mathsf{Ab}(i)$ is false. □

Therefore, one might wonder why we aren't using SBC. A key point is the lack of transitivity of the subset-based plausibility relation $\lesssim_{\mathsf{circ}}$. Recall that the framework of Shapiro et al. (2011) obeys (a slightly modified version) of the AGM postulates for belief revision (Alchourrón et al., 1985). This remains true when using CBC to describe the plausibility levels instead of the pl function. However, if we tried to make use of SBC instead of CBC, that would violate (AGM∗4), as we explain in the rest of this section.

Recall from §2.4.2 that Shapiro et al. defined the belief state $K(\sigma)$ of an agent (with respect to a situation term σ), the expansion $\sigma + \phi$, and the revision $\sigma * \phi$ (all relative to a model $\mathfrak{I}$ of the action theory Σ). Shapiro et al.'s translation of the AGM axioms into this notation included the following:

(AGM∗4) If $\neg\phi \notin K(\sigma)$, then $\sigma + \phi \subseteq K(\sigma * \phi)$

Another way to put (AGM∗4) is that if an agent believes a material conditional and doesn't believe its antecedent to be false, then after revising by the antecedent the agent should believe the consequent.

We will show that this axiom can be violated if SBC is used. Suppose that Σ is an action theory like the IAATs we considered before, except the comparison of abnormality predicates by cardinality is replaced by subset inclusion, i.e., s is more plausible than s' if the extension of each abnormality fluent in s is a subset of its extension in s' (for this example we assume all the abnormality fluents have the same priority level).

Suppose further that there are three abnormality fluents $\mathsf{Ab}_1(s), \mathsf{Ab}_2(s)$, and $\mathsf{Ab}_3(s)$, and that Σ includes

$$\mathbf{OKnow}(\mathsf{Ab}_1(now) \vee \mathsf{Ab}_2(now), \mathsf{S}_0).$$

Consider a model $\mathfrak{I}$ of Σ such that $\mathfrak{I} \models \phi_0[\mathsf{S}_0]$, where ϕ_0 stands for $\neg(\mathsf{Ab}_1 \wedge \neg\mathsf{Ab}_3)$. Note that $\mathfrak{I}$ must satisfy all of the following (by virtue of satisfying Σ):

$$\forall s.\ \mathsf{MPB}(s, \mathsf{S}_0) \supset \big[[\mathsf{Ab}_1(s) \wedge \neg\mathsf{Ab}_2(s) \wedge \neg\mathsf{Ab}_3(s)] \vee [\neg\mathsf{Ab}_1(s) \wedge \mathsf{Ab}_2(s) \wedge \neg\mathsf{Ab}_3(s)]\big]$$
$$\exists s.\ \mathsf{MPB}(s, \mathsf{S}_0) \wedge [\mathsf{Ab}_1(s) \wedge \neg\mathsf{Ab}_2(s) \wedge \neg\mathsf{Ab}_3(s)]$$
$$\exists s.\ \mathsf{MPB}(s, \mathsf{S}_0) \wedge [\neg\mathsf{Ab}_1(s) \wedge \mathsf{Ab}_2(s) \wedge \neg\mathsf{Ab}_3(s)]$$

From the last of those we can conclude that $\mathfrak{I} \not\models \mathbf{Bel}(\neg\phi_0, \mathsf{S}_0)$, that is,

$$\neg\phi_0 \notin K(\mathsf{S}_0).$$

Furthermore, observe that we have $\mathfrak{I} \models \mathbf{Bel}(\phi_0 \supset \mathsf{Ab}_2, \mathsf{S}_0)$. So $(\phi_0 \supset \mathsf{Ab}_2) \in K(\mathsf{S}_0)$, and so

$$\mathsf{Ab}_2 \in \mathsf{S}_0 + \phi_0.$$

Now suppose that α is a revision action for ϕ_0. It can be seen that $\mathfrak{I}$ satisfies each of the following:

$$\forall s.\ \mathsf{MPB}(s, \mathsf{do}(\alpha, \mathsf{S}_0)) \supset \big[[\mathsf{Ab}_1(s) \wedge \neg\mathsf{Ab}_2(s) \wedge \mathsf{Ab}_3(s)] \vee [\neg\mathsf{Ab}_1(s) \wedge \mathsf{Ab}_2(s) \wedge \neg\mathsf{Ab}_3(s)]\big]$$
$$\exists s.\ \mathsf{MPB}(s, \mathsf{do}(\alpha, \mathsf{S}_0)) \wedge [\mathsf{Ab}_1(s) \wedge \neg\mathsf{Ab}_2(s) \wedge \mathsf{Ab}_3(s)]$$
$$\exists s.\ \mathsf{MPB}(s, \mathsf{do}(\alpha, \mathsf{S}_0)) \wedge [\neg\mathsf{Ab}_1(s) \wedge \mathsf{Ab}_2(s) \wedge \neg\mathsf{Ab}_3(s)]$$

Note that for $\mathsf{Ab}_1 \wedge \neg\mathsf{Ab}_2 \wedge \mathsf{Ab}_3$ to be true is just as plausible as for $\neg\mathsf{Ab}_1 \wedge \mathsf{Ab}_2 \wedge \neg\mathsf{Ab}_3$ to be true, since to determine plausibility we are not counting abnormalities but comparing by set inclusion. Therefore, we have that $\mathfrak{I} \not\models \mathbf{Bel}(\mathsf{Ab}_2, \mathsf{do}(\alpha, \mathsf{S}_0))$, i.e.,

$$\mathsf{Ab}_2 \notin K(\mathsf{S}_0 * \phi_0),$$

contradicting (AGM$*$4).

Note that the reason that the situations accessible from (the denotation of) S_0 where $\mathsf{Ab}_1 \wedge \neg\mathsf{Ab}_2 \wedge \mathsf{Ab}_3$ was true were not then among the most plausible was because there were

also accessible situations where $\mathsf{Ab}_1 \wedge \neg\mathsf{Ab}_2 \wedge \neg\mathsf{Ab}_3$ was true. However, after the revision action, none of the latter type of situations were accessible (since ϕ_0 was not true at them). There were still most plausible accessible situations where $\neg\mathsf{Ab}_1 \wedge \mathsf{Ab}_2 \wedge \neg\mathsf{Ab}_3$ is true, but because the subset-based plausibility "ordering" is not transitive, those were not ranked as more plausible than the situations where $\mathsf{Ab}_1 \wedge \neg\mathsf{Ab}_2 \wedge \mathsf{Ab}_3$ is true.

So subset-based comparisons cannot be used in the way that we have used cardinality-based ones. For the same reason, we also could not use an alternative form of CBC that compared cardinalities for each predicate individually (see Moinard, 2000, Remark 14), instead of summing together the cardinalities of the extensions of all predicates of the same priority.

3.4.4 Lexicographic entailment

Recall that Schwering and Lakemeyer's only-believing operator determined a plausibility ordering like that given by System Z. There is no reason that we can't define versions of only-believing based on other systems from the extensive literature on using conditionals for default reasoning (e.g., Geffner and Pearl, 1992; Goldszmidt et al., 1993; Benferhat et al., 1993; Lehmann, 1995; Kern-Isberner and Eichhorn, 2014; Beierle et al., 2017). In this section we will consider using one of these systems, lexicographic entailment, in defining an alternative "only-believing" operator. We will then show that CBC is a more general approach.

Lexicographic entailment comes from the work of Benferhat et al. (1993) and Lehmann (1995). The version of lexicographic entailment we'll describe is based on the presentation by Eiter and Lukasiewicz (2000) of lex_{p}-entailment (with some notation changed to aid comparison).

In this system, a knowledge base is given as a pair $\langle \alpha, \Gamma \rangle$ where α is a sentence and

$$\Gamma = \{\phi_1 \Rightarrow \psi_1, \ldots, \phi_m \Rightarrow \psi_m\}$$

is a set of conditionals (again, '$\Rightarrow$' is not the material conditional). Traditionally, α, ϕ_i, and ψ_i were considered to be propositional, but we can let them be first-order. Each conditional $\phi_i \Rightarrow \psi_i$ is associated with a *priority* level from $\{1, \ldots, k\}$ (where 1 is the most important). Given $\langle \alpha, \Gamma \rangle$, we can associate with every interpretation $\mathfrak{I}$ a *preference vector* $\vec{\ell}(\mathfrak{I}) \in \{0, \ldots, m\}^k$, where the ith entry of $\vec{\ell}(\mathfrak{I})$ is the number of values of j for which $(\phi_j \Rightarrow \psi_j)$ is a priority i conditional and $\mathfrak{I} \not\models (\phi_j \supset \psi_j)$.

We will say that $\langle \alpha, \Gamma \rangle$ lexicographically entails $\phi \Rightarrow \psi$, written

$$\langle \alpha, \Gamma \rangle \models_{\mathsf{lex}} \phi \Rightarrow \psi,$$

if ψ is true in every interpretation $\mathfrak{I}$ with minimal $\vec{\ell}(\mathfrak{I})$ such that $\mathfrak{I} \models \alpha \wedge \phi$. As with abnormality vectors in CBC, minimality is determined by lexicographic comparison: $\vec{\ell}(\mathfrak{I}_1) < \vec{\ell}(\mathfrak{I}_2)$ if there exists an i so that $\vec{\ell}(\mathfrak{I}_1)_i < \vec{\ell}(\mathfrak{I}_2)_i$ and for all $j < i$ we have $\vec{\ell}(\mathfrak{I}_1)_j \leq \vec{\ell}(\mathfrak{I}_2)_j$.

Note that there are only a finite number of distinct vectors in the image of $\vec{\ell}(\cdot)$, so we can number them $\vec{\ell}_1, \vec{\ell}_2, \ldots \vec{\ell}_N$ so that $\vec{\ell}_i < \vec{\ell}_{i+1}$. We could define another only-believing operator, which we'll call $\mathbf{O}_{\mathsf{lex}}$, by "embedding" lexicographic entailment within it.

Definition 3.4.1 ($\mathbf{O}_{\mathsf{lex}}$). Suppose $\Gamma = \{\phi_1 \Rightarrow \psi_1, \ldots, \phi_m \Rightarrow \psi_m\}$ is a set of conditionals with associated priority levels, and $\vec{\ell}_1 < \vec{\ell}_2 < \cdots < \vec{\ell}_N$ are the distinct preference vectors in the image of $\vec{\ell}(\cdot)$ (defined w.r.t. Γ). For $e = (e_1, e_2, \ldots)$ an epistemic state (defined as in §3.4.2), we define

$$e \models \mathbf{O}_{\mathsf{lex}}(\Gamma)$$

to hold iff each e_i contains every world w where $\vec{\ell}(w) \leq \vec{\ell}_i$ (let $e_i = e_N$ when $i \geq N$).

This new form of only-believing avoids the drowning problem, insofar as lexicographic entailment does. For example, we have the following:

Proposition 3.4.4. $\mathbf{O}_{\mathsf{lex}}(\mathsf{True} \Rightarrow P, \mathsf{True} \Rightarrow Q) \models \mathbf{B}(\neg P \Rightarrow Q) \wedge \mathbf{B}(\neg Q \Rightarrow P)$

Proof. In the epistemic state $e = (e_1, e_2, \ldots)$ that satisfies the left-hand side, e_1 contains the worlds where both $(\mathsf{True} \supset P)$ and $(\mathsf{True} \supset Q)$ are true, and e_2 contains the worlds where at least one of those conditionals is true. So Q is true at the most plausible $\neg P$-worlds (which are in e_2), and similarly P is true at the most plausible $\neg Q$-worlds. □

As their similarity suggests, there is a sense in which lexicographic entailment can be easily translated into CBC.

Lemma 3.4.1. Suppose that $\mathsf{Ab}_1, \ldots \mathsf{Ab}_m$ are all the abnormality predicates and are all 0-ary, and $\phi_1, \ldots \phi_m, \psi_i, \ldots, \psi_m$ are sentences not including any Ab_i symbol. Let us define $\vec{\ell}(\mathfrak{I})$ relative to

$$\langle \alpha, \{\phi_1 \Rightarrow \psi_1, \ldots, \phi_m \Rightarrow \psi_m\} \rangle,$$

where the priority of $\psi_i \Rightarrow \phi_i$ is the same as the priority of Ab_i. Then for every interpretation $\mathfrak{I}$ such that

$$\mathfrak{I} \models \bigwedge\{\neg\mathsf{Ab}_i \equiv (\phi_i \supset \psi_i) : 1 \leq i \leq m\},$$

we have $\vec{c}(\mathfrak{I}) = \vec{\ell}(\mathfrak{I})$.

Proof. If $\mathfrak{I} \models \bigwedge\{\neg\mathsf{Ab}_i \equiv (\phi_i \supset \psi_i) : 1 \leq i \leq m\}$, then for each i such that $\mathfrak{I} \not\models (\phi_i \supset \psi_i)$, we have $\mathfrak{I} \models \mathsf{Ab}_i$ (and vice versa). The definitions of the abnormality vector $\vec{c}(\mathfrak{I})$ and preference vector $\vec{\ell}(\mathfrak{I})$ make them the same in that case. □

Proposition 3.4.5 (translating lexicographic entailment into CBC). Let $\mathsf{Ab}_1, \ldots, \mathsf{Ab}_m$, $\phi_1, \ldots \phi_m$, $\psi_i, \ldots, \psi_m$, and $\vec{\ell}(\mathfrak{I})$ be as in Lemma 3.4.1 above. Suppose that α, β_1, and β_2 are sentences not including any abnormality symbols. Then

$$\langle \alpha, \{\phi_1 \Rightarrow \psi_1, \ldots, \phi_m \Rightarrow \psi_m\}\rangle \models_{\mathsf{lex}} \beta_1 \Rightarrow \beta_2$$

if and only if

$$\{\alpha \wedge \beta_1\} \cup \{\neg\mathsf{Ab}_i \equiv (\phi_i \supset \psi_i) : 1 \leq i \leq m\} \models_{\mathsf{card}} \beta_2.$$

Proof. Immediate from Lemma 3.4.1. □

This resembles how *formula circumscription* (McCarthy, 1986) can be defined in terms of (traditional) *predicate circumscription*. Lexicographic entailment is essentially a form of cardinality-based formula circumscription. This relationship between CBC and lexicographic entailment is straightforward but to the best of our knowledge has not been previously reported on.

Note how for this translation we used only 0-ary abnormality predicates in the result. When none of the abnormality predicates take any arguments, we can also go in the other direction and translate CBC into lexicographic entailment, as the following proposition shows.

Proposition 3.4.6. Suppose that $\mathsf{Ab}_1, \ldots \mathsf{Ab}_m$ are all the abnormality predicates and are all 0-ary. Then, for any sentences α and β (possibly referring to abnormalities),

$$\{\alpha\} \models_{\mathsf{card}} \beta$$

if and only if

$$\langle \alpha, \{\mathsf{True} \Rightarrow \neg \mathsf{Ab}_1, \ldots, \mathsf{True} \Rightarrow \neg \mathsf{Ab}_m\} \rangle \models_{\mathsf{lex}} \mathsf{True} \Rightarrow \beta$$

where the priority of each conditional $\mathsf{True} \Rightarrow \neg \mathsf{Ab}_i$ is the same as the priority of Ab_i.

Proof. It's easy to see that for any interpretation $\mathfrak{I}$, $\vec{\ell}(\mathfrak{I}) = \vec{c}(\mathfrak{I})$. Therefore, β is true in every interpretation $\mathfrak{I}$ with minimal $\vec{\ell}(\mathfrak{I})$ such that $\mathfrak{I} \models \alpha \wedge \mathsf{True}$ just in case β is true in every interpretation $\mathfrak{I}$ with minimal $\vec{c}(\mathfrak{I})$ such that $\mathfrak{I} \models \alpha$. □

A consequence of this is that techniques for computing propositional lexicographic entailment, such as the MAXSAT-based approach from Borges Garcia (2005), can be applied almost directly to computing *propositional* CBC.

However, CBC isn't restricted to 0-ary abnormality predicates, and works sensibly in the first-order case. By having the abnormality predicates take arguments we can easily get an infinite number of distinct abnormality vectors (as we saw in Example 3.4.2). On the other hand, $\mathbf{O}_{\mathsf{lex}}$ only gives us at most $m + 1$ distinct plausibility levels. We should however note that there is a first-order version of lexicographic entailment from Benferhat and Baida (2004), which is similar to CBC, though defined in a more complicated way (it involves considering what is entailed by "weakened" knowledge bases in which universally quantified formulas have been syntactically modified by listing exceptions to them).

3.5 Extensions

In this section we consider other forms of action theories that, like IAATs, measure plausibility by counting abnormalities, but change some other feature. First, in §3.5.1 we consider allowing abnormalities to change over time, leading to *mutable* abnormality action theories. Then in §3.5.2 we consider "dual" IAATs, theories in which the agent may not know the true dynamics of the domain – i.e., successor state axioms, preconditions axioms, and sensing axioms – because there are separate dynamics axioms to describe what the agent believes.

3.5.1 Changing plausibility over time

Shapiro et al. specified that the plausibilities of situations never changed, and we have followed suit by keeping abnormalities fixed. An obvious alternative would be to instead allow actions to change what is abnormal. This could be useful for reasoning about *exogenous* actions, such as rain starting, or a flood occurring (we will consider handling

exogenous actions in a different way in Chapter 5). Intuitively, the situation resulting from one of those actions could be more plausible than the other.

There is one thing to be careful with when updating plausibilities in this way. The agent believes what is true in all the *currently least abnormal* accessible situations, regardless of how many abnormalities previously existed. So if we write an action theory so as to say that an action removes or adds an abnormality, we have to be careful that what we mean is that the occurrence of that action really does make the situation (with its history) more or less plausible. As we will see at the end of this section, changing abnormalities seems to lead to some quirks regarding beliefs about the past.

However, we will first show some examples involving exogenous actions in which changing abnormalities do give intuitive results. To do so, we are going to build on the approach of Shapiro and Pagnucco (2004). They generalized the framework of Shapiro et al. (2011) to allow exogenous actions, but in their work the agent could not compare the plausibility of exogenous actions, but just assumed there were as few exogenous actions in the past as possible. To be more precise, belief was defined as truth in the "minimal" situations, where minimality was defined in terms of pl values (as in Shapiro et al.) except that ties in pl values were broken by favoring situations with shorter histories. We can generalize that.

Shapiro and Pagnucco divided actions into two types, exogenous and endogenous. They had unary predicates Exo and Endo to identify them. They required that exogenous actions not provide useful sensing information, by having the axiom $\mathsf{Exo}(a) \supset (\forall s).\mathsf{SF}(a, s)$. Furthermore, instead of the axioms constraining B that we have previously seen, they used an axiom that can be written as

$$\forall s', s.\ \mathsf{B}(s', s) \ \equiv\ \mathsf{SameVisHist}(s, s'),$$

where $\mathsf{SameVisHist}(s, s')$ is an abbreviation for a formula saying that s and s' have the same endogenous actions in their histories in the same order (and with the same sensing results), but with possibly different exogenous actions interleaved among them. Intuitively, this reflects how the agent is aware what it itself does, but is not aware of exogenous actions (except of what it can infer through sensing).

As Shapiro and Pagnucco note, this axiom does more than a successor state axiom usually does – it also describes B in initial situations. In their approach the accessibility relation is domain-independent, and it is only by specifying the plausibility function that the axiomatizer gets to determine what the agent believes. This is rather the opposite of the approach we have been taking, where the plausibility of an initial situation is fixed

by what abnormalities exist there, and the beliefs of the agent are determined by how the axiomatizer specifies the accessibility relation (with only-knowing).

Instead of using their axiom for B, we can specify what the agent knows was true in the initial situation by including a sentence of the form $\mathbf{Oinit}(\phi)$ in an action theory, where

$$\mathbf{Oinit}(\phi) \stackrel{\text{def}}{=} \forall s', s.\ \mathsf{B}(s', s) \equiv \big[\mathsf{SameVisHist}(s, s') \wedge \phi[\mathsf{root}(s')]\big].$$

$\mathbf{Oinit}(\phi)$ says that accessible situations must have the same endogenous actions in the same order, and furthermore the knowledge base ϕ must have been true at the initial situations in their histories.

Remark 3.5.1. This specification of the accessibility relation allows the agent to be uncertain what exogenous actions have occurred, and so $\mathbf{Oinit}(\phi)$ holding does not necessarily mean that the agent initially believes ϕ is currently true (since it may consider it possible that exogenous actions have already made ϕ false).

So, now we can consider *mutable abnormality action theories* (MAATs).

Definition 3.5.1 (MAAT). MAATs are like IAATs, except that abnormality predicates are now allowed to have different SSAs, MAATs specify which actions are exogenous (and that those actions don't provide sensing information), and MAATs use $\mathbf{Oinit}(\phi)$ to specify B.

Example 3.5.1 (counting exogenous actions)**.**
First, let's consider how we might emulate in a MAAT the way Shapiro and Pagnucco counted exogenous actions to determine plausibility. We can define a fluent Clock that counts actions:

$$\mathsf{Clock}(i, \mathsf{do}(a, s)) \equiv \exists j.\ i = j + 1 \wedge \mathsf{Clock}(j, s).$$

We use the following SSA for $\mathsf{Ab}(i, s)$, which says (in part) that $\mathsf{Ab}(i, s)$ is true if there is a situation $s' \sqsubset s$ where $\mathsf{Clock}(i, s')$ was true and in which an exogenous action occurred:

$$\mathsf{Ab}(i, \mathsf{do}(a, s)) \equiv (\mathsf{Clock}(i, s) \wedge \mathsf{Exo}(a)) \vee \mathsf{Ab}(i, s).$$

By including

$$\mathbf{Oinit}(\mathsf{Clock}(0) \wedge \forall i.\ \neg\mathsf{Ab}(i) \wedge [i \neq 0 \supset \neg\mathsf{Clock}(i)] \wedge \alpha)$$

in the MAAT – where α is any formula, and the rest specifies that the agent knows the initial time was 0 and there were no abnormalities then – we then have that for an *accessible* situation s, $\mathsf{Ab}(i, s)$ is true iff the ith action in the history of s was exogenous. Consider how this affects the plausibility of accessible situations. If all other abnormalities have higher priority than Ab and never change, this amounts to breaking ties in plausibility by counting exogenous actions, as in Shapiro and Pagnucco's approach.

Example 3.5.2 (the plausibility of rain versus flooding)**.**
This example, in which we will model rain as more plausible than flooding, shows how we can go beyond just counting exogenous actions to determine the plausibility of situations. We have two exogenous actions, rain (rain) and flooding (flood) either of which causes the ground to be wet (Wet). For the purposes of this example, rain and flooding will be modeled as occurring independently. There is an endogenous sensing action see which checks if the ground is wet.

$$\begin{aligned} \mathsf{Wet}(\mathsf{do}(a, s)) &\equiv (a = \mathsf{rain} \vee a = \mathsf{flood}) \vee \mathsf{Wet}(s) \\ \mathsf{SF}(\mathsf{see}, s) &\equiv \mathsf{Wet}(s) \end{aligned}$$

We also have two abnormality fluents, Ab_1 and Ab_2, where Ab_1 has higher priority than Ab_2. Suppose we have an SSA for Clock as before. We can set up the SSAs for Ab_1 and Ab_2 so that flooding at time i causes $\mathsf{Ab}_1(i)$ to become true, and rain at time i causes $\mathsf{Ab}_2(i)$ to become true:

$$\begin{aligned} \mathsf{Ab}_1(i, \mathsf{do}(a, s)) &\equiv [\mathsf{Clock}(i, s) \wedge a = \mathsf{flood}] \vee \mathsf{Ab}_1(i, s) \\ \mathsf{Ab}_2(i, \mathsf{do}(a, s)) &\equiv [\mathsf{Clock}(i, s) \wedge a = \mathsf{rain}] \vee \mathsf{Ab}_2(i, s) \end{aligned}$$

Furthermore, the agent thinks that initially the ground was not wet, the time was 0, and there were no abnormalities.

$$\mathbf{Oinit}(\neg\mathsf{Wet} \wedge \mathsf{Clock}(0) \wedge \forall i.\ \neg\mathsf{Ab}_1(i) \wedge \neg\mathsf{Ab}_2(i) \wedge [i \neq 0 \supset \neg\mathsf{Clock}(i)])$$

The next proposition says that after an exogenous action occurs and the agent then senses that the ground is wet, the agent believes (possibly mistakenly) that it rained. The reason for this is that the agent knows that it either rained or flooded, but considers the rain more plausible.

Proposition 3.5.1. Let Σ be the MAAT described above. Then Σ entails each of the following:

- $\mathbf{Bel}(\exists s.\ \mathsf{do}(\mathsf{rain}, s) \sqsubset now, \mathsf{do}([\mathsf{rain}, \mathsf{see}], \mathsf{S}_0))$
- $\mathbf{Bel}(\exists s.\ \mathsf{do}(\mathsf{rain}, s) \sqsubset now, \mathsf{do}([\mathsf{flood}, \mathsf{see}], \mathsf{S}_0))$

That is, after the action sequence [rain, see] or [flood, see], the agent believes that a rain action occurred in the past.

Proof. In either $\mathsf{do}([\mathsf{rain}, \mathsf{see}])$ or $\mathsf{do}([\mathsf{flood}, \mathsf{see}], \mathsf{S}_0)$, the accessible situations all have at least one rain or flood in their history. The most plausible such situation has (just) one rain action. □

Example 3.5.3 (the fate of abandoned money)**.**
Sometimes, for an exogenous action to have occurred may seem more likely than not. For example, if there was money on the street, you might expect that it will have been taken.

Suppose that there is one exogenous action, steal (and possibly some number of endogenous actions). There is a fluent OnStreet indicating that money is on the street. The steal action results in any money on the street disappearing. For there to be money on the street is abnormal (Ab). The agent believes that initially there was money on the street (abnormally). This description is formalized below:

$$\mathsf{OnStreet}(\mathsf{do}(a, s)) \equiv (\mathsf{OnStreet}(s) \wedge a \neq \mathsf{steal})$$
$$\mathsf{Ab}(\mathsf{do}(a, s)) \equiv (\mathsf{OnStreet}(s) \wedge a \neq \mathsf{steal})$$
$$\mathbf{Oinit}(\mathsf{OnStreet} \wedge \mathsf{Ab})$$

Recall (from Remark 3.5.1) that we are no longer assuming that an agent realizes when it is in an initial situation. An agent in S_0 can believe (mistakenly) that some exogenous actions have taken place. In this example, although the agent in S_0 believes that initially there was money on the street, it also believes that the money has already been stolen.

Proposition 3.5.2. Let Σ be the MAAT described above. Then

$$\Sigma \models \mathbf{Bel}(\exists s.\ \mathsf{do}(\mathsf{steal}, s) = now, \mathsf{S}_0).$$

Proof. The initially accessible situations are initial situations (where Ab is true) and situations where steal just occurred (and Ab is false). The latter are more plausible. □

An issue with forgetting past assumptions

As we alluded to earlier, peculiar things may happen to beliefs about the past when abnormalities can change. To talk about this, let's first follow (Shapiro et al., 2011, Definition 15) in their definition of an operator Prev:

$$\mathsf{Prev}(\phi, s) \stackrel{\text{def}}{=} (\exists s', a).s = \mathsf{do}(a, s') \wedge \phi[s']$$

That is, $\mathsf{Prev}(\phi, s)$ holds if ϕ was true in the situation preceding s.

Now consider a simple MAAT Σ with endogenous actions makeAB and deleteAB (and no exogenous actions), and where the agent initially doesn't know anything.

$$\mathsf{Ab}(\mathsf{do}(a, s)) \equiv (a = \mathsf{makeAb}) \vee (\mathsf{Ab}(s) \wedge \neg\mathsf{deleteAB})$$
$$\mathbf{OKnow}(\mathsf{True}, \mathsf{S}_0)$$

As you would expect, we have that the agent believes Ab is initially false (because the initial situations where Ab is true are less plausible), and after performing makeAb, the agent believes that Ab is true.

$$\Sigma \models \mathbf{Bel}(\neg\mathsf{Ab}, \mathsf{S}_0)$$
$$\Sigma \models \mathbf{Bel}(\mathsf{Ab}, \mathsf{do}(\mathsf{makeAb}(\mathsf{S}_0))$$

However, we also have that after performing makeAb, the agent no longer believes that Ab was initially false:

$$\Sigma \models \neg\mathbf{Bel}(\mathsf{Prev}(\neg\mathsf{Ab}), \mathsf{do}(\mathsf{makeAb}, \mathsf{S}_0))$$

Why is this? From the situation $\mathsf{do}(\mathsf{makeAb}, \mathsf{S}_0)$, all accessible situations have Ab true, and so are all equally plausible (regardless of whether Ab just became true in those situations, or had been true for a while).

Similarly, if the deleteAb action is performed, the agent will believe that Ab is false, but will lose its assumption that Ab was initially false.

$$\Sigma \models \mathbf{Bel}(\neg\mathsf{Ab}, \mathsf{do}(\mathsf{deleteAb}, \mathsf{S}_0))$$
$$\Sigma \models \neg\mathbf{Bel}(\mathsf{Prev}(\neg\mathsf{Ab}), \mathsf{do}(\mathsf{deleteAb}, \mathsf{S}_0))$$

The last entailment holds because from the situation $\mathsf{do}(\mathsf{deleteAb}, \mathsf{S}_0)$ all accessible situations have Ab false, and so are equally plausible (regardless of their history). So after

performing deleteAb, the agent will have no opinion as to whether there never was an abnormality or whether there was one that was just removed.

It seems strange that the agent may retract its past assumptions based on a non-sensing action without preconditions. Perhaps there is some interpretation of the meaning of abnormalities which would justify it. Alternatively, the issue might be addressed by somehow restricting how abnormalities can change in an action theory, or by somehow considering which abnormalities that were true in the past when evaluating the plausibility of a situation. However, we will not be exploring the matter further. Instead, when we return to the topic of exogenous actions in Chapter 5, we will show how to handle them without mutable abnormalities (see in particular §5.2.6, where the new approach is related to the examples we've seen here).

3.5.2 Action theories with separate believed dynamics

In IAATs, like in many other situation calculus theories, the axioms describing the dynamics – successor state axioms, precondition axioms, and sensing axioms – apply to all situations. However as we mentioned in Chapter 2, some authors, like Lakemeyer and Levesque (1998) and Schwering and Lakemeyer (2014, 2015), have used action theories that have two collections of axioms for describing the dynamics. One collection describe the real dynamics, and the other what the agent believes. For readers interested in those sorts of theories, in this section we show that that approach is fully compatible with how we measure plausibility.

We will introduce a new form of action theories which are like IAATs, but allow for different SSAs, precondition axioms, and sensing axioms to apply to epistemically accessible situations than in the actual situation. Our action theories will have two main components: one, similar to a basic action theory, describes the way the environment actually is (i.e., in the situation tree rooted at S_0), while another describes the way the agent (possibly mistakenly) thinks the environment is. Our main result will be that these theories also mostly satisfy the AGM postulates, in much the same way as IAATs do.

Recall from Definition 2.3.2 that if Γ is a set of SSAs, precondition axioms, and/or sensing axioms, then $\Gamma{:}\sigma$ is the set of corresponding relativized axioms that only apply to situations on the situation (sub)tree with root σ. In the action theories we will be introducing, there will be "real" SSAs, preconditions, and sensing axioms relativized to S_0, and ones which the agent believes, relativized to *now*.

We finally get to our main definition.

Definition 3.5.2 (DIAAT). A *dual IAAT (DIAAT)* is a set of axioms

$$\Sigma = \Sigma_{\mathsf{basic}} \cup \{\mathbf{OKnow}(\bigwedge \Pi,\ \mathsf{S}_0)\}$$

where intuitively Σ_{basic} describes reality and Π what the agent believes. Formally,

$$\Sigma_{\mathsf{basic}} = \big(\{\Sigma_{\mathsf{ssa}} \cup \Sigma_{\mathsf{pre}} \cup \Sigma_{\mathsf{sense}}\}{:}\mathsf{S}_0\big) \cup \Sigma_0 \cup \Sigma_{\mathsf{found}} \cup \Sigma_{\mathsf{una}}$$
$$\Pi = \big(\{\Pi_{\mathsf{ssa}} \cup \Pi_{\mathsf{pre}} \cup \Pi_{\mathsf{sense}}\}{:}\mathit{now}\big) \cup \Sigma_{\mathsf{KB}}$$

Σ_{ssa} and Π_{ssa} are sets of successor state axioms, where the ones in Σ_{ssa} are the "real" ones. The SSAs that the agent believes are in Π_{ssa}, meanwhile. Σ_{pre}, Σ_{sense}, and Σ_0 are the real precondition axioms, sensing axioms, and initial state axioms, respectively. Π_{pre} and Π_{sense} are the precondition axioms and sensing axioms the agent believes. Σ_{KB} is a set of sentences uniform in *now*, as in an IAAT. Note that we don't need versions of the foundational axioms or unique names axioms in Π, since those apply to all situations (and so to all epistemically possible situations).

Both Σ_{ssa} and Π_{ssa} should contain the SSA in Equation 2.11 for B. Furthermore, we require that both Σ_{ssa} and Π_{ssa} contain the SSA in Equation 3.1 for any abnormality fluent Ab_i. Σ_{found} is the set of foundational axioms. Compared to in an IAAT, the foundational axiom about the existence of initial situations has to be modified to assert there exist initial situations where all fluents and Poss and SF take arbitrary values *and from which they evolve in arbitrary ways* (similar to Axiom F8 in Lakemeyer and Levesque (1998)). Meanwhile, Σ_{una} are unique names axioms for actions.

Example 3.5.4.
DIAATs allow for the agent to believe sensing axioms that are different from the "real" ones. Let's give a very simple illustration. Consider a DIAAT Σ with two fluents, $\mathsf{P}(s)$ and $\mathsf{Q}(s)$, and a sensing action, sense. Σ_{sense} includes

$$\mathsf{SF}(\mathsf{sense}, s) \equiv \mathsf{Q}(s)$$

but Π_{sense} (what the agent initially believes) includes

$$\mathsf{SF}(\mathsf{sense}, s) \equiv \mathsf{P}(s).$$

Intuitively, the sensing action sense really senses whether Q is true, but the agent thinks that it senses whether P is true. Furthermore, Σ_{KB} is empty, so the agent does not initially know whether P (or Q) is true. Σ_0 contains $\mathsf{Q}(\mathsf{S}_0)$. We assume that Σ is such that the

agent is certain that sensing cannot alter the truth of P or Q.

If the agent performs sense in S_0, it will get a positive response (because Q is true) but the agent will interpret that as meaning that P is true. The following proposition shows this.

Proposition 3.5.3. For Σ as described above, $\Sigma \models \mathbf{Bel}(\mathsf{P}(now), \mathsf{do}(\mathsf{sense}, \mathsf{S}_0))$.

Proof. By the SSA for B (Equation 2.11), Σ entails

$$\forall s'.\ \mathsf{B}(s', \mathsf{do}(\mathsf{sense}, \mathsf{S}_0)) \equiv$$
$$\exists s.\ \mathsf{B}(s, \mathsf{S}_0) \wedge (s' = \mathsf{do}(\mathsf{sense}, s)) \wedge \mathsf{Poss}(\mathsf{sense}, s) \wedge (\mathsf{SF}(\mathsf{sense}, s) \equiv \mathsf{SF}(\mathsf{sense}, \mathsf{S}_0)).$$

Furthermore, $\Sigma \models \mathsf{SF}(\mathsf{sense}, \mathsf{S}_0)$ because Q is true in S_0, but we also have

$$\Sigma \models \forall s.\ \mathsf{B}(s, \mathsf{S}_0) \supset (\mathsf{SF}(\mathsf{sense}, s) \equiv \mathsf{P}(s))$$

because of the sensing axiom in Π_{sense}. The result follows. □

Relation to the AGM postulates

Since DIAATs allow actions to behave differently depending on the initial situation, we cannot directly use the results from Shapiro (2005) or Shapiro et al. (2011) regarding the relation of the framework to the AGM postulates. However, through a simple modification of their definition of what revision actions and revisions are, we can recover results analogous to theirs, and we do so in this section.

As with Shapiro's approach, all revisions of belief are the result of actions. We will define *revision actions* corresponding to revising by particular formulas. Shapiro did this as well, but required that a revision action have the same preconditions, effects, and sensing results in all situations (see Definition 2.4.6). In DIAATs, no action has those properties.

Definition 3.5.3 (redefining a revision action for ϕ). Given a sentence ϕ uniform in now, a ground action term α is a revision action for ϕ with respect to a DIAAT Σ if Σ entails that

$$\forall s \sqsupseteq \mathsf{S}_0.\ \forall s'.\ \mathsf{B}(s', s) \supset$$
$$\left(\mathsf{Poss}(\alpha, s') \wedge [\mathsf{SF}(\alpha, s') \equiv \phi[s']] \wedge \left[\bigwedge_{F \text{ a fluent}} \forall \vec{x}.\ F(\vec{x}, s') \equiv F(\vec{x}, \mathsf{do}(\alpha, s'))\right]\right)$$

That is, α is a revision action for ϕ if in any situation (reachable from S_0), the agent is certain that α

- is possible,
- senses whether ϕ is true,
- and doesn't change the value of any fluent.

In contrast, Definition 2.4.6 had required each of those points to be true of α in *every* situation.

For the purposes of establishing the AGM postulates, like Shapiro we will limit our attention to a restricted set of formulas, those in the language $\mathcal{L}_{\mathsf{now}}$ defined in Definition 2.4.3. We use the same definitions of belief state (Definition 2.4.4) and expansion (Definition 2.4.5) as Shapiro. However, we redefine the revision of σ by ϕ (from Definition 2.4.7).

Definition 3.5.4 (redefining $\sigma * \phi$). Suppose that α is a revision action for ϕ and σ is a ground situation term. We define the revision of σ by ϕ (in terms of α, and w.r.t. a model $\mathfrak{I}$) as

$$\sigma * \phi \stackrel{\text{def}}{=} \begin{cases} \mathsf{do}(\alpha, \sigma) & \text{if } \mathfrak{I} \models \mathsf{SF}(\alpha, \sigma) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Revision by ϕ is not always defined, even when there is a revision action for ϕ, because any revision must come about as the result of a positive sensing result. Note that our definition differs from Definition 2.4.7 in that, where we require that the model makes $\mathsf{SF}(\sigma, \alpha)$ true for $\sigma * \phi$ to be defined, Definition 2.4.7 requires that the model makes $\phi[\sigma]$ true. The definitions would be equivalent if we assumed that α senses the *true* value of ϕ (as Shapiro did), but we don't make that assumption.[3]

Recall that Shapiro (2005) defined a version of the AGM postulates. Our framework satisfies seven of the eight postulates, the same ones that Shapiro's does (Shapiro, 2005, Theorem 3.4.25).

Proposition 3.5.4. Let Σ be a DIAAT. For any model $\mathfrak{I}$ of Σ, and any ground situation term $\sigma = \mathsf{do}(\vec{\beta}, \mathsf{S}_0)$, all the AGM postulates other than (AGM*5) are satisfied when revision is defined.

[3]One consequence of that is that in our framework, unlike Shapiro's, the agent can revise by a logically invalid sentence. This is not of great use to us, though, since after revising by such a sentence the agent's beliefs will remain inconsistent regardless of whatever else subsequently happens.

Proof. The proof is very similar to the ones by Shapiro (2005, §3.4.6) and Shapiro et al. (2011, Appendix A). Like those, it is rather long, and is included in Appendix A. □

We do not get (AGM∗5) for the same reason that Shapiro (2005, p. 79) didn't: if there are no accessible situations in which the formula to be revised by is true, then there will be no accessible situations left after revision.

3.6 Discussion and related work

Here we briefly discuss a few as-yet-unmentioned works.

Pagnucco et al. (2013) were concerned with implementing the framework of Shapiro et al. for a robotics application. They suggested a way of constraining the initial plausibility levels (for use by a robot in interpreting directions) which resembles our approach. The idea is that a number of literals referring to the initial situation are "told" to the robot, and initial situations where more of those literals are true (taking into account priorities given to the fluent symbols in the literals) are constrained to be more plausible. Pagnucco et al. do not discuss using the accessibility relation to associate more complex sentences with these "told" literals, in contrast to the way we use only-knowing to associate abnormalities with other things.

The approach of del Val and Shoham (1994) to belief revision and update in a variant of the situation calculus also (like ours) featured abnormality predicates. However, their use of circumscription was to minimize change of "persistent" properties from situation to situation (they did not have Reiter-style successor state axioms).

Demolombe (2003, p. 192) suggested specifying plausibility levels by, for each plausibility level n, having an axiom characterizing which situations have that plausibility level. They were concerned with a modal version of the situation calculus, but in our setting that might amount to having axioms of the form

$$[\mathsf{B}(s', \mathsf{S}_0) \wedge \mathsf{pl}(s') = n] \;\equiv\; \varphi(s').$$

They assumed that there were only finitely many plausibility levels, which also was a limitation of Schwering and Lakemeyer's (2014) approach.

Fang and Liu (2013) considered belief change in a multi-agent version of the situation calculus, which could also model actions that an agent was unaware of (like our exogenous actions). Following work in dynamic epistemic logic (Baltag and Smets, 2008), they made use of two plausibility orderings, one on situations and one on actions, and updated the plausibility of situations by giving priority to the plausibility of the last action to

have been performed (the so-called "action-priority update"). This is in the spirit of the importance placed on recent information in the AGM approach, but we would argue that is not the most natural way to reason about exogenous actions.

3.7 Conclusion

In order to apply Shapiro et al.'s (2011)'s framework for iterated belief change, an action theory has to be written, in which it is necessary to specify the plausibility levels of accessible situations somehow. We have provided a way of using counting abnormalities, in association with characterizing the accessibility relation using only-knowing, for this. We have shown that this approach has advantages over competitors like Schwering and Lakemeyer (2014). Note that our approach to specifying plausibility levels could also be applied outside the situation calculus – indeed, it was first used in a modal temporal logic (Klassen et al., 2017).

