Steffen Hölldobler

The Weak Completion Semantics

Technische Universität Dresden, Germany
North-Caucasus Federal University, Stavropol, Russian Federation
sh@iccl.tu-dresden.de

March 11, 2019
Unpublished Manuscript
Subject to Revision
Please do not circulate

Comments and Corrections are Very Welcome

Preface

I stumbled into the field of Cognitive Science when I gave a lecture at the summer school of the International Center of Computational Logic at the Technische Universität Dresden in the year 2007. After presenting our idea to compute semantic operators associated with logic programs by feed-forward connectionist networks and to recursively propagate interpretations through these networks until they converge to stable states I asked with caution whether these stable states, which represent least models of the programs, have something in common with mental models. Michiel van Lambalgen, who was in the audience, raised his arm and answered: *these are mental models.*

After the summer school, one of our students in the European Master's Program in Computational Logic at that time, Carroline Devi Puspa Kencana Ramli, asked me for interesting topics for a master thesis. Like me she had attended the summer school and was interested in the work by Keith Stenning and Michiel van Lambalgen. Michiel had presented main results from their upcoming book on *Human Reasoning and Cognitive Science* at the summer school. I suggested that Carroline may start by providing proofs for some of the formal results claimed by Keith and Michiel. Carroline studied these results for two weeks and came back with: *I found a counter-example.* Thus, her master thesis project became much more interesting than expected.

Bertram Fronhöfer and me cross-checked: Carroline was right. We had the feeling that the problem was related to the three-valued logic used by Keith and Michiel. As we had no background in three-valued logics we started to read the first paper ever published on this subject to the best of our knowledge, viz. the paper by Jan Łukasiewicz from 1920. I still remember that we were sitting at the round table in the front of my office for several hours discussing this paper and its implications, when somehow it became clear to us that under this logic the model intersection property holds for normal logic programs. This was the origin of the *Weak Completion Semantics.*

Carroline wrote a wonderful master thesis, probably the best I have ever seen. We published various papers based on her thesis: the main theoretical results defining the basis of the *Weak Completion Semantics*, the conditions for the semantic operator to become a contraction mapping as well as the connectionist computation of the semantic operator and its least fixed point.

We were now able to adequately model the first six experiments of Ruth Byrne's suppression task. For the remaining six experiments, the *Weak Completion Semantics* had to be extended by abduction. Christoph Wernhard and Tobias Philipp, at that time a student in our international master program in Computational Logic, extended abductive frameworks to the three-valued logic underlying the *Weak Completion Semantics.* Surprisingly, the suppression task could only be adequately solved if we apply skeptical abduction. In the meantime, in all human reasoning tasks which we have considered so far and where it was necessary to apply abduction, we had to apply skeptical abduction

Our first papers on the *Weak Completion Semantics* were published at conferences about logic programming and logic based knowledge representation and reasoning. Whenever we presented our approach colleagues immediately asked about its relation to the well-founded semantics. Emmanuelle-Anna Dietz, my PhD student at that time, looked into this problem. Together with Christoph we found that as long as programs do not have positive cycles, they can be transformed such that the *Weak Completion Semantics* of the program and the well-founded semantics of its transformation are indeed identical, but they deviate as soon as positive cylces are present.

As we were unaware of any experimental data showing the behavior of humans in the presence of positive cycles, we asked Marco Ragni to help us setting up an experiment. The cycles which we tested were probably to short, but almost none of the participants gave the answer corresponding to the predictions of the well-founded semantics.

When Emma went to the Universidade Nova de Lisboa to study with Luís Moniz Pereira, Luís became involved and asked many questions about reasoning with conditionals as well as extensions of abduction. Our research greatly benefitted from Luís' tremendous experience and background knowledge.

In the year 2015 Emma was asked by Philip N. Johnson-Laird whether we might be interested in applying the *Weak Completion Semantics* to syllogistic reasoning. Phil had published a meta-study on this subject together with Sangeet Khemlani. So we tried. Surprisingly, our first attempt was already competitive and a little later, with lots of help by Ana Oliviera da Costa, at that time a master student in the European Master's Program in Computational Logic, the *Weak Completion Semantics* outperformed the twelve cognitive theories mentioned in the meta-study. At this point, I was encouraged to call our approach a *cognitive theory*. Richard Mörbitz, at that time a student in the Diplom program in Computer Science, looked into the problem of clustering reasoners in order to explain conflicting answers given by the participants in the meta-study.

Working with Emma and Marco we had a first solution for the selection task, but we were not completely happy, as the distinction between the abstract and the social task were made by practical considerations. This changed when Isabelly Lourêdo Rocha, at that time a master student in the European Master's Program in Computational Logic, looked into obligation and factual conditionals, and showed how elegantly the selection task can be solved by making such a distinction.

Luis Palacios Medinacelli, at that time a master student in the European Master's Program in Computationpal Logic, looked into connectionist models for skeptical abduction and extended the known connectionist models for creduluos abduction. Together with Carroline and Emma we developed a complete connectionist implementation of the *Weak Completion Semantics*. In this implementation, possible explanations are generated in a predefinded sequence. Isabelly showed in her master thesis that it is possible to train connectionist networks to generate explanations in any sequence, which is a prerequisite for developing ideas on bounded skeptical abduction.

Together with Emma and Raphael Höps, at that time a bachelor student in Computer Science, we showed that the *Weak Completion Semantics* can compute preferred models of spatial reasoning task.

In 2017 Luís Moniz Pereira gave me his new book on *Programming Machine Ethics* which he had written together with Ari Saptawijaya. We offered a reading class on this subject and asked the students to implement ethical decision problems in the fluent calculus. In the end, Dominic Deckert, at that time a student in the Diplom-program in Computer Science, wrote his project on this topic.

However, in order to use the fluent calculus within the *Weak Completion Semantics* we had to extend it to handle equational theories. In fact, all theoretical result obtained so far had to be cross-checked and extended. Luckily, at that time Sibylle Schwarz and Lim Yohanes Stefanus were in Dresden. Sibylle was on a sabbatical and Stef was a guest lecturer. Together with Emma we got the job done.

My daugther Pamela developed the logo of the *Weak Completion Semantics*.

Contents

Chapter 1

Introduction

where we motivate and introduce the Weak Completion Semantics.

1.1 The Goal

Our long-term research goal is the development of a cognitive theory for adequately modelling human reasoning tasks. The theory should be *computational* in that answers to queries can be computed. The theory should be *integrated* in that different human reasoning tasks can be modelled by the theory without changing the theory.

We believe that currently the *Weak Completion Semantics (WCS)* is a very good, if not the best candidate for such an integrated and computational cognitive theory. The WCS is based on ideas initially proposed by Keith Stenning and Michiel van Lambalgen in [60], but is mathematically sound [34], has been applied to various human reasoning tasks like the suppression task [16], the selection task [17], the belief-bias effect [55], or ethical decision tasks [32], has outperformed the twelve cognitive theories considered by Philip N. Johnson-Laird and Sangeet Khemlani [41] in syllogistic reasoning [52], and can be implemented in a connectionist setting [18].

Given a human reasoning task, the first step within the WCS is to construct a logic program representing the task. The construction of these programs is based on several principles, some of which are well-established like using *licenses for inferences* [60], *existential import*, or *Gricean implicature*, whereas others are novel like *unknown generalization.* If interpreted under the three-valued logic of [46], the programs have a unique supported model, which can be computed by iterating the semantic operator introduced by [60]. Reasoning is performed and answers are computed with respect to these models. Skeptical abduction is added if some observations in the given human reasoning task can not be explained otherwise.

1.2 The Suppression Task

The suppression task is a set of twelve experiments carried out by Ruth M. J. Byrne for the first time in the 1980s [11]. The experiments have been repeated several times leading to similar results (see e.g. [20]).

1.2.1 Forward Reasoning

To start with, participants were told that

she has an essay to write (1.1)

and

if she has an essay to write then she will study late in the library (1.2)

and were asked whether given fact (1.1) and conditional (1.2) they are willing to conclude that *she will study late in the library.* 96% of the participants drew this conclusion, whereas the remaining 4% answered either *no* or *I don't know.*

A second group of participants were given then conditional

if she has a textbook to read then she will study late in the library (1.3)

together with conditional (1.2) and fact (1.1). Again, they were asked whether they are willing to conlcude that *she will study late in the library* and, again, 96% of the participants drew this conclusion.

A third group of participants were given the conditional

if the library is open, then she will study late in the library (1.4)

together with conditional (1.2) and fact (1.1). But in this case, only 38% of the participants were willing to conclude that *she will study late in the library.* All experiments are summarized in Table 1.2.

The answers of the three groups of participants cannot be explained by classical two-valued logic. The first experiment can be modeled as an example of the usage of the classical inference rule *modus ponens*: Given F and $F \to G$ then G follows in classical two-valued logic, where F and G are formulas and $\to$ denotes implication. Instantiating F by *she has an essay to write* and G by *she will study late in the library* allows to apply modus ponens leading to the conclusion that *she will study late in the library.*

Classical two-valued logic is *monotonic* in the sense that drawn conclusions persist if additional knowledge is added to the knowledge base. In particular, adding conditional (1.3) to the knowledge base consisting of conditional (1.2) and fact (1.1) still allows to apply modus ponens to (1.1) and (1.2) to conclude that *she will study late in the library.* This can be used to explain the answers given by the second group.

Unfortunately, the monotonicity of classical two-valued logic will also allow to conclude that *she will study late in the library* in the third experiment. However, the majority of the participants did not draw this conclusion. Rather, they suppressed it. The majority of the participants in the third group cannot have used classical logic to draw their conclusions. This is a first example to show that human reasoning is *nonmonotonic* in that the addition of new knowledge to a given knowledge base may lead to a revision of previously drawn conclusions.

If the participants in the third group did not use classical two-valued logic what else did they use? How did they come up with their answers? Can we formally specify a system in which the three experiments can be uniformly modeled such that the answers given by the majority of the participants can be computed? In this book we will argue that the *Weak Completion Semantics* is a good candidate for such a formal system.

The *Weak Completion Semantics* uses a representation which is quite common in logic programming and logic based knowledge representation and reasoning (see e.g. [44, 2]). The fact (1.1) is represented by

$$e \leftarrow \top, \tag{1.5}$$

where e is a (nullary) relation symbol or (propositional) atom representing that *she has an essay to write* and $\top$ is a constant denoting *truth.* Conditionals are represented as *licences for inference* [61] with the help of abnormality predicates. The conditional (1.2) is represented by

$$\ell \leftarrow e \wedge \neg ab_1,^{1} \tag{1.6}$$

where ℓ is an atom representing that *she will study late in the library* and ab_1 is an abnormality predicate. As nothing abnormal with respect to conditional (1.2) is known in Byrne's experiment, we assume that ab_1 is false. Such a (negative) assumption can be expressed by

$$ab_1 \leftarrow \bot, \tag{1.7}$$

where $\bot$ is a constant denoting *falsehood.* Altogether, we obtain a (logic) program consisting of the (program) clauses (1.5), (1.6), and (1.7). In this program, each clause defines a relation, viz. e, ℓ, and ab_1, respectively. More precisely, each clause is the if-half of a definition. These if-halves are weakly completed by adding the corresponding only-if-halves to obtain[2]

$$\begin{array}{rcl} e & \leftrightarrow & \top, \\ \ell & \leftrightarrow & e \wedge \neg ab_1, \\ ab_1 & \leftrightarrow & \bot. \end{array} \tag{1.8}$$

[1] We assume that $\neg$ is binding stronger that $\wedge$ and that $\wedge$ is binding stronger than $\leftarrow$ and $\leftrightarrow$. Hence, parenthesis can be omitted.

[2] Whenever we write programs only the first clause is numbered and that number refers to the program as a whole.