We mostly focused on immutable abnormality action theories (IAATs), involving abnormalities that do not change over time (which is consistent with Shapiro et al.'s use of fixed plausibility levels). We also considered changing abnormalities, though there would be further work needed there to determine how to make beliefs about the past work correctly. There may be some way to relate changing abnormalities to the versions of belief revision that involve changing the plausibility order (Rott, 2009). Another idea that might be considered is, instead of evaluating the plausibility of a situation by counting the abnormalities true in it, summing together the abnormalities true in that situation and all its predecessors. Note that a side-effect of that would be that if not all actions were observable by the agent, the agent would find it more plausible that there were fewer predecessors to the current situation rather than more.

Another direction for future work would be to consider the multi-agent case. That would require a multi-agent version of only-knowing (Aucher and Belle, 2015), which is much more complicated. One might also want to use separate sets of abnormalities fluents in defining the belief operators for each agent (though if the action theory in question does not make any "objective" references to abnormalities, then the same abnormality fluents could easily be reused for different purposes within the different agents' knowledge bases).

In the next chapter, we will see how IAATs can be used to model belief change about domain dynamics. We will also address how to perform regression (recall §2.2.2.4) for IAATs.

Chapter 4

Changing beliefs about domain dynamics

4.1 Introduction

In the previous chapter we described how we can use counting abnormalities to establish initial plausibility levels, in the model of belief in the situation calculus proposed by Shapiro et al. (2011). In this chapter,[1] we apply our immutable abnormality action theories (IAATs, Definition 3.3.10) to specify the plausibility of various aspects of domain dynamics: effects of physical actions, results of sensing, and action preconditions. This will support having the agent change its beliefs about dynamics in reaction to observations of the environment (i.e., the information gained from sensing actions).

First, in §4.2 we provide some general results on how what an agent believes about domain dynamics can be determined for an IAAT. We then suggest patterns to follow when writing SSAs which can control the extent to which observations change the agent's beliefs about action effects (§4.3). It will be up to the axiomatizer to specify the generality of the conclusions the agent should draw from observations (e.g., whether observing a failed attempt to pick up a cup means that that cup can never be picked up, or some broader or narrower conclusion).

We illustrate the change in beliefs that our account can support with an extended example in §4.4. In this example (previously mentioned in §1.2.2), we will describe an action theory about picking up and holding objects, where the agent changes its beliefs about how the fluent $\mathsf{Holding}(x, s)$ (x is held in s) changes over time. There's an action

[1] This chapter is based in part on a paper to appear at KR 2020 (Klassen et al., 2020).

$\mathsf{pick}(x)$ (the agent tries to pick up x). At one point the agent can believe

$$\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv a = \mathsf{pick}(x) \vee \mathsf{Holding}(x, s), \tag{4.1}$$

(i.e., that it's holding an object if it just tried to pick it up or if it was previously holding it). After sensing its failure to pick up a cup, the agent will no longer believe (4.1) but will believe something of the following form:

$$\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv [a = \mathsf{pick}(x) \wedge \neg(\ldots \wedge x = \mathsf{cup})] \vee \mathsf{Holding}(x, s), \tag{4.2}$$

where the ellipsis stands for an expression identifying when the failure occurred. That is, the agent believes that while it did fail to pick up the cup, that failure was a one-time event. So the agent believes that it will be holding anything it picks up except for that one-time failure. Furthermore, after a second time failing to pick up the cup, the agent will no longer believe (4.1) or (4.2), but will now believe the following:

$$\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv (a = \mathsf{pick}(x) \wedge x \neq \mathsf{cup}) \vee \mathsf{Holding}(x, s), \tag{4.3}$$

i.e, that it can only pick up objects other than the cup. Finally, after trying to pick up another object also doesn't result in it being held, the agent will no longer believe (4.1), (4.2) or (4.3), but will now believe the following:

$$\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv (a = \mathsf{pick}(x) \wedge \neg\mathsf{Slippery}(x, s)) \vee \mathsf{Holding}(x, s), \tag{4.4}$$

i.e., that it can only pick up non-slippery objects (it assumes the objects it couldn't pick up were slippery). As we will see in §4.4, all these beliefs will derive from a single IAAT with single SSA for the Holding fluent written using the patterns described in §4.3.

We further show in §4.5 that our approach also handles changing beliefs about sensing axioms and preconditions. In §4.6, we show how regression rewriting can be used with IAATs, and provide a result about how (potentially changed) beliefs about domain dynamics can be incorporated into regression. Then we consider related work (§4.7) before concluding.

4.2 Determining beliefs about dynamics

We will be exploring beliefs entailed by IAATs about SSAs, precondition axioms, and sensing axioms, and how to determine them. Later (§4.3) we suggest having the descriptions of SSAs in the theory refer to abnormalities, so as to describe less plausible ways

that the domain might behave. The techniques of this section can then allow us, in some cases, to find beliefs about SSAs that don't refer to abnormalities. We will also consider sensing axioms in §4.5.1 and preconditions in §4.5.2.

In this chapter, we'll make extensive use of *weights* on abnormalities. None of the examples will involve priority levels, on the other hand, though those would be compatible with the approach in this chapter as well. (Recall that the difference between weights and priorities was described on page 53.)

To get started, we will find it useful to have a symbol to denote the part of an IAAT that describes the domain dynamics.

Definition 4.2.1 (Σ_{dyn}). Given an action theory Σ including SSAs Σ_{ssa}, precondition axioms Σ_{pre}, and sensing axioms Σ_{sense}, we define

$$\Sigma_{\mathsf{dyn}} \stackrel{\text{def}}{=} \Sigma_{\mathsf{ssa}} \cup \Sigma_{\mathsf{pre}} \cup \Sigma_{\mathsf{sense}}.$$

Note that given any IAAT Σ, the agent will always believe the SSAs, precondition axioms, and sensing axioms written in it, since they hold at all situations (this is in contrast to the DIAATs from §3.5.2, which did not have dynamics axioms that applied to all situations). However, we are more interested in what the agent believes about the domain's dynamics in the situation tree it's on, i.e., in situations following from $\mathsf{root}(now)$, rather than in all situations. Therefore, we will use the notion of an axiom that holds on a (sub)tree, rooted at σ (Definition 2.3.1). Henceforth, when we informally talk about the agent believing an axiom about dynamics, we really mean that it believes the corresponding axiom relativized to $\mathsf{root}(now)$.

In terms of axioms relativized to $\mathsf{root}(now)$, the agent will still believe the axioms in Σ_{dyn}, i.e., we will always have that

$$\Sigma \models \forall s.\ \mathbf{Bel}\left(\bigwedge \Sigma_{\mathsf{dyn}}{:}\mathsf{root}(now),\ s\right).$$

However, as the agent changes its beliefs about the abnormality fluents, it may come to believe that various other axioms are equivalent to the original ones, and so also believe them. For example, if Σ includes the SSA

$$\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv (a = \mathsf{pick}(x) \wedge \neg\mathsf{Ab}_1(s)) \vee \mathsf{Holding}(x, s) \tag{4.5}$$

and the agent comes to believe that Ab_1 is true on the situation tree it's on, then the agent will as a result believe a simpler (relativized) SSA saying that Holding does not

change:

$$\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv (a = \mathsf{pick}(x) \wedge \neg\mathsf{True}) \vee \mathsf{Holding}(x, s)$$

That can be simplified to $\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv \mathsf{Holding}(x, s)$.

The following definition will be useful in describing what the agent believes about abnormalities.

Definition 4.2.2 (Ab account). Suppose we have a language with n abnormality fluents, $\mathsf{Ab}_1, \ldots, \mathsf{Ab}_n$, of possibly differing arities. An *Ab account* ξ is an expression

$$\xi(now) \stackrel{\text{def}}{=} \bigwedge_{\mathsf{Ab}_i \in R} \forall \vec{x}.\ \mathsf{Ab}_i(\vec{x}, now) \equiv \xi_i(\vec{x}),$$

where $R \subseteq \{\mathsf{Ab}_1, \ldots, \mathsf{Ab}_n\}$, containing a conjunct corresponding to each Ab_i fluent in R. If Ab_i is an $(m+1)$-ary fluent (where the last of those arguments is the situation) then the expression ξ_i is of the form

$$\left(\bigvee_{k=1}^{\ell} \bigwedge_{j=1}^{m} x_j = \tau_{jk} \right)$$

for some $\ell \geq 0$, where the τ_{jk} are ground terms that do not refer to any situation term. We call R the *range* of ξ.

Intuitively, an Ab account ξ characterizes the extension of each abnormality fluent in its range. Note that if Ab_i is a unary fluent (taking only a situation argument), the expression ξ_i in an Ab account ξ is either True or False. Also, any Ab account requires that there be only finitely many abnormalities, so there can be situations in which no Ab account is true.

Ab accounts are not normally included in action theories, but are things that may be believed or disbelieved by the agent. For example, suppose we're working with a theory including the SSA from Equation 4.5. If the agent observes that $\mathsf{pick}(x)$ fails to make $\mathsf{Holding}$ true of x, then the agent may come to believe the Ab account $(\mathsf{Ab}_1(now) \equiv \mathsf{True})$. Recall that abnormalities do not change over time, so if Ab_1 is true "now", it was always and will always be true. So, as the next lemma notes, if an agent believes an Ab account holds now, then it believes that account has held and will hold forever.

Lemma 4.2.1. For any IAAT Σ, Ab account ξ, and ground action sequence $\vec{\alpha}$,

$$\Sigma \models \mathbf{Bel}(\xi(now) \supset \forall s \sqsupseteq \mathsf{root}(now).\ \xi(s), \mathsf{do}(\vec{\alpha}, \mathsf{S}_0))$$

Proof. This follows from abnormalities not changing and the terms in $\xi(s)$ not depending on s. □

The main role to which we put abnormalities is as markers of subjective plausibility. We are typically more interested in the non-abnormality fluents, and what the agent believes about them, i.e., in beliefs about *normal formulas*.

Definition 4.2.3 (normal formula). A formula is *normal* if it doesn't refer to any abnormality fluents.

The following definition describes a syntactic transformation that can (in some cases) produce normal formulas.

Definition 4.2.4 (normalization). Given a formula ϕ and an Ab account ξ, the normalization of ϕ w.r.t. ξ is a formula ϕ' which is like ϕ but, for each Ab_i in the range of ξ, replaces each occurrence of any subformula of the form $\mathsf{Ab}_i(\vec{\tau}, \sigma)$ (where σ is a situation term and $\vec{\tau}$ are other terms) with $\xi_i(\vec{\tau})$.

For example, if ϕ is the SSA from Equation 4.5,

$$\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv (a = \mathsf{pick}(x) \wedge \neg\mathsf{Ab}_1(x, s)) \vee \mathsf{Holding}(x, s),$$

and ξ is the Ab account $\mathsf{Ab}_1(x, now) \equiv (x = \mathsf{c} \vee x = \mathsf{d})$, then the normalization of ϕ with respect to ξ is

$$\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv (a = \mathsf{pick}(x) \wedge \neg(x = \mathsf{c} \vee x = \mathsf{d})) \vee \mathsf{Holding}(x, s).$$

Note that normalization is defined for any formula ϕ, and if an Ab account ξ includes in its range every abnormality fluent mentioned by ϕ then the result of normalizing ϕ w.r.t. ξ will be a normal formula.

We will see that in some cases the agent will believe the normalizations of certain sentences it believes.

Proposition 4.2.1. Let Σ be an IAAT. Let $\forall s.\ \phi(s)$ be an SSA, precondition axiom, or sensing axiom in Σ. Let $\vec{\alpha}$ be a sequence of ground actions. If there is an Ab account ξ such that

$$\Sigma \models \mathbf{Bel}(\xi, \mathsf{do}(\vec{\alpha}, \mathsf{S}_0))$$

and ϕ' is the normalization of ϕ with respect to ξ, then

$$\Sigma \models \mathbf{Bel}(\forall s \sqsupseteq \mathsf{root}(now).\ \phi'(s), \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)).$$

Proof. Suppose that there is an Ab account ξ such that $\Sigma \models \mathbf{Bel}(\xi, \mathsf{do}(\vec{\alpha}, \mathsf{S}_0))$ and ϕ' is the normalization of ϕ with respect to ξ. By Lemma 4.2.1 we have that

$$\Sigma \models \mathbf{Bel}(\forall s \sqsupseteq \mathsf{root}(now).\ \xi(s), \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)),$$

and it's easy to see that

$$\Sigma \models \mathbf{Bel}(\forall s \sqsupseteq \mathsf{root}(now).\ \xi(s) \supset [\phi'(s) \equiv \phi(s)], \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)).$$

Therefore, since the agent believes $\forall s \sqsupseteq \mathsf{root}(now).\ \phi(s)$ in $\mathsf{do}(\vec{\alpha}, \mathsf{S}_0)$, we get the result. □

Proposition 4.2.1 can be applied to show, given particular action theories, that after certain actions the agent believes simpler dynamics axioms than those that were written in its initial knowledge base (we will put it to use in later sections).

A generalization we can make to Proposition 4.2.1 is to consider cases where the agent believes a disjunction of Ab accounts (but not necessarily any of the disjuncts). To illustrate why that is useful, consider a scenario where an agent unexpectedly fails to pick up an object and doesn't know if that failure was because the object was red or because the object was fuzzy. Then we might want the agent to believe the disjunction of "I can pick up any non-red object" and "I can pick up any non-fuzzy object". For cases like this, the more general Proposition 4.2.2 below is relevant (we won't be looking at such cases in the rest of this chapter, though).

Proposition 4.2.2. Let Σ be an IAAT. Let $\forall s.\ \phi(s)$ be an SSA, precondition axiom, or sensing axiom in Σ. Let $\vec{\alpha}$ be a sequence of ground actions. If there are Ab accounts $\xi^1, \ldots, \xi^k$ such that

$$\Sigma \models \mathbf{Bel}\left(\bigvee_{i=1}^{k} \xi^i, \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)\right)$$

and ϕ'_i is the normalization of ϕ with respect to ξ^i for each i, then

$$\Sigma \models \mathbf{Bel}\left(\bigvee_{i=1}^{k} \forall s \sqsupseteq \mathsf{root}(now).\ \phi'_i(s), \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)\right).$$

Proof. Similarly to in the proof of Proposition 4.2.1, it's easy to see that for each i,

$$\Sigma \models \mathbf{Bel}(\forall s \sqsupseteq \mathsf{root}(now).\ \xi^i(s) \supset [\phi_i'(s) \equiv \phi(s)], \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)).$$

Therefore, since the agent believes $\forall s \sqsupseteq \mathsf{root}(now).\ \phi(s)$ in $\mathsf{do}(\vec{\alpha}, \mathsf{S}_0)$, we can get the result (using Lemma 4.2.1). □

The results in Proposition 4.2.1 and Proposition 4.2.2 apply to any dynamics axioms. In the next section we consider SSAs that are written in a particular way.

4.3 Patterns to follow in writing SSAs

In this section we focus on one aspect of domain dynamics, action effects. We consider how the axiomatizer should write SSAs, so that the agent will change its beliefs by the desired amount given new evidence. We suggest some patterns to follow, based on a traditional way of writing SSAs in terms of positive and negative effects.

Following Reiter (2001, §3.2.7), an SSA for a fluent $F(x, s)$ would be written in the form

$$F(x, \mathsf{do}(a, s)) \equiv \gamma^+(x, a, s) \vee \big(\neg\gamma^-(x, a, s) \wedge F(x, a, s)\big) \tag{4.6}$$

where the formula γ^+ describes positive effects on F, i.e., conditions under which F becomes true, and the formula γ^- describes negative effects on F, i.e., conditions under which F becomes false. Our next definition generalizes that.

Definition 4.3.1 (revisable SSA). We will say that an SSA is a revisable SSA if it is written in the form

$$\begin{aligned} F(x, \mathsf{do}(a, s)) \equiv &\big(\gamma^+(x, a, s) \wedge \neg \textstyle\bigvee_i \mathsf{Imp}_i(x, a, s)\big) \vee \\ &\big(\neg\gamma^-(x, a, s) \wedge F(x, a, s)\big) \end{aligned}$$

where γ^+ and γ^- are normal formulas and each Imp_i is a formula.

The intended use of each Imp_i in a revisable SSA is to describe a less plausible case in which action a fails to make $F(x)$ true. Observe that the structure of a revisable SSA could easily be rearranged to instead describe less plausible cases in which F may fail to become false,

$$\begin{aligned} F(x, \mathsf{do}(a, s)) \equiv &\gamma^+(x, a, s) \vee \\ &\big(\neg(\gamma^-(x, a, s) \wedge \neg \textstyle\bigvee_i \mathsf{Imp}_i(x, a, s)) \wedge F(x, a, s)\big), \end{aligned}$$

may become true,

$$F(x, \mathsf{do}(a, s)) \equiv \big(\gamma^+(x, a, s) \vee \bigvee_i \mathsf{Imp}_i(x, a, s)\big) \vee \big(\neg\gamma^-(x, a, s) \wedge F(x, a, s)\big),$$

or may become false,

$$F(x, \mathsf{do}(a, s)) \equiv \gamma^+(x, a, s) \vee \big(\neg(\gamma^-(x, a, s) \vee \bigvee_i \mathsf{Imp}_i(x, a, s)) \wedge F(x, a, s)\big).$$

Those cases are similar, so we'll just consider Definition 4.3.1.

What might we want the Imp_i formulas to look like? We suggest three forms, for dealing with *exceptional objects*, *exceptional classes*, and *one-time exceptions*. We will shortly show how these influence how the agent's beliefs can change.

Exceptional objects We may want an agent to conclude from an unexpected observation involving a particular object that actions always affect that object differently. To achieve this, we could make $\mathsf{Imp}_i(x, a, s)$ take the form

$$\mathsf{Ab}_j(x, s).$$

Intuitively, if the agent comes to believe that $\mathsf{Ab}_j(\mathsf{c}, \mathit{now})$ is true of a particular object c (e.g., by sensing that F did not become true of c when expected), then the agent will conclude that all actions will fail to make F true of c. Note that it's not necessary for the action theory to say anything else about Ab_j for this to work (other than Ab_j's own SSA, specifying that it doesn't change).

Exceptional classes Another sort of generalization that we might want the agent to make on observing an unexpected (non-)effect is that that unexpected behavior will always occur when dealing with objects from a particular class. For example, an agent might conclude from failing to pick up an object that some objects are too slippery to be picked up. To achieve this, we could make $\mathsf{Imp}_i(x, a, s)$ take the form

$$[P(x, s) \wedge \mathsf{Ab}_j(s)]$$

where P is a fluent. Note that $\mathsf{Ab}_i(s)$ does not take x as an argument, so it being true would mean that *any* objects on the situation tree which s is part of behave abnormally

when they have property P.

One-time exceptions We may want an agent to, when observing an unexpected (non-) effect of an action a in situation s, just accept that a had that (non-)effect in s, while not changing its beliefs about how any action will behave in any other situation. This can be viewed as a sort of minimal way of adjusting the agent's beliefs to keep them consistent. We will call such isolated unexpected (non-)effects "one-time exceptions". We could make $\mathsf{Imp}_i(x, a, s)$ take the form

$$\mathsf{Ab}_j(\mathsf{history}(s), x, a, s)$$

(recall from Definition 3.3.10 that $\mathsf{history}(s)$ is a functional fluent whose value is the list of actions that have occurred in s). Because the abnormality depends on the sequence of actions, each new unexpected action outcome would require another abnormal atom to be true.

We call a revisable SSA that uses only these three patterns a simple SSA:

Definition 4.3.2 (simple SSA). A revisable SSA is a simple SSA if each $\mathsf{Imp}_i(x, a, s)$ is in one of the following forms (the abnormalities may have associated weights):

1. $\mathsf{Ab}_j(x, s)$ (for exceptional objects),
2. $[P(x, s) \wedge \mathsf{Ab}_j(s)]$ (for an exceptional class),
3. or $\mathsf{Ab}_j(\mathsf{history}(s), x, a, s)$ (for one-time exceptions).

We want to show that simple SSAs behave as desired. To facilitate exposition we introduce the next abbreviation.

Definition 4.3.3 ($\vec{\alpha} \rightsquigarrow \phi$). Suppose $\vec{\alpha}$ is a sequence of action terms and ϕ is a formula. Then we define

$$\vec{\alpha} \rightsquigarrow \phi \stackrel{\text{def}}{=} \mathbf{Bel}\big(\forall(\mathsf{root}(now) \sqsubseteq s \sqsupset \phi),\ \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)\big)$$

In the case where the length of $\vec{\alpha}$ is 0, we write $\rightsquigarrow \phi$.

That is, $\alpha_1, \ldots, \alpha_k \rightsquigarrow \phi$ is a formula saying that after performing the actions $\alpha_1, \ldots, \alpha_k$ starting from S_0, the agent believes the universal closure of ϕ, where the variable s is restricted to be a successor of $\mathsf{root}(now)$, what the agent thinks is the initial situation.

For example,

$$\vec{\alpha} \rightsquigarrow [\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv a = \mathsf{pick}(x) \vee \mathsf{Holding}(x, s)]$$

says that after the actions $\vec{\alpha}$, the agent believes that for any situation s which is on the tree rooted at $\mathsf{root}(now)$, and for any object x and action a, the stated relation holds (i.e., x is held after performing a in s just in case $a = \mathsf{pick}(x)$ or x was already held in s).

The following proposition illustrates what sorts of normal SSAs an agent may believe when a simple SSA is used in Σ_{ssa}. We'll see a more concrete example in the next section.

Proposition 4.3.1. Suppose Σ is an IAAT with a simple SSA for F, and $\vec{\alpha}$ is a sequence of ground actions. If there is an Ab account ξ such that

$$\Sigma \models \mathbf{Bel}(\xi, \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)),$$

and which has in its range all the abnormalities referred to by F's SSA, then

$$\Sigma \models \vec{\alpha} \rightsquigarrow F(x, \mathsf{do}(a, s)) \equiv \big[\big(\gamma^+(x, a, s) \wedge \neg\phi(x, a, s)\big) \vee \big(\neg\gamma^-(x, a, s) \wedge F(x, s)\big)\big]$$

where ϕ is a (possibly empty) disjunction, containing the following disjuncts, depending on the original simple SSA:

1. For each $\mathsf{Imp}_i(x, a, s)$ of the form $\mathsf{Ab}_j(x, s)$, ϕ contains either no corresponding disjunct, or a disjunct of the form $\big[\bigvee_{\tau \in T}(x = \tau)\big]$ for some finite set T of ground terms.

2. For each $\mathsf{Imp}_i(x, a, s)$ of the form $[P(x, s) \wedge \mathsf{Ab}_j(s)]$, ϕ contains either no corresponding disjunct, or $P(x, s)$.

3. For each $\mathsf{Imp}_i(x, a, s)$ of the form $\mathsf{Ab}_j(\mathsf{history}(s), x, a, s)$, ϕ contains either no disjunct, or a disjunct of the form

$$[\bigvee_{\langle \tau_1, \tau_2, \tau_3 \rangle \in T} (\mathsf{history}(s) = \tau_1 \wedge x = \tau_2 \wedge a = \tau_3)]$$

for some finite set T of triples of ground terms.

Proof. In the normalization of the original SSA by ξ, the abnormal atoms in each of the $\mathsf{Imp}_i(x, a, s)$ expressions will get replaced, yielding an SSA as described (for (2), there's some additional simplification needed to remove expressions that include True or False). That that SSA is believed follows from Proposition 4.2.1. □

Intuitively, in part (1) of Proposition 4.3.1, T is a list of exceptional objects, the result in (2) depends on whether the agent has determined P to be an exceptional class, and in (3), T identifies very specific circumstances for one-time exceptions. Note that a reason we used the history fluent in our one-time exception pattern, rather than just referring to a situation (which also stores a list of actions), is because the right-hand-sides of SSAs are supposed to be uniform formulas, and so cannot refer to equality of situation terms (while they can refer to expressions like $\mathsf{history}(s) = \tau_1$).

The next proposition says that if we write the SSA for F as a simple SSA, then (under some conditions) the agent will initially believe the traditional SSA from Equation 4.6.

Proposition 4.3.2. Let Σ be an IAAT. Suppose that the SSA for F in Σ_{ssa} is a simple SSA and that Σ_{KB} (the agent's initial knowledge base, from Definition 3.3.10) does not refer to any abnormality fluent. Then

$$\Sigma \models \rightsquigarrow \left[F(\vec{x}, \mathsf{do}(a, s)) \equiv \gamma^+(\vec{x}, a, s) \vee \left(\neg\gamma^-(\vec{x}, a, s) \wedge F(\vec{x}, a, s)\right)\right]$$

Proof. Since Σ_{KB} does not refer to abnormalities, it's easy to see that there are accessible situations from S_0 in which every abnormality is false. So in S_0 the agent believes the Ab account $\bigwedge_{i=1}^{n} \forall \vec{x}.\ \mathsf{Ab}_i(\vec{x}) \equiv \mathsf{False}$. The normalization of any simple SSA w.r.t. that Ab account is (after some simplification) $F(\vec{x}, \mathsf{do}(a, s)) \equiv \left[\gamma^+(\vec{x}, a, s) \vee \left(\neg\gamma^-(\vec{x}, a, s) \wedge F(\vec{x}, a, s)\right)\right]$. The result follows from Proposition 4.2.1. □

While Proposition 4.3.1 and Proposition 4.3.2 only consider SSAs dealing with less-plausible failures of positive effects, analogous results could be shown for SSAs dealing with other types of less plausible behavior. Note that in some cases it may be possible to more compactly write the SSA by distributing the less plausible conditions throughout it rather than grouping them together as we've done.

4.4 An extended example

We are now ready to formalize the revision sequence (Equations 4.1–4.4) described in the example from the introduction (§4.1). We do so by constructing an IAAT $\Sigma_{Holding}$ with the fluents $\mathsf{Holding}(x, s)$, saying that x is being held in s, and $\mathsf{Slippery}(x, s)$, that x is slippery in s. The actions are $\mathsf{pick}(x)$, the action to (try to) pick up x, and sense, which senses whether anything is held. There are constants cup and dish (to represent a cup and a dish), and Σ_0 specifies that they are distinct ($\mathsf{cup} \neq \mathsf{dish}$).

The sensing axioms are

$$\mathsf{SF}(\mathsf{sense}, s) \equiv \exists x.\ \mathsf{Holding}(x, s) \qquad \mathsf{SF}(\mathsf{pick}(x)) \equiv \mathsf{True}$$

Note that picking up does not provide sensing information. All actions are always possible to execute. The SSAs are

$$\begin{aligned}\mathsf{Slippery}(x, \mathsf{do}(a, s)) &\equiv \mathsf{Slippery}(x, s)\\ \mathsf{Holding}(x, \mathsf{do}(a, s)) &\equiv [(a = \mathsf{pick}(x) \wedge \neg \textstyle\bigvee_i \mathsf{Imp}_i(a, x, s)) \vee \mathsf{Holding}(x, s)]\end{aligned}$$

where $\bigvee_i \mathsf{Imp}_i(a, x, s))$ is

$$\mathsf{Ab}_1^2(\mathsf{history}(s), x, a, s) \vee \mathsf{Ab}_2^3(x, s) \vee [\mathsf{Slippery}(x, s) \wedge \mathsf{Ab}_3^4(s)]$$

The disjuncts with lower associated weights (superscripts) are the ones that the agent will tend to find more plausible. So, a one-time exception is more plausible than an exceptional object, which is more plausible than not being able to pick up slippery things (i.e., that slippery objects are an exceptional class). Meanwhile, what's slippery never changes. The initial state axioms include

$$\neg\mathsf{Holding}(x, \mathsf{S}_0) \qquad \mathsf{Ab}_2^3(x, \mathsf{S}_0)$$

That is, nothing is initially held, and every object is actually abnormal – a consequence of this is that no object will be held in any successor of S_0. So in reality, pick actions are ineffectual; they cannot cause anything to become held. The agent's initial knowledge base, Σ_{KB}, is empty (except for specifying that history is initially empty). The theory $\Sigma_{Holding}$ is summarized in Figure 4.1.

The first four points in Proposition 4.4.1 below show how during the action sequence pick(cup), sense, pick(cup), sense, pick(dish), and sense (trying to pick up the cup twice, then trying to pick up the dish, and sensing after each attempt) the agent believes the SSAs from Equations 4.1–4.4. The later points illustrate further aspects of the agent's beliefs over time: (5) shows a one-time exception in a different situation, (6–7) and (9–12) look at what the agent believes about what it's holding, and (8) considers a belief about the future.

Proposition 4.4.1. Let $\Sigma_{Holding}$ be the IAAT described in Figure 4.1. Then it entails each of the following:

1. $\rightsquigarrow [\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv a = \mathsf{pick}(x) \vee \mathsf{Holding}(x, s)]$

$$
\begin{aligned}
\Sigma_{\mathsf{ssa}} = \{&\mathsf{Slippery}(x, \mathsf{do}(a, s)) \equiv \mathsf{Slippery}(x, s),\\
&\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv\\
&\quad \Big[\Big(a = \mathsf{pick}(x) \wedge \neg\Big(\mathsf{Ab}_1^2(\mathsf{history}(s), x, a, s) \vee \mathsf{Ab}_2^3(x, s) \vee\\
&\qquad [\mathsf{Slippery}(x, s) \wedge \mathsf{Ab}_3^4(s)]\Big)\Big) \vee \mathsf{Holding}(x, s)\Big],\\
&\mathsf{B}(s'', \mathsf{do}(a, s)) \equiv \big[\exists s'.\ \mathsf{B}(s', s) \wedge (s'' = \mathsf{do}(a, s')) \wedge\\
&\qquad\qquad \mathsf{Poss}(a, s') \wedge (\mathsf{SF}(a, s') \equiv \mathsf{SF}(a, s))\big],\\
&\mathsf{history}(\mathsf{do}(a, s)) = \mathsf{history}(s) \cdot a\\
\} \cup \{&\mathsf{Ab}_i(\vec{x}, \mathsf{do}(a, s)) \equiv \mathsf{Ab}_i(\vec{x}, s) \quad | \quad \mathsf{Ab}_i \text{ is an abnormality fluent}\}.
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{\mathsf{pre}} = \{&\mathsf{Poss}(\mathsf{sense}, s) \equiv \mathsf{True},\\
&\mathsf{Poss}(\mathsf{pick}(x), s) \equiv \mathsf{True}\}.
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{\mathsf{sense}} = \{&\mathsf{SF}(\mathsf{sense}, s) \equiv \exists x.\ \mathsf{Holding}(x, s),\\
&\mathsf{SF}(\mathsf{pick}(x)) \equiv \mathsf{True}\}.
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_0 = \{&\neg\mathsf{Holding}(x, \mathsf{S}_0),\\
&\mathsf{Ab}_2^3(x, \mathsf{S}_0),\\
&\mathsf{cup} \neq \mathsf{dish},\\
&\mathsf{history}(\mathsf{S}_0) = \langle\rangle\\
\} \cup \{&\text{the axioms describing lists}\}.
\end{aligned}
$$

$$
\Sigma_{\mathsf{KB}} = \{\mathsf{history}(now) = \langle\rangle\}.
$$

Figure 4.1: Axioms in $\Sigma_{Holding}$

2. $\mathsf{pick}(\mathsf{cup}), \mathsf{sense} \rightsquigarrow$
   $\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv [a = \mathsf{pick}(x) \wedge \neg(\mathsf{history}(s) = \langle\rangle \wedge x = \mathsf{cup})] \vee \mathsf{Holding}(x, s)$

3. $\mathsf{pick}(\mathsf{cup}), \mathsf{sense}, \mathsf{pick}(\mathsf{cup}), \mathsf{sense} \rightsquigarrow$
   $\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv (a = \mathsf{pick}(x) \wedge x \neq \mathsf{cup}) \vee \mathsf{Holding}(x, s)$

4. $\mathsf{pick}(\mathsf{cup}), \mathsf{sense}, \mathsf{pick}(\mathsf{cup}), \mathsf{sense}, \mathsf{pick}(\mathsf{dish}), \mathsf{sense} \rightsquigarrow$
   $\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv (a = \mathsf{pick}(x) \wedge \neg\mathsf{Slippery}(x, s)) \vee \mathsf{Holding}(x, s)$

5. $\mathsf{sense}, \mathsf{pick}(\mathsf{cup}), \mathsf{sense} \rightsquigarrow \big[\mathsf{Holding}(x, \mathsf{do}(a, s)) \equiv$
   $[a = \mathsf{pick}(x) \wedge \neg(\mathsf{history}(s) = \langle \mathsf{sense} \rangle \wedge x = \mathsf{cup})] \vee \mathsf{Holding}(x, s)\big]$

6. $\mathsf{pick}(\mathsf{cup}) \rightsquigarrow \mathsf{Holding}(\mathsf{cup}, now)$

7. $\mathsf{pick}(\mathsf{cup}), \mathsf{sense} \rightsquigarrow \neg\mathsf{Holding}(\mathsf{cup}, now)$

8. $\mathsf{pick}(\mathsf{cup}), \mathsf{sense} \rightsquigarrow \mathsf{Holding}(\mathsf{cup}, \mathsf{do}(\mathsf{pick}(\mathsf{cup}), now))$

9. $\mathsf{pick}(\mathsf{cup}), \mathsf{sense}, \mathsf{pick}(\mathsf{cup}) \rightsquigarrow \mathsf{Holding}(\mathsf{cup}, now)$

10. $\mathsf{pick}(\mathsf{cup}), \mathsf{sense}, \mathsf{pick}(\mathsf{cup}), \mathsf{sense} \rightsquigarrow \neg\mathsf{Holding}(\mathsf{cup}, now)$

11. $\mathsf{pick}(\mathsf{cup}), \mathsf{sense}, \mathsf{pick}(\mathsf{cup}), \mathsf{sense}, \mathsf{pick}(\mathsf{dish}) \rightsquigarrow \mathsf{Holding}(\mathsf{dish}, now)$

12. $\mathsf{pick}(\mathsf{cup}), \mathsf{sense}, \mathsf{pick}(\mathsf{cup}), \mathsf{sense}, \mathsf{pick}(\mathsf{dish}), \mathsf{sense} \rightsquigarrow \neg\mathsf{Holding}(\mathsf{dish}, now)$

Proof. We sketch the reason for each entailment. By using Proposition 4.2.1, believed SSAs can be determined by showing which abnormalities the agent believes in the relevant situations. We use the notation $\langle \alpha_1, \ldots, \alpha_k \rangle$ for the term representing the sequence of actions $\alpha_1, \ldots, \alpha_k$.

1. In the initial situation, it's consistent with the agent's knowledge that all abnormalities are false.

2. After the actions $\mathsf{pick}(\mathsf{cup})$ and sense, the agent knows that executing $\mathsf{pick}(\mathsf{cup})$ (from a situation with an empty history) failed to cause $\mathsf{Holding}(\mathsf{cup})$. So

   $$\mathsf{Ab}_1^2(\langle\rangle, \mathsf{cup}, \mathsf{pick}(\mathsf{cup})) \vee \mathsf{Ab}_2^3(\mathsf{cup}) \vee [\mathsf{Slippery}(\mathsf{cup}) \wedge \mathsf{Ab}_3^4]$$

   must be true at all accessible situations. The most plausible of those are where $\mathsf{Ab}_1^2(\langle\rangle, \mathsf{cup}, \mathsf{pick}(\mathsf{cup}))$ is true and all other abnormalities are false (because Ab_1^2 has the lowest weight). (Note that an $a = \mathsf{pick}(\mathsf{cup})$ condition could be include in the believed SSA but is redundant.)

3. After $\mathsf{pick}(\mathsf{cup}), \mathsf{sense}, \mathsf{pick}(\mathsf{cup}), \mathsf{sense}$, the agent has observed two cases in which picking up cup failed. The most plausible accessible situations are those where $\mathsf{Ab}_2^3(\mathsf{cup})$ is true and all other abnormalities are false. Note that situations where instead there were two one-time exceptions, i.e.,

$$\mathsf{Ab}_1^2(\langle\rangle, \mathsf{cup}, \mathsf{pick}(\mathsf{cup})) \qquad \mathsf{Ab}_1^2(\langle \mathsf{pick}(\mathsf{cup}), \mathsf{sense}\rangle, \mathsf{cup}, \mathsf{pick}(\mathsf{cup}))$$

   are less plausible, as the sum of their weights is four.

4. After these actions, the agent has seen two failures to pick up cup and one to pick up dish. The most plausible accessible situations are those where slippery objects can't be picked up (and cup and dish are slippery), i.e., where $\mathsf{Ab}_3^4 \wedge \mathsf{Slippery}(\mathsf{cup}) \wedge \mathsf{Slippery}(\mathsf{dish})$ is true, and there are no other abnormalities.

5. This is like point (2) above, except that the $\mathsf{pick}(\mathsf{cup})$ action was executed in a situation with history $\langle \mathsf{sense}\rangle$ instead of $\langle\rangle$.

6–7, 9–12. For (6), (9), and (11), note that in the situations involved, the agent still believes the SSAs from (1), (2), and (3), respectively (the agent does not gain information from trying to pick something up). From each of those SSAs, and the actions that have occurred, it follows that the agent must be holding the relevant object.

   For (7), (10), and (12), the result follows because the agent has just performed a sensing action, and so no accessible situation has the agent holding anything.

8. This follows from the agent believing the SSA from (2). □

So, as promised in the introduction, we have demonstrated how the axiomatizer can control how the agent's beliefs are changed by observations. Our approach could also easily handle other changes of beliefs beyond those we've shown. For example, if we wanted the agent to not conclude the cup was abnormal until observing that it had failed to pick up the cup three times (instead of just twice), we could achieve that by changing the relative weights associated with the abnormalities. Also, while we wrote the action theory so that in actuality nothing could ever be picked up, that of course is not essential. Note that if the agent observes enough failures to pick up the cup to conclude that the cup is abnormal, but then senses that it's successfully picked up the cup, the agent would be forced to retract its belief that the cup was abnormal (and instead explain the past failures as each being a one-time exception).

4.5 Beyond SSAs

The previous section was concerned with SSAs, but beliefs about other aspects of the dynamics of the domain – sensing and preconditions – can change as well. We consider some examples in this section.

4.5.1 Changing beliefs about sensing

By having sensing axioms refer to abnormalities, we can easily allow for the agent to change its beliefs about what sensing tells it. The following examples show how we can model inaccurate and noisy sensors.

Example 4.5.1.
Suppose that we have another IAAT Σ describing a setting where there are two actions (corresponding to two different sensors), sense_1 and sense_2. The agent's initial knowledge base, Σ_{KB}, is empty (except for specifying that $\mathsf{history}$ is initially empty). Σ_{sense} contains

$$\mathsf{SF}(\mathsf{sense}_1, s) \equiv \mathsf{Q}(s) \qquad\qquad \mathsf{SF}(\mathsf{sense}_2, s) \equiv [\mathsf{Q}(s) \vee \mathsf{Ab}(s)]$$

That is, sense_1 senses whether Q is true, and the agent knows that. However, the agent initially believes that sense_2 does the same (because the agent assumes Ab is false). But Σ_0 includes

$$\mathsf{Ab}(\mathsf{S}_0) \wedge \neg\mathsf{Q}(\mathsf{S}_0),$$

so in reality the sensor represented with sense_2 is broken and always returns a positive result. The SSA for Q says Q never changes. By using both sensors and comparing their results, the agent can come to learn the truth about sense_2, as the following proposition says.

Proposition 4.5.1. For Σ as described above,

$$\Sigma \models \mathsf{sense}_1, \mathsf{sense}_2 \rightsquigarrow [\mathsf{SF}(\mathsf{sense}_2, s) \equiv \mathsf{True}]$$

Proof. The result will follow (using Proposition 4.2.1) from showing that $\Sigma \models \mathbf{Bel}(\mathsf{Ab}, \mathsf{do}([\mathsf{sense}_1, \mathsf{sense}_2]))$.

It can be shown that any situation accessible from $\mathsf{do}([\mathsf{sense}_1, \mathsf{sense}_2], \mathsf{S}_0)$ must have

the same action history, and the same sensing results on that history. That is, we have

$$\Sigma \models \forall s'.\ \mathsf{B}(s', \mathsf{do}([\mathsf{sense}_1, \mathsf{sense}_2], \mathsf{S}_0)) \supset \big(\exists s.\ s' = \mathsf{do}([\mathsf{sense}_1, \mathsf{sense}_2], s) \wedge \\ [\mathsf{SF}(\mathsf{sense}_1, s) \equiv \mathsf{SF}(\mathsf{sense}_1, \mathsf{S}_0)] \wedge \\ [\mathsf{SF}(\mathsf{sense}_2, \mathsf{do}(\mathsf{sense}_1, s)) \equiv \mathsf{SF}(\mathsf{sense}_2, \mathsf{do}(\mathsf{sense}_1, \mathsf{S}_0))]\big)$$

It can be seen that $\Sigma \models \mathsf{SF}(\mathsf{sense}_1, \mathsf{S}_0) \equiv \mathsf{False}$ and $\Sigma \models \mathsf{SF}(\mathsf{sense}_2, \mathsf{do}(\mathsf{sense}_1, \mathsf{S}_0)) \equiv \mathsf{True}$. Therefore,

$$\Sigma \models \forall s'.\ \mathsf{B}(s', \mathsf{do}([\mathsf{sense}_1, \mathsf{sense}_2], \mathsf{S}_0)) \supset \big(\exists s.\ s' = \mathsf{do}([\mathsf{sense}_1, \mathsf{sense}_2], s) \wedge \\ [\mathsf{SF}(\mathsf{sense}_1, s) \equiv \mathsf{False}] \wedge [\mathsf{SF}(\mathsf{sense}_2, \mathsf{do}(\mathsf{sense}_1, s)) \equiv \mathsf{True}]\big).$$

Using the sensing axioms and SSAs we can equivalently rewrite that as

$$\Sigma \models \forall s'.\ \mathsf{B}(s', \mathsf{do}([\mathsf{sense}_1, \mathsf{sense}_2], \mathsf{S}_0)) \supset \big(\exists s.\ s' = \mathsf{do}([\mathsf{sense}_1, \mathsf{sense}_2], s) \wedge \\ [\mathsf{Q}(s) \equiv \mathsf{False}] \wedge [(\mathsf{Q}(s) \vee \mathsf{Ab}(s)) \equiv \mathsf{True}]\big).$$

The result that the agent believes Ab in $\mathsf{do}([\mathsf{sense}_1, \mathsf{sense}_2], \mathsf{S}_0)$ then follows easily. □

If the agent can change its beliefs both about SSAs and about sensing axioms, should it explain an unexpected sensor reading by concluding that the sensor behaves differently from expected, or by concluding that some prior action had an unexpected effect and the sensor is working as expected? Or should the agent be uncertain which of those is the case? Any of those might be a reasonable outcome, and so we allow the axiomatizer to arrange for what they want. The following example illustrates this.