Programs as well as their weak completions will be interpreted under the three-valued Łukasiewicz logic [46] (see Table 1.1). In a three-valued logic, all atoms are interpreted by mapping them to either $\top$, $\bot$, or U, where U is a constant denoting *unknowability.* Such a three-valued interpretation I can be represented by two sets $I^\top$ and $I^\bot$, where $I^\top$ contains all relation symbols mapped to true and $I^\bot$ contains all relation symbols mapped to false. Because an interpretation is a mapping, the intersection of $I^\top$ and $I^\bot$ must be empty. Moreover, if a relation symbol does neither occur in $I^\top$ nor in $I^\bot$, then it is mapped to unknown. For example, the interpretation

$$\langle I^\top, I^\bot \rangle = \langle \emptyset, \emptyset \rangle \tag{1.9}$$

maps the relation symbols e, ℓ, and ab_1 to unknown. This interpretation is often called the *empty interpretation.* The interpretation

$$\langle I^\top, I^\bot \rangle = \langle \{e\}, \{ab_1\} \rangle \tag{1.10}$$

maps e to true, ab_1 to false, and ℓ to unknown, whereas the interpretation

$$\langle I^\top, I^\bot \rangle = \langle \{e, \ell\}, \{ab_1\} \rangle \tag{1.11}$$

maps e and ℓ to true and ab_1 to false.

The interpretation (1.11) maps each equivalence occurring in the weakly completed program (1.8) to true. For the first and the last equivalence this follows immediately from the definition of the interpretation. The right-hand-side $e \wedge \neg ab_1$ of the second equivalence evaluates to true because the negation of false is true and the conjunction of true and true is true (see Table 1.1). Because the left-hand-side ℓ is also mapped to true by (1.11), the equivalence is mapped to true as well. Interpretations which map all equivalences of a weakly completed program to true are called *models.* In particular, interpretation (1.11) is a model for (1.8). Interpretations (1.9) and (1.10) are not models for (1.8). The first and the last equivalence are not mapped to true under (1.9) because e and ab_1 are mapped to unknown. Under (1.10) the first and the last equivalence are mapped to true, however, the second equivalence is not. Its left-hand-side is mapped to unknown, whereas its right-hand-side is mapped to true.

As shown in Chapter 3 each weakly completed program admits a least model if interpreted under the three-valued Łukasiewicz logic [46]. In particular, the interpretation (1.11) is the least model of the weakly completed program (1.8). Moreover, the least model can be computed by iterating a semantic operator. Starting with the empty interpretation (1.9), this operator computes immediate consequences as follows:

- Because $e \leftrightarrow \top$ is contained in (1.8), e must be mapped to true; because $ab_1 \leftrightarrow \bot$ is contained in (1.8), ab_1 must be mapped to false; thus, interpretation (1.10) is obtained.

[3] Kleene has defined various logics in [42]. Herein, we will refer to the logic presented in this table as *Kleene logic.*

[4] Although this logic has already been considered in [42], it has received much attention in the logic programming community after the publication of Fitting's paper [21]. Therefore, we will refer to the logic presented herein as *Fitting logic.*

Classical two-valued logic

F	$\neg F$
$\top$	$\bot$
$\bot$	$\top$

$\wedge$	$\top$	$\bot$
$\top$	$\top$	$\bot$
$\bot$	$\bot$	$\bot$

$\vee$	$\top$	$\bot$
$\top$	$\top$	$\top$
$\bot$	$\top$	$\bot$

$\leftarrow$	$\top$	$\bot$
$\top$	$\top$	$\top$
$\bot$	$\bot$	$\top$

$\leftrightarrow$	$\top$	$\bot$
$\top$	$\top$	$\bot$
$\bot$	$\bot$	$\top$

Łukasiewicz three-valued logic

F	$\neg F$
$\top$	$\bot$
$\bot$	$\top$
U	U

$\wedge$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	U	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$

$\vee$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	$\top$	U	U
$\bot$	$\top$	U	$\bot$

$\leftarrow$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	U	$\top$	$\top$
$\bot$	$\bot$	U	$\top$

$\leftrightarrow$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	$\top$	U
$\bot$	$\bot$	U	$\top$

Kleene three-valued logic[3]

F	$\neg F$
$\top$	$\bot$
$\bot$	$\top$
U	U

$\wedge$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	U	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$

$\vee$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	$\top$	U	U
$\bot$	$\top$	U	$\bot$

$\leftarrow$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	U	U	$\top$
$\bot$	$\bot$	U	$\top$

$\leftrightarrow$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	U	U
$\bot$	$\bot$	U	$\top$

Fitting three-valued logic[4]

F	$\neg F$
$\top$	$\bot$
$\bot$	$\top$
U	U

$\wedge$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	U	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$

$\vee$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	$\top$	U	U
$\bot$	$\top$	U	$\bot$

$\leftarrow$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	U	U	$\top$
$\bot$	$\bot$	U	$\top$

$\leftrightarrow$	$\top$	U	$\bot$
$\top$	$\top$	$\bot$	$\bot$
U	$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\bot$	$\top$

Table 1.1: Truth tables for classical two-valued logic as well as three-valued logics introduced by Łukasiewicz and Kleene and used by Fitting. F denotes a formula. Under Łukasiewicz logic, $\mathsf{U} \leftarrow \mathsf{U} = \top$, whereas under Kleene and Fitting logic, $\mathsf{U} \leftarrow \mathsf{U} = \mathsf{U}$. Under Łukasiewicz logic, $\mathsf{U} \leftrightarrow \mathsf{U} = \top$, whereas under Kleene logic, $\mathsf{U} \leftrightarrow \mathsf{U} = \mathsf{U}$.

- Under interpretation (1.10), the right-hand-side $e \wedge \neg ab_1$ of the second equivalence occurring in (1.8) is mapped to true; consequently, its left-hand-side ℓ must also be mapped to true; thus, interpretation (1.11) is obtained.

Because (1.11) is the least model of the weakly completed program (1.8) and because ℓ is mapped to true under this interpretation, the system concludes that *she will study late in the library.*

In a nutshell, under the *Weak Completion Semantics* the following steps are taken given a scenario:

1. Reasoning towards a (logic) program.
2. Weakly completing the program.
3. Computing its least model under Łukasiewicz logic.
4. Reasoning with respect to the least model.
5. If necessary, applying skeptical abduction.

The first four steps have already been demonstrated in modeling the first experiment. The fifth step will be necessary in Subsections 1.2.3 and 1.2.4, which deal with experiments seven to twelve.

In the second experiment, the additional conditional (1.3) is represented by

$$\ell \leftarrow t \wedge \neg ab_2, \tag{1.12}$$

where t is an atom representing that *she has a textbok to read* and ab_2 is another abnormality predicate. As nothing abormal with respect to conditional (1.3) is known in Byrne's experiment, we assume that ab_2 is represented by

$$ab_2 \leftarrow \bot. \tag{1.13}$$

Altogether, we obtain a program consisting of the clauses (1.5), (1.6), (1.7), (1.12), and (1.13). The weak completion of this program is

$$\begin{array}{rcl} e & \leftrightarrow & \top, \\ \ell & \leftrightarrow & (e \wedge \neg ab_1) \vee (t \wedge \neg ab_2), \\ ab_1 & \leftrightarrow & \bot, \\ ab_2 & \leftrightarrow & \bot. \end{array} \tag{1.14}$$

To obtain the least model of (1.14) we start with the empty interpretation and proceed as follows:

- Because $e \leftrightarrow \top$ is contained in (1.14), e must be mapped to true; because $ab_1 \leftrightarrow \bot$ and $ab_2 \leftrightarrow \bot$ are contained in (1.14), ab_1 and ab_2 must be mapped to false; thus, we obtain the interpretation
$$\langle \{e\}, \{ab_1, ab_2\}\rangle.$$
- Under this interpretation the right-hand-side $(e \wedge \neg ab_1) \vee (t \wedge \neg ab_2)$ of the second equivalence occurring in (1.14) is mapped to true; consequently, its left-hand-side ℓ must be mapped to true; thus, we obtain the interpretation
$$\langle \{e, \ell\}, \{ab_1, ab_2\}\rangle.$$

As this interpretation is the least model of the weakly completed program (1.14), the system concludes that *she will study late in the library.*

In the third experiment, the additional conditional (1.4) is represented by

$$\ell \leftarrow o \wedge \neg ab_3, \tag{1.15}$$

where o is an atom representing the fact that *the library is open* and ab_3 is yet another abnormality predicate. As before we assume that ab_3 is mapped to unknown:

$$ab_3 \leftarrow \bot. \tag{1.16}$$

But as soon as the additional conditional (1.4) is given, we believe – following [61] – that most participants recognize that the antecedent of (1.4) is in fact an additional condition for going into the library or, as we prefer, that the *the library not being open* is an abnormality with respect to conditional (1.2). This can be represented by

$$ab_1 \leftarrow \neg o. \tag{1.17}$$

Again following [61] we believe that not having a reason for going to the library prevents somebody to go to the library, which can be represented by

$$ab_3 \leftarrow \neg e \tag{1.18}$$

with *having an essay to write* being the only reason mentioned in the third experiment.

Robert Anthony Kowalski discusses this example in [43] as well. In particular, in the context of legal reasoning the two conditionals can be represented as *context independent* or *context dependent rules.* In the former form they read:

> *If she has an essay to write and the library is open then she will study late in the library.*
>
> *If the library is open and she has a reason for studying in the library then she will study late in the library.*

whereas in the latter we obtain:

> *If she has an essay to write then she will study late in the library. However, if the library is not open then she will not study late in the library.*
>
> *If the library is open then she will study late in the library. However, if she has no reason for studying in the library then she will not study late in the library.*

We believe that the context dependent form is more approprate as we usually do not know all exceptions. We will come back to this problem in Section 4.5.

Altogether, we obtain a program consisting of the clauses (1.5), (1.6), (1.7), (1.15), (1.16), (1.17), and (1.18). The weak completion of this program is

$$\begin{array}{rcl} e & \leftrightarrow & \top, \\ \ell & \leftrightarrow & (e \wedge \neg ab_1) \vee (o \wedge \neg ab_3), \\ ab_1 & \leftrightarrow & \bot \vee \neg o, \\ ab_3 & \leftrightarrow & \bot \vee \neg e. \end{array} \tag{1.19}$$

To obtain the least model of (1.19) we start with the empty interpretation and proceed as follows:

- Because $e \leftrightarrow \top$ is contained in (1.19), e must be mapped to true; thus, we obtain the interpretation
$$\langle \{e\}, \emptyset \rangle.$$
- Under this interpretation the right-hand-side $\bot \vee \neg e$ of the last equivalence occurring in (1.19) is mapped to false; thus, we obtain the interpretation
$$\langle \{e\}, \{ab_3\} \rangle.$$

One should observe that this interpretation maps all equivalences occurring in (1.19) to true. In particular, both sides of the second and third equivalence are mapped to unknown and $\mathsf{U} \leftrightarrow \mathsf{U} = \top$ under Łukasiewicz logic. As this interpretation is the least model of the weakly completed program (1.19), the system concludes that *it does not know whether she will study late in the library.* In other words, the *Weak Completion Semantics* models the suppression effect demonstrated by the participants of the third experiment.

1.2.2 Denial of the Antecedent

Byrne's experiments four to six are repetitions of the experiments one to three except that the fact (1.1) is replaced by

$$\textit{she does not have an essay to write} \tag{1.20}$$

and the participants were asked whether they are willing to conclude that *she will not study late in the library.*

In the fourth experiment the fact (1.20) and the conditional (1.2) were given. 46% of the participants were willing to conclude that *she will not study late in the library.*

In the fifth experiment the conditional (1.3) was added to the conditional (1.2) and the fact (1.20). In this case, only 4% of the participants were willing to conclude that *she will not study late in the library.*

In the sixth experiment the conditional (1.4) was added to the conditional (1.2) and the fact (1.20). In this case, only 63% of the participants were willing to conclude that *she will not study late in the library.*

The answers of the participants can again not be modeled in classical two valued logic. Given $\neg F$ and $F \rightarrow G$ does not allow to conclude $\neg G$. Consider a two-valued interpretation which maps F to false and G to true as well as the usual classical two-valued truth tables for the negation $\neg$ and implication $\rightarrow$ (see Table 1.1). Then, $\neg F$ as well as $F \rightarrow G$ are mapped to true, but $\neg G$ is mapped to false. Clearly, the participants who concluded that *she will not study late in the library* did not use classical logic.

On the other hand, the fifth experiment demonstrates that the conlusion drawn in the forth experiment is suppressed if additional knowledge becomes available.