Example 4.5.2.
We consider another action theory Σ about picking up objects, this time using this SSA for $\mathsf{Holding}$:

$$\mathsf{Holding}(x, \mathsf{do}(a, s)) \ \equiv \ (a = \mathsf{pick}(x) \wedge \neg \mathsf{Ab}_1(\mathsf{history}(s), x, a, s)) \vee \mathsf{Holding}(x, s)$$

(which says pick-up actions might implausibly fail in a one-time way). Furthermore, the domain has one sensing action, sense, which now has this corresponding sensing axiom:

$$\mathsf{SF}(\mathsf{sense}, s) \ \equiv \ \exists x.\ \mathsf{Holding}(x) \wedge \neg \mathsf{Ab}_2(\mathsf{history}(s), x, s)$$

So, not only does the agent think that pick actions may implausibly fail, the agent also thinks that sense actions may implausibly give a false negative result, i.e., indicate that

nothing is being held even though something is really being held. The agent's initial knowledge base, Σ_{KB}, is empty (except for specifying that $\mathsf{history}$ is initially empty). Finally, in actuality, pick-ups always fail: Σ_0 includes $\forall x, y, a.\ \mathsf{Ab}_1(x, y, a, \mathsf{S}_0)$.

The following proposition considers the agent's beliefs after trying to pick up the cup and then sensing that it's not holding anything. Depending on the relative weights of Ab_1 and Ab_2, the agent will either conclude that the pick action failed (so the cup is now not being held) or that the sense action gave a false negative result (so the cup is now being held), or be unsure which of those occurred.

Proposition 4.5.2. Let Σ be the IAAT described above. Then

- If Ab_1 has higher weight than Ab_2, then $\Sigma \models \mathsf{pick}(\mathsf{cup}), \mathsf{sense} \rightsquigarrow \mathsf{Holding}(now)$.
- If Ab_1 has lower weight than Ab_2, then $\Sigma \models \mathsf{pick}(\mathsf{cup}), \mathsf{sense} \rightsquigarrow \neg\mathsf{Holding}(now)$.
- If Ab_1 and Ab_2 have the same weight, neither of the above entailments holds.

Proof. After the two actions, the agent knows that either the pick-up failed or the sensor gave a false negative result. Therefore, at least one of

$$\mathsf{Ab}_1(\langle\rangle, \mathsf{cup}, \mathsf{pick}(\mathsf{cup})) \qquad \text{and} \qquad \mathsf{Ab}_2(\langle \mathsf{pick}(\mathsf{cup})\rangle, \mathsf{cup})$$

is true in all accessible situations. Giving a higher weight to Ab_1 and a lower one to Ab_2 makes the more plausible situations those in which the first is false and the second is true. Assigning weights in the opposite way gives the opposite result. Assigning the same weights to each will result in there being most plausible situations in which either one is true. □

So we see that beliefs about the combination of world-altering actions and sensing actions behave in a sensible and controllable way.

4.5.2 Changing beliefs about preconditions

Beliefs about the preconditions of actions can change over time, similarly to what we have already seen for SSAs and sensing axioms.

Example 4.5.3.
Suppose we have an IAAT Σ where Σ_{ssa} includes

$$\mathsf{Poss}(\mathsf{pick}(x), s) \ \equiv\ (\neg\mathsf{Ab}_1(s) \vee \forall y.\ \neg\mathsf{Holding}(y, s)),$$

saying that it's possible to pick up x if either a plausible condition ($\neg\mathsf{Ab}_1$) holds or nothing is held. Furthermore, Σ_{sense} specifies that the sense action senses whether Ab_1 is true,

$$\mathsf{SF}(\mathsf{sense}, s) \equiv \mathsf{Ab}_1(s),$$

and Σ_0 specifies that Ab_1 really is true:

$$\mathsf{Ab}_1(\mathsf{S}_0).$$

It can be seen that the agent will initially believe that it's always possible to execute the pick action (because the agent will assume Ab_1 is false). However, after a sense action the agent will believe that objects can always be picked up just in case nothing is already held (because it will have concluded that Ab_1 is true). The following proposition formalizes this.

Proposition 4.5.3. The IAAT Σ described above entails each of the following:

1. $\rightsquigarrow \mathsf{Poss}(\mathsf{pick}(x), s) \equiv \mathsf{True}$
2. $\mathsf{sense} \rightsquigarrow \mathsf{Poss}(\mathsf{pick}(x), s) \equiv \forall y.\ \neg\mathsf{Holding}(y, s)$

Proof. Initially, the most plausible accessible situations have Ab_1 false in them, but after the sense action all accessible situations have Ab_1 true in them. The result then follows from Proposition 4.2.1. □

In that example, the agent came to believe that a precondition was more restrictive than initially thought – the action can be executed in fewer situations. Coming to believe that a precondition is less restrictive can be handled similarly (consider what happens if you remove the negation before Ab_1 in the example's original SSA).

4.6 Regression

We now turn to considering regression, the syntactic procedure often used in automated reasoning about situation calculus formulas. As described in §2.2.2.4, Pirri and Reiter (1999) showed that a certain class of formulas, the *regressable* formulas, can be rewritten using regression so as not to refer to any non-initial situations (this can make them easier to prove, since some axioms will no longer be needed). Recall that for a basic action theory Σ (Definition 2.2.5) we have that $\Sigma_0 \cup \Sigma_{\mathsf{una}}$ will entail the regression rewriting of a regressable formula iff Σ entails the original formula (Proposition 2.2.2).

When using IAATs, we'll often want to regress formulas referring to belief. For that, it's fairly straight-forward to adapt the approach by Schwering and Lakemeyer (2015) from the modal situation calculus, which involves the use of conditional beliefs. We do so in §4.6.2. (Note that we cannot just use the procedure for regressing formulas referring to *knowledge* from Scherl and Levesque (2003), since we have to take plausibility into account.)

More interestingly, though, we first present a way beliefs about SSAs and other domain dynamics could be taken advantage of in regression. Recall that the essential feature of regression is recursively replacing substitution instances of the left-hand-sides of SSAs with their right-hand-sides. In regression as it's usually considered, the SSAs used are those the axiomatizer wrote. A novel alternative that our work suggests is to use other SSAs that the agent happens to believe at a given time. A computational advantage might be gained in some cases, because some believed SSAs may lead to much smaller or larger regression rewritings than others. To illustrate, an agent could believe both the SSA

$$\mathsf{P}(x, \mathsf{do}(a, s)) \equiv (\mathsf{P}(\mathsf{f}(x), s) \wedge \mathsf{P}(\mathsf{g}(x), s))$$

and the SSA

$$\mathsf{P}(x, \mathsf{do}(a, s)) \equiv \mathsf{P}(x, s).$$

The first SSA's right-hand-side has twice as many atoms as its left-hand-side, so regressing with it could cause an exponential (in the number of applied actions) blowup, while that doesn't happen using the second SSA. For IAATs, the SSAs given by the axiomatizer will often refer to various implausible conditions, and in many situations the agent will believe simpler SSAs.

We will prove (in §4.6.1) that an agent can use a form of regression, working with any set of SSAs (and precondition axioms and sensing axioms) it believes, to reason about its beliefs. Note that here we apply regression to formulas only *within* belief operators. To regress the whole formula, you would need to additionally apply another form of regression – the one that we previously mentioned, described in §4.6.2, which we will call *full regression* because it can be applied to a complete sentence including belief operators.

To illustrate the distinction between regression within beliefs and full regression, suppose that we have an IAAT Σ and we want to regress a sentence

$$\mathbf{Bel}(F(\mathsf{do}(\vec{\beta}, now)), \mathsf{do}(\vec{\alpha}, \mathsf{S}_0))$$

where $\vec{\alpha}$ and $\vec{\beta}$ are sequences of action terms. We have the option to use regression *within* beliefs, using any dynamics axioms that the agent believes, to rewrite that expression as

$$\mathbf{Bel}(\phi, \mathsf{do}(\vec{\alpha}, \mathsf{S}_0))$$

where ϕ is the regression of $F(\mathsf{do}(\vec{\beta}, now))$, and is uniform in *now* (and so no longer refers to the future). Then, we can apply full regression (using the actual dynamics axioms from Σ_{dyn}) to further rewrite that formula to remove the reference to the non-initial situation $\mathsf{do}(\vec{\alpha}, \mathsf{S}_0)$.

We could alternatively just have applied full regression to the original sentence, $\mathbf{Bel}(F(\mathsf{do}(\vec{\beta}, now)), \mathsf{do}(\vec{\alpha}, \mathsf{S}_0))$. However, again, by doing some of the computation with believed SSAs, there could potentially be computational savings. We leave to future work the important question of how to automatically choose a set of believed SSAs for which regression will be more efficient.

4.6.1 Regression within beliefs

Formulas within beliefs typically refer to *now*. To regress them, we will require them to be "*now*-regressable", which we define similarly to *regressable* (Definition 2.2.7).

Definition 4.6.1 (r-regressable). Given a situation term r (e.g., *now*), a first-order formula ϕ is r-regressable if

- for each term of sort situation mentioned by ϕ, the term has the syntactic form $\mathsf{do}(\vec{\alpha}, r)$ where $\vec{\alpha}$ is a sequence of 0 or more action terms
- for each atom of the form $\mathsf{Poss}(\alpha, \sigma)$ or $\mathsf{SF}(\alpha, \sigma)$ mentioned by ϕ, α has the syntactic form $\alpha'(\vec{\tau})$ where α' is an action function symbol
- ϕ does not have quantification over situations
- ϕ does not mention $\sqsubset$ or compare situations for equality
- ϕ does not mention the B predicate.
- ϕ does not mention any functional fluents (this is just for simplicity)

The definition of regression is as follows (based closely on (Reiter, 2001, Definition 4.5.3)).

Definition 4.6.2. Let $\Delta = \Delta_{\mathsf{ssa}} \cup \Delta_{\mathsf{pre}} \cup \Delta_{\mathsf{sense}}$ be a set of sentences including SSAs, precondition axioms, and sensing axioms for all the fluents and actions. Let ϕ be a *now*-regressable formula, where WLOG we assume that any variables appearing in ϕ are distinct from those mentioned by Δ. Then the regression of ϕ with respect to Δ is written $\mathcal{R}_1^{\Delta}[\phi]$ and defined case-by-case as follows:

1. ϕ is a situation-independent atom, or a relational fluent atom of the form $F(\vec{\tau}, now)$. Then $\mathcal{R}_1^{\Delta}[\phi] = \phi$.

2. ϕ is a relational fluent atom $F(\vec{\tau}, \mathsf{do}(\alpha, \sigma))$, where the SSA for F in Δ_{ssa} is

$$F(\vec{x}, \mathsf{do}(a, s)) \equiv \phi_F(\vec{x}, a, s).$$

   Then $\mathcal{R}_1^{\Delta}[\phi] = \mathcal{R}_1^{\Delta}[\phi_F(\vec{\tau}, \alpha, \sigma)]$.

3. ϕ is a formula of the form $\mathsf{Poss}(\alpha(\vec{\tau}), \sigma)$ where α is an action function symbol and the precondition axiom for α in Δ_{pre} is $\mathsf{Poss}(\alpha(\vec{x}), s) \equiv \phi_\alpha(\vec{x}, s)$. Then $\mathcal{R}_1^{\Delta}[\mathsf{Poss}(\alpha(\vec{\tau}), \sigma)] = \mathcal{R}_1^{\Delta}[\phi_\alpha(\vec{\tau}, \sigma)]$.

4. ϕ is a formula of the form $\mathsf{SF}(\alpha(\vec{\tau}), \sigma)$ where α is an action function symbol and the sensing axiom for α in Δ_{sense} is $\mathsf{SF}(\alpha(\vec{x}), s) \equiv \phi_\alpha(\vec{x}, s)$. Then $\mathcal{R}_1^{\Delta}[\mathsf{SF}(\alpha(\vec{\tau}), \sigma)] = \mathcal{R}_1^{\Delta}[\phi_\alpha(\vec{\tau}, \sigma)]$.

5. ϕ is a non-atomic formula. Regression is defined inductively as follows:

$$\begin{aligned}
\mathcal{R}_1^{\Delta}[\neg\phi] &= \neg\mathcal{R}_1^{\Delta}[\phi] \\
\mathcal{R}_1^{\Delta}[\phi_1 \wedge \phi_2] &= \mathcal{R}_1^{\Delta}[\phi_1] \wedge \mathcal{R}_1^{\Delta}[\phi_2] \\
\mathcal{R}_1^{\Delta}[\exists x.\ \phi] &= \exists x.\ \mathcal{R}_1^{\Delta}[\phi]
\end{aligned}$$

This is a traditional regression procedure, with *now* serving the role that S_0 usually plays. It can be shown that regressing a *now*-regressable formula yields a formula uniform in *now*. The next proposition says that an agent can reason using regression using any set of SSAs that it believes, in the following sense: the agent will believe that any *now*-regressable formula is equivalent to its regression with respect to those SSAs.

Proposition 4.6.1. Let $\Delta = \Delta_{\mathsf{ssa}} \cup \Delta_{\mathsf{pre}} \cup \Delta_{\mathsf{sense}}$ be any set of sentences including SSAs, precondition axioms, and sensing axioms for all the fluents and actions. Suppose that σ^* is a ground situation term such that

$$\Sigma \models \mathbf{Bel}\big(\bigwedge \Delta{:}now,\ \sigma^*\big),$$

i.e., the agent in situation σ^* believes that the axioms in Δ apply to future situations. Then for any *now*-regressable formula ϕ (which WLOG uses distinct variables from Δ),

$$\Sigma \models \mathbf{Bel}\big(\forall(\phi \equiv \mathcal{R}_1^{\Delta}[\phi]),\ \sigma^*\big).$$

Proof. Our proof resembles that of the related (Pirri and Reiter, 1999, Theorem 2). We assign any *now*-regressable formula ϕ a triple of numbers, $\mathsf{index}(\phi) = \langle b, d, c\rangle$, where b is 1 if an atom of the form $\mathsf{Poss}(\alpha, \sigma)$ or $\mathsf{SF}(\alpha, \sigma)$ appears in ϕ (and 0 otherwise), d is the greatest depth of nesting of do functions in ϕ, and c is the number of logical connectives/quantifiers in ϕ. The proof is by induction on $\mathsf{index}(\phi)$, with respect to a lexicographic ordering, which we call $\leq_3$.

1. When its index is $\langle 0, 0, 0\rangle$, ϕ is either a situation-independent atom or a relational fluent atom $F(\vec{\tau}, \mathit{now})$. In either case, $\mathcal{R}_1^{\Delta}[\phi] = \phi$, so the result is trivial.

2. When its index is $\langle 0, d, 0\rangle$ for $d > 0$, ϕ is a relational fluent atom $F(\vec{\tau}, \mathsf{do}(\alpha, \sigma))$. We want to show that Σ entails $\mathbf{Bel}\big(\forall(F(\vec{\tau}, \mathsf{do}(\alpha, \sigma)) \equiv \mathcal{R}_1^{\Delta}[\phi_F(\vec{\tau}, \alpha, \sigma)]), \sigma^*\big)$ where ϕ_F is from the RHS of the SSA for F in Δ_{ssa}. First, because the agent believes that that SSA applies to *now* and its successors (and σ is one of those), we get that

   $$\Sigma \models \mathbf{Bel}\big(\forall(F(\vec{\tau}, \mathsf{do}(\alpha, \sigma)) \equiv \phi_F(\vec{\tau}, \alpha, \sigma)), \sigma^*\big)$$

   It can be seen that $\mathsf{index}(\phi_F(\vec{t}, \alpha, \sigma)) \leq_3 \langle 0, d-1, c\rangle$ for some c, and since

   $$\langle 0, d-1, c\rangle <_3 \langle 0, d, 0\rangle,$$

   by the inductive hypothesis we get that

   $$\Sigma \models \mathbf{Bel}\big(\forall(\phi_F(\vec{\tau}, \alpha, \sigma) \equiv \mathcal{R}_1^{\Delta}[\phi_F(\vec{\tau}, \alpha, \sigma)]), \sigma^*\big)$$

   Since belief is closed under logical consequence we can put this together with the previous entailment to get the result we want.

3. When its index is $\langle 1, d, 0\rangle$, ϕ is an atom either of the form $\mathsf{Poss}(\alpha(\vec{\tau}), \sigma)$ or $\mathsf{SF}(\alpha(\vec{\tau}), \sigma)$. In either case, the regression of ϕ is $\mathcal{R}_1^{\Delta}[\phi_\alpha(\vec{\tau})]$ where ϕ_α comes from the RHS of a precondition or sensing axiom. It can be seen that $\mathsf{index}(\phi_\alpha(\vec{t}, \sigma)) \leq_3 \langle 0, d, c\rangle$ for some c, and $\langle 0, d, c\rangle <_3 \langle 1, d, 0\rangle$. Therefore, this case can be shown similarly to the previous one.

4. When its index is $\langle b, d, c\rangle$ with $c > 0$, ϕ is a non-atomic formula. The result can be seen to follow from the inductive hypothesis and belief being deductively closed. □

So we can preform regression within belief using believed SSAs. Again, this may be advantageous because believed SSAs may be much simpler than the ones written in the action theory. The next section will consider regression outside of beliefs as well.

4.6.2 Fully regressing formulas

To fully regress formulas containing beliefs (and not just regress formulas within beliefs), we adapt the approach by Schwering and Lakemeyer (2015) from the modal situation calculus. This will *not* subsume the previously described procedure $\mathcal{R}_1$, since for full regression we will not in general be able to make use of axioms that are merely believed. Instead, the relation between the two approaches is complementary; we can (optionally) first use $\mathcal{R}_1$ to make formulas within beliefs uniform in *now*, and then apply the full regression procedure, which we'll call $\mathcal{R}_2$, to the entire formula. (Also, $\mathcal{R}_1$ is used as a subprocedure by $\mathcal{R}_2$ in a limited way.)

The main result of this section is Proposition 4.6.4, which is a version of the regression theorem (Proposition 2.2.2) that applies to IAATs. To get there, following Schwering and Lakemeyer's approach we make use of *conditional beliefs*. The full regression procedure involves both regressing formulas within conditional beliefs, and regressing formulas which refer to conditional beliefs. We have results for each of those aspects (Lemma 4.6.1 and Lemma 4.6.2, respectively), adapting work by Schwering and Lakemeyer. Finally, to show that not all axioms from an IAAT are needed to entail a fully regressed formula, we make use of another result that we prove, Proposition 4.6.3. (The final result is still not quite as strong as the regression theorem for BATs, as not all second-order components are eliminated, as we will see.)

Recall from §3.4.1 that a conditional belief in ψ given ϕ, which we write as $\mathbf{Bel}(\phi \Rightarrow \psi, s)$, intuitively means that in the most plausible accessible situations from s where ϕ is true, ψ is also true. Belief can be related to conditional belief in the usual way, i.e., $\mathbf{Bel}(\phi, s)$ could equivalently be defined as $\mathbf{Bel}(\mathsf{True} \Rightarrow \phi, s)$. When fully regressing formulas containing beliefs, we will assume that any expression of the form $\mathbf{Bel}(\phi, \sigma)$ has been replaced with $\mathbf{Bel}(\mathsf{True} \Rightarrow \phi, \sigma)$.

Let's consider regression *within* conditional beliefs (this will be a part of the full regression procedure). It turns out that we can use the regression operator $\mathcal{R}_1$ that we previously defined within conditional beliefs, though unlike in Proposition 4.6.1 it will not suffice for the agent to just *believe* the dynamics axioms Δ used by regression,

because in the most plausible accessible situations where the conditional's antecedent is true, merely believed axioms may not hold. It will suffice, though, to use axioms Δ that the agent is *certain* of, in that they hold in all accessible situations (not just the most plausible). Recall that $\mathbf{Know}(\phi, s)$ is true if ϕ is true in all situations accessible from s (Equation 2.10). Therefore, we can use **Know** to indicate what the agent is certain of (however, we are not assuming what is "known" must be true). Lemma 4.6.1 below shows how regression within conditional beliefs can be performed using known dynamics axioms.

Lemma 4.6.1. Let Σ be an IAAT and $\Delta = \Delta_{\mathsf{ssa}} \cup \Delta_{\mathsf{pre}} \cup \Delta_{\mathsf{sense}}$ a set of axioms such that

$$\Sigma \models \mathbf{Know}\left(\bigwedge \Delta{:}now, \mathsf{S}_0\right).$$

Then for any *now*-regressable formulas ψ_1 and ψ_2 using distinct variables from Δ,

$$\Sigma \models \forall[\mathbf{Bel}\left(\psi_1 \Rightarrow \psi_2, \mathsf{S}_0\right) \equiv \mathbf{Bel}\left(\mathcal{R}_1^{\Delta}\left[\psi_1\right] \Rightarrow \mathcal{R}_1^{\Delta}\left[\psi_2\right], \mathsf{S}_0\right)].$$

Proof. The key is to note that it would suffice to show that

$$\Sigma \models \forall[\mathbf{Know}((\psi_1 \equiv \mathcal{R}_1^{\Delta}\left[\psi_1\right]) \wedge (\psi_2 \equiv \mathcal{R}_1^{\Delta}\left[\psi_2\right]), \mathsf{S}_0)].$$

This is because that would mean that the most plausible accessible situations where ψ_1 is true are exactly the most plausible accessible situations where $\mathcal{R}_1^{\Delta}\left[\psi_1\right]$ is true, and whether ψ_2 is true at those situations is equivalent to whether $\mathcal{R}_1^{\Delta}\left[\psi_2\right]$ is true at those situations. The proof is similar to that of Proposition 4.6.1 but substitutes **Know** for **Bel**. □

In Lemma 4.6.1, we considered conditional beliefs only in S_0, because that's all we'll need for the role that $\mathcal{R}_1$ plays within the broader procedure $\mathcal{R}_2$ that we're going to define.

For $\mathcal{R}_2$ we need to establish how conditional beliefs in a situation are related to the previous situation. Schwering and Lakemeyer (2015, Theorem 5) described this, and we adapt their result below. Note that Lakemeyer and Levesque (2011, Theorem 4) had earlier presented a similar result about how *knowledge* in a situation is related to the previous situation.

Lemma 4.6.2. For any IAAT Σ and *now*-regressable formulas ψ_1 and ψ_2,

$$\begin{aligned}\Sigma \models \forall a, s.\ \mathbf{Bel}\,(\psi_1 \Rightarrow \psi_2, \mathsf{do}(a, s)) \equiv \\ \big([\mathsf{SF}(a, s) \wedge \mathbf{Bel}\,\big(\chi^+(a) \Rightarrow \psi_2[\mathsf{do}(a, now)], s\big)] \vee \\ [\neg\mathsf{SF}(a, s) \wedge \mathbf{Bel}\,\big(\chi^-(a) \Rightarrow \psi_2[\mathsf{do}(a, now)], s\big)]\big)\end{aligned}$$

where

- $\chi^+(a)$ abbreviates $\mathsf{SF}(a, now) \wedge \mathsf{Poss}(a, now) \wedge \psi_1[\mathsf{do}(a, now)]$, and
- $\chi^-(a)$ abbreviates $\neg\mathsf{SF}(a, now) \wedge \mathsf{Poss}(a, now) \wedge \psi_1[\mathsf{do}(a, now)]$.

Proof. For readability in this proof, let's introduce the abbreviations

$$\begin{aligned}\mathsf{C}(\phi, s', s) &\stackrel{\text{def}}{=} \mathsf{B}(s', s) \wedge \phi[s'] \\ \mathsf{MPC}(\phi, s', s) &\stackrel{\text{def}}{=} \mathsf{C}(\phi, s', s) \wedge \forall s''.\ \mathsf{C}(\phi, s'', s) \supset s' \leq_{\mathsf{pl}} s''\end{aligned}$$

That is, $\mathsf{MPC}(\phi, s', s)$ means that s' is one of the most plausible accessible situations from s where ϕ is true. Observe that $\mathbf{Bel}\,(\psi_1 \Rightarrow \psi_2, s)$ expands to the same thing as $\forall s'.\ \mathsf{MPC}(\psi_1, s', s) \supset \psi_2[s']$.

Now consider any model $\mathfrak{I}$ of Σ and an arbitrary variable assignment μ. We'll assume that

$$\mathfrak{I}, \mu \models \mathsf{SF}(a, s)$$

(the other case is symmetric). Then what we want to show is that

$$\begin{aligned}\mathfrak{I}, \mu \models \mathbf{Bel}\,(\psi_1 \Rightarrow \psi_2, \mathsf{do}(a, s)) \equiv \\ \mathbf{Bel}\,(\mathsf{SF}(a, now) \wedge \mathsf{Poss}(a, now) \wedge \psi_1[\mathsf{do}(a, now)] \Rightarrow \psi_2[\mathsf{do}(a, now)], s)\,.\end{aligned}$$

To establish that, it will suffice to show that

$$\begin{aligned}\mathfrak{I}, \mu \models \mathsf{MPC}(\psi_1, \mathsf{do}(a, s'), \mathsf{do}(a, s)) \equiv \\ \mathsf{MPC}(\mathsf{SF}(a, now) \wedge \mathsf{Poss}(a, now) \wedge \psi_1[\mathsf{do}(a, now)], s', s)\end{aligned} \tag{4.7}$$

(note that there can't be situations accessible from the situation denoted by $\mathsf{do}(a, s)$ where the action denoted by a has not just occurred).

Because $\mathfrak{I}$ satisfies SSA for B (Equation 2.11), it's easy to see that

$$\mathfrak{I}, \mu \models \mathsf{B}(\mathsf{do}(a, s'), \mathsf{do}(a, s)) \equiv \mathsf{B}(s', s) \wedge \mathsf{SF}(a, s') \wedge \mathsf{Poss}(a, s')$$

Therefore, we can conjoin $\psi_1[\mathsf{do}(a, s')]$ to both sides of the equivalence, yielding

$$\begin{aligned}\mathfrak{I}, \mu \models \big(\mathsf{B}(\mathsf{do}(a, s'), \mathsf{do}(a, s)) \wedge \psi_1[\mathsf{do}(a, s')]\big) \equiv \\ \big(\mathsf{B}(s', s) \wedge \mathsf{SF}(a, s') \wedge \mathsf{Poss}(a, s') \wedge \psi_1[\mathsf{do}(a, s')]\big).\end{aligned}$$

Observe that that can be rewritten as

$$\mathfrak{I}, \mu \models \mathsf{C}(\psi_1, \mathsf{do}(a, s'), \mathsf{do}(a, s)) \equiv \mathsf{C}(\mathsf{SF}(a, now) \wedge \mathsf{Poss}(a, now) \wedge \psi_1[\mathsf{do}(a, now)], s', s).$$

The desired result (Equation 4.7) then follows from the plausibility level of a situation not changing as a result of doing an action. □

Now we are almost ready to describe the full regression procedure, $\mathcal{R}_2$. First, we define a class of formula that can be fully regressed. Note that, for simplicity, we're not allowing nested beliefs.

Definition 4.6.3 (fully-regressable). A formula ϕ is *fully-regressable* if the following hold:

- for each term of sort situation mentioned by ϕ (outside beliefs), the term has the syntactic form $\mathsf{do}(\vec{\alpha}, \mathsf{S}_0)$ where $\vec{\alpha}$ is a sequence of 0 or more action terms
- for each atom of the form $\mathsf{Poss}(\alpha, \sigma)$ or $\mathsf{SF}(\alpha, \sigma)$ mentioned by ϕ, α has the syntactic form $\alpha'(\vec{\tau})$ where α' is an action function symbol
- ϕ does not have quantification over situations (except in the expansions of conditional beliefs)
- ϕ does not mention $\sqsubset$ or compare situations for equality
- the only uses of B or second-order quantification in ϕ are in the expansions of conditional beliefs
- for any expression $\mathbf{Bel}(\psi_1 \Rightarrow \psi_2, \sigma)$ appearing in ϕ,
  - ψ_1 and ψ_2 are *now*-regressable

  - not only is σ of the form $\mathsf{do}(\vec{\alpha}, \mathsf{S}_0)$, but each action term α in the sequence $\vec{\alpha}$ has the syntactic form $\alpha'(\vec{\tau})$ where α' is an action function symbol[2]

- ϕ does not refer to functional fluents (as with *now*-regressable formulas, this is just for simplicity)

We can now describe the regression procedure, which is much like that from Schwering and Lakemeyer (2015). Note that the first five cases are analogues of the ones in Definition 4.6.2.

Definition 4.6.4. Let $\Gamma = \Gamma_{\mathsf{ssa}} \cup \Gamma_{\mathsf{pre}} \cup \Gamma_{\mathsf{sense}}$ be a set of sentences including SSAs, precondition axioms, and sensing axioms for all the fluents and actions. The (full) regression of ϕ with respect to Γ, where ϕ is fully-regressable (and uses distinct variables from Γ), is written $\mathcal{R}_2^\Gamma[\phi]$ and defined case-by-case as follows:

1. ϕ is a situation-independent atom, or a relational fluent atom of the form $F(\vec{\tau}, \mathsf{S}_0)$. Then $\mathcal{R}_2^\Gamma[\phi] = \phi$.

2. ϕ is a relational fluent atom $F(\vec{\tau}, \mathsf{do}(\alpha, \sigma))$, where the SSA for F in Γ is

$$F(\vec{x}, \mathsf{do}(a, s)) \equiv \phi_F(\vec{x}, a, s).$$

   Then $\mathcal{R}_2^\Gamma[\phi] = \mathcal{R}_2^\Gamma[\phi_F(\vec{\tau}, \alpha, \sigma)]$.

3. ϕ is a formula of the form $\mathsf{Poss}(\alpha(\vec{\tau}), \sigma)$ where α is an action function symbol and the precondition axiom for α in Γ is $\mathsf{Poss}(\alpha(\vec{x}), s) \equiv \phi_\alpha(\vec{x}, s)$. Then $\mathcal{R}_2^\Gamma[\mathsf{Poss}(\alpha(\vec{\tau}), \sigma)] = \mathcal{R}_2^\Gamma[\phi_\alpha(\vec{\tau}, \sigma)]$.

4. ϕ is a formula of the form $\mathsf{SF}(\alpha(\vec{\tau}), \sigma)$ where α is an action function symbol and the sensing axiom for α in Γ is $\mathsf{SF}(\alpha(\vec{x}), s) \equiv \phi_\alpha(\vec{x}, s)$. Then $\mathcal{R}_2^\Gamma[\mathsf{SF}(\alpha(\vec{\tau}), \sigma)] = \mathcal{R}_2^\Gamma[\phi_\alpha(\vec{\tau}, \sigma)]$

5. ϕ is a non-atomic formula. Regression is defined inductively as usual:

$$\mathcal{R}_2^\Gamma[\neg\phi] = \neg\mathcal{R}_2^\Gamma[\phi]$$
$$\mathcal{R}_2^\Gamma[\phi_1 \wedge \phi_2] = \mathcal{R}_2^\Gamma[\phi_1] \wedge \mathcal{R}_2^\Gamma[\phi_2]$$
$$\mathcal{R}_2^\Gamma[\exists x.\ \phi] = \exists x.\ \mathcal{R}_2^\Gamma[\phi]$$

[2]This will ensure that when regressing conditional beliefs, which produces atoms using Poss and SF, which precondition/sensing axioms are relevant to regress those atoms can be determined. Schwering and Lakemeyer (2015) do not require this, but in their theories write precondition and sensing axioms differently, so that there is only one precondition axiom and one sensing axiom for all actions.

6. ϕ is a formula of the form $\mathbf{Bel}(\psi_1 \Rightarrow \psi_2, \mathsf{do}(\alpha, \sigma))$. Let $\beta(a, s)$ abbreviate the following expression from Lemma 4.6.2:

$$[\mathsf{SF}(a, s) \wedge \mathbf{Bel}\left(\chi^+(a) \Rightarrow \psi_2[\mathsf{do}(a, now)], s\right)] \vee$$
$$[\neg\mathsf{SF}(a, s) \wedge \mathbf{Bel}\left(\chi^-(a) \Rightarrow \psi_2[\mathsf{do}(a, now)], s\right)].$$

Then

$$\mathcal{R}_2^\Gamma[\mathbf{Bel}(\psi_1 \Rightarrow \psi_2, \mathsf{do}(\alpha, \sigma))] = \mathcal{R}_2^\Gamma[\beta(\alpha, \sigma)]$$

7. ϕ is a formula of the form $\mathbf{Bel}(\psi_1 \Rightarrow \psi_2, \mathsf{S}_0)$. Then

$$\mathcal{R}_2^\Gamma[\mathbf{Bel}(\psi_1 \Rightarrow \psi_2, \mathsf{S}_0)] = \mathbf{Bel}\left(\mathcal{R}_1^\Gamma[\psi_1] \Rightarrow \mathcal{R}_1^\Gamma[\psi_2], \mathsf{S}_0\right)$$

where $\mathcal{R}_1$ is the regression operator from Definition 4.6.2.

Proposition 4.6.2. Suppose that Σ is an IAAT. For any fully-regressable formula ϕ (not sharing variables with Σ_{dyn}), $\Sigma \models \forall(\phi \equiv \mathcal{R}_2^{\Sigma_{\mathsf{dyn}}}[\phi])$.

Proof. This can be proved by induction. The correctness of case (6) can be shown using Lemma 4.6.2. Observe that $\beta(\alpha, \sigma)$ will be a fully regressable formula, because α will be of the form $\alpha'(\vec{\tau})$ where α' is an action function symbol. For case (7), the result follows from Lemma 4.6.1. □

Note that while the result of case (6) is a complicated-looking expression, the number of actions referred to by situation terms outside of belief is reduced (the number of actions referred to by situation terms *inside* beliefs may be increased, but those can later be removed through applications of case (7)). It can be shown that the result of full regression (on a fully-regressable formula) will be a formula where all the situation terms outside of conditional beliefs are S_0, and all the ones inside are *now*. If there are conditional beliefs, it will not be a formula uniform in S_0; we will instead call it *quasi-uniform* in S_0. This is defined below.

Definition 4.6.5 (quasi-uniform). A situation calculus formula ϕ is quasi-uniform in a situation term σ if ϕ satisfies the conditions of being uniform in σ with the exception that ϕ can include subformulas of the form $\mathbf{Bel}(\psi_1 \Rightarrow \psi_2, \sigma)$, where ψ and ψ_2 are uniform in *now*.

We'll conclude by showing how not all axioms from the theory are needed to entail the regressed formula – similarly to in the regression theorem for basic action theories

(Proposition 2.2.2). In proving the regression theorem, Pirri and Reiter used an intermediate result, the "relative satisfiability" of BATs (Pirri and Reiter, 1999, Theorem 1). We will prove a similar result for IAATs – actually, we will need it for a slightly broader class of action theories, that we will call *quasi*-IAATs.

Definition 4.6.6 (quasi-IAAT). A quasi-IAAT is an action theory like an IAAT except that Σ_0 is only required to be quasi-uniform in S_0, rather than uniform in S_0. (Note that every IAAT is also a quasi-IAAT.)

Let's name the conjunction of the axiom specifying what initial situations exist and the sentence $\mathsf{Init}(\mathsf{S}_0)$ as "initials". We now prove a version of relative satisfiability for (quasi-)IAATs.

Proposition 4.6.3 (Relative satisfiability for quasi-IAATs). An quasi-IAAT Σ is satisfiable iff $\Sigma_0 \cup \Sigma_{\mathsf{una}} \cup \{\mathbf{OKnow}(\bigwedge \Sigma_{\mathsf{KB}}, \mathsf{S}_0)\} \cup \{\text{initials}\}$ is satisfiable.

Proof. Given a model of $\Sigma_0 \cup \Sigma_{\mathsf{una}} \cup \{\mathbf{OKnow}(\bigwedge \Sigma_{\mathsf{KB}}, \mathsf{S}_0)\} \cup \{\text{initials}\}$, a model of Σ can be constructed. The proof is similar to the analogous result for basic action theories (Pirri and Reiter, 1999, Theorem 1). The most significant difference is that, unlike with BATs, there are multiple initial situations to deal with, and so the domain of situations that we construct is different. We also have to deal with interpreting the B and SF predicates.

Suppose $\mathfrak{I}_0 = \langle \mathcal{D}_0, \mathcal{I}_0 \rangle$ is a model of $\Sigma_0 \cup \Sigma_{\mathsf{una}} \cup \{\mathbf{OKnow}(\bigwedge \Sigma_{\mathsf{KB}}, \mathsf{S}_0)\} \cup \{\text{initials}\}$, where the domain $\mathcal{D}_0$ is the disjoint union of the domain of situations $\mathcal{D}_0^S$, domain of actions $\mathcal{D}_0^A$, and domain of objects $\mathcal{D}_0^O$. Then we construct a model $\mathfrak{I} = \langle \mathcal{D}, \mathcal{I} \rangle$ of Σ where $\mathcal{D}$ is the disjoint union of the domain of situations $\mathcal{D}_S$ (defined below) and the same domains of actions and of objects from $\mathcal{D}_0$.

The domain of situations $\mathcal{D}^S$ we construct is the smallest set such that the following holds: for every $\hat{s} \in \mathcal{D}_0^S$ such that $\hat{s}$ is an initial situation according to $\mathfrak{I}_0$ (i.e., such that $\mathcal{I}_0[\mathsf{do}](\hat{a}, \hat{s}') \neq \hat{s}$ for all $\hat{a} \in \mathcal{D}_0^A$ and $\hat{s}' \in \mathcal{D}_0^S$), and for every finite (possible empty) sequence $\langle \hat{a}_1, \ldots, \hat{a}_k \rangle$ of elements each from $\mathcal{D}_0^A$, the sequence $\langle \hat{s}, \hat{a}_1, \ldots, \hat{a}_k \rangle$ is an element of $\mathcal{D}^S$.

We define interpretations of the symbols S_0, $\sqsubset$, do, and root as follows:

$$\mathcal{I}[\mathsf{S}_0] = \langle \mathcal{I}_0[\mathsf{S}_0] \rangle$$

$$\mathcal{I}[\sqsubset] = \{\langle \hat{s}, \hat{s}' \rangle \in \mathcal{D}^S \times \mathcal{D}^S : \hat{s} \text{ is a proper initial subsequence of } \hat{s}'\}$$

$$\mathcal{I}[\mathsf{do}](\hat{a}_{k+1}, \langle \hat{s}, \hat{a}_1, \ldots, \hat{a}_k \rangle) = \langle \hat{s}, \hat{a}_1, \ldots, \hat{a}_n, \hat{a}_{k+1} \rangle, \text{ for each } \langle \hat{s}, \hat{a}_1, \ldots, \hat{a}_k \rangle \in \mathcal{D}^S \text{ and } \hat{a}_{k+1} \in \mathcal{D}_0^A$$

$$\mathcal{I}[\mathsf{root}](\langle \hat{s}, \hat{a}_1, \ldots, \hat{a}_k \rangle) = \hat{s}, \text{ for each } \langle \hat{s}, \hat{a}_1, \ldots, \hat{a}_k \rangle \in \mathcal{D}^S$$

It can be seen $\mathfrak{I}$ therefore satisfies all of the foundational axioms, other than the one regarding the existence of initial situations with all combinations of fluent values (we have not yet specified how to interpret fluent values with $\mathfrak{I}$).

Next, we define $\mathfrak{I}$ to agree with $\mathfrak{I}_0$ on the interpretations of non-fluent predicates and functions. Therefore, $\mathfrak{I}$ will satisfy Σ_{una}, since $\mathfrak{I}_0$ does.

We also define $\mathfrak{I}$ so that for each initial situation $\langle \hat{s} \rangle \in \mathcal{D}^S$, relational and functional fluents (which do not include the special B predicate) take the same value there as $\mathfrak{I}_0$ gives the fluents in the situation $\hat{s} \in \mathcal{D}_0^S$. We now can show that $\mathfrak{I}$ satisfies the axiom about existence of initial situations – its initial situations are in a one-to-one correspondence with those of $\mathfrak{I}_0$, and get the same fluent values as their counterparts there. So all the foundational axioms have been satisfied at this stage of the construction.

Next, we define the interpretation of B in S_0 so that the accessible situations are exactly the (counterparts of) the situations accessible from S_0 according to $\mathfrak{I}_0$. We thereby get that $\mathfrak{I}$ satisfies $\{\mathbf{OKnow}(\bigwedge \Sigma_{\mathsf{KB}}, \mathsf{S}_0)\}$, since $\mathfrak{I}_0$ does.