Under the *Weak Completion Semantics* the fact (1.20) is modeled by

$$e \leftarrow \bot. \tag{1.21}$$

For the fourth experiment we obtain the program consisting of the clauses (1.21), (1.6), and (1.7). The weak completion of this program is

$$\begin{array}{rcl} e & \leftrightarrow & \bot, \\ \ell & \leftrightarrow & e \wedge \neg ab_1, \\ ab_1 & \leftrightarrow & \bot. \end{array} \tag{1.22}$$

To obtain the least model of (1.22) we start with the empty interpretation and proceed as follows:

- Because $e \leftrightarrow \bot$ and $ab_1 \leftrightarrow \bot$ are contained in (1.22), both, e and ab_1, must be mapped to false; thus, we obtain the interpretation

  $$\langle \emptyset, \{e, ab_1\} \rangle.$$

- Because under this interpretation the right-hand-side $e \wedge \neg ab_1$ of the second equivalence occurring in (1.22) is mapped to false, its left-hand-side ℓ must be mapped to false as well; thus, we obtain the interpretation

  $$\langle \emptyset, \{e, ab_1, \ell\} \rangle.$$

As this interpretation is the least model for (1.22) the system concludes that *she will not study late in the library.*

For the fifth experiment we obtain the program consisting of the clauses (1.21), (1.6), (1.7), (1.12), and (1.13). The weak completion of this program is

$$\begin{aligned} e &\leftrightarrow \bot, \\ \ell &\leftrightarrow (e \wedge \neg ab_1) \vee (t \wedge \neg ab_2), \\ ab_1 &\leftrightarrow \bot, \\ ab_2 &\leftrightarrow \bot. \end{aligned} \tag{1.23}$$

To obtain the least model of (1.23) we start with the empty interpretation and proceed as follows:

- Because $e \leftrightarrow \bot$, $ab_1 \leftrightarrow \bot$, and $ab_2 \leftrightarrow \bot$ are contained in (1.23), e, ab_1, and ab_2 must be mapped to false; thus, we obtain the interpretation
$$\langle \emptyset, \{e, ab_1, ab_2\} \rangle.$$

As this interpretation is the least model for (1.23), the system concludes that *it does not know whether she will not study late in the library.*

For the sixth experiment we obtain the program consisting of the clauses (1.21), (1.6), (1.7), (1.15), (1.16), (1.17), and (1.18). The weak completion of this program is

$$\begin{aligned} e &\leftrightarrow \bot, \\ \ell &\leftrightarrow (e \wedge \neg ab_1) \vee (o \wedge \neg ab_3), \\ ab_1 &\leftrightarrow \bot \vee \neg o, \\ ab_3 &\leftrightarrow \bot \vee \neg e. \end{aligned} \tag{1.24}$$

To obtain the least model of (1.24) we start with the empty interpretation and proceed as follows:

- Because $e \leftrightarrow \bot$ is contained in (1.24), e must be mapped to false; thus, we obtain the interpretation
$$\langle \emptyset, \{e\} \rangle.$$
- Because under this interpretation the right-hand-side $\bot \vee \neg e$ of the last equivalence occurring in (1.24) is mapped to true, ab_3 must be mapped to true as well; thus we obtain the interpretation
$$\langle \{ab_3\}, \{e\} \rangle.$$
- Because under this interpretation the right-hand-side $(e \wedge \neg ab_1) \vee (o \wedge \neg ab_3)$ of the second equivalence occurring in (1.24) is mapped to false, ℓ must be mapped to false as well; thus, we obtain the interpretation
$$\langle \{ab_3\}, \{e, \ell\} \rangle.$$

As this interpretation is the least model for (1.24), the system concludes that *she will not study late in the library.*

1.2.3 Affirmation of the Consequent

Byrne's experiments seven to nine are repetitions of the experiments one to three except that the fact (1.1) is replaced by

$$\textit{she will study late in the library} \tag{1.25}$$

and the participants were asked whether they are willing to conclude that *she has an essay to write.*

In the seventh experiment the fact (1.25) and the conditional (1.2) were given. 72% of the participants were willing to conclude that *she has an essay to write.*

In the eighth experiment the conditional (1.3) was added to the conditional (1.2) and the fact (1.25). In this case, only 13% of the participants were willing to conclude that *she has an essay to write.*

In the ninth experiment the conditional (1.4) was added to the conditional (1.2) and the fact (1.25). In this case, only 71% of the participants were willing to conclude that *she has an essay to write.*

These experiments are different than the experiments considered in the previous subsections as conditional (1.2) is represented by $\ell \leftarrow e \wedge \neg ab_1$ in (1.6), which is already a definition for ℓ. Just adding $\ell \leftarrow \top$ to (1.6) does not seem to be particular meaningful. Rather, this seems to be a case for applying *abduction* [29]. Can ℓ be explained by adding certain facts to the knowledge base? In the context of logic programming [38], fact (1.25) is consider to be an *observation* which needs to be explained by adding *abducibles* to a given program. Possible abducibles are usually atoms which are undefined in the given program. *Explanations* are subsets of the set of abducibles such that these subsets together with the given program entail the observation. In most cases, minimal subsets are preferred.

For the seventh experiment the program consists of the clauses (1.6) and (1.7). Hence, the atoms ℓ and ab_1 are defined, but the atom e is undefined. As we are using a three-valued logic, the set of abducibles is

$$\{e \leftarrow \top,\ e \leftarrow \bot\}.$$

e can be added either as a positive fact or as a negative assumption to the program. Adding the first abducible $e \leftarrow \top$ to (1.6) and (1.7) and weakly completing the program we obtain (1.8), whose least model is

$$\langle \{e, \ell\}, \{ab_1\} \rangle.$$

This model does not only explain the observation ℓ by mapping it to true, but e is mapped to true as well. Hence, the system will conclude that *she has an essay to write.* The answer is quite intuitive. In the given context the only reason for studying late in the library is the need to write an essay.

One should observe that adding the second abducible $e \leftarrow \bot$ to (1.6) and (1.7) and weakly completing the program we obtain

$$\begin{array}{rcl} e & \leftrightarrow & \bot, \\ \ell & \leftrightarrow & e \wedge \neg ab_1, \\ ab_1 & \leftrightarrow & \bot. \end{array}$$

The least model of this weakly completed program is

$$\langle \emptyset, \{ab_1, e, \ell\} \rangle.$$

This model does not explain the observation ℓ as it maps ℓ to false.

One should also observe, that adding both abducibles to (1.6) and (1.7) and weakly completing the program will lead to the equivalence

$$e \leftrightarrow \top \vee \bot,$$

which is equivalent to

$$e \leftrightarrow \top.$$

Hence, this case is equivalent to considering only the first abducible. But, it shows another feature of the *Weak Completion Semantics*: If a positive fact like $e \leftarrow \top$ and a negative assumption like $e \leftarrow \bot$ occur together in a program and the program is weakly completed, then the negative assumption is overwritten by the positive fact.

For the eighth experiment the program consists of the clauses (1.6), (1.7), (1.12), and (1.13). Hence, the atoms ℓ, ab_1, and ab_2 are defined, whereas the atoms e and t are undefined. Thus, the set of abducibles is

$$\{e \leftarrow \top,\ e \leftarrow \bot,\ t \leftarrow \top, t \leftarrow \bot\}.$$

Adding the first abducible $e \leftarrow \top$ to (1.6), (1.7), (1.12), and (1.13) and weakly completing the program we obtain (1.14), whose least model is

$$\langle \{e, \ell\}, \{ab_1, ab_2\} \rangle. \tag{1.26}$$

This model explains ℓ. The reason for going to the library is also spelled out, viz. that *she has an essay to write.*

But in the context of this experiment, another reason for going to the library is mentioned explicitely, viz. that *she has a textbook to read.* Indeed, adding the third abducible $t \leftarrow \top$ to (1.6), (1.7), (1.12), and (1.13) and weakly completing the program we obtain

$$\begin{array}{rcl} t & \leftrightarrow & \top, \\ \ell & \leftrightarrow & (e \wedge \neg ab_1) \vee (t \wedge \neg ab_2), \\ ab_1 & \leftrightarrow & \bot, \\ ab_2 & \leftrightarrow & \bot, \end{array}$$

whose least model is

$$\langle\{t, \ell\}, \{ab_1, ab_2\}\rangle. \tag{1.27}$$

This model explains ℓ as well and the reason for going to the library in this model is that *she has a textbook to read.*

In fact, $e \leftarrow \top$ and $t \leftarrow \top$ are the only minimal explanations for ℓ in this case. However, the models (1.26) and (1.27) differ in their interpretation of the atom e. Whereas (1.26) maps e to $\top$, (1.27) maps e to $\cup$. Taking both models into account and reasoning skeptically [**?**], the system concludes that *it does not know whether she has an essay to write.*

One should observe that in this case a creduluous reasoner would have concluded that *she has an essy to write*, because there is a model, viz. (1.26), which maps the observation ℓ and the atom e to true. As reported in [11], only 13% of the partipants drew this conclusion. This is the first example discussed in this book indicating that humans seem to reason skeptically.

For the ninth experiment the program consists of the clauses (1.6), (1.7), (1.15), (1.16), (1.17), and (1.18). Hence, the atoms ℓ, ab_1, and ab_3 are defined, whereas the atoms e and o are undefined. Thus, the set of abducibles is

$$\{e \leftarrow \top,\ e \leftarrow \bot,\ o \leftarrow \top,\ o \leftarrow \bot\}.$$

Adding the abducibles $e \leftarrow \top$ and $o \leftarrow \top$ to (1.6), (1.7), (1.15), (1.16), (1.17), and (1.18) and weakly completing the program we obtain

$$\begin{array}{rcl} e & \leftrightarrow & \top, \\ o & \leftrightarrow & \top, \\ \ell & \leftrightarrow & (e \wedge \neg ab_1) \vee (o \wedge \neg ab_3), \\ ab_1 & \leftrightarrow & \bot \vee \neg o, \\ ab_3 & \leftrightarrow & \bot \vee \neg e, \end{array}$$

whose least model is

$$\langle\{e, o, \ell\}, \{ab_1, ab_3\}\rangle.$$

This model explains the observation ℓ. As it maps e to true, the system concludes that *she has an essay to write.* One should observe that the set

$$\{e \leftarrow \top,\ o \leftarrow \top\}$$

is the only minimal explanation for the observation ℓ. Deleting, for example, $o \leftarrow \top$ from this set would lead to the weakly completed program (1.19), whose least model is

$$\langle\{o\}, \{ab_3\}\rangle.$$

It does not explain the observation ℓ.

1.2.4 Denial of the Consequent

Byrne's experiments ten to twelve are repetitions of the experiments one to three except that the fact (1.1) is replaced by

$$\textit{she will not study late in the library} \tag{1.28}$$

and the participants were asked whether thet are willing to conclude that *she does not have an essay to write.*

In the tenth experiment the fact (1.28) and the conditional (1.2) were given. 92% of the participants were willing to conclude that *she does not have an essay to write.*

In the eleventh experiment the conditional (1.3) was added to the conditional (1.2) and the fact (1.28). In this case, only 96% of the participants were willing to conclude that *she does not have an essay to write.*

In the twelvth experiment the conditional (1.4) was added to the conditional (1.2) and the fact (1.28). In this case, only 33% of the participants were willing to conclude that *she does not have an essay to write.*

As in the previous Subsection 1.2.3 the clause $\ell \leftarrow e \wedge \neg ab_1$ representing conditional (1.2) is a definition for ℓ and, hence, (1.28) should again be considered as an observation which needs to be explained.

For the tenth experiment the program consists of the clauses (1.6) and (1.7). Hence, the set of abducibles is

$$\{e \leftarrow \top,\ e \leftarrow \bot\}.$$

Adding the second abducible $e \leftarrow \bot$ to (1.6) and (1.7) and weakly completing the program we obtain (1.22), whose least model is

$$\langle \emptyset, \{e, ab_1, \ell\} \rangle.$$

This model does not only explain the observation by mapping ℓ to $\bot$, but it also maps e to false. Hence, the system will conclude that *she does not have an essay to write.* One should observe that

$$\{e \leftarrow \bot\}$$

is the only explanation for the observation (1.28).