Also, $\mathfrak{I}$ satisfies Σ_0 (since $\mathfrak{I}_0$ does), so it only remains to complete the construction of $\mathfrak{I}$ so that Σ_{dyn} (i.e., $\Sigma_{\mathsf{ssa}} \cup \Sigma_{\mathsf{pre}} \cup \Sigma_{\mathsf{sense}}$) is satisfied. This proceeds similarly to (Pirri and Reiter, 1999, Theorem 1). We first determine the interpretations of Poss and SF in initial situations:

- How Poss should be interpreted in an initial situation can be determined from the precondition axioms and the values of fluents there. Because $\mathfrak{I}_0$ satisfies Σ_{una}, the precondition axioms cannot contradict each other. A complication that Pirri and Reiter note is that there may be actions in the domain which do not correspond to any action function symbol (and therefore aren't covered by any precondition axiom). Their approach was to assume that these actions are always possible.

- Pirri and Reiter did not have to deal with the SF predicate (which does not appear in BATs), but its interpretation in an initial situation can be constructed exactly analogously to Poss's, using sensing axioms instead of precondition axioms.

Finally, the interpretations of fluents, B, Poss, and SF in non-initial situations are constructed inductively. Suppose we have interpreted fluents, B, Poss, and SF in situations in which k actions have been performed. We then can interpret them in situations in which $k+1$ actions have been performed as follows:

- Using the SSAs, the values of fluents (and B) in a situation in which $k+1$ actions have been performed will be uniquely determined by its predecessor situation, in which k actions have been performed.

- The values of Poss and SF in non-initial situations will be determined analogously to in initial situations.

This construction leads to Σ_{dyn} being satisfied, and completes the proof. □

Finally, we can prove our version of the regression theorem for IAATs.

Proposition 4.6.4. Suppose that Σ is an IAAT. For any fully-regressable sentence ϕ (not sharing variables with Σ_{dyn}),

$$\Sigma \models \phi \qquad \text{iff} \qquad \Sigma_0 \cup \Sigma_{\mathsf{una}} \cup \{\mathbf{OKnow}(\bigwedge \Sigma_{\mathsf{KB}}, \mathsf{S}_0)\} \cup \{\text{initials}\} \models \mathcal{R}_2^{\Sigma_{\mathsf{dyn}}}[\phi]\,.$$

Proof. Observe that $\Sigma \models \phi$ iff $\Sigma \cup \{\neg\phi\}$ is unsatisfiable, which holds iff $\Sigma \cup \{\neg\mathcal{R}_2^{\Sigma_{\mathsf{dyn}}}[\phi]\}$ is unsatisfiable, which (by Proposition 4.6.3) holds iff

$$\Sigma_0 \cup \Sigma_{\mathsf{una}} \cup \{\mathbf{OKnow}(\bigwedge \Sigma_{\mathsf{KB}}, \mathsf{S}_0)\} \cup \{\text{initials}\} \cup \{\neg\mathcal{R}_2^{\Sigma_{\mathsf{dyn}}}[\phi]\}$$

is unsatisfiable. The reason Proposition 4.6.3 applies is that $\Sigma_0 \cup \{\neg\mathcal{R}_2^{\Sigma_{\mathsf{dyn}}}[\phi]\}$ is quasi-uniform in S_0 and so could be the set of initial state axioms in a quasi-IAAT. □

So regression removes the need for further use of the dynamics axioms, as with basic action theories.

Discussion

Proposition 4.6.4 is not quite as easy to make practical use of as the regression theorem for basic action theories. Recall that no second-order reasoning is needed to determine if a BAT entails a regressable sentence. In contrast, applying Proposition 4.6.4 to a reasoning problem still leaves some second-order components. In particular,

- The "initials" axiom regarding what initial situations exist is second-order.
- The fully-regressed sentence can refer to conditional beliefs, which are abbreviations for second-order expressions (that involve counting abnormalities).

Note that we cannot just dispense with the "initials" axiom after regression, as the following proposition shows.

Proposition 4.6.5. There exists a quasi-IAAT Σ such that

$$\Sigma_0 \cup \Sigma_{\mathsf{una}} \cup \{\mathbf{OKnow}(\bigwedge \Sigma_{\mathsf{KB}}, \mathsf{S}_0)\} \cup \{\text{initials}\}$$

is not satisfiable, but $\Sigma_0 \cup \Sigma_{\mathsf{una}} \cup \{\mathbf{OKnow}(\bigwedge \Sigma_{\mathsf{KB}}, \mathsf{S}_0)\}$ is satisfiable.

Proof. Consider a quasi-IAAT Σ where $\Sigma_{\mathsf{KB}} = \{\mathsf{True}\}$ and

$$\Sigma_0 = \{\mathbf{Bel}\,(\mathsf{True} \Rightarrow \mathsf{P}(now), \mathsf{S}_0)\}$$

(and Σ_{una} is arbitrary). The "initials" axioms requires that there be initial situations where P and all abnormalities are false, and $\mathbf{OKnow}(\mathsf{True}, \mathsf{S}_0)$ requires those situations to be accessible from the situation denoted by S_0. That contradicts Σ_0, which requires that P be true in all the most plausible accessible situations from the situation denoted by S_0. However, $\Sigma_0 \cup \Sigma_{\mathsf{una}} \cup \{\mathbf{OKnow}(\bigwedge \Sigma_{\mathsf{KB}}, \mathsf{S}_0)\}$ can be seen to be satisfiable (e.g., with a model in which there is only one initial situation, the denotation of S_0, where P is true). □

In the modal situation calculus, Schwering and Lakemeyer (2015, §5) were able to reduce entailments about conditional beliefs to entailments about objective formulas (though they relied on there being finitely many plausibility levels). We leave to future work how to do something similar for IAATs.

Another topic for future work is how to accomplish a greater share of the reasoning process using believed SSAs instead of the ones written in the theory. It's worth observing that we could get a similar result to Proposition 4.6.1 with a variant of the regression algorithm $\mathcal{R}_1$ that applied to $\mathsf{root}(now)$-regressable formulas within beliefs, rewriting them into formulas uniform in $\mathsf{root}(now)$. (We would need not just Δ:now but Δ:$\mathsf{root}(now)$ to be believed at the start, where Δ are the dynamics axioms used by the algorithm.) How would that be useful? Suppose that we want to know whether an IAAT entails a formula like

$$\mathbf{Bel}\big(F(\mathsf{do}(\vec{\beta}, now)), \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)\big).$$

We could rewrite that formula as $\mathbf{Bel}\big(F(\mathsf{do}([\vec{\alpha}, \vec{\beta}], \mathsf{root}(now))), \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)\big)$ (since the agent knows what actions have occurred) and then regress $F(\mathsf{do}([\vec{\alpha}, \vec{\beta}], \mathsf{root}(now)))$, which is $\mathsf{root}(now)$-regressable, to get an expression $\mathbf{Bel}(\phi, \mathsf{do}(\vec{\alpha}, \mathsf{S}_0))$ where ϕ is uniform in $\mathsf{root}(now)$. Intuitively, compared to if we had regressed $F(\mathsf{do}(\vec{\beta}, now))$ to get a formula uniform in now, this might leave less work for a subsequent full regression procedure (so the SSAs written in the theory, as opposed to believed SSAs, would get used less). That's because within belief we've already regressed through all the actions in $\vec{\alpha}$ and $\vec{\beta}$, instead of just $\vec{\beta}$.

Finally, another thing to note is that formulas that are $\mathsf{root}(now)$-regressable are much more expressive than now-regressable ones. While now-regressable expressions can

only talk about the present and future, $\mathsf{root}(now)$-regressable expressions can also talk about the past and counterfactual action histories. For example, we might be interested in whether the agent believes, after performing action α_1, whether F would have been true had action α_2 been performed instead, i.e., whether the action theory entails

$$\mathbf{Bel}\big(F(\mathsf{do}(\alpha_2, \mathsf{root}(now))), \mathsf{do}(\alpha_1, \mathsf{S}_0)\big).$$

Through regression that could be transformed into the question of whether the theory entails

$$\mathbf{Bel}\big(\phi, \mathsf{do}(\alpha_1, \mathsf{S}_0)\big)$$

where ϕ is uniform in $\mathsf{root}(now)$.

4.7 Discussion and related work

Past approaches to belief revision in the situation calculus have supported having SSAs describing conditional effects and the agent revising its beliefs about when those conditions hold. For instance, Schwering et al. (2017, §4.2) gave an example where there is an SSA saying that dropping fragile objects breaks them, and the agent revises its beliefs about whether a particular object is fragile. However, the effect of such revisions on what SSAs the agent believes was not discussed (and so neither was regression with SSAs that the agent believes but were not written by the axiomatizer).

Delgrande and Levesque (2013) considered actions which could fail (and non-deterministic actions more generally). Their formalization (also in the situation calculus) was rather different from ours, as the failure of an action was represented by the agent "intending" to execute one action but actually executing another. Fang and Liu (2013) similarly had an approach, in a multi-agent setting, where agents could be uncertain about what actions had occurred. These works did not discuss having the agent generalize from past failures to reach new conclusions about future action behavior.

A limitation of our approach is that beliefs about domain dynamics are only changed in response to observations of the present state, as opposed to in response to being given arbitrary facts about dynamics, such as you might read in a physics textbook or a fantasy story. For propositional languages, there has been some work about revising or contracting by beliefs about dynamics (e.g., Herzig et al., 2006; Eiter et al., 2010; Varzinczak, 2010; Van Zee et al., 2015). However, they have not usually been concerned with how to specify the generality of conclusions the agent should draw. An exception may be Eiter

et al. (2007, 2010), who describe how a preference order can be defined on propositional transition diagrams by valuing a diagram as the weighted sum of the "query" formulas it entails (Eiter et al., 2007, §4.2). The queries are written in a propositional temporal-logic-like language. It appears this approach could describe preferences on how general of effects action have. However, unlike our work theirs is in a propositional setting and there are no sensing actions.

Another limitation of our approach is that the generalizations the agent can draw from observations have to be specified in advance, as opposed to being determined by some general inductive principles (e.g., for the example in §4.4, the theory had to explicitly identify the possibility of one-time exceptions, that objects could be exceptional, and that slippery objects could be an exceptional class). In contrast, research in *inductive logic programming (ILP)* (Muggleton and de Raedt, 1994; De Raedt, 2017) has dealt with the problem of inducing general first-order rules given examples. ILP has been applied to learning event calculus theories (e.g., Moyle and Muggleton, 1997; Katzouris et al., 2019), and also to learning action models in the field of *relational reinforcement learning* (e.g., Walker et al., 2007; Rodrigues et al., 2010). On the other hand, we have focused on providing a way for the axiomatizer to precisely and explicitly control the plausibility assigned to different possible dynamics.

Working within the event calculus, Mueller (2006, Chapter 12) used abnormality predicates within descriptions of the environment dynamics, so as to model phenomena like default effects and default events. That was not combined with explicitly modelling belief or belief revision, though.

Britz and Varzinczak (2018) distinguish in an example between two reasons a light might fail to turn on, "either because the light bulb is blown (the current situation is abnormal) or because an overcharge resulted from switching the light (the action behaves abnormally)." In our framework, we would represent both cases as abnormal situations (with the latter using an abnormality fluent that also takes as arguments the action and history, so as to treat overcharges like "one-time exceptions").

4.8 Conclusion

People can change their beliefs about how the world works, and this is a desirable property for artificial agents as well. In this chapter, we have shown how changes of beliefs about SSAs, precondition axioms, and sensing axioms can be modelled using IAATs. We described several patterns for writing SSAs that refer to abnormalities, to allow for more general or less general changes of belief in response to unexpected observations. We have

also shown how beliefs about domain dynamics can be incorporated in regression, raising the prospect of computational consequences.

As has been mentioned, with IAATs the original dynamics axioms from the theory will always be believed (though others may also be). If it were desired to have the agent not believe the actual dynamics, the approach of this chapter could be adapted to be used with DIAATs (§3.5.2) instead.

We've assumed that the agent always knows what actions have occurred. However, it would be natural for the agent to also change its beliefs about that. For example, perhaps the reason it's not now holding the cup is that someone else took it. We did consider unobserved exogenous actions in §3.5.1. The next chapter will consider a more general epistemic accessibility relation, allowing for varying degrees of information about the actions that have occurred, and also considers using a program to describe what can exogenously occur.

Chapter 5

Environment processes and knowing how

5.1 Introduction

In the previous chapters, we've seen how we can use abnormalities in theories to express plausibility of initial state properties and domain dynamics. In this chapter, we consider another aspect of the environment that an agent can have plausible beliefs about, the exogenous processes that are occurring around them. (We did previously briefly consider exogenous actions in §3.5.1.)

We present a modified version of belief, where plausibility is still taken into account by counting abnormalities as before, but where accessible situations are constrained to be ones reachable through the execution of a program. Furthermore, we also allow for actions that aren't observed by the agent, following an approach by Kelly and Pearce (2015) (they had also suggested using a program as we are as future work). The resulting new type of action theory, which we call *programmed action theories* (PATs), allow for easily representing beliefs about what happens in the environment (at a potentially longer time-scale than just single-step transitions). The program is written in the ConGolog programming language, a standard language for use with the situation calculus (see §2.2.2.5). ConGolog programs can be non-deterministic, giving one way to represent uncertainty about the various things that are happening concurrently in the environment.

We also give a formalization of *knowing how* to achieve goals in such a setting, generalizing a definition by Lespérance et al. (2000) to take exogenous processes into account. Since our model of belief incorporates a notion of plausibility, we allow for beliefs (including beliefs about how to achieve a goal) to be revised when things are seen by the

Figure 5.1: The fox-chicken-grain problem, after the farmer has carried the chicken north across the river.

agent to change in unexpected ways.

We will illustrate our approach using a version of the classic fox-chicken-grain problem (Ascher, 1990) where a farmer is trying to transport a fox, a chicken, and some grain across a river one at a time (see Figure 5.1).[1] A solution cannot allow either the fox and chicken or the chicken and the grain to be left alone together (because the fox may eat the chicken, and the chicken may eat the grain). The problem is usually formalized in a way that does not explicitly represent what the chicken and fox are doing. We will show how their actions can be modelled, as well as how certain unexpected events that occur while the farmer is solving the problem (including bad weather) can cause the farmer to change his beliefs about whether the goal is achievable at all, or in some cases, whether the goal is still achievable but with a modified plan.

This chapter is structured as follows. In §5.2, we describe our model of belief with its program-based accessibility relation. In §5.3 we define our new form of knowing-how in terms of belief, and prove some formal properties of our approach in §5.3.4. We formalize the fox-chicken-grain problem in §5.4, and also consider an example requiring a potentially unbounded number of actions in §5.5. We discuss related work in §5.6 before concluding.

5.2 Belief in the presence of exogenous processes

In this section we present our model of belief that takes into account exogenous actions taking place according to a program. We first describe how exogenous processes are represented, then how we incorporate them into the definition of our new accessibility relation for belief.

We will suppose that our language includes a predicate $\mathsf{Exo}(a)$ to identify which

[1]The emoji in this chapter are from the Twitter Emoji library (`https://github.com/twitter/twemoji`) and are copyright Twitter, Inc and other contributors, licensed under CC-BY 4.0 (`https://creativecommons.org/licenses/by/4.0/`). The chicken emoji was modified.

actions are exogenous, and define

$$\mathsf{Endo}(a) \stackrel{\text{def}}{=} \neg\mathsf{Exo}(a),$$

so that $\mathsf{Endo}(a)$ means that an action is endogenous.

5.2.1 The exogenous program

As we've said, in the framework that we're presenting, there will be a (ConGolog) program that governs what occurs in the environment. We will borrow an idea from Lespérance et al. (2008) and have that program be of the form

$$\delta_{Exo} \rangle\!\rangle \delta_{Endo}$$

where the program δ_{Exo} describes exogenous actions (by the environment) and the program δ_{Endo} describes endogenous actions (by the agent). Recall that "$\rangle\!\rangle$" is the prioritized concurrency operator in ConGolog, so δ_{Exo} is run with higher priority (so endogenous actions only occur when δ_{Exo} is blocked, i.e., cannot execute an action). The endogenous process δ_{Endo} will always be the one we define here:

$$\delta_{Endo} \stackrel{\text{def}}{=} (\pi a.\ \mathsf{Endo}(a)?; a)^*.$$

That is, the agent repeatedly selects an arbitrary endogenous action and executes it. In other words, the agent just does whatever it wants (later we'll talk about achieving goals). The exogenous process, on the other hand, will vary from one domain to another.

Giving the environment process higher priority should be understood as a convention for axiomatizing domains, and does not mean that our approach is limited to modelling real world problems in which the agent is "less important" than its environment. The point is just to give the axiomatizer a way to specify what interleavings of endogenous and exogenous actions are possible. An alternative convention would be to give the endogenous process higher priority. In that case, to allow the environment to take turns, the axiomatizer would have to ensure that the precondition axioms were such that in appropriate situations there were no endogenous actions that were possible.

Note also that while the environment is described with a single program, since we are using ConGolog, that program can contain multiple concurrent processes (e.g., corresponding to different elements of the environment, such as the weather or the actions of an animal). It will be up to the axiomatizer to specify the theory and program in such a

way so as to restrict, as desired, how any such processes can interleave with each other. (We do not consider "true" concurrency, where multiple actions are executed at the same instant, in this chapter.)

Moving on, a *legal* or executable situation was traditionally defined as one where all the actions that had been executed were possible (Reiter, 2001, p. 53). We will define a more restricted set of situations – those which can be arrived at by following a particular ConGolog program.

We will find this abbreviation useful:

$$\mathsf{Reachable}(s, \delta, s') \stackrel{\text{def}}{=} \exists \gamma.\ \mathsf{Trans}^*(\delta, s, \gamma, s')$$

That is, situation s' can be reached from s by following δ (not necessarily to completion). Now, we define the Legal^+ situations, relative to a given exogenous program term δ, to be those that can be reached by following that program (and the agent's choice of actions, when the agent gets to act) from $\mathsf{root}(s)$, the initial situation preceding s.

Definition 5.2.1 (Legal^+).

$$\mathsf{Legal}^+(\delta, s) \stackrel{\text{def}}{=} \mathsf{Reachable}(\mathsf{root}(s), [\delta \rangle\!\rangle\ \delta_{Endo}], s)$$

Note that a Legal^+ situation is always also legal, because a primitive action taken in a program transition must be possible (see §2.2.2.5).

In this chapter, we're going to assume that the new form of action theories we consider (PATs) contain a sentence of the form

$$\mathsf{exoProgram} = \delta_{Exo}$$

for some ground literal program term δ_{Exo}, to indicate how the environment will behave (recall that a literal program term is defined in Definition 2.2.8). We will assume that δ_{Exo} is written so that the only actions that can be produced in a run of δ_{Exo} are exogenous actions.

5.2.2 The accessibility relation for belief

We now turn to defining the accessibility relation B for belief. In previous chapters, B was a fluent; now, we will be defining it as an abbreviation that takes into account the exogenous program. Furthermore, we want to allow for some actions (especially exogenous ones) to potentially not be fully observable to the agent. To do so, we will follow Kelly

and Pearce (2015) and suppose that there is a functional fluent $\mathsf{view}(s)$, whose value in a situation s describes how much the agent has observed as a result of the actions that have occurred.[2]

If all actions are observable and there are no sensing actions, then we can set the view in a situation to just be equal to the list of actions that have occurred. To do so, we can first initialize view to be an empty list with this axiom:

$$\mathsf{Init}(s) \supset [\mathsf{view}(s) = \langle\rangle] \tag{5.1}$$

Then, we can use this SSA for view (where $\cdot$ is a concatenation operator):

$$\mathsf{view}(\mathsf{do}(a, s)) = a \cdot \mathsf{view}(s) \tag{5.2}$$

(Note how this version of view is like the $\mathsf{history}$ fluent from the previous chapters.)

Kelly and Pearce give the SSA for view in a more general form by introducing an obs function that describes what the agent observes, and described (in their §5) various ways that observations can work. Most of our general results won't depend on a particular SSA for view. For now we'll just point out two other possible SSAs for view that can serve useful roles.

Sensing results The SSA for view in Equation 5.2 does not allow the agent to gain information from sensing. However, to allow for sensing, all we have to do is record a representation of the sensing outcome of an action along with that action, e.g., with this SSA:

$$\begin{aligned}&\mathsf{view}(\mathsf{do}(a, s)) = y \equiv \\ &\quad [(\mathsf{SF}(a, s) \wedge y = \langle a, 1\rangle \cdot \mathsf{view}(s)) \vee (\neg\mathsf{SF}(a, s) \wedge y = \langle a, 0\rangle \cdot \mathsf{view}(s))]\end{aligned} \tag{5.3}$$

That is, if action a gets a positive sensing result s, $\langle a, 1\rangle$ will be recorded; if a gets a negative sensing result, $\langle a, 0\rangle$ is recorded.

Unobservable exogenous actions We might want the agent to not observe exogenous actions when they occur. We can model that by having view only record the endogenous actions, using this SSA:

$$\mathsf{view}(\mathsf{do}(a, s)) = y \equiv [(\mathsf{Endo}(a) \wedge y = a \cdot \mathsf{view}(s)) \vee (\mathsf{Exo}(a) \wedge y = \mathsf{view}(s))] \tag{5.4}$$

[2] $\mathsf{view}(s)$ can be thought of as the agent's "local state" in s, in the sense of Halpern and Fagin (1989).

We now define the epistemic accessibility relation B – unlike in previous chapters, this is now a three-place relation, which also considers the relevant exogenous program. Note that in the definition we make use of a special predicate $\mathsf{B}_0(s)$ to restrict the accessibility relation.

Definition 5.2.2 ($\mathsf{B}(\delta, s', s)$)**.**

$$\mathsf{B}(\delta, s', s) \stackrel{\text{def}}{=} [\mathsf{view}(s') = \mathsf{view}(s)] \wedge \mathsf{B}_0(\mathsf{root}(s')) \wedge \mathsf{Legal}^+(\delta, s')$$

That is, B-accessible situations must both match the agent's view of what has happened, and be reachable – from an initial situation that B_0 is true of – by following the appropriate program. (It's not relevant what non-initial situations B_0 is true of, which is why we don't give it an SSA.) Note that the B relation is very similar to the K_p relation suggested in Kelly and Pearce's (2015) future work section. It's also worth noting that the encodings of formulas in ConGolog programs can't refer to program terms, and so a program cannot refer to the accessibility relation B. So it is not circular to define accessibility in terms of a program. Furthermore, when $\delta = \mathsf{nil}$, and view is defined appropriately, the accessibility relation can behave like the one from previous chapters (under some conditions), as will be later shown in Proposition 5.2.1.

We now define a new version of the **Bel** operator which specifies the program as an argument. As with the belief operator in previous chapters, belief is still defined as what's true in the most plausible accessible situations:

$$\mathbf{Bel}(\delta, \phi, s) \stackrel{\text{def}}{=} \forall s'.\ [\mathsf{B}(\delta, s', s) \wedge \forall s''.\ \mathsf{B}(\delta, s'', s) \supset s' \leq_{\mathsf{pl}} s''] \supset \phi[s'].$$

We can abbreviate the antecedent of that conditional:

$$\mathsf{MPB}(\delta, s', s) \stackrel{\text{def}}{=} \mathsf{B}(\delta, s', s) \wedge \forall s''.\ \mathsf{B}(\delta, s'', s) \supset s' \leq_{\mathsf{pl}} s''$$

Intuitively, $\mathsf{MPB}(\delta, s', s)$ means that s' is one of the most plausible situations accessible from s, for an agent that thinks the program δ is running.

Since the relevant program will typically be $\mathsf{exoProgram}$, for brevity we make the following definitions (which within this chapter replace the definitions from previous chapters):

Definition 5.2.3 (redefining $\mathsf{B}(s', s)$, $\mathsf{MPB}(s', s)$, and $\mathbf{Bel}(\phi, s)$).

$$\mathsf{B}(s', s) \stackrel{\text{def}}{=} \mathsf{B}(\mathsf{exoProgram}, s', s)$$
$$\mathsf{MPB}(s', s) \stackrel{\text{def}}{=} \mathsf{MPB}(\mathsf{exoProgram}, s', s)$$
$$\mathbf{Bel}(\phi, s) \stackrel{\text{def}}{=} \mathbf{Bel}(\mathsf{exoProgram}, \phi, s)$$

We will also find it convenient to have a "true belief" operator:

Definition 5.2.4 (TBel).

$$\mathbf{TBel}(\phi, s) \stackrel{\text{def}}{=} (\mathbf{Bel}(\phi, s) \wedge \phi[s]).$$

Note that because belief is still defined in terms of truth in a set of situations, beliefs behave in a fairly standard way, with usual properties like closure under logical consequence. Furthermore, positive and negative introspection (previously discussed in §2.3) are built-in, as we will see later.

Often, we'll want an action theory to completely characterize the predicate B_0 (used in the definition of B), so as to say exactly what the agent initially considers possible (similarly to what IAATs did with only-knowing in the previous chapters). We can do so with a formula $\mathsf{Init}(s) \supset (\mathsf{B}_0(s) \equiv \phi[s])$. Note that it doesn't matter which non-initial situations are included in the extension of B_0, since that won't affect B (which only applies B_0 to root situations). We will introduce this abbreviation:

Definition 5.2.5 (InitB).

$$\mathbf{InitB}(\phi) \stackrel{\text{def}}{=} \forall s.\ \mathsf{Init}(s) \supset (\mathsf{B}_0(s) \equiv \phi[s])$$

Recalling how B_0 is used in the definition of B, what $\mathbf{InitB}(\phi)$ specifies is that the roots of the accessible situations have ϕ true at them.

5.2.3 Programmed action theories (PATs)

Finally, we define the action theories we are considering in this chapter:

Definition 5.2.6 (PAT). A *programmed action theory* (PAT) is a set of axioms

$$\Sigma_{\mathsf{found}} \cup \Sigma_{\mathsf{ssa}} \cup \Sigma_{\mathsf{pre}} \cup \Sigma_{\mathsf{sense}} \cup \Sigma_0 \cup \Sigma_{\mathsf{una}} \cup \Sigma_{\mathsf{ConGolog}} \cup \{\mathbf{InitB}(\phi)\}$$

Most of the components are familiar from IAATs. We allow Σ_0 to contain, in addition to formulas uniform in S_0, formulas of the form $\forall s.\ \mathsf{Init}(s) \supset \phi(s)$, where $\phi(s)$ is uniform in s.

In particular, we require that Σ_0 always includes Equation 5.1 (initializing view to be an empty list). In $\mathbf{InitB}(\phi)$, ϕ is an expression uniform in *now* that describes what the agent is initially certain of. $\Sigma_{\mathsf{ConGolog}}$ consists of general axioms describing ConGolog programs (see §2.2.2.5) and Σ_0 includes $\mathsf{exoProgram} = \delta_{Exo}$ for some (ground) literal program term δ_{Exo}, whose executions can only produce exogenous actions. We require that Σ_0 specifies which actions are exogenous. As in an IAAT, we require that Σ_{ssa} contains axioms for each abnormality fluent in the form of Equation 3.1, specifying that what's abnormal doesn't change. Σ_0 should also include an axiomatization of lists, as in an IAAT. Finally, for later bookkeeping purposes we suppose that there is an exogenous action null with the precondition axiom $\mathsf{Poss}(\mathsf{null}, s) \equiv \mathsf{False}$.

The proposition below shows how the accessibility relation we're using can be related to the one we considered in previous chapters, which we've renamed to $\mathsf{B}_{old}(s', s)$.

Proposition 5.2.1. Suppose that Σ is a PAT including

$$\mathbf{InitB}(\phi) \qquad \mathsf{exoProgram} = \mathsf{nil} \qquad \forall a.\ \mathsf{Endo}(a)$$

and the SSA for view from Equation 5.3. Let Γ be the set of sentences comprised of

$$\mathsf{Init}(s) \supset \forall s'.\ \mathsf{B}_{old}(s', s) \equiv (\mathsf{Init}(s') \wedge \phi[s'])$$

and the SSA for our old accessibility relation, Equation 2.11 (renaming B to B_{old}). Then

$$\Sigma \cup \Gamma \models \forall s, s'.\ \mathsf{B}(s', s) \equiv \mathsf{B}_{old}(s', s). \tag{5.5}$$

Proof. We will prove this using the induction axiom (Equation 2.6). First, we want to establish that the accessibility relations are equivalent initially, i.e., that

$$\Sigma \cup \Gamma \models \forall s.\ \mathsf{Init}(s) \supset \forall s'.\ \mathsf{B}(s', s) \equiv \mathsf{B}_{old}(s', s). \tag{5.6}$$

That holds because using either accessibility relation, the situations accessible from an initial situation are exactly the initial situations where ϕ is true (for B, note that only initial situations have the value of view as an empty list).

Next, we will show that if the accessibility relations are equivalent in one situation, then they remain equivalent after performing any action, i.e., that

$$\Sigma \cup \Gamma \models \forall a, s.\ [\forall s'.\ \mathsf{B}(s', s) \equiv \mathsf{B}_{old}(s', s)] \supset \forall s''.\ \mathsf{B}(s'', \mathsf{do}(a, s)) \equiv \mathsf{B}_{old}(s'', \mathsf{do}(a, s)). \tag{5.7}$$

To get started, by the definition of $\mathsf{B}(s', s)$ we have that

$$\Sigma \models \forall s, a, s''.\ \mathsf{B}(s'', \mathsf{do}(a, s)) \equiv [\mathsf{view}(s'') = \mathsf{view}(\mathsf{do}(a, s))] \wedge \mathsf{B}_0(\mathsf{root}(s'')) \wedge \mathsf{Legal}^+(\mathsf{exoProgram}, s'').$$

Using the SSA for view, it can be shown that

$$\Sigma \models \forall s, a, s''.\ \mathsf{view}(s'') = \mathsf{view}(\mathsf{do}(a, s)) \equiv \exists s'.\ (s'' = \mathsf{do}(a, s')) \wedge (\mathsf{view}(s') = \mathsf{view}(s)) \wedge (\mathsf{SF}(a, s') \equiv \mathsf{SF}(a, s)).$$

Furthermore, because $\mathsf{exoProgram} = \mathsf{nil}$ and all actions are endogenous, it can be seen that

$$\Sigma \models \forall a, s'.\ \mathsf{Legal}^+(\mathsf{exoProgram}, \mathsf{do}(a, s')) \equiv (\mathsf{Legal}^+(\mathsf{exoProgram}, s') \wedge \mathsf{Poss}(a, s')).$$

Putting all that together (along with the fact that $\Sigma \models \forall a, s'.\ \mathsf{root}(\mathsf{do}(a, s')) = \mathsf{root}(s')$), we have that

$$\Sigma \models \forall s, a, s''.\ \mathsf{B}(s'', \mathsf{do}(a, s)) \equiv \exists s'.\ (s'' = \mathsf{do}(a, s') \wedge (\mathsf{view}(s') = \mathsf{view}(s)) \wedge (\mathsf{SF}(a, s') \equiv \mathsf{SF}(a, s)) \wedge \mathsf{B}_0(\mathsf{root}(s')) \wedge (\mathsf{Legal}^+(\mathsf{exoProgram}, s') \wedge \mathsf{Poss}(a, s'))$$

Rearranging that expression and applying the definition of B gives us

$$\Sigma \models \forall s, a, s''.\ \mathsf{B}(s'', \mathsf{do}(a, s)) \equiv \exists s'.\ \mathsf{B}(s', s) \wedge (s'' = \mathsf{do}(a, s')) \wedge \mathsf{Poss}(a, s') \wedge (\mathsf{SF}(a, s') \equiv \mathsf{SF}(a, s)).$$

So we have related what's B-accessible from the situation denoted by $\mathsf{do}(a, s)$ to what's B-accessible from the situation denoted by s – in a way that exactly parallels the SSA for B_{old} (Equation 2.11). Therefore, if B and B_{old} agree on what's accessible from the situation denoted by s, they will also agree on what's accessible from the situation denoted by $\mathsf{do}(a, s)$, and so we get Equation 5.7.

In conclusion, to get the final result (Equation 5.5), consider any model $\mathfrak{I}$ of $\Sigma \cup \Gamma$ and variable assignment μ mapping the second-order predicate variable P such that

$$\mathfrak{I}, \mu \models \forall s.\ P(s) \equiv [\forall s'.\ \mathsf{B}(s', s) \equiv \mathsf{B}_{old}(s', s)].$$

Because Σ includes the induction axiom (Equation 2.6) as a foundational axiom, we get that

$$\mathfrak{I}, \mu \models \big([\forall s.\ \mathsf{Init}(s) \supset P(s)] \wedge [\forall a, s.\ P(s) \supset P(\mathsf{do}(a, s))]\big) \supset \forall s.\ P(s)$$

It can be seen from Equation 5.6 and Equation 5.7 that the antecedent of that conditional is satisfied, and therefore so is the consequent, from which we can conclude Equation 5.5. □

Of course, the more interesting cases, which we'll spend most of this chapter considering, are ones where exoProgram is not equal to nil. We next establish that there is positive and negative introspection of beliefs (recall introspection was discussed in §2.3.1), by first showing that the B relation is transitive and Euclidean, and then that the MPB relation is also.

Lemma 5.2.1. Let δ be any ground program term. For any PAT Σ, and any model $\mathfrak{I}$ of Σ and variable assignment μ,

$$\mathfrak{I}, \mu \models \mathsf{B}(\delta, s', s) \supset \forall s''.\ \mathsf{B}(\delta, s'', s) \equiv \mathsf{B}(\delta, s'', s')$$

Proof. We'll show that $\mathfrak{I}, \mu \models \mathsf{B}(\delta, s', s) \supset \forall s''.\ \mathsf{B}(\delta, s'', s) \supset \mathsf{B}(\delta, s'', s')$. The other direction, that $\mathfrak{I}, \mu \models \mathsf{B}(\delta, s', s) \supset \forall s''.\ \mathsf{B}(\delta, s'', s') \supset \mathsf{B}(\delta, s'', s)$, is symmetric.

Suppose that $\mathfrak{I}, \mu \models \mathsf{B}(\delta, s', s)$. Therefore, $\mathfrak{I}, \mu \models \mathsf{view}(s') = \mathsf{view}(s)$. Now suppose that μ' is a variable assignment that differs from μ at most on s'', and that $\mathfrak{I}, \mu' \models \mathsf{B}(\delta, s'', s)$. Then $\mathfrak{I}, \mu' \models \mathsf{view}(s'') = \mathsf{view}(s)$, and so $\mathfrak{I}, \mu' \models \mathsf{view}(s'') = \mathsf{view}(s')$. Also, it must be the case that $\mathfrak{I}, \mu' \models \mathsf{B}_0(\mathsf{root}(s''))$ and $\mathfrak{I}, \mu' \models \mathsf{Legal}^+(\delta, s'')$. Then $\mathfrak{I}, \mu' \models \mathsf{B}(\delta, s'', s')$ by definition. Therefore, $\mathfrak{I}, \mu \models \forall s''.\ \mathsf{B}(\delta, s'', s) \supset \mathsf{B}(\delta, s'', s')$. □

Lemma 5.2.2. Let δ be any ground program term. For any PAT Σ, and any model $\mathfrak{I}$ of Σ and variable assignment μ,

$$\mathfrak{I}, \mu \models \mathsf{MPB}(\delta, s', s) \supset \forall s''.\ \mathsf{MPB}(\delta, s'', s) \equiv \mathsf{MPB}(\delta, s'', s')$$

Proof. Suppose that $\mathfrak{I}, \mu \models \mathsf{MPB}(\delta, s', s)$. By Lemma 5.2.1, $\mathfrak{I}, \mu \models \forall s''.\ \mathsf{B}(\delta, s'', s) \equiv \mathsf{B}(\delta, s'', s')$. Therefore, the set of situation objects in the domain of $\mathfrak{I}$ that are accessible from $\mu[s]$ is the same set that is accessible from $\mu[s']$. It follows that the most plausible situation objects in each of those sets are the same, from which the result can be concluded. □

Proposition 5.2.2 (positive and negative introspection). For any PAT Σ, Σ entails each of the following:

$$\forall s.\ \mathbf{Bel}(\phi, s) \supset \mathbf{Bel}(\mathbf{Bel}(\phi, now), s)$$
$$\forall s.\ \neg\mathbf{Bel}(\phi, s) \supset \mathbf{Bel}(\neg\mathbf{Bel}(\phi, now), s)$$

Proof. This follows from the MPB relation being transitive and Euclidean, as shown in Lemma 5.2.2. □

We will make use of introspection later, when knowing-how is defined in terms of nested beliefs.

5.2.4 Beliefs about the running program

The accessible situations are constrained to be reachable by following $\mathsf{exoProgram}\rangle\!\rangle\, \delta_{Endo}$ in our model of belief, so the agent is certain that exoProgram is what the environment is following. However, the agent may believe that other exogenous programs would determine the same set of Legal^+ situations. To talk about this, we introduce the following abbreviation:

$$\mathsf{ExoRunning}(\delta, s) \stackrel{\text{def}}{=} \forall s' \sqsupseteq s.\ \mathsf{Legal}^+(\mathsf{exoProgram}, s') \equiv \mathsf{Legal}^+(\delta, s')$$

That is, $\mathsf{ExoRunning}(\delta, s)$ holds if the situations in s's future reachable when the environment follows δ (starting from the root of s) are exactly those that are reachable when the environment follows exoProgram. So there is a sense in which δ is equivalent to exoProgram, and can be said to be (also) running.

To illustrate, if a PAT Σ includes

$$\mathsf{exoProgram} = \mathbf{if}\ \mathsf{P}\ \mathbf{then}\ \delta_1\ \mathbf{else}\ \delta_2\ \mathbf{endIf;}$$

for some ground ConGolog program terms δ_1 and δ_2, and $\Sigma \models \mathsf{P}(\mathsf{S}_0)$ then

$$\Sigma \models \mathsf{ExoRunning}(\delta_1, \mathsf{S}_0).$$

Furthermore, if the agent initially believes that $\mathsf{P}(now)$ is true, i.e. $\Sigma \models \mathbf{Bel}(\mathsf{P}(now), \mathsf{S}_0)$, then

$$\Sigma \models \mathbf{Bel}(\mathsf{ExoRunning}(\delta_1, now), \mathsf{S}_0),$$

that is, the agent believes δ_1 is running.

Example 5.2.1 (Plan recognition)**.**
It's worth noting that we can use ExoRunning to describe a simple form of *plan recognition*, i.e., recognizing a particular action sequence that is being followed based on observations (see e.g. Goultiaeva and Lespérance, 2007). Consider the following proposition, where the agent uses its observation of the first action to distinguish between two possible cooking plans that might be being followed in a kitchen.

Proposition 5.2.3. Let Σ be a PAT including

$$\mathsf{exoProgram} = (\underbrace{(\mathsf{boilWater}; \mathsf{addPasta})}_{\text{plan 1}} \mid \underbrace{(\mathsf{breakEggs}; \mathsf{fry})}_{\text{plan 2}})$$

where boilWater, addPasta, breakEggs, and fry are exogenous actions. Finally, suppose that Σ includes the SSA in Equation 5.2 for view (so that actions are observable). Then Σ entails each of the following:

$$\mathbf{Bel}(\mathsf{ExoRunning}(\mathsf{boilWater}; \mathsf{addPasta}, now), \mathsf{do}(\mathsf{boilWater}, \mathsf{S}_0))$$
$$\mathbf{Bel}(\mathsf{ExoRunning}(\mathsf{breakEggs}; \mathsf{fry}, now), \mathsf{do}(\mathsf{breakEggs}, \mathsf{S}_0))$$

Proof. The two cases are symmetric; consider the first one. From $\mathsf{do}(\mathsf{boilWater}, \mathsf{S}_0)$ any accessible situation s is such that boilWater (and no other action) has occurred. Any $s' \sqsupseteq s$ that is Legal^+ is a situation reachable from $\mathsf{root}(s)$ by following

$$((\mathsf{boilWater}; \mathsf{addPasta}) \mid (\mathsf{breakEggs}; \mathsf{fry})) \gg\!\!\gg \delta_{Endo}$$

and in which the action boilWater is the first action to have occurred. Such situations are exactly those successors of s that can be reached from $\mathsf{root}(s)$ by following the program $(\mathsf{boilWater}; \mathsf{addPasta}) \gg\!\!\gg \delta_{Endo}$. □

That example did not involve any abnormalities, but for much of this chapter we'll be looking at programs that refer to abnormalities. That will give a natural way of specifying that some program executions are more plausible than others.

5.2.5 Normalized programs

In the previous section, we saw how the agent could believe that various programs were running, even though it's certain that exoProgram is running. This is analogous to how in

Chapter 4 the agent could believe other dynamics axioms than those written in the theory. In Chapter 4 we also saw that the agent would sometimes believe *normalized* axioms that did not refer to abnormalities, and we can do something similar with programs. For example, if exoProgram is **if** Ab **then** sun **else** rain (where sun and rain are exogenous actions), then the agent may believe that the environment is running a program just saying that there will be rain (we will formalize this in Example 5.2.2).

Now, we can expand the definition of *normalization* (Definition 4.2.4) with respect to an *Ab account* (Definition 4.2.2) to include ConGolog programs.

Definition 5.2.7 (normalization of a program). Given a literal program term δ and an Ab account $\xi = \bigwedge_{\mathsf{Ab}_i \in R} \forall \vec{x}.\ \mathsf{Ab}_i(\vec{x}, now) \equiv \xi_i(\vec{x})$, the normalization of δ with respect to ξ is a program δ' which is like δ but, for each $\mathsf{Ab}_i \in R$ (the range of the Ab account), replaces any reference to $\mathsf{Ab}_i(\vec{\tau}, \sigma)$ with $\xi_i(\vec{\tau}, \sigma)$. (Programs are typically written in a situation-suppressed way, in which case that transformation amounts to replacing $\mathsf{Ab}_i(\vec{\tau})$ with $\xi_i(\vec{\tau})$.)

To illustrate, again consider the program

if Ab **then** sun **else** rain.