For the eleventh experiment the program consists of the clauses (1.6), (1.7), (1.12), and (1.13). Hence, the set of abducibles is

$$\{e \leftarrow \top,\ e \leftarrow \bot,\ t \leftarrow \top, t \leftarrow \bot\}.$$

The only minimal explanation for the observation (1.28) is

$$\{e \leftarrow \bot,\ t \leftarrow \bot\}.$$

Adding the abducibles $e \leftarrow \bot$ and $t \leftarrow \bot$ to (1.6), (1.7), (1.12), and (1.13) and weakly completing the program we obtain

$$\begin{array}{rcl} e & \leftrightarrow & \bot, \\ t & \leftrightarrow & \bot, \\ \ell & \leftrightarrow & (e \wedge \neg ab_1) \vee (t \wedge \neg ab_2), \\ ab_1 & \leftrightarrow & \bot, \\ ab_2 & \leftrightarrow & \bot, \end{array}$$

whose least model is

$$\langle \emptyset, \{e, t, ab_1, ab_2, \ell\} \rangle.$$

This model maps ℓ and e to false. Hence, the system will conclude that *she does not have an essay to write.*

For the twelvth experiment the program consists of the clauses (1.6), (1.7), (1.15), (1.16), (1.17), and (1.18). The set of abducibles is

$$\{e \leftarrow \top,\ e \leftarrow \bot,\ o \leftarrow \top,\ o \leftarrow \bot\}.$$

There are two minimal explanations for the observation (1.28), viz.

$$\{e \leftarrow \bot\}$$

and

$$\{o \leftarrow \bot\}.$$

Adding $e \leftarrow \bot$ to (1.6), (1.7), (1.15), (1.16), (1.17), and (1.18) and weakly completing the program we obtain (1.24), whose least model is

$$\langle \{ab_3\}, \{e, \ell\} \rangle. \tag{1.29}$$

Adding $o \leftarrow \bot$ to (1.6), (1.7), (1.15), (1.16), (1.17), and (1.18) and weakly completing the program we obtain

$$\begin{array}{rcl} o & \leftrightarrow & \bot, \\ \ell & \leftrightarrow & (e \wedge \neg ab_1) \vee (o \wedge \neg ab_3), \\ ab_1 & \leftrightarrow & \bot \vee \neg o, \\ ab_3 & \leftrightarrow & \bot \vee \neg e \end{array}$$

whose least model is

$$\langle \{ab_1\}, \{o, \ell\} \rangle. \tag{1.30}$$

Comparing (1.29) and (1.30) and reasoning skeptically, the system will conclude that *it does not know whether she does not have an essay to write.* One should observe, that because of (1.29) a creduluous reasoner would have concluded *she does not have an essay to write.* However, as reported by Byrne, only 33% of the participants were willing to do so. This is the second example discussed in this book showing that humans seem to reason skeptically.

The results are summarized in Table 1.2.

Ex.	facts				conditionals			queries				WCS
	(1.1)	(1.20)	(1.25)	(1.28)	(1.2)	(1.3)	(1.4)					
	e	$\neg e$	ℓ	$\neg\ell$				ℓ	$\neg\ell$	e	$\neg e$	
1	X				X			96%				$\top$
2	X				X	X		96%				$\top$
3	X				X		X	38%				U
4		X			X				46%			$\top$
5		X			X	X			4%			U
6		X			X		X		63%			$\top$
7			X		X					72%		$\top$
8			X		X	X				13%		U
9			X		X		X			71%		$\top$
10				X	X						92%	$\top$
11				X	X	X					96%	$\top$
12				X	X		X				33%	U

Table 1.2: The results of the suppression task as reported in [11] and the conclusions drawn by the *Weak Completion Semantics* (WCS).

Chapter 2

Foundations

where we define some basic notions concerning logics, fixed point theory, metric methods, the fluent calculus, and connectionist networks. Experienced readers may skip this chapter. It was added to make the book self-contained.

2.1 Logics

We consider an *alphabet* consisting of

- a finite set of *function symbols* with arity greater or equal than 0,
- a countably infinite set of *variables*,
- a finite or countably infinite set of *relation symbols* with arity greater or equal than 0,
- the connectives $\neg$, $\wedge$, $\vee$, $\leftarrow$, and $\leftrightarrow$ called *negation*, *conjunction*, *disjunction*, *implication*, and *equivalence*, respectively,
- the *existential quantifier* $\exists$,
- the *universal quantifier* $\forall$,
- the special symbols $'('$, $')'$, and $','$, i.e. the parenthesis and the comma.

Nullary function symbols are often called *constant symbols*, whereas nullary relation symbols are often called *propositional variables*. We usually assume that the alphabet is implicitely given.

The set of *terms* is the smallest set satisfying the following conditions:

1. Each variable is a term.
2. If f is an n-ary function symbol, $n \geq 0$, and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term as well.

In case of nullary function symbols we write c instead of $c()$.

The set of *atoms* consists of all expressions of the form $p(t_1, \ldots, t_n)$, where p is an n-ary relation symbol, $n \geq 0$, and $t_1, \ldots, t_n$ are terms. In case of nullary relation symbols we write p instead of $p()$.

A *literal* is either an atom or its negation. For example, let ab be a nullary relation symbol. Then, ab and $\neg ab$ are literals

A term, atom or literal is said to be *ground* if and only if it does not contain the occurrence of a variable. For example, let p be a unary relation symbol, a a constant symbol, and X a variable. Then, $p\,a$[1] and a are ground, whereas $p\,X$ and X are not ground.

The set of *formulas* is the smallest set satisfying the following conditions:

1. Each atom is a formula.
2. If F is a formula, then so is $\neg F$.
3. If F and G are formulas, then so are $(F \wedge G)$, $(F \vee G)$, $(F \leftarrow G)$, and $(F \leftrightarrow G)$.
4. If F is a formula and X is a variable, then $(\forall X)\, F$ and $(\exists X)\, F$ are formulas as well.

We assume the following *precedence hierarchy* $\succ$ among the connectives and the quantifiers:

$$\{\forall, \exists\} \succ \neg \succ \wedge \succ \vee \succ \{\leftarrow, \leftrightarrow\}.$$

For example, let e, ℓ, and ab_1 be propositional variables. Then,

$$e,\ \ell,\ ab_1,\ \neg ab_1,\ e \wedge \neg ab_1,\ \ell \leftarrow e \wedge \neg ab_1$$

are formulas, where the last formula corresponds to

$$(\ell \leftarrow (e \wedge \neg ab_1))$$

because of the precedence hierarchy.