If ξ is the Ab account $\mathsf{Ab}(now) \equiv \mathsf{False}$, then the normalization of that program with respect to ξ is

if False **then** sun **else** rain.

We next get the following result about a general case in which the agent will believe that a normalization of exoProgram is running.

Proposition 5.2.4. Let Σ be a PAT including $\mathsf{exoProgram} = \delta_{Exo}$, where δ_{Exo} can be written in a situation-suppressed way (i.e., the only situation term it refers to is now). Given a ground situation term σ, if there is an Ab account ξ such that

$$\Sigma \models \mathbf{Bel}(\xi, \sigma)$$

and δ is the normalization of δ_{Exo} w.r.t ξ, then

$$\Sigma \models \mathbf{Bel}(\mathsf{ExoRunning}(\delta, now), \sigma)$$

Proof. We prove this by showing a stronger result, that under the conditions described

above $\Sigma \models \mathbf{Bel}(\mathsf{ExoRunning}^*(\delta, now), \sigma)$, where we use $\mathsf{ExoRunning}^*(\delta, s)$ to abbreviate

$$\forall s' \sqsupseteq \mathsf{root}(s).\ \mathsf{Legal}^+(\mathsf{exoProgram}, s') \equiv \mathsf{Legal}^+(\delta, s').$$

Note that the difference between $\mathsf{ExoRunning}^*$ and $\mathsf{ExoRunning}$ is that in $\mathsf{ExoRunning}^*$, s' is a successor of $\mathsf{root}(s)$ instead of s.

As was shown for IAATs in Lemma 4.2.1, if the agent believes an Ab account ξ, the agent believes that ξ was always and will always be true. Therefore, if $\Sigma \models \mathbf{Bel}(\xi, \sigma)$, then the agent in σ believes that whenever $\mathsf{exoProgram}$ makes a choice depending on whether an abnormal atom $\mathsf{Ab}_i(\vec{\tau})$ is true, δ makes a choice depending on whether $\xi_i(\vec{\tau})$ is true, and that those conditions are equivalent. More formally, the result can be shown to follow from how, for every k,

$$\begin{aligned}\Sigma \models \mathbf{Bel}(\forall a_1, \ldots, a_k.\ &\mathsf{Legal}^+(\mathsf{exoProgram}, \mathsf{do}([a_1, \ldots, a_k], \mathsf{root}(now))) \equiv \\ &\mathsf{Legal}^+(\delta, \mathsf{do}([a_1, \ldots, a_k], \mathsf{root}(now))), \sigma),\end{aligned}$$

which can be shown using induction. □

This proposition will be useful in various results in this chapter, starting with the example below.

Example 5.2.2 (beliefs about the future)**.**
The following proposition gives an example where the agent believes that the more plausible branch of the environment's program will run (in this case, rain is more plausible than sun).

Proposition 5.2.5. Let Σ be a PAT including

$$\begin{aligned}&\mathsf{exoProgram} = \mathbf{if}\ \mathsf{Ab}\ \mathbf{then}\ \mathsf{sun}\ \mathbf{else}\ \mathsf{rain}\ \mathbf{endIf}\\ &\mathbf{InitB}(\mathsf{True})\end{aligned}$$

where $\mathsf{Ab}(s)$ is an abnormality fluent and sun and rain are always-possible exogenous actions. Then $\Sigma \models \mathbf{Bel}(\mathsf{ExoRunning}(\mathsf{rain}, now), \mathsf{S}_0)$.

Proof. In the most plausible accessible situations from S_0, Ab is false. As previously pointed out, the normalization of (the value of) $\mathsf{exoProgram}$ in S_0 with respect to the Ab account $\mathsf{Ab}(now) \equiv \mathsf{False}$ is

$$\mathbf{if}\ \mathsf{False}\ \mathbf{then}\ \mathsf{sun}\ \mathbf{else}\ \mathsf{rain}\ \mathbf{endIf}$$

By Proposition 5.2.4, $\Sigma \models \mathbf{Bel}(\mathsf{ExoRunning}(\delta, now), \mathsf{S}_0)$ where δ is that normalized program. The overall result follows from noting that that program can be simplified to rain. □

In general, writing abnormalities within a program gives a convenient way to specify what executions the agent will expect.

5.2.6 A note on changing abnormalities

In PATs, what's abnormal cannot change. We previously, in §3.5.1, considered exogenous actions in action theories in which abnormalities could change over time. It turns out we can still model the sorts of examples we considered back in §3.5.1, without needing changing abnormalities, as we describe in this section.

One example (Example 3.5.1) was just about how to specify, as in (Shapiro and Pagnucco, 2004), that for fewer exogenous actions to occur is more plausible than for more exogenous actions to occur. To model this with a PAT, suppose we have a $\mathsf{history}(s)$ fluent recording all the actions that have occurred (like we used in IAATs in previous chapters), and that we use the axiom

$$\mathsf{exoProgram} = (\mathsf{Ab}(\mathsf{history})?;\ \pi a.\ \mathsf{Exo}(a)?; a)^*$$

which specifies that the environment program is blocked except in situations whose history is abnormal. So the fewer action sequences are abnormal (i.e., the more plausible the situation is), the fewer times the environment will get to act.

Another example was about saying how one exogenous action will more plausibly occur than another (Example 3.5.2); we have already seen things similar to that in this chapter (Example 5.2.2). The last and perhaps most interesting example we considered was the one about the agent believing that money left on the street has been stolen (Example 3.5.3), which we revisit in detail below.

The fate of abandoned money, revisited

Recall that Example 3.5.3 involves the agent not knowing how many actions have occurred, and comparing the plausibility of an initial situation and the situation resulting from doing the steal action. For the steal action to have occurred is considered more plausible because it switches an abnormality from true to false.

Note that using PATs it's straight-forward to have (without using mutable abnormalities) the agent believe that money *will* be stolen (by having the environment program

include the steal action), and that the money possibly already has been stolen (since we can make exogenous actions invisible to the agent). However, that doesn't quite capture the original example, where the agent initially thinks that the money *has already been* stolen, i.e., it's more plausible that it's a later time rather an earlier time. If abnormalities *don't* change, if there is any initial situation s such that both s and $\mathsf{do}(\mathsf{steal}, s)$ are accessible, they will be equally plausible, which makes it tricky to have the agent believe that the steal action has already happened.

To circumvent this problem, we relax the requirement from PATs that view be initialized to an empty list. Below we describe an action theory that is like a PAT but sets up $\mathsf{view}(s)$ so that in any situation s, $\mathsf{view}(s)$ will be a pair, where the second element is the list of endogenous actions that have occurred (exogenous ones are invisible to the agent) and the first element of the pair is a value that intuitively says what time the agent thinks it is (i.e., how many actions have occurred).

The view fluent is described by these axioms (note the reference to a numeric-valued functional fluent clock):

$$\mathsf{Init}(s) \supset [\mathsf{view}(s) = \langle \mathsf{clock}(s), \langle\rangle\rangle]$$
$$\mathsf{view}(\mathsf{do}(a, s)) = y \equiv \exists y_1, y_2, y_3.\ [\mathsf{view}(s) = \langle y_1, y_2\rangle \wedge y = \langle y_1 + 1, y_3\rangle] \wedge [(\mathsf{Endo}(a) \wedge y_3 = a \cdot y_2) \vee (\mathsf{Exo}(a) \wedge y_3 = y_2)]$$

So in any initial situation, view stores the value of clock in that situation, which intuitively represents the time in that situation. Furthermore, after any action (even an exogenous one), the stored time value gets incremented. (Also, endogenous actions are recorded in the list.)

For clock we have these axioms:

$$\mathsf{clock}(\mathsf{do}(a, s)) = \mathsf{clock}(s) + 1$$
$$\mathsf{clock}(\mathsf{S}_0) = 1$$

So the clock fluent's value is increased by one by any action. Note that (because of the foundational axioms) there are initial situations (other than S_0) where the clock fluent takes any numeric value.

We specify that the agent believes that most plausibly the clock starts at 0 (recall that in S_0 the clock actually starts at 1).

$$\mathbf{InitB}([\mathsf{clock}(now) = 0] \vee \mathsf{Ab}(now))$$

Finally, the exogenous program just says that the steal action will be performed:

$$\mathsf{exoProgram} = \mathsf{steal}$$

The proposition below shows that this all results in the agent believing in S_0 that the steal action has already occurred.

Proposition 5.2.6. Let Σ be the action theory described above. Then

$$\Sigma \models \mathbf{Bel}(\exists s.\ \mathsf{do}(\mathsf{steal}, s) = now, \mathsf{S}_0).$$

Proof. Suppose that $\mathfrak{I}$ is a model of Σ and μ a variable assignment such that $\mathfrak{I}, \mu \models \mathsf{B}(s, \mathsf{S}_0)$, i.e.,

$$\mathfrak{I}, \mu \models [\mathsf{view}(s) = \mathsf{view}(\mathsf{S}_0)] \wedge \mathsf{B}_0(\mathsf{root}(s)) \wedge \mathsf{Legal}^+(\mathsf{exoProgram}, s)$$

Suppose further that $\mathfrak{I}, \mu \models \mathsf{MPB}(s, \mathsf{S}_0)$, which can be seen to require that $\mathfrak{I}, \mu \models \neg\mathsf{Ab}(s)$. Then we can conclude that

$$\mathfrak{I}, \mu \models \mathsf{clock}(\mathsf{root}(s)) = 0.$$

Therefore,

$$\mathfrak{I}, \mu \models \mathsf{view}(\mathsf{root}(s)) = \langle 0, \langle\rangle\rangle.$$

However,

$$\mathfrak{I}, \mu \models \mathsf{view}(\mathsf{S}_0) = \langle 1, \langle\rangle\rangle.$$

Therefore, $\mathfrak{I}, \mu \models \neg\mathsf{B}(\mathsf{root}(s), \mathsf{S}_0)$, and so, since $\mathfrak{I}, \mu \models \mathsf{B}(s, \mathsf{S}_0)$, we have

$$\mathfrak{I}, \mu \models s \neq \mathsf{root}(s),$$

(so $\mu[s]$ is not an initial situation). It can be seen that no endogenous actions can have occurred in $\mu[s]$, because those would be recorded by view and so make $\mu[s]$ inaccessible from the situation denoted by S_0. Therefore, the actions that have occurred in $\mu[s]$ must be exogenous. Since $\mathfrak{I}, \mu \models \mathsf{Legal}^+(\mathsf{exoProgram}, s)$, we can conclude that the exogenous action was the one denoted by steal. □

So we see that we don't need mutable abnormalities to model the phenomena that

we considered in §3.5.1.

5.3 Knowing how

Intuitively, an agent knows how to accomplish a goal if it can choose actions so as to bring the goal about. One definition of knowing-how from the literature, which we'll take as our starting point, is the one by Lespérance et al. (2000), which defines knowing-how in terms of knowledge. Unlike in previous parts of this thesis, we will now have a reason to distinguish between beliefs (which can be false) and knowledge which has to be true. We will first consider the difference between defining knowing-how in terms of beliefs instead of knowledge, and then introduce another definition that also deals with exogenous actions.

5.3.1 Knowing-how in terms of belief

We want to define knowing-how in terms of belief. To get started, we'll look at how Lespérance et al. (2000) did so in terms of *knowledge*. They used a knowledge operator with positive and negative introspection. We'll call this knowledge operator $\mathbf{Know}_L(\phi, s)$, and in general we'll use an L subscript on the operators they defined. Note that $\mathbf{Know}_L(\phi, s)$ is an abbreviation for the formula $\forall s'.\ \mathsf{K}_L(s', s) \supset \phi[s']$, where K_L is the accessibility relation.

Lespérance et al. did not seem to explicitly say whether knowledge had to be true (they mentioned the accessibility relation K_L being transitive and euclidean, but didn't say whether it had to be reflexive). However, if knowledge could be untrue, then their account would allow for knowing how to do impossible things. Therefore, we will assume that the $\mathbf{Know}_L$ operator describes true knowledge.

To say that the agent knew how to make ϕ true in situation s, they introduced a $\mathbf{Can}_L(\phi, s)$ operator. It was defined using the sequence of definitions below, where π is a second-order variable for a function mapping situations to actions – what they called an "action selection function". We will call such functions *policies*, following similar use in the planning literature (e.g., Ghallab et al., 2004). The first definition needed is OnPath_L, which is used in defining CanGet_L, which shortly will be used in defining $\mathbf{Can}_L$.

$$\mathsf{OnPath}_L(\pi, s, s') \stackrel{\text{def}}{=} s \leq s' \wedge \forall a, s^*.\ (s < \mathsf{do}(a, s^*) \leq s') \supset (\pi(s^*) = a)$$

$$\begin{aligned}\mathsf{CanGet}_L(\phi, \pi, s) \stackrel{\text{def}}{=}\ & \exists s'.\ [\mathsf{OnPath}_L(\pi, s, s') \wedge \mathbf{Know}_L(\phi, s') \wedge \\ & \forall s^*.\ (s \leq s^* < s') \supset \exists a.\ \mathbf{Know}_L(\pi(now) = a, s^*)]\end{aligned}$$

Informally, $\mathsf{OnPath}_L(\pi, s, s')$ means that situation s' can be reached from s by following the policy π, and $\mathsf{CanGet}_L(\phi, \pi, s)$ means that the agent can make ϕ true by following π from s.

Definition 5.3.1 ($\mathbf{Can}_L$ (Lespérance et al., 2000, p. 170)).

$$\mathbf{Can}_L(\phi, s) \stackrel{\text{def}}{=} \exists\pi.\ \mathbf{Know}_L(\mathsf{CanGet}_L(\phi, \pi, now), s)$$

Intuitively, $\mathbf{Can}_L(\phi, s)$ means that the agent knows a policy by which it can make ϕ true from s. Note that $\mathsf{CanGet}_L(\phi, \pi, s)$ requires the agent to know when the goal ϕ is achieved, and also to always know what action the policy π selects until then. $\mathbf{Can}_L(\phi, s)$ defines knowing-how in terms of knowing of a particular policy π such that $\mathsf{CanGet}_L(\phi, \pi, now)$ holds.

Remark 5.3.1. This definition of knowing-how requires that the agent knows that eventually they'll know that the goal has been achieved, i.e., that they're done. An alternative would just have the agent know that the goal will be achieved eventually (without requiring them to recognize the point when it happens). More generally, one could consider, instead of goals that will be completed, properties that hold with respect to the entire infinite run that results from following the policy forever. For this chapter, however, those are not the sort of generalizations that we will be exploring.

We now would like to instead define knowing-how in terms of the belief operator, **Bel**. For now, let's suppose that no exogenous actions can occur (we will consider those in the next section). If we just substituted **Bel** for $\mathbf{Know}_L$ in Lespérance et al.'s definitions, the result would not ensure that the agent would actually be able to do what it "knew how" to do, since beliefs can be false. Even substituting **TBel** for $\mathbf{Know}_L$ in Lespérance et al.'s definitions wouldn't give the result we want, because even if the agent correctly believes that a particular policy π will let it achieve the goal, it may also incorrectly believe that another policy π' would also work (and so in practice the agent might fail to act effectively). Therefore, we instead define a new version of knowing-how that requires *every* policy the agent believes in must actually work:

Definition 5.3.2 ($\mathbf{KHow}_0$). Let $\mathsf{CanGet}_L^{\mathbf{TBel}}$ be the abbreviation that is defined like CanGet_L but substitutes **TBel** for $\mathbf{Know}_L$. Then we define

$$\begin{aligned}\mathbf{KHow}_0(\phi, s) \stackrel{\text{def}}{=}\ & \exists\pi\ \mathbf{Bel}(\mathsf{CanGet}_L^{\mathbf{TBel}}(\phi, \pi, now), s) \wedge \\ & \forall\pi.\ \mathbf{Bel}(\mathsf{CanGet}_L^{\mathbf{TBel}}(\phi, \pi, now), s) \supset \mathsf{CanGet}_L^{\mathbf{TBel}}(\phi, \pi, s)\end{aligned}$$

The subscript 0 is just to distinguish this operator from another operator we will introduce, that takes into account exogenous actions.

5.3.2 Taking exogenous actions into account

Recall that we give higher priority to the environment's process. So, if there is an exogenous action a that could be executed in situation s (leading to a Legal^+ situation), then it's the environment's "turn" to act in s. Otherwise, it's the agent's turn in s. Hence we define the following:

Definition 5.3.3 (ExoTurn)**.**

$$\mathsf{ExoTurn}(\delta, s) \stackrel{\text{def}}{=} \exists a.\ \mathsf{Exo}(a) \wedge \mathsf{Legal}^+(\delta, \mathsf{do}(a, s))$$

While the environment is constrained to follow its program, we can use a policy to describe how the environment chooses to act (as for the agent). We define $\mathsf{ExoOption}(\delta, \rho, s)$ to mean that starting in situation s, the environment could use the policy ρ to select actions that would follow δ:

Definition 5.3.4 (ExoOption)**.**

$$\begin{aligned}\mathsf{ExoOption}(\delta, \rho, s) \stackrel{\text{def}}{=} \forall s' \sqsupseteq s.\ &[\mathsf{ExoTurn}(\delta, s') \supset (\mathsf{Exo}(\rho(s')) \wedge \mathsf{Legal}^+(\delta, \mathsf{do}(\rho(s'), s')))] \wedge \\ &[\neg\mathsf{ExoTurn}(\delta, s') \supset (\rho(s') = \mathsf{null})]\end{aligned}$$

That is, $\mathsf{ExoOption}(\delta, \rho, s)$ means that starting in s, the action selected by ρ on the environment's turn will always be an exogenous one that would produce a Legal^+ situation. When it's not the environment's turn, we adopt the convention that ρ must pick the special non-executable null action.

We will now define variants of Lespérance et al.'s OnPath_L and CanGet_L that take an additional argument – the environment's policy.

We use $\mathsf{OnPath}(\pi, \rho, s, s')$ to mean that s' is a situation that will be reached from s by following the two policies π and ρ (intuitively, π represents the agent's choices, and ρ the environment's). The way the two policies get combined is that ρ picks the action to be executed, except that if ρ picks the special null action, then π gets to pick the action

to execute.

$$\begin{aligned}&\mathsf{OnPath}(\pi, \rho, s, s') \stackrel{\text{def}}{=} \\ &\quad s \leq s' \wedge \forall a, s^*.\ (s < \mathsf{do}(a, s^*) \leq s') \supset \\ &\qquad \big[\big(\rho(s^*) \neq \mathsf{null} \wedge \rho(s^*) = a \wedge \mathsf{Exo}(a)\big) \vee \\ &\qquad\ (\rho(s^*) = \mathsf{null} \wedge \pi(s^*) = a \wedge \mathsf{Endo}(a))\big]\end{aligned}$$

As the following observation formalizes, if ρ selects actions that follow δ, and $\mathsf{OnPath}(\pi, \rho, s, s')$ holds, and s is Legal^+, then s' must also be Legal^+.

Observation 5.3.1. For any PAT Σ and ground program term δ,

$$\Sigma \models \forall s, \rho, \pi, s'.\ [\mathsf{ExoOption}(\delta, \rho, s) \wedge \mathsf{OnPath}(\pi, \rho, s, s') \wedge \mathsf{Legal}^+(\delta, s)] \supset \mathsf{Legal}^+(\delta, s').$$

Next, we can define $\mathsf{CanGet}(\phi, \pi, \rho, s)$ to mean that the agent can make ϕ true by following the policy π from s while the environment acts according to the policy ρ.

$$\begin{aligned}&\mathsf{CanGet}(\phi, \pi, \rho, s) \stackrel{\text{def}}{=} \\ &\quad \exists s'.\ \mathsf{OnPath}(\pi, \rho, s, s') \wedge \mathbf{TBel}(\phi, s') \wedge \\ &\qquad \forall s^*.\ (s \leq s^* < s') \supset \exists a.\ \mathbf{TBel}(\pi(now) = a, s^*)\end{aligned}$$

We then define $\mathsf{CanAlwaysGet}(\delta, \phi, \pi, s)$ to mean that the agent can achieve ϕ by following π from s regardless of what the environment chooses to do (so long as the environment follows δ):

$$\mathsf{CanAlwaysGet}(\delta, \phi, \pi, s) \stackrel{\text{def}}{=} \forall \rho.\ \mathsf{ExoOption}(\delta, \rho, s) \supset \mathsf{CanGet}(\phi, \pi, \rho, s)$$

(We've made δ a parameter, rather than just fixing it at $\mathsf{exoProgram}$, for use in some propositions later.)

For brevity, we can define this operator:

Definition 5.3.5 (BHow).

$$\mathbf{BHow}(\phi, \pi, s) \stackrel{\text{def}}{=} \mathbf{Bel}(\mathsf{CanAlwaysGet}(\mathsf{exoProgram}, \phi, \pi, now), s)$$

We can read $\mathbf{BHow}(\phi, \pi, s)$ as saying that the agent believes it can (or "believes how to") accomplish ϕ with policy π.

Lastly, we can define knowing-how analogously to how we did before (with $\mathbf{KHow}_0$),

except now considering everything the environment can do (constrained by following its program):

Definition 5.3.6 (KHow).

$$\mathbf{KHow}(\phi, s) \stackrel{\text{def}}{=} \exists \pi\ \mathbf{BHow}(\phi, \pi, s) \wedge \forall \pi.\ \mathbf{BHow}(\phi, \pi, s) \supset \mathsf{CanAlwaysGet}(\mathsf{exoProgram}, \phi, \pi, s)$$

So there has to be a particular policy that the agent believes works, and any policy that the agent believes works should work. The latter part is what makes this an "objective" definition of knowing-how.

Remark 5.3.2. If we wanted a subjective "believing how" version of **KHow**, we could define

$$\mathbf{BHow'}(\phi, s) \stackrel{\text{def}}{=} \exists \pi.\ \mathbf{BHow}(\phi, \pi, s)$$

However, that's not really necessary, since as we'll later see (Proposition 5.3.5), $\mathbf{BHow'}(\phi, s)$ would be equivalent to $\mathbf{Bel}(\mathbf{KHow}(\phi, now), s)$. So determining beliefs about know-how is simpler than determining what the agent actually knows how to do.

In general some combination of exogenous and endogenous actions will be needed to bring about a goal. As the agent's beliefs about what the environment will do evolve, so too will the agent's beliefs about what it knows how to do. To illustrate, let's consider a (highly abstracted) restaurant scenario. The agent (a customer at a restaurant) is concerned with whether it knows how to get served lasagna. The only endogenous actions is $\mathsf{order}(x)$ – the customer orders x. This does not directly cause the agent to have been served. Instead, the customer has to rely on exogenous actions (by the waiter). The exogenous actions are as follows: greet – the waiter greets the agent; $\mathsf{greet'}$ – the waiter greets the agent, but in a way that somehow causes the agent to question the waiter's competence; and $\mathsf{serve}(x)$ – the waiter serves x to the customer.

We will not present the entire PAT for the restaurant scenario, but suppose the exogenous program $\mathsf{exoProgram}$ is set to

$$\Big[\neg\mathsf{Ab}?;\ \mathsf{greet};\ \pi x.\ \mathsf{Ordered}(x)?;\ \mathsf{serve}(x)\Big] \mid \Big[\mathsf{Ab}?;\ \mathsf{greet'};\ (\exists x\ \mathsf{Ordered}(x))?;\ \pi y.\ \mathsf{serve}(y)\Big]$$

where $\mathsf{Ordered}(x, s)$ is a fluent that becomes true when the agent performs $\mathsf{order}(x)$. What the program says is that there are two courses of action that are possible. The more plausible course of action (under some assumptions about what else the theory

contains) is described by the part of the program on the left side of the | operator, and involves the waiter greeting the customer in the normal way, and the customer getting the dish x that was ordered. The less plausible course of action (which only occurs if Ab is true) involves the customer being served a random dish y that may not be what was ordered. Note that the way the program works is that the environment acts first, the agent performs one action (when the environment program blocks waiting for an order), and then the environment performs one action (after which the agent could perform more actions, though those won't help it get served anything).

This could be fleshed out to get an PAT entailing each of the following (where $\mathsf{Served}(x, s)$ is a fluent that $\mathsf{serve}(x)$ makes true):

$$\mathbf{Bel}(\mathbf{KHow}(\mathsf{Served}(\mathsf{lasagna}), now), \mathsf{S}_0)$$
$$\mathbf{Bel}(\mathbf{KHow}(\mathsf{Served}(\mathsf{lasagna}), now), \mathsf{do}(\mathsf{greet}, \mathsf{S}_0))$$
$$\mathbf{Bel}(\neg\mathbf{KHow}(\mathsf{Served}(\mathsf{lasagna}), now), \mathsf{do}(\mathsf{greet}', \mathsf{S}_0))$$

That is, the agent initially believes it knows how to get served lasagna, and still does after being greeted by the waiter in the normal way. However, if the greet' action is performed, then the agent comes to question the waiter's competence and no longer believes it knows how to get served lasagna.

The way this works is that initially the agent assumes that Ab is false, and so believes that the first branch of the exogenous program gets executed, in which the agent will get served whatever it orders (e.g., lasagna). So it could be shown that the agent believes it can bring about its goal with, e.g., a policy that just always orders lasagna:

$$\pi(s) = \mathsf{order}(\mathsf{lasagna}).$$

In $\mathsf{do}(\mathsf{greet}, \mathsf{S}_0)$ the agent gets confirmation that that expected branch of the program is being followed, and so would still believe that it knows how to achieve its goal. On the other hand, in $\mathsf{do}(\mathsf{greet}', \mathsf{S}_0)$ the agent concludes that Ab must be true, and so expects that what it will be served will be random. Therefore, the agent believes it can't guarantee that it will get lasagna.

Note that while in this example it was easy to describe a policy by which the agent could achieve its goal (in the cases where it could), in general policies may have to be much more complicated and may be awkward to describe. In the next section, we will consider a somewhat more concrete specification of agent behavior than policies: sequential plans.

5.3.3 Achieving goals by sequential plans

We will sometimes be interested in saying not just that the agent knows how to accomplish a goal ϕ, but that they know that they can do so by following some *plan* – a sequence of actions. To describe achieving things with sequential plans, we will need another set of definitions, analogous to the ones for policies.

First, we define $\mathsf{AfterSeq}([\alpha_1, \ldots, \alpha_k], \rho, s, s')$ to mean, intuitively, that s' is a situation resulting from the interleaving of the environment's actions (determined by the policy ρ) and the agent's actions (which are the sequence $\alpha_1, \ldots, \alpha_k$), starting from s:

$$\begin{aligned}&\mathsf{AfterSeq}([\alpha_1, \ldots, \alpha_k], \rho, s, s') \stackrel{\text{def}}{=} \\ &\quad \exists s_1, \ldots, s_k.\ \Big[(s < \mathsf{do}(\alpha_1, s_1) < \cdots < \mathsf{do}(\alpha_k, s_k) \leq s') \wedge \textstyle\bigwedge_{i=1}^{k} \rho(s_i) = \mathsf{null} \wedge \mathsf{Endo}(\alpha_i)\Big] \wedge \\ &\qquad \forall a, s^*.\ \Big[(s < \mathsf{do}(a, s^*) \leq s') \wedge \textstyle\bigwedge_{i=1}^{k} s^* \neq s_i\Big] \supset \Big[\rho(s^*) \neq \mathsf{null} \wedge \rho(s^*) = a \wedge \mathsf{Exo}(a)\Big]\end{aligned}$$

That is, between s and s' all of $\alpha_1, \ldots, \alpha_k$ must occur in order (each when ρ selects the null action), and except in the situations $s_1, \ldots, s_k$ where those actions are executed, the action executed is the exogenous one selected by ρ. This operator looks a bit different from OnPath but will play a similar role.

We previously defined $\mathsf{CanGet}(\phi, \pi, \rho, s)$ to say that the agent can make ϕ true by following the policy π from s while the environment acts according to the policy ρ. We now define $\mathsf{CanSeq}(\phi, [\alpha_1, \ldots, \alpha_k], \rho, s)$, where $\alpha_1, \ldots, \alpha_k$ are actions, to say that the agent can make ϕ true by following the sequence $\alpha_1, \ldots, \alpha_k$ from s while the environment acts according to ρ.

$$\mathsf{CanSeq}(\phi, [\alpha_1, \ldots, \alpha_k], \rho, s) \stackrel{\text{def}}{=} \exists s'.\ \mathsf{AfterSeq}([\alpha_1, \ldots, \alpha_k], \rho, s, s') \wedge \mathbf{TBel}(\phi, s')$$

That is, there must be a situation s' reached by the interleaving of (all) the actions from $[\alpha_1, \ldots, \alpha_k]$ and ρ such that in it the agent truly believes that ϕ is true. (This definition assumes that the agent remembers which of $\alpha_1, \ldots, \alpha_k$ it has already executed, and so will know which action to take on its turn.)

We can then define $\mathsf{CanAlwaysSeq}(\delta, \phi, [\alpha_1, \ldots, \alpha_k], s)$ in terms of CanSeq analogously to how we defined $\mathsf{CanAlwaysGet}$ in terms of CanGet:

$$\begin{aligned}&\mathsf{CanAlwaysSeq}(\delta, \phi, [\alpha_1, \ldots, \alpha_k], s) \stackrel{\text{def}}{=} \\ &\qquad \forall \rho.\ \mathsf{ExoOption}(\delta, \rho, s) \supset \mathsf{CanSeq}(\phi, [\alpha_1, \ldots, \alpha_k], \rho, s)\end{aligned}$$

Finally, analogously to $\mathbf{BHow}(\phi, \pi, s)$ where π is a policy, we can define the following:

Definition 5.3.7 (BHowSeq).

$$\mathbf{BHowSeq}(\phi, [\alpha_1, \ldots, \alpha_k], s) \stackrel{\text{def}}{=} \mathbf{Bel}(\mathsf{CanAlwaysSeq}(\mathsf{exoProgram}, \phi, [\alpha_1, \ldots, \alpha_k], now), s).$$

That is, $\mathbf{BHowSeq}(\phi, [\alpha_1, \ldots, \alpha_k], s)$ means that the agent believes in s that it can bring about ϕ by performing $\alpha_1, \ldots, \alpha_k$ in order (while the environment acts according to its program).

Remark 5.3.3. We could go one step further and define an operator analogous to **KHow**, that specifies that there is a sequential plan that the agent believes achieves the goal, and any sequential plan that the agent believes achieves the goal does so. It's not clear that this would be very useful, however, as the point of **KHow** was to guarantee that the agent would act correctly, and the agent isn't limited to acting in a sequential way.

5.3.4 Properties

In this section we prove some properties of our operators relating to knowing-how. In particular, we relate **KHow** and $\mathbf{KHow}_0$ to Lespérance et al.'s (2000) $\mathbf{Can}_L$ operator and each other, prove various properties relating to introspection, relate **BHow** and **BHowSeq** to **KHow**, and consider how beliefs about the attainment of goals by policies and sequential plans relate to beliefs about what exogenous programs are running.

Our first result (Proposition 5.3.1) will be that for domains with no exogenous actions, under some conditions on the accessibility relation, our $\mathbf{KHow}_0$ operator is equivalent to Lespérance et al.'s (2000) $\mathbf{Can}_L$ operator. Afterwards, we will show that when there are no exogenous actions, $\mathbf{KHow}_0$ is equivalent to **KHow** (Proposition 5.3.2), and so by transitivity, we will have related **KHow** to $\mathbf{Can}_L$.

In preparation for that, we introduce the notion of an endogenous-only PAT, which intuitively involves no exogenous actions.

Definition 5.3.8 (endogenous-only). A PAT Σ is *endogenous-only* if

$$\Sigma \models \mathsf{exoProgram} = \mathsf{nil} \wedge \forall a.\ \mathsf{Endo}(a).$$

Observe that for an endogenous-only PAT Σ, we have $\Sigma \models \forall s.\ \mathsf{Legal}^+(\mathsf{exoProgram}, s) \equiv \mathsf{Legal}(s)$.

Proposition 5.3.1. Suppose Σ is an endogenous-only PAT for a language without ab-

normality fluents (so the MPB and B relations are equivalent), such that

$$\Sigma \models \forall s \geq \mathsf{S}_0.\ \mathsf{B}(s,s) \wedge \forall s'.\ \mathsf{B}(s',s) \supset \forall s'' \geq s'.\ \mathsf{B}(s'',s''),$$

i.e., B is reflexive at legal successors of S_0, and at legal successors of situations that are accessible from legal successors of S_0. Suppose further Γ is a set of axioms describing the accessibility relation $\mathsf{K}_L(s',s)$ (used in defining $\mathbf{Know}_L$) so that $\Sigma \cup \Gamma$ entails

$$\begin{aligned}\forall s_1 \geq \mathsf{S}_0.\ \forall s_2.\ &[\mathsf{B}(s_2,s_1) \equiv \mathsf{K}_L(s_2,s_1)] \wedge \\ &[\mathsf{K}_L(s_2,s_1) \supset \forall s_3 \geq s_2.\ \forall s_4.\ [\mathsf{B}(s_4,s_3) \equiv \mathsf{K}_L(s_4,s_3)]],\end{aligned}$$

i.e., that the K_L and B relations agree on what's accessible from legal successors of S_0, and also on what's accessible from legal successors of what's accessible from legal successors of S_0. Then

$$\Sigma \cup \Gamma \models \forall s \geq \mathsf{S}_0.\ [\mathbf{Can}_L(\phi,s) \equiv \mathbf{KHow}_0(\phi,s)].$$

Proof. The first thing to note is that

$$\Sigma \cup \Gamma \models \forall s \geq \mathsf{S}_0, \pi, s'.\ \mathsf{K}_L(s',s) \supset [\mathsf{CanGet}_L(\phi,\pi,s') \equiv \mathsf{CanGet}_L^{\mathbf{TBel}}(\phi,\pi,s')]$$

since $\mathsf{CanGet}_L(\phi,\pi,s')$ and $\mathsf{CanGet}_L^{\mathbf{TBel}}(\phi,\pi,s')$ only differ in whether they use $\mathbf{Know}_L$ or $\mathbf{TBel}$, and only apply those operators to situations that are legal successors of s' – where by assumption the K_L and B relations agree, and where B is reflexive (so $\mathbf{Know}_L$ is equivalent to $\mathbf{Bel}$ which is equivalent to $\mathbf{TBel}$). Since the K_L and B relations also agree at legal successors of S_0, we then get

$$\begin{aligned}\Sigma \cup \Gamma \models \forall s \geq \mathsf{S}_0.\ [&\exists \pi\ \mathbf{Know}_L(\mathsf{CanGet}_L(\phi,\pi,now),s)\ \equiv \\ &\exists \pi\ \mathbf{Bel}(\mathsf{CanGet}_L^{\mathbf{TBel}}(\phi,\pi,now),s)].\end{aligned}$$

This is almost the result we wanted to show, except that it remains to show that

$$\Sigma \models \forall s \geq \mathsf{S}_0.\ \forall \pi.\ \mathbf{Bel}(\mathsf{CanGet}_L^{\mathbf{TBel}}(\phi,\pi,now),s) \supset \mathsf{CanGet}_L^{\mathbf{TBel}}(\phi,\pi,s),$$

i.e., that any policy the agent believes works, actually does. That follows from the assumption that $\Sigma \models \forall s \geq \mathsf{S}_0.\ \mathsf{B}(s,s)$, which means that the agent's beliefs are correct in legal successors of S_0. □

This result shows that $\mathbf{KHow}_0$ can be used as a replacement for $\mathbf{Can}_L$. So, for instance, we can also model the unbounded tree-chopping scenario from Lespérance et al. (2000, Examples 3–4), in which an agent is said to know how to cut down a tree even if it doesn't know in advance how many chops are needed, provided that it can sense whether the tree is down. Lespérance et al.'s point was that the $\mathbf{Can}_L$ operator handles unbounded iteration. $\mathbf{KHow}_0$ would handle the example the same. We consider a slightly more complicated variant of that problem in §5.5.

Our next proposition says that for endogenous-only PATs, **KHow** and $\mathbf{KHow}_0$ behave the same.

Proposition 5.3.2. Let Σ be an endogenous-only PAT. Then $\Sigma \models \forall s.\ \mathbf{KHow}(\phi, s) \equiv \mathbf{KHow}_0(\phi, s)$.

Proof. Since the environment program is nil, it is never the environment's turn to act. It can be seen that Σ entails each of

$$\forall \pi, \rho, s, s'.\ \mathsf{ExoOption}(\mathsf{nil}, \rho, s) \supset [\mathsf{OnPath}(\pi, \rho, s, s') \equiv \mathsf{OnPath}_L(\pi, s, s')]$$
$$\forall \pi, \rho, s, s'.\ \mathsf{ExoOption}(\mathsf{nil}, \rho, s) \supset [\mathsf{CanGet}(\phi, \pi, \rho, s) \equiv \mathsf{CanGet}_L^{\mathbf{TBel}}(\phi, \pi, s)]$$
$$\forall \pi, s, s'.\ \mathsf{CanAlwaysGet}(\mathsf{nil}, \phi, \pi, s) \equiv \mathsf{CanGet}_L^{\mathbf{TBel}}(\phi, \pi, s)$$

from which the result follows. □

It follows that **KHow** can also be used as a replacement for $\mathbf{Can}_L$. Of course, for PATs which are not endogenous-only, **KHow** would take the exogenous actions into account whereas $\mathbf{Can}_L$ would not.

The next two lemmas establish some properties relating to introspection that we will be using.

Lemma 5.3.1 (the agent believes its own beliefs are true). For any PAT Σ and formula ϕ (possibly with free variables, even second-order ones, though not the situation variable s),

$$\Sigma \models \forall s.\ \mathbf{Bel}(\forall(\mathbf{Bel}(\phi, now) \supset \phi), s).$$

Proof. Let $\mathfrak{I}$ be a model of Σ and μ a variable assignment. Suppose for contradiction that, for the situation variable s,

$$\mathfrak{I}, \mu \models \neg\mathbf{Bel}(\forall(\mathbf{Bel}(\phi, now) \supset \phi), s).$$

Then there is a variable assignment μ', differing from μ at most on s' (WLOG assume that s' does not appear free in ϕ), such that $\mathfrak{I}, \mu \models \mathsf{MPB}(s', s)$ and

$$\mathfrak{I}, \mu' \models \neg\forall(\mathbf{Bel}(\phi, s') \supset \phi[s']).$$

It follows that there is some variable assignment μ'' differing from μ at most on the free variables in ϕ (if any), such that

$$\mathfrak{I}, \mu'' \models \mathbf{Bel}(\phi, s') \wedge \neg\phi[s'].$$

By Lemma 5.2.2 we have that $\mathfrak{I}, \mu'' \models \mathsf{MPB}(s', s')$. So from $\mathfrak{I}, \mu'' \models \mathbf{Bel}(\phi, s')$ we would be forced to conclude $\mathfrak{I}, \mu' \models \phi[s']$, which is a contradiction. □

Lemma 5.3.2 (another introspective property). Suppose that π is a (possibly second-order) variable of any sort, and $\phi(\pi)$ is a formula with π as its only free variable. For any PAT Σ,

$$\Sigma \models \forall s.\ \mathbf{Bel}(\exists\pi.\ \mathbf{Bel}(\phi(\pi), now), s) \ \equiv\ \exists\pi.\ \mathbf{Bel}(\phi(\pi), s)$$

Proof. We prove each direction of the equivalence.

1: $\Sigma \models \forall s.\ \mathbf{Bel}(\exists\pi.\ \mathbf{Bel}(\phi(\pi), now), s) \supset \exists\pi.\ \mathbf{Bel}(\phi(\pi), s)$

Suppose that $\mathfrak{I}$ is a model of Σ and μ a variable assignment so that, for the situation variable s,

$$\mathfrak{I}, \mu \models \mathbf{Bel}(\exists\pi.\ \mathbf{Bel}(\phi(\pi), now), s)$$

We want to show that $\mathfrak{I}, \mu \models \exists\pi.\ \mathbf{Bel}(\phi(\pi), s)$.

Suppose μ' is a variable assignment differing from μ at most on s', and such that $\mathfrak{I}, \mu' \models \mathsf{MPB}(s', s)$. We then get that

$$\mathfrak{I}, \mu' \models \exists\pi.\ \mathbf{Bel}(\phi(\pi), s').$$

Therefore, there is some variable assignment μ'', agreeing with μ' except possibly on π, such that

$$\mathfrak{I}, \mu'' \models \mathbf{Bel}(\phi(\pi), s').$$

By Lemma 5.2.2 we have that $\mathfrak{I}, \mu'' \models \forall s''.\ \mathsf{MPB}(s'', s') \equiv \mathsf{MPB}(s'', s)$. Therefore,

$$\mathfrak{I}, \mu'' \models \mathbf{Bel}(\phi(\pi), s).$$

We can conclude that $\mathfrak{I}, \mu' \models \exists \pi.\ \mathbf{Bel}(\phi(\pi), s)$, and so also $\mathfrak{I}, \mu \models \exists \pi.\ \mathbf{Bel}(\phi(\pi), s)$.

2: $\Sigma \models \forall s.\ (\exists \pi\ \mathbf{Bel}(\phi(\pi), s)) \supset \mathbf{Bel}(\exists \pi.\ \mathbf{Bel}(\phi(\pi), now), s)$

Suppose that $\mathfrak{I}$ is a model of Σ and μ a variable assignment so that, for the situation variable s,

$$\mathfrak{I}, \mu \models \exists \pi.\ \mathbf{Bel}(\phi(\pi), s).$$

We want to show that $\mathfrak{I}, \mu \models \mathbf{Bel}(\exists \pi.\ \mathbf{Bel}(\phi(\pi), now), s)$.

By definition (the definition of the semantics of existential quantifiers) we have that there must be some variable assignment μ', differing from μ at most on π, such that

$$\mathfrak{I}, \mu' \models \mathbf{Bel}(\phi(\pi), s).$$

Consider any variable assignment μ'' differing from μ' at most on s', for which $\mathfrak{I}, \mu'' \models \mathsf{MPB}(s', s)$. By Lemma 5.2.2, we have that $\mathfrak{I}, \mu'' \models \forall s''.\ \mathsf{MPB}(s'', s') \equiv \mathsf{MPB}(s'', s)$. Therefore,

$$\mathfrak{I}, \mu'' \models \mathbf{Bel}(\phi(\pi), s'),$$

and we can weaken that statement to get

$$\mathfrak{I}, \mu'' \models \exists \pi.\ \mathbf{Bel}(\phi(\pi), s').$$

Since μ'' assigned s' to an arbitrary situation object, subject to the restriction that $\mathfrak{I}, \mu'' \models \mathsf{MPB}(s', s)$, we can conclude that

$$\mathfrak{I}, \mu' \models \mathbf{Bel}(\exists \pi.\ \mathbf{Bel}(\phi(\pi), now), s).$$

Since μ and μ' differ at most on π, which does not appear free on the RHS of the $\models$ operator above, we can conclude

$$\mathfrak{I}, \mu \models \mathbf{Bel}(\exists \pi.\ \mathbf{Bel}(\phi(\pi), now), s)$$

and are done. □

Those last two lemmas are used in a further lemma below, which will then be used to show that the agent always will believe that it has the know-how it does have (Proposition 5.3.3).

Lemma 5.3.3 (a sufficient condition for believing in know-how). For any PAT Σ,

$$\Sigma \models \forall s.\ [\exists \pi\ \mathbf{BHow}(\phi, \pi, s)] \supset \mathbf{Bel}(\mathbf{KHow}(\phi, now), s).$$

Proof. Fix some model $\mathfrak{I}$ of Σ, and a variable assignment μ such that

$$\mathfrak{I}, \mu \models \exists \pi\ \mathbf{BHow}(\phi, \pi, s).$$

We want to show that in the situation denoted by s it is believed that $\mathbf{KHow}(\phi, now)$, i.e., the conjunction

$$[\exists \pi\ \mathbf{BHow}(\phi, \pi, now)] \wedge [\forall \pi.\ \mathbf{BHow}(\phi, \pi, now) \supset \mathsf{CanAlwaysGet}(\mathsf{exoProgram}, \phi, \pi, now)].$$

To establish that the first conjunct is believed, by Lemma 5.3.2 we get that

$$\mathfrak{I}, \mu \models (\exists \pi\ \mathbf{BHow}(\phi, \pi, s)) \supset \mathbf{Bel}(\exists \pi.\ \mathbf{BHow}(\phi, \pi, now), s).$$

Furthermore, believing the second conjunct amounts to a special case of the agent believing that its own beliefs are true (Lemma 5.3.1). □

Proposition 5.3.3 (positive introspection for knowing-how). For any PAT Σ,

$$\Sigma \models \forall s.\ \mathbf{KHow}(\phi, s) \supset \mathbf{Bel}(\mathbf{KHow}(\phi, now), s).$$

Proof. This follows from Lemma 5.3.3, since it's part of the definition of **KHow** that if $\mathbf{KHow}(\phi, s)$ holds, then $\exists \pi\ \mathbf{BHow}(\phi, \pi, s)$. □

However, the agent may think that it knows how to do more than it really can, as the next proposition shows.

Proposition 5.3.4 (failure of negative introspection for knowing-how). There exists a PAT Σ for which

$$\Sigma \not\models \forall s.\ \neg\mathbf{KHow}(\phi, s) \supset \mathbf{Bel}(\neg\mathbf{KHow}(\phi, now), s).$$

Proof. Consider a PAT Σ including $\mathbf{InitB}(\mathsf{P})$, $\neg\mathsf{P}(\mathsf{S}_0)$, and the SSA $\mathsf{P}(\mathsf{do}(a,s)) \equiv \mathsf{P}(s)$. Then $\Sigma \models \neg\mathbf{KHow}(\mathsf{P},\mathsf{S}_0)$ since there is no way to make P true starting in S_0. However, in S_0 the agent believes that P is already true, and so believes $\mathbf{KHow}(\mathsf{P}, now)$. □

The following proposition shows that the task of proving that the agent believes that it knows how to do something can be simplified to showing that the agent believes that some policy π works.

Proposition 5.3.5. For any PAT Σ,

$$\Sigma \models \forall s.\ \mathbf{Bel}(\mathbf{KHow}(\phi, now), s) \equiv \exists \pi.\ \mathbf{BHow}(\phi, \pi, s)$$

Proof. One direction of the equivalence was shown by Lemma 5.3.3. What remains to be shown is that

$$\Sigma \models \forall s.\ \mathbf{Bel}(\mathbf{KHow}(\phi, now), s) \supset \exists \pi.\ \mathbf{BHow}(\phi, \pi, s).$$

From the definition of $\mathbf{KHow}$ we can conclude that

$$\Sigma \models \forall s.\ \mathbf{Bel}(\mathbf{KHow}(\phi, now), s) \supset \mathbf{Bel}(\exists \pi.\ \mathbf{BHow}(\phi, \pi, now), s).$$

Recall that $\mathbf{BHow}(\phi, \pi, now)$ expands to an expression of the form $\mathbf{Bel}(\psi(\pi), now)$. Hence, we get the desired result from the introspective properties of belief (Lemma 5.3.2). □

We now turn to considering how the $\mathsf{ExoRunning}$ operator relates to knowing-how. The following lemma says that if $\mathsf{ExoRunning}(\delta, s)$ holds, we can use δ instead of $\mathsf{exoProgram}$ when determining whether a policy ρ is an option that the environment could follow (starting from s).

Lemma 5.3.4. Let Σ be a PAT. Then for any ground program term δ,

$$\Sigma \models \forall s.\ \mathsf{ExoRunning}(\delta, s) \supset [\mathsf{ExoOption}(\mathsf{exoProgram}, \rho, s) \equiv \mathsf{ExoOption}(\delta, \rho, s)]$$

Proof. The only way $\mathsf{ExoOption}$ depends on its first argument is in determining what future situations are Legal^+. If $\mathsf{ExoRunning}(\delta, s)$ holds, then for any situation s' following s, we have that $\mathsf{Legal}^+(\mathsf{exoProgram}, s')$ holds iff $\mathsf{Legal}^+(\delta, s')$ holds. □

We can use that result to get that if the agent believes $\mathsf{ExoRunning}(\delta, now)$, then the agent will believe that whether it can achieve a goal ϕ by executing a policy π would be the same whether the environment acted according to $\mathsf{exoProgram}$ or δ.

Proposition 5.3.6. Let Σ be a PAT. Then for any ground program term δ,

$$\Sigma \models \forall s.\ \mathbf{Bel}(\mathsf{ExoRunning}(\delta, now), s) \supset \\ \mathbf{Bel}(\forall \pi.\ \mathsf{CanAlwaysGet}(\mathsf{exoProgram}, \phi, \pi, now) \equiv \mathsf{CanAlwaysGet}(\delta, \phi, \pi, now), s)$$

Proof. The only way CanAlwaysGet depends on its first argument is through ExoOption. The result follows from Lemma 5.3.4. □

We now return to considering normalizations:

Proposition 5.3.7. Let Σ be a PAT including $\mathsf{exoProgram} = \delta_{Exo}$, where δ_{Exo} can be written in a situation-suppressed way (i.e., the only situation term it refers to is now). Given a ground situation term σ, if there is an Ab account ξ such that

$$\Sigma \models \mathbf{Bel}(\xi, \sigma)$$

and δ is the normalization of δ_{Exo} w.r.t ξ, then

$$\Sigma \models \mathbf{Bel}(\forall \pi.\ \mathsf{CanAlwaysGet}(\mathsf{exoProgram}, \phi, \pi, now) \equiv \mathsf{CanAlwaysGet}(\delta, \phi, \pi, now), \sigma)$$

Proof. This follows from Proposition 5.2.4 and Proposition 5.3.6. □

Because the normalized program δ may be much simpler than exoProgram, this result can give an easier way of determining what the agent believes it knows how to do.

Sequential plans

We now consider properties relating to sequential plans. First, we want to relate being able to make ϕ true with a sequential plan to being able to make it true with a policy.

Proposition 5.3.8. Let Σ be a PAT where the SSA for view is any one of Equations 5.2, 5.3, or 5.4. Then for any k,

$$\Sigma \models \forall s, a_1, \ldots, a_k, \rho\ \exists \pi.\ \mathbf{Bel}(\mathsf{CanSeq}(\phi, [a_1, \ldots, a_k], \rho, now) \supset \\ \mathsf{CanGet}(\phi, \pi, \rho, now), s)$$

Proof. Suppose that $\mathfrak{I}$ is a model of Σ and μ an arbitrary variable assignment. We want to show that

$$\mathfrak{I}, \mu \models \exists \pi.\ \mathbf{Bel}(\mathsf{CanSeq}(\phi, [a_1, \ldots, a_k], \rho, now) \supset \mathsf{CanGet}(\phi, \pi, \rho, now), s)$$

Let's say that the subsequence of endogenous actions in the history of the situation $\mu[s]$ is $\hat{b}_1, \ldots, \hat{b}_m$ (note that the $\hat{b}_i$ are action objects in the domain, not terms). Now consider a variable assignment μ', differing from μ at most on the second-order variable π. Specifically, let's say that $\mu'[\pi] = \hat{\pi}$, the function from situation objects to action objects that is defined below (note that $\hat{s}$ ranges over situation objects and is not related to the variable s):

$$\hat{\pi}(\hat{s}) = \begin{cases} \mu'[a_1] & \text{if the endogenous actions in } \hat{s}\text{'s history are } \hat{b}_1, \ldots, \hat{b}_m \\ \mu'[a_2] & \text{if the endogenous actions in } \hat{s}\text{'s history are } \hat{b}_1, \ldots, \hat{b}_m, \mu'[a_1] \\ \vdots & \\ \mu'[a_{k-1}] & \text{if the endogenous actions in } \hat{s}\text{'s history are } \hat{b}_1, \ldots, \hat{b}_m, \mu'[a_1], \ldots, \\ & \mu'[a_{k-2}] \\ \mu'[a_k] & \text{otherwise} \end{cases}$$

Now consider a variable assignment μ'' differing from μ' at most on s', and suppose that

$$\mathfrak{I}, \mu'' \models \mathsf{MPB}(s', s).$$

Furthermore, suppose that

$$\mathfrak{I}, \mu'' \models \mathsf{CanSeq}(\phi, [a_1, \ldots, a_k], \rho, s').$$

We will show that

$$\mathfrak{I}, \mu'' \models \mathsf{CanGet}(\phi, \pi, \rho, s'),$$

which will establish the overall result that we want.

Observe that the endogenous actions in the history of $\mu''[s']$ must be the same as in the history of $\mu''[s]$ (because any of the possible SSAs for view listed in the statement of this proposition will result in the agent always knowing what endogenous actions have been performed). We can get the desired result by noting that

$$\Sigma, \mu'' \models \forall s''.\ \mathsf{AfterSeq}([\alpha_1, \ldots, \alpha_k], \rho, s', s'') \supset \mathsf{OnPath}(\pi, \rho, s', s'')$$

and that starting in $\mu''[s']$ the agent will always know what action $\hat{\pi}$ recommends. □

We may note that the result of Proposition 5.3.8 would also apply to PATs with other

SSAs for view, so long as the agent always knows which endogenous actions have been performed. Furthermore, we also get this easy corollary:

Corollary 5.3.1. Under the conditions of Proposition 5.3.8, for any k,

$$\Sigma \models \forall s, a_1, \ldots, a_k.\ \mathbf{BHowSeq}(\phi, [\alpha_1, \ldots, \alpha_k], s) \supset \mathbf{Bel}(\mathbf{KHow}(\phi, now), s)$$

Finally, the last results of this section are the analogue of Proposition 5.3.6 and Proposition 5.3.7 for sequential plans.

Proposition 5.3.9. Let Σ be a PAT. Then for any ground program term δ and number k,

$$\begin{aligned}\Sigma \models \forall s.\ \mathbf{Bel}(\mathsf{ExoRunning}(\delta, now), s) \supset \\ \forall a_1, \ldots, a_k.\ \mathbf{Bel}(\mathsf{CanAlwaysSeq}(\mathsf{exoProgram}, \phi, [a_1, \ldots, a_k], now) \equiv \\ \mathsf{CanAlwaysSeq}(\delta, \phi, [a_1, \ldots, a_k], now), s).\end{aligned}$$

Proof. Similarly to what we had with CanAlwaysGet in Proposition 5.3.6, the only way CanAlwaysSeq depends on its first argument is through ExoOption. The result follows from Lemma 5.3.4. □

Proposition 5.3.10. Let Σ be a PAT including $\mathsf{exoProgram} = \delta_{Exo}$, where δ_{Exo} can be written in a situation-suppressed way (i.e., the only situation term it refers to is now). Given a ground situation term σ, if there is an Ab account ξ such that

$$\Sigma \models \mathbf{Bel}(\xi, \sigma)$$

and δ is the normalization of δ_{Exo} w.r.t ξ, then, for any number k,

$$\begin{aligned}\Sigma \models \forall a_1, \ldots, a_k.\ \mathbf{Bel}(\mathsf{CanAlwaysSeq}(\mathsf{exoProgram}, \phi, [a_1, \ldots, a_k], now) \equiv \\ \mathsf{CanAlwaysSeq}(\delta, \phi, [a_1, \ldots, a_k], now), \sigma)\end{aligned}$$

Proof. This follows from Proposition 5.2.4 and Proposition 5.3.9. □

Having established all these properties, we now are ready to apply some of them to examples.

5.4 An extended example

To illustrate how our framework allows for describing beliefs about knowing how and the change of those beliefs, we consider a version of the well-known fox-chicken-grain problem (Ascher, 1990) that we mentioned in the introduction, where a farmer is trying to carry a fox, a chicken, and some grain north across a river, but can only carry them one at a time in his boat. The fox will eat the chicken if the farmer isn't on the same river bank as them, and similarly the chicken will eat the grain if the farmer isn't with them.

In the classic version of the problem, it's well-known that the goal can be accomplished if the farmer does the following:

1. carry the chicken across,
2. come back,
3. carry the fox across,
4. come back with the chicken,
5. carry the grain across,
6. come back,
7. and carry the chicken across.

With this sequence of actions, there is no point at which the chicken and grain, or the fox and the chicken, are left together unattended by the farmer.

In our version, exogenous actions (like eating) are explicitly represented, and also some (implausible) things can happen that may affect the farmer's ability to achieve his goal. The chicken can fly across the river (which may help or hinder the farmer, depending on the circumstances), though this is an implausible event that the farmer assumes won't happen. The fox might be sleepy, in which case it won't eat the chicken but may spend some time sleeping. The weather may act as well – a storm may damage the boat, causing a leak. A leaky boat can only be used for two more crossings before it fills with water. Below we construct a PAT for this problem, before describing what the agent (the farmer) believes he knows how to do in various situations.

The endogenous actions are $\uparrow(x)$ (the farmer crosses to the north side of the river, taking x with him) and $\downarrow(x)$ (the farmer crosses to the south side of the river, taking x with him). The exogenous actions are 🚀 (the chicken flies to the opposite side of the river), $\mathsf{eat}(x, y)$ (x eats y), zzz (the fox sleeps), and 🌧 (a storm damages the boat, causing it to leak).

The fluents are $\mathsf{Eaten}(x, s)$ (x has been eaten), $\mathsf{FarmerMoved}(s)$ (the farmer just performed an action), $\mathsf{Damaged}(s)$ (the boat has a leak), $\mathsf{North}(x, s)$ (x is on the north side of the river), and $\mathsf{level}(s)$ (a numeric value indicating how much water is in the boat). We will also make use of three abnormality fluents, $\mathsf{Ab}_1(x, s)$, $\mathsf{Ab}_2(x, s)$, and $\mathsf{Ab}_3(s)$. We consider any object not on the north side of the river to be on the south side (we do not model the river itself as a location) and so we use

$$\mathsf{SameSide}(x, y, s) \stackrel{\text{def}}{=} (\mathsf{North}(x, s) \equiv \mathsf{North}(y, s))$$

to indicate that x and y are on the same side of the river.

We have the constant symbols 🧑‍🌾, 🦊, 🐔, 🌾 for the farmer, fox, chicken, and grain, respectively. We use another constant, "_", as an argument to $\uparrow$ and $\downarrow$ when the farmer crosses empty-handed. The theory specifies that all these symbols denote distinct objects.

The PAT is described in Figure 5.2. We use the SSA from Equation 5.2 for view, so that the agent always knows what actions have occurred. The precondition axioms make it so that all (non-null) actions are always possible, except for $\uparrow(x)$ and $\downarrow(x)$, which require the objects crossing the river to start on the appropriate side, and that the water level in the boat not be too high.

Observe that in Σ_0, the following formula $\phi[\mathsf{S}_0]$ describes S_0:

$$\begin{aligned}&\forall x.\ \neg\mathsf{North}(x, \mathsf{S}_0) \wedge \neg\mathsf{Eaten}(x, \mathsf{S}_0) \wedge \\ &\neg\mathsf{Damaged}(\mathsf{S}_0) \wedge \mathsf{level}(\mathsf{S}_0) = 0 \wedge \neg\mathsf{FarmerMoved}(\mathsf{S}_0)\end{aligned}$$

That is, the objects start on the south side of the river and are not eaten, the boat is not leaking, and the agent has not yet moved. Furthermore, the agent knows these things; we have $\mathbf{InitB}(\phi[now])$.

The environmental program $\mathsf{exoProgram}$ specifies that the weather, fox and chicken take turns acting according to their programs; that is, the theory includes $\mathsf{exoProgram} = \delta_{Exo}$ where δ_{Exo} is

$$\big(\mathsf{weatherProg};\ \mathsf{chickenProg};\ \mathsf{foxProg};\ \mathsf{FarmerMoved}?\big)^*$$

where each of the procedures ($\mathsf{weatherProg}$, $\mathsf{chickenProg}$, and $\mathsf{foxProg}$) is described in Figure 5.3. (For simplicity we have the programs run in sequence rather than concurrently.) Note that the environment program blocks after giving the weather, chicken, and fox a turn so as to give the farmer a turn.

The weather program says that implausibly a storm may occur. Note that the purpose

$$
\begin{aligned}
\Sigma_{\mathsf{ssa}} = \{&\mathsf{FarmerMoved}(\mathsf{do}(a,s)) \equiv \mathsf{Endo}(a),\\
&\mathsf{Eaten}(x,\mathsf{do}(a,s)) \equiv \exists y.\ a = \mathsf{eat}(y,x) \vee \mathsf{Eaten}(x,s),\\
&\mathsf{Damaged}(\mathsf{do}(a,s)) \equiv a = \text{🌧} \vee \mathsf{Damaged}(s),\\
&\mathsf{North}(x,\mathsf{do}(a,s)) \equiv\\
&\quad \big([a = \uparrow(x) \vee (x = \text{🧑‍🌾} \wedge \exists y.\ a = \uparrow(y))] \vee\\
&\quad [\neg\mathsf{North}(x,s) \wedge x = \text{🐔} \wedge a = \text{🚀}] \vee\\
&\quad [\mathsf{North}(x,s) \wedge ([x = \text{🧑‍🌾} \wedge \neg\exists y.\ a = \downarrow(y)] \vee\\
&\qquad\qquad [x \neq \text{🧑‍🌾} \wedge a \neq \downarrow(x) \wedge \neg(x = \text{🐔} \wedge a = \text{🚀})])]\big),\\
&\mathsf{level}(\mathsf{do}(a,s)) = y \equiv\\
&\quad \big([(y = \mathsf{level}(s)+1) \wedge \mathsf{Damaged}(s) \wedge \exists x\ (a = \uparrow(x) \vee a = \downarrow(x))] \vee\\
&\quad [(y = \mathsf{level}(s)) \wedge \neg[\mathsf{Damaged}(s) \wedge \exists x\ (a = \uparrow(x) \vee a = \downarrow(x))]]\big),\\
&\mathsf{view}(\mathsf{do}(a,s)) = a \cdot \mathsf{view}(s),\\
\} \cup \{&\mathsf{Ab}_i(\vec{x},\mathsf{do}(a,s)) \equiv \mathsf{Ab}_i(\vec{x},s) \quad | \quad \mathsf{Ab}_i \text{ is an abnormality fluent}\}.
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{\mathsf{pre}} = \{&\mathsf{Poss}(\uparrow(x),s) \equiv \neg\mathsf{North}(\text{🧑‍🌾},s) \wedge (\neg\mathsf{North}(x,s) \vee x = \_) \wedge (\mathsf{level}(s) < 2),\\
&\mathsf{Poss}(\downarrow(x),s) \equiv \mathsf{North}(\text{🧑‍🌾},s) \wedge (\mathsf{North}(x,s) \vee x = \_) \wedge (\mathsf{level}(s) < 2),\\
&\mathsf{Poss}(\mathsf{null},s) \equiv \mathsf{False}\\
\} \cup \{&\mathsf{Poss}(\alpha(\vec{x}),s) \equiv \mathsf{True} \quad | \quad \alpha \text{ is not } \uparrow, \downarrow, \text{ or } \mathsf{null}\}.
\end{aligned}
$$

$$
\Sigma_{\mathsf{sense}} = \{\mathsf{SF}(\alpha(\vec{x}),s) \equiv \mathsf{True} \quad | \quad \alpha \text{ is an action function symbol}\}.
$$

$$
\begin{aligned}
\Sigma_0 = \{&\neg\mathsf{North}(x,\mathsf{S}_0) \wedge \neg\mathsf{Eaten}(x,\mathsf{S}_0) \wedge\\
&\quad \neg\mathsf{Damaged}(\mathsf{S}_0) \wedge \mathsf{level}(\mathsf{S}_0) = 0 \wedge \neg\mathsf{FarmerMoved}(\mathsf{S}_0),\\
&\mathsf{exoProgram} = (\mathsf{weatherProg};\ \mathsf{chickenProg};\ \mathsf{foxProg};\ \mathsf{FarmerMoved}?)^*,\\
&\mathsf{Endo}(\uparrow(x)) \wedge \mathsf{Endo}(\downarrow(x)),\\
&\mathsf{Exo}(\text{🚀}) \wedge \mathsf{Exo}(\mathsf{eat}(x,y)) \wedge \mathsf{Exo}(\mathsf{zzz}) \wedge \mathsf{Exo}(\text{🌧}) \wedge \mathsf{Exo}(\mathsf{null}),\\
&\mathsf{Init}(s) \supset \mathsf{view}(s) = \langle\rangle\\
\} \cup \{&\tau_1 \neq \tau_2 \mid \tau_1,\tau_2 \in \{\text{🧑‍🌾},\text{🦊},\text{🐔},\text{🌾},\_\}\} \cup \{\text{the axioms describing lists}\}.
\end{aligned}
$$

$$
\mathbf{InitB}(\forall x.\ \neg\mathsf{North}(x) \wedge \neg\mathsf{Eaten}(x) \wedge \neg\mathsf{Damaged} \wedge \mathsf{level} = 0 \wedge \neg\mathsf{FarmerMoved}).
$$

Figure 5.2: Axioms in the PAT for the fox-chicken-grain problem. The procedures referred to in the exogenous program are given in Figure 5.3.

proc weatherProg **if** Ab_1(view) **then** 🌧 **endIf**; **endProc**;

proc chickenProg
  if Ab_2(view) **then** 🚀 **endIf**;
  if SameSide(🐔, 🌾) ∧ ¬SameSide(🧑‍🌾, 🌾) **then** eat(🐔, 🌾);
  endIf;
endProc;

proc foxProg
  if ¬Ab_3 **then**
    if SameSide(🦊, 🐔) ∧ ¬SameSide(🧑‍🌾, 🐔) **then** eat(🦊, 🐔);
    endIf;
  else (nil | zzz); **endIf**;
endProc;

Figure 5.3: The procedures used by the exogenous program.

of making view the argument to Ab_1 is that view, as a list of the actions that have been executed, takes different values over time, and so a new abnormal atom is required to be true for each storm that happens (so after one storm, another storm is still implausible).[3] Meanwhile, the chicken's program says it may implausibly fly across the river (and as with storms, each flight requires a separate abnormality to occur), and will eat grain if on the same bank as it when the farmer isn't there. Finally, according to the fox's program, the fox will eat the chicken if it can, unless it's sleepy (corresponding to Ab_3 being true) in which case the fox non-deterministically does nothing or sleeps. Note that whether the fox is sleepy doesn't change over time.

The farmer's goal can be given by this formula:

$$\mathsf{Goal}(s) \stackrel{\text{def}}{=} \mathsf{North}(🦊, s) \wedge \mathsf{North}(🐔, s) \wedge \mathsf{North}(🌾, s) \wedge \neg\mathsf{Eaten}(🐔, s) \wedge \neg\mathsf{Eaten}(🌾, s)$$

We now turn to considering whether the farmer believes he can accomplish his goal, and how. For brevity in talking about beliefs about knowing how, let's introduce some

[3] In Chapter 4, we used the value of the history fluent as an argument to abnormalities for similar purposes.

notation (where $\vec{\alpha}$ and $\vec{\beta}$ are sequences of actions):

$$\begin{aligned}
\vec{\alpha} \twoheadrightarrow \phi &\stackrel{\text{def}}{=} \mathbf{Bel}(\mathbf{KHow}(\phi, now), \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)) \\
\vec{\alpha} \twoheadrightarrow\!\!\!| \, \phi &\stackrel{\text{def}}{=} \mathbf{Bel}(\neg\mathbf{KHow}(\phi, now), \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)) \\
\vec{\alpha} \twoheadrightarrow \phi : \vec{\beta} &\stackrel{\text{def}}{=} \mathbf{BHowSeq}(\phi, \vec{\beta}, \mathsf{do}(\vec{\alpha}, \mathsf{S}_0)) \\
\vec{\alpha} \not\twoheadrightarrow \phi : \vec{\beta} &\stackrel{\text{def}}{=} \neg\mathbf{BHowSeq}(\phi, \vec{\beta}, \mathsf{do}(\vec{\alpha}, \mathsf{S}_0))
\end{aligned}$$

So, for example, $\vec{\alpha} \twoheadrightarrow \phi : \vec{\beta}$ is a formula that says that after the actions $\vec{\alpha}$ have occurred, the agent believes that it can bring about ϕ by performing $\vec{\beta}$ (possibly interleaved with exogenous actions).

Proposition 5.4.1. The action theory described above has the following belief formulas as logical consequences:

1. $\twoheadrightarrow$ Goal : [↑(🐔), ↓(_), ↑(🦊), ↓(🐔), ↑(🌾), ↓(_), ↑(🐔)]
2. 🚀 $\not\twoheadrightarrow$ Goal : [↑(🐔), ↓(_), ↑(🦊), ↓(🐔), ↑(🌾), ↓(_), ↑(🐔)]
3. 🚀 $\twoheadrightarrow$ Goal : [↑(🦊), ↓(🐔), ↑(🌾), ↓(_), ↑(🐔)]
4. [↑(🐔), ↓(_), ↑(🦊), 🚀] $\twoheadrightarrow\!\!\!|$ Goal
5. 🌧️ $\twoheadrightarrow\!\!\!|$ Goal
6. zzz $\twoheadrightarrow$ Goal : [↑(🐔), ↓(_), ↑(🦊), ↓(_), ↑(🌾)]
7. [↑(🐔), ↓(_), ↑(🦊), zzz] $\twoheadrightarrow$ Goal : [↓(_), ↑(🌾)]
8. [↑(🐔), ↓(_), ↑(🦊), 🌧️, zzz] $\twoheadrightarrow$ Goal : [↓(_), ↑(🌾)]
9. [↑(🐔), ↓(_), ↑(🦊), ↓(🐔), 🚀] $\twoheadrightarrow\!\!\!|$ Goal
10. [↑(🐔), zzz, ↓(_), ↑(🦊), ↓(🐔), 🚀] $\twoheadrightarrow$ Goal : ↑(🌾)

Below, we informally describe the points in this proposition and why they are true. Somewhat more detailed proof sketches follow.

1. The farmer initially assumes all abnormalities are false and so believes he can achieve his goal by the usual sequence of crossings (the same as in the classic problem).

2. If the first thing that happens is that the chicken flies across the river (🚀), the farmer no longer believes the original plan will work (he can't take the chicken that is already on the other side).

3. However, if the first thing that happens is that the chicken flies across the river, the farmer believes a simpler plan will work.

4. In the situation considered here, the chicken flies back to the original bank after the farmer arrives with the fox, and will be able to eat the grain before the farmer gets another turn. Therefore, the farmer now believes he does not know how to achieve his goal.

5. If a storm is the first thing to happen, the farmer knows he doesn't have enough crossings left to finish.

6. After learning (from the zzz action) that the fox is sleepy and won't eat the chicken, the farmer knows that it's safe for the fox and chicken to be alone together, and can follow a simpler plan.

7. This is like the previous point except that the farmer only comes to know that the fox is sleepy at a later point.

8. Because of the storm, the farmer knows he only has two crossings left, but because the fox is sleepy, the farmer knows that suffices.

9. In this situation, the chicken has flown to be alone with the fox on the north bank, and so will be eaten before the farmer gets to do anything. Hence the farmer believes his goal cannot be achieved.

10. This resembles (9) above, but this time the farmer still believes he knows how to achieve his goal, because the fox has been observed to sleep (zzz), showing that it is sleepy and won't eat the chicken.

Proof. We consider each of the points.

1. In S_0 the agent believes the Ab account according to which all abnormalities are false. The normalization of δ_{Exo} with respect to that Ab account has the following definitions for the procedures:

$$\textbf{proc}\ \mathsf{weatherProg}\ \textbf{if}\ \mathsf{False}\ \textbf{then}\ \text{🌧}\ \textbf{endIf};\ \textbf{endProc};$$

proc chickenProg
  if False **then** 🚀 **endIf**;
  if SameSide(🐔, 🌾) ∧ ¬SameSide(🧑‍🌾, 🌾) **then** eat(🐔, 🌾);
  endIf;
endProc;
proc foxProg
  if ¬False **then**
    if SameSide(🦊, 🐔) ∧ ¬SameSide(🧑‍🌾, 🐔) **then** eat(🦊, 🐔);
    endIf;
  else (nil | zzz); **endIf**;
endProc;

Simplifying that program further by removing branches that can never be taken, we get the following:

proc weatherProg nil; **endProc**;
proc chickenProg
  if SameSide(🐔, 🌾) ∧ ¬SameSide(🧑‍🌾, 🌾) **then** eat(🐔, 🌾);
  endIf;
endProc;
proc foxProg
  if SameSide(🦊, 🐔) ∧ ¬SameSide(🧑‍🌾, 🐔) **then** eat(🦊, 🐔);
  endIf;
endProc;

That is, the weather does nothing, the only thing the chicken does is eat the grain when it can, and only thing the fox does is eat the chicken when it can.

Let δ be the simplified program with those procedures. By Proposition 5.2.4, we have

$$\Sigma \models \mathbf{Bel}(\mathsf{ExoRunning}(\delta, now), \mathsf{S}_0).$$

We want to show that

$$\Sigma \models \textbf{BHowSeq}(\mathsf{Goal}, [\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(🐔), \uparrow(🌾), \downarrow(\_), \uparrow(🐔)], \mathsf{S}_0).$$

By Proposition 5.3.9 it will suffice to show

$$\begin{aligned}\Sigma \models \textbf{Bel}(\mathsf{CanAlwaysSeq}(&\\ &\delta, \mathsf{Goal}, [\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(🐔), \uparrow(🌾), \downarrow(\_), \uparrow(🐔)], \mathit{now}), \mathsf{S}_0).\end{aligned} \tag{5.8}$$

To show that, let $\mathfrak{I}$ be any model of Σ, and μ a variable assignment such that

$$\mathfrak{I}, \mu \models \mathsf{MPB}(s, \mathsf{S}_0).$$

We have to show that

$$\mathfrak{I}, \mu \models \mathsf{CanAlwaysSeq}(\delta, \mathsf{Goal}, [\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(🐔), \uparrow(🌾), \downarrow(\_), \uparrow(🐔)], s),$$

i.e., that $\mathfrak{I}, \mu$ satisfy

$$\begin{aligned}&\forall \rho.\ \mathsf{ExoOption}(\delta, \rho, s) \supset\\ &\qquad \mathsf{CanSeq}(\mathsf{Goal}, [\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(🐔), \uparrow(🌾), \downarrow(\_), \uparrow(🐔)], \rho, s).\end{aligned}$$

Therefore, let μ' be any variable assignment agreeing with μ except possibly on ρ, and such that

$$\mathfrak{I}, \mu' \models \mathsf{ExoOption}(\delta, \rho, s).$$

From the definition of ExoOption and what δ is, it's easy to see that (starting from the situation denoted by s) the policy denoted by ρ must always select the action denoted by null when neither the chicken and grain nor the fox and chicken are left together unattended. That is, if we let

$$\begin{aligned}\mathsf{Safe}(t) \stackrel{\text{def}}{=}\ \neg\big([&\mathsf{SameSide}(🐔, 🌾, t) \wedge \neg\mathsf{SameSide}(🧑‍🌾, 🌾, t)] \vee\\ &[\mathsf{SameSide}(🦊, 🐔, t) \wedge \neg\mathsf{SameSide}(🧑‍🌾, 🐔, t)]\big)\end{aligned}$$

then we have

$$\mathfrak{I}, \mu' \models \forall s' \sqsupseteq s.\ \mathsf{Safe}(s') \supset (\rho(s') = \mathsf{null}).$$

It's also straight-forward to verify that

$$\mathfrak{I}, \mu' \models \forall s'.\ s' \sqsubseteq \mathsf{do}([\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(🐔), \uparrow(🌾), \downarrow(\_), \uparrow(🐔)], s) \supset \mathsf{Safe}(s').$$

That is, starting in the situation denoted by s, if the farmer was able to follow his plan without interruption by exogenous actions, at every situation along the way, that situation would be "safe". This means that the policy denoted by ρ must select the action denoted by null, and so the farmer does in fact get to follow his plan without interruption (note that the precondition of each action will be satisfied when it's executed). That means that we have

$$\mathfrak{I}, \mu' \models \mathsf{AfterSeq}([\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(🐔), \uparrow(🌾), \downarrow(\_), \uparrow(🐔)], \rho, s, \\ \mathsf{do}([\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(🐔), \uparrow(🌾), \downarrow(\_), \uparrow(🐔)], s)).$$

It's also straight-forward to verify that

$$\mathfrak{I}, \mu' \models \mathbf{TBel}(\mathsf{Goal}, \mathsf{do}([\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(🐔), \uparrow(🌾), \downarrow(\_), \uparrow(🐔)], s)).$$

Therefore, we can conclude that

$$\mathfrak{I}, \mu' \models \mathsf{CanSeq}(\mathsf{Goal}, [\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(🐔), \uparrow(🌾), \downarrow(\_), \uparrow(🐔)], \rho, s)$$

and so we can get Equation 5.8 and are done.

2. Consider the situation $\mathsf{do}(🚀, \mathsf{S}_0)$. Note that in all accessible situations from there, the 🚀 action has been performed (and no others), because all accessible situations have the same value for the view fluent, which records the actions that have occurred. Therefore, the farmer believes that the chicken is on the North side, i.e., we have

$$\Sigma \models \mathbf{Bel}(\mathsf{North}(🐔), \mathsf{do}(🚀, \mathsf{S}_0)).$$

Therefore, the farmer believes that the first action of his original plan, $\uparrow(🐔)$, is not currently executable:

$$\Sigma \models \mathbf{Bel}(\neg\mathsf{Poss}(\uparrow(🐔), now), \mathsf{do}(🚀, \mathsf{S}_0)).$$

Finally, it can be shown that the farmer believes the next action to be executed

will be his own:

$$\Sigma \models \mathbf{Bel}(\forall \rho.\ \mathsf{ExoOption}(\mathsf{exoProgram}, \rho, now) \supset [\rho(now) = \mathsf{null}], \mathsf{do}(🚀, \mathsf{S}_0)).$$

From these we can conclude that Σ entails

$$\mathbf{Bel}(\forall \rho.\ \mathsf{ExoOption}(\mathsf{exoProgram}, \rho, now) \supset \neg\exists s.\ \mathsf{AfterSeq}([\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(🐔), \uparrow(🌾), \downarrow(\_), \uparrow(🐔)], \rho, now, s), \mathsf{do}(🚀, \mathsf{S}_0))$$

and so the result follows.

3. In $\mathsf{do}(🚀, \mathsf{S}_0)$, it can be shown that the farmer believes the Ab account specifying that $\mathsf{Ab}_2(\langle\rangle, now)$ is true and every other abnormality is false (note that if that one abnormality had not been true, then 🚀 would not have been possible). Therefore, the farmer believes that a simplified program is running as was shown in (1) above, except that chickenProg is slightly more complicated:

   proc chickenProg
   if view $= \langle\rangle$ **then** 🚀 **endIf**;
   if SameSide(🐔, 🌾) $\wedge$ ¬SameSide(🧑‍🌾, 🌾) **then** eat(🐔, 🌾);
   endIf;
   endProc;

   However, the value of view will not be equal to $\langle\rangle$ in future situations, so this says that the chicken won't fly again. Hence, trying to achieve Goal from $\mathsf{do}(🚀, \mathsf{S}_0)$ is like the classical problem but with a starting position where the chicken is already across. The result can be shown similarly to in (1) above.

4. Consider the situation $\sigma = \mathsf{do}([\uparrow(🐔), \downarrow(\_), \uparrow(🦊), 🚀], \mathsf{S}_0)$. It's straight-forward to verify that the farmer believes that the chicken and grain are now on the same side (opposite the farmer):

   $$\Sigma \models \mathbf{Bel}(\mathsf{SameSide}(🐔, 🌾, now) \wedge \neg\mathsf{SameSide}(🧑‍🌾, 🌾, now), \sigma)$$

   Therefore, it can be seen that

   $$\Sigma \models \mathbf{Bel}(\forall \rho.\ \mathsf{ExoOption}(\mathsf{exoProgram}, \rho, now) \supset [\rho(now) = \mathsf{eat}(🐔, 🌾)], \sigma)$$

So the next action to occur will be the chicken eating the grain, which will result in a situation from which the goal is not achievable.

5. It can be seen that

$$\Sigma \models \mathbf{Bel}(\mathsf{Damaged}(now) \wedge \mathsf{level}(now) = 0, \mathsf{do}(🌧, \mathsf{S}_0)).$$

As a result of this, the farmer believes that in any legal future situation in which he's already performed two-river crossing actions, he won't be able to cross a third time. That is, it can be shown that

$$\begin{aligned}\Sigma \models \mathbf{Bel}(\forall s_1, a_1, s_2, a_2, s_3.\ \big[(now \leq \mathsf{do}(a_1, s_1) < \mathsf{do}(a_2, s_2) \leq s_3) \wedge \\ \exists x_1, x_2.\ (a_1 = \uparrow(x_1) \vee a_1 = \downarrow(x_1)) \wedge (a_2 = \uparrow(x_2) \vee a_2 = \downarrow(x_2))\big] \supset \\ \forall x_3.\ \neg[\mathsf{Poss}(\uparrow(x_3), s_3) \vee \mathsf{Poss}(\downarrow(x_3), s_3)].\end{aligned}$$

It can be verified (without even considering the program) that no sequence of endogenous and exogenous actions in which the farmer crosses the river at most twice will result in all three of the fox, chicken, and grain being taken across the river.

6. In $\mathsf{do}(\mathsf{zzz}, \mathsf{S}_0)$ it can be shown the farmer believes the Ab account according to which $\mathsf{Ab}_3(now)$ is true and every other abnormality is false. The normalization of δ_{Exo} with respect to that Ab account has the same definitions for the procedures that we saw in (1) above, except that the fox's normalized procedure is this:

proc foxProg
  if ¬True **then**
    if SameSide(🦊, 🐔) ∧ ¬SameSide(🧑‍🌾, 🐔) **then** eat(🦊, 🐔);
    endIf;
  else (nil | zzz); **endIf**;
endProc

Simplifying that program further by removing branches that can never be taken, we get the following:

proc foxProg (nil | zzz); **endProc**

So the fox can only do nothing or sleep. As in (1), the weather does nothing and all the chicken can do is eat the grain if on the same side of the river as it while the farmer is on the other side.

The proof proceeds similarly to in (1), but considering the simpler plan $[\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(\_), \uparrow(🌾)]$. Note that unlike in (1), during the plan the farmer may be interrupted by an exogenous action, but that can only be the fox performing zzz again, which doesn't change anything.

7. This is similar to (6). In the situation considered here the farmer has seen the fox sleep, and believes the same Ab account, the one according to which the only true abnormality is $\mathsf{Ab}_3(now)$. So the farmer believes the same simplified programs are running that we considered in (6).

8. This is similar to (7). The only difference is that there's been a storm, so the boat is damaged and the farmer only has two crossings left. However, the plan to achieve the goal from this point (which is the same as in (7)) only requires two crossings, so is unaffected.

9. It can be shown that in $\mathsf{do}([\uparrow(🐔), \downarrow(\_), \uparrow(🦊), \downarrow(🐔), 🚀], \mathsf{S}_0)$ the farmer does *not* believe $\mathsf{Ab}_3(now)$. Therefore, the farmer believes the next action to occur will be the fox eating the chicken, after which the goal is not achieved in any future situation.

10. It can be shown that in $\mathsf{do}([\uparrow(🐔), \mathsf{zzz}, \downarrow(\_), \uparrow(🦊), \downarrow(🐔), 🚀], \mathsf{S}_0)$ the farmer believes the Ab account according to which $\mathsf{Ab}_2(\langle\uparrow(🐔), \mathsf{zzz}, \downarrow(\_), \uparrow(🦊)\rangle, now)$ and $\mathsf{Ab}_3(now)$ are true, and all other abnormalities are false. The important thing to note is that because $\mathsf{Ab}_3(now)$ is believed, the farmer believes the fox is running a program which says it won't eat the chicken (the program we previously saw in (6)). Therefore, given the current situation (in which the fox and chicken are already on the north side), the farmer believes he can achieve his goal by taking the grain across. □

This example has illustrated how the agent's beliefs about what it can achieve, and how it can achieve it, can change over time. The same principles would apply to more complicated problems; e.g., one could consider a variant where the animals can move around on the banks of the river and have to be caught.

5.5 Knowing-how in the unbounded case

In the examples we've considered so far, the agent has had simple (sequential) ways of bringing about its goal (or no way). However, our definition of knowing-how is in terms of policies, and so can be applied to a much broader range of problems. For example, it could be applied to problems which require policies that behave like *conditional plans*, which branch on sensing results. (Note that Lespérance et al. (2008) had already formalized conditional planning in the presence of exogenous actions, though without plausibility.)

More interestingly, our approach also is designed to handle problems which require policies with iterative behavior – unlike the approach in (Lespérance et al., 2008), but like the earlier (Lespérance et al., 2000). Lespérance et al. (2000) demonstrated that their approach to modelling knowing-how, which ours is based on, could handle loops by considering a problem where the agent is trying to chop down a tree but doesn't know how many chops it will take. If the agent can sense whether the tree is down, then it will be said to know how to bring it down on their account (using $\mathbf{Can}_L$), because it can continue chopping and sensing until it knows that it's done.

We now consider an analogous problem – trying to rake away all the leaves under a tree without knowing how many leaves there are – but with the extra complication that more leaves can exogenously fall while the agent acts. Note that this example does not involve any abnormalities, and is just meant to showcase the combination of iterative planning and exogenous actions.

We construct a PAT to model the scenario. There's a fluent $\mathsf{AgentTurn}(s)$ to indicate when the agent gets to act, and numeric-valued functional fluents $\mathsf{leavesOnGround}(s)$ and $\mathsf{leavesOnTree}(s)$ to model how many leaves are on the ground and the tree, respectively. Furthermore, there are these actions:

- rake – an endogenous action that reduces by one the number of leaves under the tree
- sense – an endogenous sensing action that checks if there are still leaves under the tree
- drop – an exogenous action that moves one leaf from the tree to the ground
- pass – an exogenous action that passes the turn to the agent

The PAT is shown in Figure 5.4. Note how the actions are always possible to execute, except that leaves can only be raked if there are some on the ground, and can only fall

if there are still some on the tree. Furthermore, the only action that senses anything is sense, and it's described by this sensing axiom:

$$\mathsf{SF}(\mathsf{sense}, s) \equiv \mathsf{leavesOnGround}(s) > 0.$$

We use the SSA for view from Equation 5.3 so that the agent observes all actions and also gets sensing results.

The exogenous program specifies that whenever it's the environment's turn, either a leaf can fall or the turn can be passed to the agent:

$$\mathsf{exoProgram} = (\neg\mathsf{AgentTurn}?; (\mathsf{drop} \mid \mathsf{pass}))^*$$

Initially, the agent doesn't know anything. In particular, the agent doesn't know how many leaves are on the ground or in the tree.

Proposition 5.5.1. Let Σ be the PAT described in Figure 5.4. Then

$$\Sigma \models \mathbf{KHow}(\mathsf{leavesOnGround}(now) = 0, \mathsf{S}_0)$$

Proof. Consider any model $\mathfrak{I} = \langle \mathcal{D}, \mathcal{I} \rangle$ of Σ. Let μ be an arbitrary variable assignment such the second-order variable π is mapped to this function from situations to actions:

$$\hat{\pi}(\hat{s}) = \begin{cases} \mathcal{I}[\mathsf{rake}] & \text{if the last endogenous action in } \hat{s} \text{ was } \mathcal{I}[\mathsf{sense}], \text{ with a positive} \\ & \text{sensing result} \\ \mathcal{I}[\mathsf{sense}] & \text{otherwise} \end{cases}$$

We will show that

$$\mathfrak{I}, \mu \models \mathbf{BHow}(\mathsf{leavesOnGround}(now) = 0, \pi, \mathsf{S}_0).$$

To further show that every policy the action believes will bring about the goal actually does, note that $\mathfrak{I}, \mu \models \mathsf{B}(\mathsf{S}_0, \mathsf{S}_0)$, and so (since there are no abnormality fluents) also $\mathfrak{I}, \mu \models \mathsf{MPB}(\mathsf{S}_0, \mathsf{S}_0)$. So the agent has no false beliefs initially, and in particular, any policy that the agent believes works must actually work starting in the situation denoted by S_0.

Suppose that

$$\mathfrak{I}, \mu \models \mathsf{B}(s, \mathsf{S}_0).$$

$$
\begin{aligned}
\Sigma_{\mathsf{ssa}} = \{&\mathsf{AgentTurn}(\mathsf{do}(a,s)) \equiv a = \mathsf{pass},\\
&\mathsf{leavesOnGround}(\mathsf{do}(a,s)) = n \equiv\\
&\qquad\big[[a = \mathsf{rake} \wedge (n+1 = \mathsf{leavesOnGround}(s))] \vee\\
&\qquad\ [a = \mathsf{drop} \wedge (n = \mathsf{leavesOnGround}(s)+1)] \vee\\
&\qquad\ [(a \neq \mathsf{rake} \wedge a \neq \mathsf{drop}) \wedge n = \mathsf{leavesOnGround}(s)]\big],\\
&\mathsf{leavesOnTree}(\mathsf{do}(a,s)) = n \equiv [a = \mathsf{drop} \wedge (n+1 = \mathsf{leavesOnTree}(s))] \vee\\
&\qquad\qquad [a \neq \mathsf{drop} \wedge n = \mathsf{leavesOnTree}(s)],\\
&\mathsf{view}(\mathsf{do}(a,s)) = y \equiv [(\mathsf{SF}(a,s) \wedge y = \langle a, 1\rangle \cdot \mathsf{view}(s)) \vee\\
&\qquad\qquad (\neg\mathsf{SF}(a,s) \wedge y = \langle a, 0\rangle \cdot \mathsf{view}(s))]\\
\}.&
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{\mathsf{pre}} = \{&\mathsf{Poss}(\mathsf{rake}, s) \equiv \mathsf{leavesOnGround}(s) > 0,\\
&\mathsf{Poss}(\mathsf{sense}, s) \equiv \mathsf{True},\\
&\mathsf{Poss}(\mathsf{drop}, s) \equiv \mathsf{leavesOnTree}(s) > 0,\\
&\mathsf{Poss}(\mathsf{pass}, s) \equiv \mathsf{True},\\
&\mathsf{Poss}(\mathsf{null}, s) \equiv \mathsf{False}\}.
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{\mathsf{sense}} = \{&\mathsf{SF}(\mathsf{sense}, s) \equiv \mathsf{leavesOnGround}(s) > 0\\
&\} \cup \{\mathsf{SF}(\alpha, s) \equiv \mathsf{True} \quad | \quad \alpha \text{ is an action function symbol other than } \mathsf{sense}\}.
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_0 = \{&\mathsf{exoProgram} = (\neg\mathsf{AgentTurn}?; (\mathsf{drop} \mid \mathsf{pass}))^*,\\
&\mathsf{Endo}(\mathsf{rake}) \wedge \mathsf{Endo}(\mathsf{sense}) \wedge \mathsf{Exo}(\mathsf{drop}) \wedge \mathsf{Exo}(\mathsf{pass}) \wedge \mathsf{Exo}(\mathsf{null}),\\
&\mathsf{Init}(s) \supset \mathsf{view}(s) = \langle\rangle\\
&\} \cup \{\text{the axioms describing lists}\}.
\end{aligned}
$$

$$\mathbf{InitB}(\mathsf{True}).$$

Figure 5.4: Axioms for the leaf-raking domain.

(Note that s must denote an initial situation.) We want to show that

$$\mathfrak{I}, \mu \models \mathsf{CanAlwaysGet}(\mathsf{exoProgram}, \mathsf{leavesOnGround}(now) = 0, \pi, s).$$

First, it's easy to see that $\hat{\pi}$ will always recommend possible actions – the environment can't take leaves off the ground, so if the last sensing action showed leaves were on the ground, then leaves are still on the ground, and the action denoted by rake is possible. Furthermore, since the agent remembers its own actions and sensing results it will always know what $\hat{\pi}$ recommends.

Next, note that there are some natural numbers n and m such that

$$\mathfrak{I}, \mu \models \mathsf{leavesOnGround}(s) = n \wedge \mathsf{leavesOnTree}(s) = m.$$

It can be shown that, starting from s, it will *always eventually* be the agent's turn. In fact, it will always be the agent's turn in at most $m + 1$ steps, because the environment can only avoid passing the turn to the agent while there are still leaves on the tree. That is, we can get the following:

$$\begin{aligned}\mathfrak{I}, \mu \models \forall \rho, s'.\ [\mathsf{ExoOption}(\mathsf{exoProgram}, \rho, s) \wedge \mathsf{OnPath}(\pi, \rho, s, s')] \supset \\ \textstyle\bigvee_{k=0}^{m+1} \exists a_1, \ldots, a_k.\ \mathsf{OnPath}(\pi, \rho, s, \mathsf{do}([a_1, \ldots, a_k], s')) \wedge \\ \mathsf{AgentTurn}(\mathsf{do}([a_1, \ldots, a_k], s'))\end{aligned}$$

Therefore, the agent will eventually have performed $2 \times (n+m+1)$ actions. If the results of all sensing during those $2 \times (n + m + 1)$ actions was always positive, then the agent would have performed the action denoted by rake $n+m+1$ times, which is not possible, since there were at most $n + m$ leaves that could potentially be raked. Therefore, the agent must have got a negative sensing result at some point, at which point it would correctly believe that the goal of $\mathsf{leavesOnGround}(now) = 0$ was satisfied. □

Note that in this example the agent does not believe that any sequential plan will achieve the goal, since any finite sequence of actions may be too short to rake away all the leaves. Though we do not consider it here, the policy described in the proof could be represented syntactically with something like an *FSA plan* (Hu and Levesque, 2010; Hu, 2012). We leave that to future work. Also note that a similar result can be shown with a modified domain with infinitely many leaves on the tree, so long as the environment program has them fall at a slower rate than the agent can rake them away and sense.

5.6 Discussion and related work

As has been mentioned, Kelly and Pearce (2015) had suggested a notion of belief like ours (though without plausibility) in their section on future work. That follows a line of work (outside the situation calculus) in distributed systems and multi-agent epistemic logics, where what occurs is constrained by a *protocol* that agents may know (see e.g. Halpern and Fagin, 1989; Fagin et al., 1995; Pacuit and Simon, 2011; van Lee et al., 2019). Protocols and plausibility have been combined in a few frameworks, though apparently not with the same purpose as ours. For example, van Benthem and Dégremont (2010) treated protocols only as semantic objects (sets of histories) and did not discuss how to formally specify them. Halpern and Moses (2004) introduced belief and plausibility only for the purpose of reasoning about counterfactuals.

A general approach in AI for representing exogenous activity is to embed the results of possible (implicit) exogenous actions within the outcomes of non-deterministic endogenous actions (e.g., Kuter and Nau, 2004). We argue that explicitly representing exogenous actions and the program controlling them can allow for some domains to be more naturally described. *Reactive synthesis* does involve explicitly modeling the activity of the environment (Pnueli and Rosner, 1989), sometimes with constraints on what the environment can do expressed using a temporal logic (see, e.g., Chatterjee et al., 2008; Bloem et al., 2014; Camacho et al., 2018; Aminof et al., 2018), but this line of work has not featured plausible beliefs.

Our notion of Legal^+ situations makes use of a ConGolog program to specify a subset of a situation tree (the situations that can be reached from the root by following the program). Another way to describe a subtree of situations was given by Pinto (1998), who suggested a collection of predicates for describing constraints on the occurrences of events in "legal" situations. Yet another option is to use the *non-Markovian* precondition axioms of Gabaldon (2011), which allow for the possibility of executing an action to depend on more than just the current state.

Knowing how has been studied in both philosophy and artificial intelligence. There are surveys by Gochet (2013) and Ågotnes et al. (2015). We now turn to considering how our work fits in this area. A common distinction is how we've incorporated plausibility.

A closely related work that we've already mentioned is (Lespérance et al., 2008), from which we take how we model the interaction between the exogenous and endogenous process using prioritized concurrency. They used "knowledge" in a metalogical sense (i.e., what is known to the agent is what the action theory entails), and did not consider plausibility or belief revision. Furthermore, as they note their approach to knowing-how

does not support iterative plans, unlike the earlier work (Lespérance et al., 2000) which our approach generalizes.

The joint ability of a group of agents has also been considered in the situation calculus, including in work by Ghaderi and collaborators (Ghaderi et al., 2007; Ghaderi, 2011). Note that their definition of a joint ability operator **JCan** involves (potentially false) beliefs, like our definitions of **KHow** and $\mathbf{KHow}_0$, and also universal quantification over policies. They do not specifically discuss how joint ability behaves when the group has only one agent. We leave it to future work to investigate the exact relation between their joint ability operator and our knowing-how operators.

Another related approach is De Giacomo et al.'s (2010) "GameGolog" language, which allows for describing which agent has control over which aspects of non-determinism in a program. De Giacomo et al. describe properties (e.g. that a group has a strategy to achieve a goal) using a language based on the μ-calculus. However, they did not deal with plausibility or revising beliefs, and as they note, actions are fully observable and there aren't sensing actions.

Also working in the situation calculus, Xiong and Liu (2016) considered strategies in a multiagent setting with partial observability of actions (though agents know how many actions occurred). In spite of introducing a "true belief" operator they made very little use of it, and none of their four alternative definitions of individual ability (p. 1325) requires that the goal can actually be made true (but just that the agent believes it). They also did not consider plausibility or revising beliefs.

Alur et al. (1997, 2002) proposed alternating time temporal logic (ATL), which extends the branching-time temporal logic CTL with a parameterized operator $\langle\langle A\rangle\rangle$, where A is a set of agents. The formula $\langle\langle A\rangle\rangle\varphi$ means that the group A has a collective strategy that ensures φ is satisfied (note that φ is not a final goal, but a formula that can describe temporally extended properties). ATL does not include operators for knowledge or belief, but alternating-time temporal epistemic logic (ATEL) extends ATL with knowledge operators (van der Hoek and Wooldridge, 2003). Many other variants of ATL have been studied in the literature. There is a logic called "ATL with plausibility" (Bulling et al., 2008), but the purpose of plausibility there seems rather different from ours; Bulling et al. were concerned with making such plausible assumptions as requiring "the agents to play only Nash equilibria, Pareto-optimal profiles or undominated strategies".

The requirement in Lespérance et al.'s (2000) CanGet_L operator (and our CanGet operator) that the agent knows what actions the policy recommends is much like how a *uniform strategy* in a game with partial information must select the same move at all nodes in the player's information set (van Benthem, 2001, p. 230). Uniform strategies

have often been considered in the literature on knowing how (e.g., Jamroga and van der Hoek, 2004; Fervari et al., 2017).

Our definition of knowing how to does not try to distinguish between what is *caused* by the agent and environment. For example, if the agent can predict that the sun will rise, and tell when it has risen, then we may say that the agent knows how to achieve having the sun being up. Other approaches are possible. Consider the "see to it that" (stit) operator of Belnap and Perloff (1988). To say that agent "saw to it that φ" requires, roughly speaking, that there was a choice the agent made which guaranteed φ, and that when that choice was made, the agent could have done something different which wouldn't have guaranteed φ.

Naumov and Tao (2019) proposed a modal logic with separate modalities for knowledge and knowing-how, each of which was indexed with an "uncertainty parameter". This parameter does not appear to be very closely related to our notion of plausibility, though, as they considered that it "represents the precision with which the agent can determine the position (state) of the whole system in an arbitrary metric space". Naumov and Tao were concerned with handling examples such as whether a self-driving truck knows how to avoid a collision, depending on the precision of the truck's radar in estimating the speed of other vehicles.

The approach proposed in this chapter has defined knowing-how in a situation-dependent way, following Lespérance et al. (2000). In contrast, Wang (2018) was concerned with capturing the following intuition (p. 4422):

> Knowing how to achieve a goal may not entail that you can realize the goal now [...] a broken-arm pianist may still know how to play piano even if he cannot play right now [...]

To deal with this, Wang proposed a modal logic with a knowing-how operator $\mathbf{Kh}(\psi, \varphi)$ that means that the agent can achieve φ whenever ψ is true. The truth of $\mathbf{Kh}(\psi, \varphi)$ does not depend on the state in which it is evaluated (in particular, it doesn't matter if ψ is currently true). If in this chapter's framework it were desired to have a situation-independent knowing-how operator, one could be defined with something like the following:

$$\mathbf{KHow'}(\psi, \varphi) \stackrel{\text{def}}{=} \forall s.\ (\mathsf{Legal}^{+}(\mathsf{exoProgram}, s) \wedge \psi[s]) \supset \mathbf{KHow}(\varphi, s).$$

However, that $\mathbf{KHow'}$ operator does not allow for what the agent knows how to do to change over time. A more interesting operator might be

$$\mathbf{KHow''}(\psi, \varphi, s) \stackrel{\text{def}}{=} \forall s' \sqsupseteq s.\ (\mathsf{Legal}^{+}(\mathsf{exoProgram}, s') \wedge \psi[s']) \supset \mathbf{KHow}(\varphi, s').$$

which intuitively says that henceforth (starting from s) the agent can achieve φ from any point where ψ is true.

Before concluding, it's worth noting that a motivating example for McCarthy's original work on circumscription (McCarthy, 1980) was the "missionaries and cannibals" problem, another river-crossing problem similar to the fox-chicken-grain one we discussed.

5.7 Conclusion

We have presented two main contributions in this chapter. First, we presented an approach to modeling defeasible belief in the situation calculus where the accessible situations over time are constrained to be reachable by following a ConGolog program. This allows for representing what the agent believes the environment is doing (and that some alternatives are more plausible than others), and for the agent to change or retract such beliefs. Second, our new definition of knowing how to achieve goals, made in terms of belief, takes into account both how beliefs may be false and the running of exogenous processes. These beliefs can also be changed or retracted in response to observations.

We also considered sequential plans in some detail. As mentioned previously, future work could relate syntactic representations for non-sequential plans – e.g., FSA plans (Hu and Levesque, 2010; Hu, 2012) or robot programs (Lin and Levesque, 1998) – to knowing-how in our setting.

Another future direction would be to address the computation of beliefs and plans. Kelly and Pearce (2015) described how regression could be used with a knowledge operator defined with a view fluent in some cases. Also, in their future work section, where they suggested using a ConGolog program to constrain the accessibility relation (as we have done), they also suggested that a technique of Fritz et al. (2008) – which compiles an action theory and ConGolog program into a new action theory whose legal situations are those reachable by the program in the original theory – could be used to simplify the problem of computing entailed beliefs. Note that the possibility of compiling away the program does not imply that it's not useful to use programs as a high-level specification for exogenous behavior.

The relation between plausibility and knowing-how could be further explored. One might want to consider a more robust version of knowing-how, where the agent does not just think that a policy will succeed in the most plausible cases, but also in others that aren't too implausible (similar to "fault-tolerant planning" (Jensen et al., 2004; Domshlak, 2013)). Similarly, for some outcomes of the policy to be certain could be useful. For example, one might want a policy that will most plausibly cause the coffee

cup to be clean and in the cupboard, and is certain not to break the cup.

We have only been modelling the beliefs of one agent (though other actors can be considered to some extent by representing their behavior as ConGolog processes). Future work could include generalizing the approach to a truly multi-agent setting. Finally, while we have (as in the rest of this thesis) taken a qualitative approach to uncertainty where plausibility levels give rise to categorical beliefs, the general idea of using a program to describe the environment would also be compatible with probabilistic representations of uncertainty.

Chapter 6

Conclusion

6.1 Summary and contributions

In this thesis, we have seen how abnormality fluents can be used in situation calculus action theories to describe the plausibility of various aspects of dynamic environments – states, actions, and processes. In each case, this supports belief change about that aspect, as once the more plausible options are ruled out, the agent will believe the next most plausible options. We now review what each of the last three chapters has accomplished.

Specifying plausibility levels We extended the framework of Shapiro et al. (2011) by assigning plausibility levels to initial situations by counting abnormalities (also taking into account priorities). We introduced a form of action theories, called IAATs, that specify what the agent considers plausible by employing the counting of (unchanging) abnormalities along with only-knowing. We saw that this approach has advantages over alternatives like Schwering and Lakemeyer's (2014) "only-believing" operator. We also considered a couple variants of IAATs. First, we considered DIAATs, which don't require that the agent know the true dynamics axioms, and showed that those mostly satisfied the AGM postulates. With MAATs we explored allowing abnormality fluents to change over time.

Changing beliefs about domain dynamics We studied how to represent the plausibility of aspects of actions – their effects, preconditions, and sensing results. We proposed a number of patterns to follow when writing successor state axioms that refer to abnormalities in an IAAT, so as to control the extent to which beliefs about action effects would change as a result of observations. We also presented results about using regression with IAATs, and in particular showed how believed SSAs (and not just those written in the

action theory) could be incorporated into the regression procedure.

Environment processes and knowing-how We followed an idea from Kelly and Pearce (2015) and considered a model of belief in which the agent knows that a ConGolog program is running. We showed that by having the program refer to abnormalities, some executions could be believed more plausible than others. Finally, we generalized the definition of knowing-how from Lespérance et al. (2000) to accommodate false beliefs and exogenous actions occurring according to a program. We also considered a version of knowing-how that was limited to goals that could be achieved by sequential plans.

6.2 Future work

In this section we suggest a few ways the approach of this thesis could be extended or applied.

6.2.1 Plausibility in other frameworks

We've been considering plausibility within the framework of the situation calculus. However, as was pointed out in Chapter 3, the approach of specifying plausibility levels by counting abnormalities was first used in a modal temporal logic (Klassen et al., 2017). Future work could look at applying the approach in other formalisms, like the fluent calculus (Thielscher, 1998).

6.2.2 Belief update

We have not focused very much on belief update in this thesis. Recall that belief update (as opposed to belief revision) involves belief change in response to changes in the world as opposed to merely gaining information (Katsuno and Mendelzon, 1991). Shapiro et al. (2011) did have results about belief update, but that basically just amounted to the agent knowing what the effect of so-called "update actions" were. Delgrande and Levesque (2012) had the following to say about belief update (in reference to their own framework, but which could have been said of Shapiro et al.'s):

> Katsuno-Mendelzon style update doesn't make much sense from the point of view of the agent. Recall that in update, a formula ϕ is recorded as being true following the execution of some action, and the task is to determine what else is true. In our framework, an agent is fully aware of the effects of the actions it believes that it has executed; and so its beliefs are simply the image of its previous beliefs under this intended action.

On the other hand, another way of thinking about belief update was suggested by Boutilier (1996). In his framework, exogenous events occur, unobserved by the agent, and the agent then updates its beliefs after making an observation. The way the agent updates its beliefs is by considering what events could have occurred to result in the observation, and what other changes those events would have produced. Since we've also considered exogenous events in this thesis, it seems like a natural direction for future work to translate Boutilier's conception of belief update into the situation calculus.

6.2.3 Elaboration tolerance and applications to fiction

Elaboration tolerance (McCarthy, 2003; Amir, 2001; Parmar, 2003) is a desirable property of formalisms. McCarthy (2003) wrote that

> A formalism is *elaboration tolerant* to the extent that it is convenient to modify a set of facts expressed in the formalism to take into account new phenomena or changed circumstances. [..] The simplest kind of elaboration is the addition of new formulas.

An obvious way to look at elaboration tolerance in the setting of this thesis would be to ask, for example, whether can we easily add new sentences to the agent's knowledge base Σ_{KB} in an IAAT to change what the agent believes in desired ways. However, we can look deeper, and ask if by adding new sentences to Σ_{KB} we can get a desired modification to the *plausibility ordering* (not just to what's most plausible).

One potential application of this is for interpreting fiction. What's plausible in fiction is somewhat based on what's plausible in reality, but there can be differences. Philosophers have suggested that when people read fiction, they

1. "carry over" knowledge of the real world into the fiction (when the story does not contradict it) so as to conclude, for example, that there are no purple gnomes in the world of Sherlock Holmes (see, e.g., Lewis, 1978; Ryan, 1991; Rapaport and Shapiro, 1995; Badura and Berto, 2018);

2. and may also bring in knowledge of what other stories are like – for example, a fictional dragon may be presumed to breathe fire (see, e.g., Lewis, 1978; Walton, 1990; Bonomi and Zucchi, 2003; Abell, 2012).

The examples are from Lewis (1978).

So, suppose we start with a plausibility ordering on possible situations induced by Σ_{KB}. This ordering describes what the agent thinks is plausible in reality. We might want to modify that ordering (by modifying Σ_{KB}) to get a plausibility ordering to describe

what's plausible in fiction (for use in an automated story understanding system, for example). We still want to retain much of the information from Σ_{KB} (so as to carry over knowledge about the real world).

We'll discuss an example to illustrate how this might work. Let's say the agent is reading a (thus-far) realistic story. We may want it to assume that pigs don't talk, because they don't in reality – but the agent should be able to revise its belief about pigs in the story depending on what it reads next. If Σ_{KB} already contains a sentence like

$$(\mathsf{Pig}(x, now) \wedge \neg \mathsf{Ab}_1(x, now)) \supset \neg \mathsf{Talks}(x, now) \tag{6.1}$$

(because the agent allows for pigs implausibly talking in reality) then perhaps that can be used as-is for describing fictional plausibility as well.[1] If however Σ_{KB} contains sentences that don't admit exceptions (e.g., if it categorically states that pigs don't talk), then to get the desired plausibility ordering for fiction we may have to modify those sentences.

One approach would be to replace a sentence of the form $\forall \vec{x}.\ \phi(\vec{x}, now)$ appearing in Σ_{KB} with a "defeasible copy" $\forall \vec{x}.\ \neg \mathsf{Ab}_i(\vec{x}, now) \supset \phi(\vec{x}, now)$. This was proposed by Klassen et al. (2017, §4.1), and can be thought of as a first-order version of the transformation suggested by Amir (2001, §4.2) from a propositional theory to the "associated abnormality theory". Parmar (2003, §11.4) also suggested a similar first-order version. (Note that Amir and Parmar were not using cardinality-based minimization of abnormalities, though.)

To illustrate, if we started with the sentence $\mathsf{Pig}(x, now) \supset \neg \mathsf{Talks}(x, now)$ we might replace that with $\neg \mathsf{Ab}_1(x, now) \supset [\mathsf{Pig}(x, now) \supset \neg \mathsf{Talks}(x, now)]$ which is logically equivalent to Equation 6.1. That is perhaps a more surgical change than just adding a sentence to the knowledge base, but it might be automated (though there is a choice to make regarding how many arguments the new abnormality predicate should take).

Finally, consider that in certain genres of fiction, talking pigs are not surprising. In general, we may want to overrule axioms in Σ_{KB}. Note that if every sentence in Σ_{KB} is (or is rewritten to be) of the form $\neg \mathsf{Ab}_i(now) \supset \phi_i(now)$ (where Ab_i does not otherwise appear in the knowledge base), then the effect of each can be cancelled by adding another sentence saying $\mathsf{Ab}_i(now)$ is true. That affords a degree of elaboration tolerance in a simple way (Amir (2001) and Parmar (2003) made similar points). We leave further investigation to future work.

[1] One might want to change the priority of the abnormality.

Appendix A

Dual theories and the AGM postulates

The purpose of this appendix is to prove this proposition from Chapter 3:

Proposition 3.5.4.
Let Σ be a DIAAT. For any model $\mathfrak{I}$ of Σ, and any ground situation term $\sigma = \mathsf{do}(\vec{\beta}, \mathsf{S}_0)$, all the AGM postulates other than (AGM$*$5) are satisfied when revision is defined.

We will prove each of the postulates separately in §A.2. First, we'll establish some preparatory results. The proofs in this appendix closely parallel those given by Shapiro (2005, §3.4.6) and Shapiro et al. (2011, Appendix A), which however used slightly different definitions and didn't apply to DIAATs.

A.1 Preparatory results

Let's first note that the agent will be certain that revision actions don't change the world.

Lemma A.1.1. Let $\sigma = \mathsf{do}(\vec{\beta}, \mathsf{S}_0)$ be a ground situation term, and suppose that α is a revision action for ϕ. For any $\psi \in \mathcal{L}_{\mathsf{now}}$,

$$\Sigma \models \forall s.\ \mathsf{B}(s, \sigma) \supset \Big(\psi[s] \equiv \psi[\mathsf{do}(\alpha, s)]\Big)$$

Proof. Recall that a revision action is defined so that in accessible situations it doesn't change the value of any fluent. That it doesn't change the truth of a sentence in $\mathcal{L}_{\mathsf{now}}$ can be proved by induction. □

Note that this differs from the analogous result by Shapiro (2005, Lemma 3.4.12), which was that revision actions (as defined there) actually don't change the world.

The next lemma (similar to Shapiro, 2005, Lemma 3.4.28) says that if a most plausible accessible situation gives the same sensing results for a revision action α as the actual situation the agent is in, then after performing α that situation (technically, its successor) is still a most plausible accessible situation.

Lemma A.1.2. Suppose α is a revision action for ϕ. Then

$$\Sigma \models \forall s \sqsupseteq \mathsf{S}_0, s'.\ [\mathsf{SF}(\alpha, s) \wedge \mathsf{SF}(\alpha, s') \wedge \mathsf{MPB}(s', s)] \supset \mathsf{MPB}(\mathsf{do}(\alpha, s'), \mathsf{do}(\alpha, s)).$$

Proof. Let $\mathfrak{I}$ be a model of Σ, and μ any variable assignment such that

$$\mathfrak{I} \models (\mathsf{S}_0 \sqsubseteq s) \wedge \mathsf{SF}(\alpha, s) \wedge \mathsf{SF}(\alpha, s') \wedge \mathsf{MPB}(s', s)$$

(if there are no such variable assignments, then the result is trivial). By the definition of revision actions, the agent is certain that the action denoted by α is possible, so using the SSA for B (which $\mathfrak{I}$ makes true) it can be seen that $\mathfrak{I}, \mu \models \mathsf{B}(\mathsf{do}(\alpha, s'), \mathsf{do}(\alpha, s))$. We need to additionally show that the situation denoted by $\mathsf{do}(\alpha, s')$ is one of the most plausible among the accessible situations from the situation denoted by $\mathsf{do}(\alpha, s)$. That follows from s' denoting one of the most plausible accessible situations from the denotation of s (the plausibility of the situation denoted by $\mathsf{do}(\alpha, s')$ is the same as that of the situation denoted by s'). □

So, the only way a most plausible accessible situation can cease to be a most plausible accessible situation is if it becomes inaccessible.

Finally, the next lemma (similar to Shapiro, 2005, Lemma 3.4.30) shows that if ϕ is not disbelieved before a revision action for ϕ is performed, then afterwards any most plausible accessible situation is the successor of a situation that was previously one of the most plausible accessible situations.

Lemma A.1.3. Suppose α is a revision action for ϕ. Then

$$\Sigma \models \forall s \sqsupseteq \mathsf{S}_0.\ \Big(\mathsf{SF}(\alpha, s) \wedge \neg \mathbf{Bel}(\neg\phi, s)\Big) \supset \Big(\forall s''.\ \mathsf{MPB}(s'', \mathsf{do}(\alpha, s)) \supset \exists s'.\ \mathsf{MPB}(s', s) \wedge (s'' = \mathsf{do}(\alpha, s)) \wedge \phi[s']\Big).$$

Proof. Let $\mathfrak{I}$ be a model of Σ, and μ_1 any variable assignment such that

$$\mathfrak{I}, \mu_1 \models (s \sqsupseteq \mathsf{S}_0) \wedge \mathsf{SF}(\alpha, s) \wedge \neg \mathbf{Bel}(\neg\phi, s).$$

We want to show that

$$\mathfrak{I}, \mu_1 \models \forall s''.\ \mathsf{MPB}(s'', \mathsf{do}(\alpha, s)) \supset \exists s'.\ \mathsf{MPB}(s', s) \wedge (s'' = \mathsf{do}(\alpha, s)) \wedge \phi[s']$$

Let μ_2 be any variable assignment agreeing with μ_1, except possibly on s''. Suppose that $\mathfrak{I}, \mu_2 \models \mathsf{MPB}(s'', \mathsf{do}(\alpha, s))$. We now want to show that

$$\mathfrak{I}, \mu_2 \models \exists s'.\ \mathsf{MPB}(s', s) \wedge (s'' = \mathsf{do}(\alpha, s')) \wedge \phi[s'].$$

From the supposition that $\mathfrak{I}, \mu_2 \models \mathsf{MPB}(s'', \mathsf{do}(\alpha, s))$ and the SSA for B we can conclude that there must be a variable assignment μ_3, agreeing with μ_2 except possibly on s', such that

$$\mathfrak{I}, \mu_3 \models (s'' = \mathsf{do}(\alpha, s')) \wedge \mathsf{B}(s', s) \wedge \mathsf{Poss}(\alpha, s) \wedge (\mathsf{SF}(\alpha, s) \equiv \mathsf{SF}(\alpha, s'))$$

Furthermore, because α is a revision action for ϕ, we can conclude that $\mathfrak{I}, \mu_3 \models (\mathsf{SF}(\alpha, s') \equiv \phi[s'])$ and so $\mathfrak{I}, \mu_3 \models \phi[s']$. So all that remains is to show that $\mathfrak{I}, \mu_3 \models \mathsf{MPB}(s', s)$.

We have supposed that $\mathfrak{I}, \mu_3 \models \neg\mathbf{Bel}(\neg\phi, s)$, so there is a variable assignment μ_4, agreeing with μ_3 except possibly on s^*, such that $\mathfrak{I}, \mu_4 \models \mathsf{MPB}(s^*, s) \wedge \phi[s^*]$. Because α is a revision action for α, it can be seen that $\mathfrak{I}, \mu_4 \models \mathsf{B}(\mathsf{do}(\alpha, s^*), \mathsf{do}(\alpha, s))$. However, since we had supposed that s'' (which denotes the same thing as $\mathsf{do}(\alpha, s')$) denoted one of the most plausible situations accessible from the situation denoted by $\mathsf{do}(\alpha, s)$, it must be that the situation denoted by $\mathsf{do}(\alpha, s')$ is at least as plausible as the situation denoted by $\mathsf{do}(\alpha, s^*)$, and so the situation denoted by s' must be at least as plausible as the situation denoted by s^*. In conclusion, $\mathfrak{I}, \mu_4 \models \mathsf{MPB}(s', s)$. □

A.2 Proving the AGM properties

We now are ready to prove the AGM postulates. As previously mentioned, the proofs are very similar to the corresponding ones by Shapiro (2005) and Shapiro et al. (2011).

Proposition A.2.1 (AGM∗1). Under the conditions of Proposition 3.5.4, $K(\sigma * \phi)$ is deductively closed.

Proof. This follows from belief being deductively closed. □

Proposition A.2.2 (AGM∗2). Under the conditions of Proposition 3.5.4, $\phi \in K(\sigma * \phi)$.

Proof. Let α be a revision action for ϕ. We want to show that, given that $\mathfrak{I} \models \mathsf{SF}(\alpha, \sigma)$, then $\mathfrak{I} \models \mathbf{Bel}(\phi, \mathsf{do}(\alpha, \sigma))$.

If there are no accessible situations in $\mathsf{do}(\alpha, \sigma)$, the result follows trivially. Otherwise, let μ be any variable assignment such that $\mathfrak{I}, \mu \models \mathsf{MPB}(s'', \mathsf{do}(\alpha, \sigma))$. We will have established what we want to show if we get that $\mathfrak{I}, \mu \models \phi[s'']$. By the SSA for B (Equation 2.11), we have that there is some variable assignment μ' (agreeing with μ except possibly on s') such that

$$\mathfrak{I}, \mu' \models \mathsf{B}(s', \sigma) \wedge (s'' = \mathsf{do}(\alpha, s')) \wedge \mathsf{Poss}(\alpha, s') \wedge (\mathsf{SF}(\alpha, s') \equiv \mathsf{SF}(\alpha, \sigma))$$

Since $\mathfrak{I} \models \mathsf{SF}(\alpha, \sigma)$, we can simplify that to get

$$\mathfrak{I}, \mu' \models \mathsf{B}(s', \sigma) \wedge (s'' = \mathsf{do}(\alpha, s')) \wedge \mathsf{Poss}(\alpha, s') \wedge \mathsf{SF}(\alpha, s')$$

Since α is a revision action for ϕ,

$$\mathfrak{I}, \mu' \models \mathsf{Poss}(\alpha, s') \wedge [\mathsf{SF}(\alpha, s') \equiv \phi[s']] \wedge \left[\bigwedge_{F \text{ a fluent}} \forall \vec{x}.\ F(\vec{x}, s') \equiv F(\vec{x}, \mathsf{do}(\alpha, s')) \right]$$

Therefore, $\mathfrak{I}, \mu' \models \phi[s']$. Finally, recalling that $\mathfrak{I}, \mu' \models s'' = \mathsf{do}(\alpha, s')$, and that μ' and μ agree on s'', by Lemma A.1.1 we get that $\mathfrak{I}, \mu \models \phi[s'']$. □

Proposition A.2.3 (AGM∗3). Under the conditions of Proposition 3.5.4, $K(\sigma * \phi) \subseteq \sigma + \phi$.

Proof. Suppose that $\psi \in K(\sigma * \phi)$, i.e., $\mathfrak{I} \models \mathbf{Bel}(\psi, \mathsf{do}(\alpha, \sigma))$. We want to show that $\psi \in \sigma + \phi$, i.e. $\mathfrak{I} \models \mathbf{Bel}(\phi \supset \psi, \sigma)$. If $\mathfrak{I} \models \mathbf{Bel}(\neg\phi, \sigma)$, then it's trivial that $\mathfrak{I} \models \mathbf{Bel}(\phi \supset \psi, \sigma))$. Otherwise, let μ be a variable assignment such that $\mathfrak{I}, \mu \models \mathsf{MPB}(s', \sigma) \wedge \phi[s']$. Since $\sigma * \phi$ is defined, $\mathfrak{I} \models \mathsf{SF}(\alpha, \sigma)$, and since α is a revision action for ϕ and s' is an accessible situation where ϕ is true, $\mathfrak{I} \models \mathsf{SF}(\alpha, s')$. Therefore, by Lemma A.1.2, $\mathfrak{I} \models \mathsf{MPB}(\mathsf{do}(\alpha, s'), \mathsf{do}(\alpha, \sigma))$. Furthermore, since $\mathfrak{I} \models \mathbf{Bel}(\psi, \mathsf{do}(\alpha, \sigma))$, we get that $\mathfrak{I} \models \psi[\mathsf{do}(\alpha, s')]$. The result that $\mathfrak{I} \models \psi[s']$ follows from Lemma A.1.1. □

Proposition A.2.4 (AGM∗4). Under the conditions of Proposition 3.5.4, if $\neg\phi \notin K(\sigma)$, then $\sigma + \phi \subseteq K(t * \phi)$.

Proof. Suppose that $\neg\phi \notin K(\sigma)$, i.e., $\mathfrak{I} \models \neg\mathbf{Bel}(\neg\phi, \sigma)$. Now, consider any $\psi \in \sigma + \phi$, i.e., any ψ such that $\mathfrak{I} \models \mathbf{Bel}(\phi \supset \psi, \sigma)$. We want to show that $\mathfrak{I} \models \mathbf{Bel}(\psi, \mathsf{do}(\alpha, \sigma))$. Because $\sigma * \phi$ is defined, $\mathfrak{I} \models \mathsf{SF}(\alpha, \sigma)$, and so by Lemma A.1.3 we can conclude that

$$\mathfrak{I} \models \forall s''.\ \mathsf{MPB}(s'', \mathsf{do}(\alpha, \sigma)) \supset \exists s'.\ \mathsf{MPB}(s', \sigma) \wedge (s'' = \mathsf{do}(\alpha, \sigma)) \wedge \phi[s'].$$

From the assumption that $\Sigma \models \mathbf{Bel}(\phi \supset \psi, \sigma)$ we can replace ϕ with ψ in that expression:

$$\mathfrak{I} \models \forall s''.\ \mathsf{MPB}(s'', \mathsf{do}(\alpha, \sigma)) \supset \exists s'.\ \mathsf{MPB}(s', \sigma) \wedge (s'' = \mathsf{do}(\alpha, \sigma)) \wedge \psi[s'].$$

That is, any most plausible accessible situation from the situation denoted by $\mathsf{do}(\alpha, \sigma)$ has a predecessor where ψ is true. The result follows from the action denoted by α not changing the value of ψ (Lemma A.1.1). □

Recall that the postulate (AGM∗5) is not claimed by Proposition 3.5.4. Therefore, the next postulate to prove is (AGM∗6).

Proposition A.2.5 (AGM∗6). Under the conditions of Proposition 3.5.4, if $\models \phi \equiv \psi$, then $K(\sigma * \phi) = K(\sigma * \psi)$.

Proof. Since revision by both ϕ and ψ is defined, the corresponding revision actions are such that the agent is certain in σ that each is possible, senses whether ϕ or ψ is true, and doesn't change the value of any fluent. The result follows from the agent believing that logically equivalent sentences are equivalent. □

Proposition A.2.6 (AGM∗7). Under the conditions of Proposition 3.5.4, $K(\sigma * (\phi \wedge \psi)) \subseteq (\sigma * \phi) + \psi$.

Proof. Suppose that α_ϕ is the revision action for ϕ and $\alpha_{\phi\wedge\psi}$ is the revision action for $\phi \wedge \psi$. Now, suppose that $\gamma \in K(\sigma * (\phi \wedge \psi))$, i.e., $\mathfrak{I} \models \mathbf{Bel}(\gamma, \mathsf{do}(\alpha_{\phi\wedge\psi}, \sigma))$. We want to show that $\gamma \in (\sigma * \phi) + \psi$, i.e., $\mathfrak{I} \models \mathbf{Bel}(\psi \supset \gamma, \mathsf{do}(\alpha_\phi, \sigma))$. Suppose for contradiction that there is a variable assignment μ_1 such that

$$\mathfrak{I}, \mu_1 \models \mathsf{MPB}(s'', \mathsf{do}(\alpha_\phi, \sigma)) \wedge (\psi \wedge \neg\gamma)[s''] \tag{A.1}$$

Since $\sigma * \phi$ is defined and α_ϕ is a revision action for ϕ, we can conclude that there is a variable assignment μ_2 (agreeing with μ_1 except possibly on s') such that

$$\mathfrak{I}, \mu' \models (s'' = \mathsf{do}(\alpha_\phi, s')) \wedge \mathsf{B}(s', \sigma) \wedge \phi[s']$$

Furthermore, by Lemma A.1.1, $\mathfrak{I}, \mu_2 \models (\psi \wedge \neg\gamma)[s']$. By applying Lemma A.1.1 again, we also get that $\mathfrak{I}, \mu_2 \models \neg\gamma[\mathsf{do}(\alpha_{\phi\wedge\psi}, s')]$, since α' is also a revision action. If we can show that $\mathfrak{I}, \mu_2 \models \mathsf{MPB}(\mathsf{do}(\alpha_{\phi\wedge\psi}, s'), \mathsf{do}(\alpha_{\phi\wedge\psi}, \sigma))$, this will complete our proof by contradicting the assumption that $\mathfrak{I} \models \mathbf{Bel}(\gamma, \mathsf{do}(\alpha_{\phi\wedge\psi}, \sigma))$.

Since $\mathfrak{I}, \mu_2 \models \mathsf{B}(s', \sigma) \wedge (\phi \wedge \psi)[s']$ and $\alpha_{\phi\wedge\psi}$ is a revision action for $(\phi \wedge \psi)$, we get that $\mathfrak{I}, \mu_2 \models \mathsf{SF}(\alpha_{\phi\wedge\psi}, s')$. Furthermore, because the revision operator is defined,

$\mathfrak{I} \models \mathsf{SF}(\alpha_{\phi\wedge\psi}, \sigma)$. Since $\alpha_{\phi\wedge\psi}$ is a revision action, $\mathfrak{I}, \mu_2 \models \mathsf{Poss}(\alpha_{\phi\wedge\psi}, s')$, so it can be seen using the SSA for B that $\mathfrak{I}, \mu_2 \models \mathsf{B}(\mathsf{do}(\alpha_{\phi\wedge\psi}, s'), \mathsf{do}(\alpha_{\phi\wedge\psi}, \sigma))$. We next show that the situation denoted by $\mathsf{do}(\alpha_{\phi\wedge\psi}, s')$ is as plausible as any other situation accessible from the situation denoted by $\mathsf{do}(\alpha_{\phi\wedge\psi}, \sigma)$.

Suppose that there is another variable assignment μ_3 (agreeing with μ_2 except possibly on s^{**}) such that

$$\mathfrak{I}, \mu_3 \models \mathsf{B}(s^{**}, \mathsf{do}(\alpha_{\phi\wedge\psi}, \sigma))$$

Then there is a variable assignment μ_4 (agreeing with μ_3 except possibly on s^*) such that

$$\mathfrak{I}, \mu_4 \models (s^{**} = \mathsf{do}(\alpha_{\phi\wedge\psi}, s^*)) \wedge \mathsf{B}(s^*, \sigma) \wedge (\phi \wedge \psi)[s^*]$$

Then it can be seen that $\mathfrak{I}, \mu_4 \models \mathsf{B}(\mathsf{do}(\alpha_\phi, s^*), \mathsf{do}(\alpha_\phi, \sigma))$. From Equation A.1 we know that the plausibility of the situation denoted by $\mathsf{do}(\alpha_\phi, s^*)$ is not greater than that of the situation denoted by s'' (which is the same situation denoted by $\mathsf{do}(\alpha_\phi, s')$). Therefore, the plausibility of the denotation of s^* is not greater than that of the denotation of s', and so the plausibility of the denotation of $\mathsf{do}(\alpha_{\phi\wedge\psi}, s^*)$ is not greater than that of the denotation of $\mathsf{do}(\alpha_{\phi\wedge\psi}, s')$. □

Proposition A.2.7 (AGM∗8). Under the conditions of Proposition 3.5.4, if $\neg\psi \notin K(\sigma * \phi)$, then $(\sigma * \phi) + \psi \subseteq K(\sigma * \phi \wedge \psi)$.

Proof. Let α_ϕ and $\alpha_{\phi\wedge\psi}$ be the revision actions for ϕ and $\phi \wedge \psi$, respectively. Suppose that $\neg\psi \notin K(\sigma * \phi)$. Now consider any $\gamma \in (\sigma * \phi) + \psi$, i.e. any γ such that $\mathfrak{I} \models \mathbf{Bel}(\psi \supset \gamma, \mathsf{do}(\alpha_\phi, \sigma))$. We want to show that $\gamma \in K(\sigma * \phi \wedge \psi)$, i.e., $\mathfrak{I} \models \mathbf{Bel}(\gamma, \mathsf{do}(\alpha_{\phi\wedge\psi}, \sigma))$.

Consider a variable assignment μ_1 such that

$$\mathfrak{I}, \mu_1 \models \mathsf{MPB}(s'', \mathsf{do}(\alpha_{\phi\wedge\psi}, \sigma))$$

(if there is no such assignment, then no situation is accessible from $\mathsf{do}(\alpha_{\phi\wedge\psi}, \sigma)$, so the result that γ is believed trivially follows). We want to show that $\mathfrak{I}, \mu_1 \models \gamma[s'']$. There must be a variable assignment μ_2, agreeing with μ_1 except possibly on s', such that

$$\mathfrak{I}, \mu_2 \models (s'' = \mathsf{do}(\alpha_{\phi\wedge\psi}, s')) \wedge \mathsf{B}(s', \sigma) \wedge (\phi \wedge \psi)[s']$$

By (two applications of) Lemma A.1.1, $\mathfrak{I}, \mu_2 \models (\phi \wedge \psi)[\mathsf{do}(\alpha_\phi, s')]$. If we can show that

$$\mathfrak{I}, \mu_2 \models \mathsf{MPB}(\mathsf{do}(\alpha_\phi, s'), \mathsf{do}(\alpha_\phi, \sigma)),$$

that would mean that $\mathfrak{I}, \mu_2 \models \gamma[\mathsf{do}(\alpha_\phi, s')]$ (because $\psi \supset \gamma$ is believed in the situation denoted by $\mathsf{do}(\alpha_\phi, \sigma)$). That $\mathfrak{I}, \mu_2 \models \gamma[\mathsf{do}(\alpha_{\phi\wedge\psi}, s')]$ would then follow by (two applications of) Lemma A.1.1, and we would be done.

To show that, recall the premise that $\neg\psi \notin K(\sigma * \phi)$, i.e., $\mathfrak{I} \models \neg\mathbf{Bel}(\neg\phi, \mathsf{do}(\alpha_\phi, \sigma))$. Therefore, there is a variable assignment μ_3, agreeing with μ_2 except possibly on s^{**}, such that

$$\mathfrak{I}, \mu_3 \models \mathsf{MPB}(s^{**}, \mathsf{do}(\alpha_\phi, \sigma)) \wedge \psi[s^{**}]$$

So, (recalling Lemma A.1.1) there must be a variable assignment μ_4, agreeing with μ_3 except possibly on s^*, such that

$$\mathfrak{I}, \mu_4 \models (s^{**} = \mathsf{do}(\alpha_\phi, s^*)) \wedge \mathsf{B}(s^*, \sigma) \wedge (\phi \wedge \psi)[s^*]$$

Since $\alpha_{\phi\wedge\psi}$ is a revision action and revision by $(\phi \wedge \psi)$ is defined in σ, it can be seen that

$$\mathfrak{I}, \mu_4 \models \mathsf{B}(\mathsf{do}(\alpha_{\phi\wedge\psi}, s^*), \mathsf{do}(\alpha_{\phi\wedge\psi}, \sigma))$$

It follows that the situation denoted by $\mathsf{do}(\alpha_{\phi\wedge\psi}, s^*)$ is not more plausible than the situation denoted by $\mathsf{do}(\alpha_{\phi\wedge\psi}, s') = s''$. Hence, the denotation of $\mathsf{do}(\alpha_\phi, s^*) = s^{**}$ is not more plausible than the denotation of $\mathsf{do}(\alpha_\phi, s')$. So the denotation of $\mathsf{do}(\alpha_\phi, s')$ is one of the most plausible accessible situations from the situation denoted by $\mathsf{do}(\alpha_\phi, \sigma)$. □

Bibliography

Catharine Abell. Comics and genre. In Aaron Meskin and Roy T. Cook, editors, *The Art of Comics: A Philosophical Approach*. Blackwell Publishing Ltd., 2012. doi:10.1002/9781444354843.ch4.

Thomas Ågotnes, Valentin Goranko, Wojciech Jamroga, and Michael Wooldridge. Knowledge and ability. In Hans van Ditmarsch, Joseph Halpern, Wiebe van der Hoek, and Barteld Kooi, editors, *Handbook of Epistemic Logic*, pages 543–589. College Publications, 2015.

Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530, 1985. doi:10.2307/2274239.

Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 100–109, 1997. doi:10.1109/SFCS.1997.646098.

Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, September 2002. doi:10.1145/585265.585270.

Benjamin Aminof, Giuseppe De Giacomo, Aniello Murano, and Sasha Rubin. Synthesis under assumptions. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018*, pages 615–616, 2018.

Eyal Amir. Toward a formalization of elaboration tolerance: Adding and deleting axioms. In Mary-Anne Williams and Hans Rott, editors, *Frontiers in Belief Revision*, volume 22 of *Applied Logic Series*, pages 147–162. Springer, 2001. doi:10.1007/978-94-015-9817-0_7.

Marcia Ascher. A river-crossing problem in cross-cultural perspective. *Mathematics Magazine*, 63(1):26–29, 1990. doi:10.1080/0025570X.1990.11977478.

Guillaume Aucher and Vaishak Belle. Multi-agent only knowing on Planet Kripke. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, pages 2713–2719, 2015.

Christopher Badura and Francesco Berto. Truth in fiction, impossible worlds, and belief revision. *Australasian Journal of Philosophy*, 2018. doi:10.1080/00048402.2018.1435698.

Alexandru Baltag and Sonja Smets. A qualitative theory of dynamic interactive belief revision. In *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, Texts in Logic and Games 3, pages 11–58. Amsterdam University Press, 2008.

Ruth C. Barcan. A functional calculus of first order based on strict implication. *The Journal of Symbolic Logic*, 11(1):1–16, 1946. doi:10.2307/2269159.

Christoph Beierle, Tobias Falke, Steven Kutsch, and Gabriele Kern-Isberner. System ZFO: Default reasoning with system Z-like ranking functions for unary first-order conditional knowledge bases. *International Journal of Approximate Reasoning*, 90: 120–143, 2017. doi:10.1016/j.ijar.2017.07.005.

Nuel Belnap and Michael Perloff. Seeing to it that: a canonical form for agentives. *Theoria*, 54:175–199, December 1988. doi:10.1111/j.1755-2567.1988.tb00717.x.

Salem Benferhat and Rania El Baida. A stratified first order logic approach for access control. *International Journal of Intelligent Systems*, 19(9):817–836, 2004. doi:10.1002/int.20026.

Salem Benferhat, Claudette Cayrol, Didier Dubois, Jerome Lang, and Henri Prade. Inconsistency management and prioritized syntax-based entailment. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'93, pages 640–645, 1993.

Roderick Bloem, Rüdiger Ehlers, Swen Jacobs, and Robert Könighofer. How to handle assumptions in synthesis. In *Proceedings 3rd Workshop on Synthesis*, volume 157 of *Electronic Proceedings in Theoretical Computer Science*, pages 34–50. Open Publishing Association, 2014. doi:10.4204/EPTCS.157.7.

Andrea Bonomi and Sandro Zucchi. A pragmatic framework for truth in fiction. *Dialectica*, 57(2):103–120, 2003. doi:10.1111/j.1746-8361.2003.tb00259.x.

Berilhes Borges Garcia. New tractable classes for default reasoning from conditional knowledge bases. *Annals of Mathematics and Artificial Intelligence*, 45(3-4):275–291, 2005. doi:10.1007/s10472-005-9000-3.

Craig Boutilier. Normative, subjunctive and autoepistemic defaults. In Gerhard Lakemeyer and Bernhard Nebel, editors, *Foundations of Knowledge Representation and Reasoning*, pages 74–97. Springer Berlin Heidelberg, 1994. doi:10.1007/3-540-58107-3_5.

Craig Boutilier. Abduction to plausible causes: an event-based model of belief update. *Artificial Intelligence*, 83(1):143–166, 1996. doi:10.1016/0004-3702(94)00097-2.

Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann, 2004. doi:10.1016/B978-1-55860-932-7.X5083-3.

Katarina Britz and Ivan Varzinczak. From KLM-style conditionals to defeasible modalities, and back. *Journal of Applied Non-Classical Logics*, 28(1):92–121, 2018. doi:10.1080/11663081.2017.1397325.

Nils Bulling, Wojciech Jamroga, and Jürgen Dix. Reasoning about temporal properties of rational play. *Annals of Mathematics and Artificial Intelligence*, 53:51–114, 2008. doi:10.1007/s10472-009-9110-4.

Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1):3–55, 1999. doi:10.1016/S0004-3702(99)00070-3.

Diego Calvanese, Giuseppe De Giacomo, and Moshe Y. Vardi. Reasoning about actions and planning in LTL action theories. In *Proceedings of the Eighth International Conference on Principles and Knowledge Representation and Reasoning (KR-02)*, pages 593–602, 2002.

Alberto Camacho, Meghyn Bienvenu, and Sheila A. McIlraith. Finite LTL synthesis with environment assumptions and quality measures. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018*, pages 454–463, 2018.

Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. Environment assumptions for synthesis. In Franck van Breugel and Marsha Chechik, editors, *CON-*

CUR 2008 - Concurrency Theory, pages 147–161. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-85361-9_14.

E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, April 1986. doi:10.1145/5397.5399.

Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89(1):1–29, 1997. doi:10.1016/S0004-3702(96)00038-0.

Giuseppe De Giacomo, Yves Lespérance, and Hector J. Levesque. ConGolog, a concurrent programming language based on the situation calculus. *Artificial Intelligence*, 121(1-2): 109–169, 2000. doi:10.1016/S0004-3702(00)00031-X.

Giuseppe De Giacomo, Yves Lespérance, and Adrian R. Pearce. Situation calculus based programs for representing and reasoning about game structures. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010*, 2010.

Luc De Raedt. Inductive logic programming. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 648–656. Springer US, 2017. doi:10.1007/978-1-4899-7687-1_135.

Alvaro del Val and Yoav Shoham. A unified view of belief revision and update. *Journal of Logic and Computation*, 4(5):797–810, 1994. doi:10.1093/logcom/4.5.797.

James P. Delgrande and Hector J. Levesque. Belief revision with sensing and fallible actions. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012*, pages 148–157, 2012.

James P. Delgrande and Hector J. Levesque. A formal account of nondeterministic and failed actions. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 861–868, 2013.

Robert Demolombe. Belief change: from situation calculus to modal logic. *Journal of Applied Non-Classical Logics*, 13(2):187–198, 2003. doi:10.3166/jancl.13.187-198.

Robert Demolombe and Pilar Pozos Parra. Belief revision in the situation calculus without plausibility levels. In *Foundations of Intelligent Systems. ISMIS 2006*, pages 504–513, 2006. doi:10.1007/11875604_57.

Pedro Domingos and Daniel Lowd. Unifying logical and statistical AI with Markov logic. *Communications of the ACM*, 62(7):74–83, June 2019. doi:10.1145/3241978.

Carmel Domshlak. Fault tolerant planning: Complexity and compilation. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 64–72, 2013.

Thomas Eiter and Thomas Lukasiewicz. Default reasoning from conditional knowledge bases: Complexity and tractable cases. *Artificial Intelligence*, 124(2):169–241, 2000. doi:10.1016/S0004-3702(00)00073-4.

Thomas Eiter, Esra Erdem, Michael Fink, and Ján Senko. Comparing action descriptions based on semantic preferences. *Annals of Mathematics and Artificial Intelligence*, 50 (3):273–304, Aug 2007. doi:10.1007/s10472-007-9077-y.

Thomas Eiter, Esra Erdem, Michael Fink, and Ján Senko. Updating action domain descriptions. *Artificial Intelligence*, 174(15):1172–1221, 2010. doi:10.1016/j.artint.2010.07.004.

Jennifer J. Elgot-Drapkin and Donald Perlis. Reasoning situated in time I: basic concepts. *Journal of Experimental & Theoretical Artificial Intelligence*, 2(1):75–98, 1990. doi:10.1080/09528139008953715.

Herbert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 2nd edition, 2001. doi:10.1016/C2009-0-22107-6.

Christopher Ewin, Adrian R. Pearce, and Stavros Vassos. Optimizing long-running action histories in the situation calculus through search. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems*, pages 85–100, 2015. doi:10.1007/978-3-319-25524-8_6.

Ronald Fagin and Joseph Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988. doi:10.1016/0004-3702(87)90003-8.

Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

Liangda Fang and Yongmei Liu. Multiagent knowledge and belief change in the situation calculus. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 304–312, 2013.

Raul Fervari, Andreas Herzig, Yanjun Li, and Yanjing Wang. Strategically knowing how. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1031–1038, 2017. doi:10.24963/ijcai.2017/143.

Melvin Fitting. Barcan both ways. *Journal of Applied Non-Classical Logics*, 9(2-3): 329–344, 1999. doi:10.1080/11663081.1999.10510970.

Nir Friedman and Joseph Y. Halpern. Modeling belief in dynamic systems, part II: Revision and update. *Journal of Artificial Intelligence Research (JAIR)*, 10:117–167, 1999a. doi:10.1613/jair.506.

Nir Friedman and Joseph Y. Halpern. Belief revision: A critique. *Journal of Logic, Language and Information*, 8(4):401–420, Oct 1999b. doi:10.1023/A:1008314832430.

Christian Fritz. *Monitoring the Generation and Execution of Optimal Plans*. PhD thesis, University of Toronto, April 2009. URL `http://hdl.handle.net/1807/17763`.

Christian Fritz, Jorge A. Baier, and Sheila A. McIlraith. ConGolog, Sin Trans: Compiling ConGolog into basic action theories for planning and beyond. In *Proceedings on the 11th International Conference on Principles of Knowledge Representation and Reasoning*, pages 600–610, 2008.

Alfredo Gabaldon. Non-Markovian control in the situation calculus. *Artificial Intelligence*, 175(1):25–48, 2011. doi:10.1016/j.artint.2010.04.012.

Hector Geffner and Judea Pearl. Conditional entailment: Bridging two approaches to default reasoning. *Artificial Intelligence*, 53(2):209–244, 1992. doi:10.1016/0004-3702(92)90071-5.

Michael Gelfond and Vladimir Lifschitz. Action languages. *Electronic Transactions on Artificial Intelligence*, 2(3-4):193–210, 1998. URL `http://www.ep.liu.se/ej/etai/1998/007/`.

Hojjat Ghaderi. *A Logical Theory of Joint Ability in the Situation Calculus*. PhD thesis, University of Toronto, 2011. URL `http://hdl.handle.net/1807/26272`.

Hojjat Ghaderi, Hector J. Levesque, and Yves Lespérance. A logical theory of coordination and joint ability. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 421–426, 2007.

Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann, 2004. doi:10.1016/B978-1-55860-856-6.X5000-5.

Paul Gochet. An open problem in the logic of knowing how. In Jaakko Hintikka, editor, *Open Problems in Epistemology*. Philosophical Society of Finland, 2013.

Moisés Goldszmidt, Paul Morris, and Judea Pearl. A maximum entropy approach to nonmonotonic reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):220–232, Mar 1993. doi:10.1109/34.204904.

Noah T. Goodman, Joshua B. Tenenbaum, and Tobias Gerstenberg. Concepts in a probabilistic language of thought. CBMM Memo 010, Center for Brains, Minds, and Machines, 2014. URL `http://hdl.handle.net/1721.1/100174`.

Alexandra Goultiaeva and Yves Lespérance. Incremental plan recognition in an agent programming framework. In *Working Notes of the AAAI 2007 Workshop on Plan, Activity, and Intent Recognition (PAIR'07)*, 2007.

Martin Grohe. Generalized model-checking problems for first-order logic. In *18th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2001)*, pages 12–26, 2001. doi:10.1007/3-540-44693-1_2.

Adam Grove. Two modellings for theory change. *Journal of Philosophical Logic*, 17(2): 157–170, 1988. doi:10.1007/BF00247909.

Joeseph Y. Halpern. *Reasoning about Uncertainty*. MIT Press, 2003.

Joseph Y. Halpern and Ronald Fagin. Modelling knowledge and action in distributed systems. *Distributed Computing*, 3(4):159–177, Dec 1989. doi:10.1007/BF01784885.

Joseph Y. Halpern and Yoram Moses. Using counterfactuals in knowledge-based programming. *Distributed Computing*, 17(2):91–106, Aug 2004. doi:10.1007/s00446-004-0108-1.

Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. Algorithmic knowledge. In *Theoretical Aspects of Reasoning about Knowledge: Proceedings of the 5th Conference (TARK 1994)*, 1994.

Joseph Y. Halpern, Dov Samet, and Ella Segev. Defining knowledge in terms of belief: The modal logic perspective. *The Review of Symbolic Logic*, 2(3):469–487, 2009. doi:10.1017/S1755020309990141.

Andreas Herzig, Laurent Perrussel, and Ivan José Varzinczak. Elaborating domain descriptions. In *ECAI 2006, 17th European Conference on Artificial Intelligence*, pages 397–401, 2006.

Jaakko Hintikka. *Knowledge and Belief*. Cornell University Press, 1962.

Jaakko Hintikka. Impossible possible worlds vindicated. *Journal of Philosophical Logic*, 4(4):475–484, 1975. doi:10.1007/BF00558761.

Jerry R. Hobbs. Ontological promiscuity. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, ACL '85, 1985. doi:10.3115/981210.981218.

Yuxiao Hu. *Generation and Verification of Plans with Loops*. PhD thesis, University of Toronto, 2012. URL `http://hdl.handle.net/1807/32740`.

Yuxiao Hu and Hector J. Levesque. A correctness result for reasoning about one-dimensional planning problems. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010*, 2010.

Daniel Hunter. On the relation between categorical and probabilistic belief. *Noûs*, 30(1): 75–98, 1996. doi:10.2307/2216304.

Jonathan Jenkins Ichikawa and Matthias Steup. The analysis of knowledge. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2018 edition, 2018. URL `https://plato.stanford.edu/archives/sum2018/entries/knowledge-analysis/`.

Wojciech Jamroga and Wiebe van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2-3):185–219, 2004.

Rune M. Jensen, Manuela M. Veloso, and Randal E. Bryant. Fault tolerant planning: Toward probabilistic uncertainty models in symbolic non-deterministic planning. In *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004)*, pages 335–344, 2004.

Yi Jin and Michael Thielscher. Representing beliefs in the fluent calculus. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004*, pages 823–827, 2004.

P. N. Johnson-Laird, Sangeet S. Khemlani, and Geoffrey P. Goodwin. Logic, probability, and human reasoning. *Trends in Cognitives Sciences*, 19:201–214, April 2015. doi:10.1016/j.tics.2015.02.006.

David Kaplan. Quantifying in. *Synthese*, 19(1/2):178–214, 1968. doi:10.1007/BF00568057.

Mark Kaplan. Decision theory and epistemology. In Paul K. Moser, editor, *The Oxford Handbook of Epistemology*. Oxford University Press, 2005. doi:10.1093/oxfordhb/9780195301700.003.0016.

Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 387–394, 1991.

Nikos Katzouris, Alexander Artikis, and Georgios Paliouras. Parallel online event calculus learning for complex event recognition. *Future Generation Computer Systems*, 94:468–478, 2019. doi:10.1016/j.future.2018.11.033.

Ryan F. Kelly and Adrian R. Pearce. Property persistence in the situation calculus. *Artificial Intelligence*, 174(12):865–888, 2010. doi:10.1016/j.artint.2010.05.003.

Ryan F. Kelly and Adrian R. Pearce. Asynchronous knowledge with hidden actions in the situation calculus. *Artificial Intelligence*, 221:1–35, 2015. doi:10.1016/j.artint.2014.12.005.

Gabriele Kern-Isberner and Christian Eichhorn. Structural inference from conditional knowledge bases. *Studia Logica*, 102(4):751–769, Aug 2014. doi:10.1007/s11225-013-9503-6.

Toryn Q. Klassen, Sheila A. McIlraith, and Hector J. Levesque. Towards tractable inference for resource-bounded agents. In *Logical Formalizations of Commonsense Reasoning: Papers from the AAAI Spring Symposium*, pages 89–95. AAAI Press, 2015.

Toryn Q. Klassen, Hector J. Levesque, and Sheila A. McIlraith. Towards representing what readers of fiction believe. In *Proceedings of the Thirteenth International Symposium on Commonsense Reasoning, COMMONSENSE 2017*, 2017. URL `http://ceur-ws.org/Vol-2052/paper12.pdf`.

Toryn Q. Klassen, Sheila A. McIlraith, and Hector J. Levesque. Specifying plausibility levels for iterated belief change in the situation calculus. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018*, pages 257–266, 2018.

Toryn Q. Klassen, Sheila A. McIlraith, and Hector J. Levesque. Changing beliefs about domain dynamics in the situation calculus. In *17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020)*, 2020. (To appear).

Toryn Qwyllyn Klassen. *Resource-bounded inference with three-valued neighborhood semantics*. MSc paper, University of Toronto, 2015.

Laura Kovács and Andrei Voronkov. First-order theorem proving and Vampire. In *Computer Aided Verification - 25th International Conference, CAV 2013*, pages 1–35, 2013. doi:10.1007/978-3-642-39799-8_1.

Robert Kowalski and Marek Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–95, 1986. doi:10.1007/BF03037383.

Saul A. Kripke. Semantical analysis of modal logic I: Normal modal propositional calculi. *Mathematical Logic Quarterly*, 9(5-6):67–96, 1963. doi:10.1002/malq.19630090502.

Ugur Kuter and Dana S. Nau. Forward-chaining planning in nondeterministic domains. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 513–518, 2004.

Gerhard Lakemeyer. The situation calculus: A case for modal logic. *Journal of Logic, Language and Information*, 19(4):431–450, Oct 2010. doi:10.1007/s10849-009-9117-6.

Gerhard Lakemeyer and Hector J. Levesque. AOL: a logic of acting, sensing, knowing, and only knowing. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 316–327, 1998.

Gerhard Lakemeyer and Hector J. Levesque. A semantic characterization of a useful fragment of the situation calculus with knowledge. *Artificial Intelligence*, 175(1):142–164, 2011. doi:10.1016/j.artint.2010.04.005.

Gerhard Lakemeyer and Hector J. Levesque. Decidable reasoning in a fragment of the epistemic situation calculus. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference (KR 2014)*, 2014.

Gerhard Lakemeyer and Hector J. Levesque. A tractable, expressive, and eventually complete first-order logic of limited belief. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, pages 1764–1771, 2019. doi:10.24963/ijcai.2019/244.

Daniel Lehmann. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence*, 15(1):61–82, 1995. doi:10.1007/BF01535841.

Yves Lespérance, Hector J. Levesque, Fangzhen Lin, and Richard B. Scherl. Ability and knowing how in the situation calculus. *Studia Logica*, 66(1):165–186, 2000. doi:10.1023/A:1026761331498.

Yves Lespérance, Giuseppe De Giacomo, and Atalay Nafi Ozgovde. A model of contingent planning for agent programming languages. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Volume 1*, pages 477–484, 2008.

Hector Levesque, Fiora Pirri, and Ray Reiter. Foundations for a calculus of situations. *Electronic Transactions on Artificial Intelligence*, 2(3–4):159–178, 1998. URL `http://www.ep.liu.se/ej/etai/1998/005/`. Originally titled "Foundations for the Situation Calculus".

Hector J. Levesque. A logic of implicit and explicit belief. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-84)*, pages 198–202, 1984.

Hector J. Levesque. All I know: A study in autoepistemic logic. *Artificial Intelligence*, 42(2-3):263–309, 1990. doi:10.1016/0004-3702(90)90056-6.

Hector J. Levesque. What is planning in the presence of sensing? In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, pages 1139–1146, 1996.

Hector J. Levesque, Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl. GOLOG: A logic programming language for dynamic domains. *The Journal of Logic Programming*, 31(1):59–83, 1997. doi:10.1016/S0743-1066(96)00121-5.

David Lewis. *Counterfactuals*. Harvard University Press, 1973.

David Lewis. Truth in fiction. *American Philosophical Quarterly*, 15(1):37–46, 1978.

Paolo Liberatore and Marco Schaerf. Relating belief revision and circumscription. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95*, pages 1557–1566, 1995.

Paolo Liberatore and Marco Schaerf. Reducing belief revision to circumscription (and vice versa). *Artificial Intelligence*, 93(1):261–296, 1997. doi:10.1016/S0004-3702(97)00016-7.

Vladimir Lifschitz. Circumscription. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 297–352. Oxford University Press, 1994.

Fangzhen Lin. Chapter 16: Situation calculus. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 649–669. Elsevier, 2008. doi:10.1016/S1574-6526(07)03016-7.

Fangzhen Lin and Hector J. Levesque. What robots can do: Robot programs and effective achievability. *Artificial Intelligence*, 101(1-2):201–226, 1998. doi:10.1016/S0004-3702(98)00041-1.

Fangzhen Lin and Ray Reiter. How to progress a database. *Artificial Intelligence*, 92(1): 131–167, 1997. doi:10.1016/S0004-3702(96)00044-6.

Yongmei Liu, Gerhard Lakemeyer, and Hector J. Levesque. A logic of limited belief for reasoning with disjunctive information. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR 2004)*, pages 587–597, 2004.

D. C. Makinson. The paradox of the preface. *Analysis*, 25(6):205–207, 1965.

Gary F. Marcus and Ernest Davis. How robust are probabilistic models of higher-level cognition? *Psychological Science*, 24(12):2351–2360, 2013. doi:10.1177/0956797613495418.

John McCarthy. Situations, actions, and causal laws. Memo AIM-002, Stanford Artificial Intelligence Project, 1963. URL `https://purl.stanford.edu/kf190cg0706`.

John McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1–2):27–39, 1980. doi:10.1016/0004-3702(80)90011-9.

John McCarthy. Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, 28(1):89–116, 1986. doi:10.1016/0004-3702(86)90032-9.

John McCarthy. Elaboration tolerance, 2003. URL `http://www-formal.stanford.edu/jmc/elaboration.pdf`. An earlier version was presented at Commonsense 1998.

John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

Sheila A. McIlraith, Tran Cao Son, and Honglei Zeng. Semantic web services. *IEEE Intelligent Systems*, 16(2):46–53, 2001. doi:10.1109/5254.920599.

Yves Moinard. Note about cardinality-based circumscription. *Artificial Intelligence*, 119 (1):259–273, 2000. doi:10.1016/S0004-3702(00)00018-7.

Robert C. Moore. Reasoning about knowledge and action. Technical Note 191, SRI International, 1980. URL `https://apps.dtic.mil/sti/citations/ADA126244`.

Robert C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75–94, 1985. doi:10.1016/0004-3702(85)90042-6.

Stephen Moyle and Stephen Muggleton. Learning programs in the event calculus. In *Inductive Logic Programming (ILP-97)*, pages 205–212, 1997. doi:10.1007/3540635149_49.

Erik T. Mueller. *Commonsense Reasoning*. Morgan Kaufmann Publishers, 2006. doi:10.1016/B978-0-12-369388-4.X5054-1.

Stephen Muggleton and Luc de Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19–20:629–679, 1994. doi:10.1016/0743-1066(94)90035-3.

Pavel Naumov and Jia Tao. Knowing-how under uncertainty. *Artificial Intelligence*, 276: 41–56, 2019. doi:10.1016/j.artint.2019.06.007.

Eric Pacuit and Sunil Simon. Reasoning with protocols under imperfect information. *The Review of Symbolic Logic*, 4(3):412—444, 2011. doi:10.1017/S1755020311000190.

Maurice Pagnucco, David Rajaratnam, Hannes Strass, and Michael Thielscher. Implementing belief change in the situation calculus and an application. In *Logic Programming and Nonmonotonic Reasoning. LPNMR 2013*, pages 439–451, 2013. doi:10.1007/978-3-642-40564-8_44.

Aarati Dinesh Parmar. *Formalizing Elaboration Tolerance*. PhD thesis, Stanford University, 2003.

Judea Pearl. System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning About Knowledge*, TARK '90, pages 121–135, 1990.

Pavlos Peppas. Chapter 8: Belief revision. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*, volume 3, pages 317–359. Elsevier, 2008. doi:10.1016/S1574-6526(07)03008-8.

Pavlos Peppas and Mary-Anne Williams. Parametrised difference revision. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018*, pages 277–286, 2018.

Javier Pinto. Occurrences and narratives as constraints in the branching structure of the situation calculus. *Journal of Logic and Computation*, 8(6):777–808, 1998. doi:10.1093/logcom/8.6.777.

Fiora Pirri and Ray Reiter. Some contributions to the metatheory of the situation calculus. *Journal of the ACM*, 46(3):325–361, May 1999. doi:10.1145/316542.316545.

Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science*, pages 46–57. IEEE Computer Society, 1977. doi:10.1109/SFCS.1977.32.

Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In *Proceedings of the 16th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '89, pages 179–190, 1989. doi:10.1145/75277.75293.

W. V. Quine. Quantifiers and propositional attitudes. *The Journal of Philosophy*, 53(5): 177–187, 1956. doi:10.2307/2022451.

William J. Rapaport and Stuart C. Shapiro. Cognition and fiction. In Judith F. Duchan, Gail A. Bruder, and Lynne E. Hewitt, editors, *Deixis in Narrative: A Cognitive Science Perspective*, pages 107–128. Lawrence Erlbaum Associates, Inc., 1995.

R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, April 1980. doi:10.1016/0004-3702(80)90014-4.

Raymond Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.

Christophe Rodrigues, Pierre Gerard, Celine Rouveirol, and Henry Soldano. Incremental learning of relational action rules. In *2010 Ninth International Conference on Machine Learning and Applications (ICMLA)*, pages 451–458, 2010. doi:10.1109/ICMLA.2010.73.

Hans Rott. Shifting priorities: Simple representations for twenty-seven iterated theory change operators. In David Makinson, Jacek Malinowski, and Heinrich Wansing, editors, *Towards Mathematical Philosophy*, volume 28 of *Trends in Logic*, pages 269–296. Springer Netherlands, 2009. doi:10.1007/978-1-4020-9084-4_14.

Marie-Laure Ryan. *Possible Worlds, Artificial Intelligence, and Narrative Theory*. Indiana University Press, 1991.

Roger C. Schank and Robert P. Abelson. Scripts, plans, and knowledge. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'75, pages 151–157, 1975.

Richard B. Scherl and Hector J. Levesque. Knowledge, action, and the frame problem. *Artificial Intelligence*, 144(1):1–39, 2003. doi:10.1016/S0004-3702(02)00365-X.

Christoph Schwering. *Conditional Beliefs in Action*. PhD thesis, RWTH Aachen University, 2016. URL `http://publications.rwth-aachen.de/record/660817/files/660817.pdf`.

Christoph Schwering and Gerhard Lakemeyer. A semantic account of iterated belief revision in the situation calculus. In *ECAI 2014 - 21st European Conference on Artificial Intelligence*, pages 801–806, 2014.

Christoph Schwering and Gerhard Lakemeyer. Projection in the epistemic situation calculus with belief conditionals. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1583–1589, 2015.

Christoph Schwering, Gerhard Lakemeyer, and Maurice Pagnucco. Belief revision and projection in the epistemic situation calculus. *Artificial Intelligence*, 251:62–97, 2017. doi:10.1016/j.artint.2017.07.004.

Krister Segerberg. Belief revision from the point of view of doxastic logic. *Logic Journal of the IGPL*, 3(4):535–553, 1995. doi:10.1093/jigpal/3.4.535.

Krister Segerberg. Two traditions in the logic of belief: Bringing them together. In Hans Jürgen Ohlbach and Uwe Reyle, editors, *Logic, Language and Reasoning: Essays in Honour of Dov Gabbay*, pages 135–147. Springer Netherlands, 1999. doi:10.1007/978-94-011-4574-9_8.

Steven Shapiro. *Specifying and Verifying Multiagent Systems Using the Cognitive Agents Specification Language (CASL)*. PhD thesis, University of Toronto, 2005.

Steven Shapiro and Maurice Pagnucco. Iterated belief change and exogeneous actions in the situation calculus. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004*, pages 878–882, 2004.

Steven Shapiro, Maurice Pagnucco, Yves Lespérance, and Hector J. Levesque. Iterated belief change in the situation calculus. *Artificial Intelligence*, 175(1):165–192, 2011. doi:10.1016/j.artint.2010.04.003.

Nirad Sharma and Robert Colomb. Towards an integrated characterisation of model-based diagnosis and configuration through circumscription policies. Technical Report 364, Department of Computer Science, University of Queensland, 1997.

Anthia Solaki, Francesco Berto, and Sonja Smets. The logic of fast and slow thinking. *Erkenntnis*, 2019. doi:10.1007/s10670-019-00128-z.

Wolfgang Spohn. Ordinal conditional functions: A dynamic theory of epistemic states. In *Causation in Decision, Belief Change, and Statistics: Proceedings of the Irvine Conference on Probability and Causation*, volume 2, pages 105–134, 1988. doi:10.1007/978-94-009-2865-7_6.

Robert Stalnaker. The Problem of Logical Omniscience, I. *Synthese*, 89(3):425–440, 1991. doi:10.1007/BF00413506.

Christian Strasser and G. Aldo Antonelli. Non-monotonic logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2016 edition, 2016. URL `https://plato.stanford.edu/archives/win2016/entries/logic-nonmonotonic/`.

Michael Thielscher. Introduction to the fluent calculus. *Electronic Transactions on Artificial Intelligence*, 2(3-4):179–192, 1998. URL `http://www.ep.liu.se/ej/etai/1998/006/`.

Michael Thielscher. From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. *Artificial Intelligence*, 111(1):277–299, 1999. doi:10.1016/S0004-3702(99)00033-8.

Nikoleta Tsampanaki, Theodore Patkos, Giorgos Flouris, and Dimitris Plexousakis. Revising event calculus theories to recover from unexpected observations. *Annals of Mathematics and Artificial Intelligence*, 2019. doi:10.1007/s10472-019-09663-5.

Johan van Benthem. Games in dynamic-epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001. doi:10.1111/1467-8586.00133.

Johan van Benthem and Cédric Dégremont. Bridges between dynamic doxastic and doxastic temporal logics. In Giacomo Bonanno, Benedikt Löwe, and Wiebe van der Hoek, editors, *Logic and the Foundations of Game and Decision Theory – LOFT 8*, pages 151–173. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-15164-4_8.

Wiebe van der Hoek and Michael Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica: An International Journal for Symbolic Logic*, 75(1):125–157, 2003. doi:10.1023/A:1026185103185.

Hans P. van Ditmarsch. Prolegomena to dynamic logic for belief revision. *Synthese*, 147 (2):229–275, 2005. doi:10.1007/s11229-005-1349-7.

Hanna S. van Lee, Rasmus K. Rendsvig, and Suzanne van Wijk. Intensional protocols for dynamic epistemic logic. *Journal of Philosophical Logic*, May 2019. doi:10.1007/s10992-019-09508-w.

Marc Van Zee, Dragan Doder, Mehdi Dastani, and Leendert Van Der Torre. AGM revision of beliefs about action and time. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 3250–3256, 2015.

Ivan José Varzinczak. On action theory change. *Journal of Artificial Intelligence Research*, 37:189–246, 2010. doi:10.1613/jair.2959.

Stavros Vassos and Hector J. Levesque. How to progress a database III. *Artificial Intelligence*, 195:203–221, 2013. doi:10.1016/j.artint.2012.10.005.

Trevor Walker, Lisa Torrey, Jude W. Shavlik, and Richard Maclin. Building relational world models for reinforcement learning. In *Inductive Logic Programming, 17th International Conference, ILP 2007*, pages 280–291, 2007. doi:10.1007/978-3-540-78469-2_27.

Kendall L. Walton. *Mimesis as Make-Believe: On the Foundations of the Representational Arts*. Harvard University Press, 1990.

Yanjing Wang. A logic of goal-directed knowing how. *Synthese*, 195(10):4419–4439, Oct 2018. doi:10.1007/s11229-016-1272-0.

Liping Xiong and Yongmei Liu. Strategy representation and reasoning for incomplete information concurrent games in the situation calculus. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1322–1329, 2016.

Wael Yehia, Hongkai Liu, Marcel Lippmann, Franz Baader, and Mikhail Soutchanski. Experimental results on solving the projection problem in action formalisms based on description logics. In *Proceedings of the 2012 International Workshop on Description Logics, DL-2012*, 2012. URL `http://ceur-ws.org/Vol-846/paper_15.pdf`.

Richard Zach. *Incompleteness and Computability: An Open Introduction to Gödel's Theorems*. Independently published, 2020. URL `https://ic.openlogicproject.org/`. Revision 368c6d0 (2020-04-28).