Equality is a particular relation enjoying certain properties which require particular attention. An *equation* is an atom of the form $s \approx t$, where s and t are terms and $\approx$ is a binary relation symbol written infix. Let 1 be a constant symbol and $\circ$ a binary function symbol written infix. Then, the equations

$$\begin{array}{rcl} X \circ 1 & \approx & X, \\ X \circ Y & \approx & Y \circ X, \\ (X \circ Y) \circ Z & \approx & X \circ (Y \circ Z) \end{array} \tag{2.1}$$

[1] If p is a unary relation symbol and is applied to a constant symbol a or a variable X, then we write $p\,a$ and $p\,X$ instead of $p(a)$ and $p(X)$, respectively.

state that 1 is a unit with respect to $\circ$ and that $\circ$ is commutative as well as associative. In other words, equations (2.1) specify an *AC1-theory*. The equations presented in (2.1) are assumed to be *universally closed* in that each variable occurring in an equation is assumed to be prefixed by a universal quantifier. For example,

$$X \circ 1 \approx X$$

should be understood as

$$(\forall X)\, X \circ 1 \approx X.$$

Because equations, atoms, formulas (see below) are usually assumed to be unviversally closed, we omit the universal quantifiers.

The equality relation enjoys some typical properties, viz. *reflexivity*, *symmetry*, *transitivity* as well as *substitutivity* for function and relation symbols. These properties can be expressed by the following (universally closed) *axioms of equality*:

$$\begin{array}{rcl} X \approx X & \leftarrow & \top, \\ X \approx Y & \leftarrow & Y \approx X, \\ X \approx Z & \leftarrow & X \approx Y \wedge Y \approx Z, \\ f(X_1,\ldots,X_n) \approx f(Y_1,\ldots,Y_n) & \leftarrow & \bigwedge_{i=1}^{n} X_i \approx Y_i, \\ r(Y_1,\ldots,Y_n) & \leftarrow & r(X_1,\ldots,X_n) \wedge \bigwedge_{i=1}^{n} X_i \approx Y_i. \end{array} \tag{2.2}$$

The substitutivity axioms are defined for each function symbol f and each relation symbol r occurring in the underlying alphabet. Universal quantifiers have been omitted.

An *equational theory* consists of set of (universally closed) equations together with the (universally closed) axioms of equality. It is specified by the set of equations.

An equational theory defines a finest congruence relation $\equiv$ on the set of ground terms (see e.g. [30]). Let t be a ground term. $[t]$ denotes the congruence class defined by $\equiv$ and containing t. Suppose that the set of equations is

$$\begin{array}{rcl} a & \approx & b, \\ b & \approx & c, \end{array}$$

where a, b, and c are constant symbols. Then,

$$a \equiv b \equiv c$$

and

$$[a] = [b] = [c].$$

If the set of equations is empty, then the congruence class defined by $\equiv$ and containing t consists only of t. In this case, we write t instead of $[t]$.

We abbreviate $p([t_1], \ldots, [t_n])$ by $[p(t_1, \ldots, t_n)]$. Furthermore, $[p(t_1, \ldots, t_n)] = [q(s_1, \ldots, s_m)]$ if and only if $p = q$, $n = m$, and $[t_i] = [s_i]$ for all $1 \leq i \leq n$. If the set of equations is empty, then we write $p(t_1, \ldots, t_n)$ instead of $[p(t_1, \ldots, t_n)]$; furthermore, $p(t_1, \ldots, t_n) = q(s_1, \ldots, s_m)$ if and only if $p = q$, $n = m$, and $t_i = s_i$ for all $1 \leq i \leq n$.

For example, consider the set (2.1) of equations. Let d, t_1, and t_2 be constant symbols and p a binary relation symbol. Then,

$$\begin{array}{rcl} [d \circ t_2] & = & [t_2 \circ d], \\ [d \circ t_1 \circ d] & = & [t_1 \circ d \circ d \circ 1], \\ [p(d \circ t_2, d \circ t_1 \circ d)] & = & [p(t_2 \circ d, t_1 \circ d \circ d \circ 1)]. \end{array}$$

The *Herbrand universe* is the quotient of the set of ground terms modulo $\equiv$. In the literature, the notion of a Herbrand universe is often restriced to the case where the equational theory is empty, whereas if the equational theory is not empty, then this set is called the *Herbrand $\mathcal{E}$-universe* (see e.g. [37]). But this distinction seems to be superfluous as by definition, if $\mathcal{E}$ is empty, then the Herbrand $\mathcal{E}$-universe becomes the Herbrand universe. Hence, in this book we opted for dropping the prefix '$\mathcal{E}$'.

The *Herbrand base* is the set of all expressions of the form $[p(t_1, \ldots, t_n)]$, where p is an n-ary function symbol and $[t_i]$, $1 \leq i \leq n$, are elements of the Herbrand universe. In the literature, the notion of a Herbrand base is often restricted to the case where the equational theory is empty, whereas if this is not the case, then this set is called *Herbrand $\mathcal{E}$-base*. For the reasons discussed above we are dropping the prefix '$\mathcal{E}$'. As before, if the equational theory is empty, we omit the squared brackets and write $p(t_1, \ldots, tn)$ instead of $[p(t_1, \ldots, t_n)]$.

An *interpretation* is a mapping from the set of formulas into the set of truth values. In this book we only consider interpretations which map a given equational theory to the truth value $\top$ denoting truth. In other words, we consider only interpretations which *satisfy* a given equational theory. Such interpretations are often called *$\mathcal{E}$-interpretations* in the literature (see e.g. [37]). But as we will only consider $\mathcal{E}$-interpretations, we are again dropping the prefix '$\mathcal{E}$'.

We will usually consider the truth values $\top$, $\bot$, and $\cup$ denoting *truth*, *falsehood*, and *unknowability*, respectively. The *truth ordering* on $\{\bot, \cup, \top\}$ is defined by

$$\bot <_t \cup <_t \top.$$

An interpretation is defined by the truth tables for the connectives and the mapping of the ground atoms to the truth values. Several examples are given in Table 1.1.

It remains to represent the mapping of the ground atoms to truth values. A three-valued interpretation I can be represented by $\langle I^\top, I^\bot \rangle$, where $I^\top$ and $I^\bot$ are disjoint subsets of the Herbrand base such that $[A] \in I^\top$ if and only if $I(A) = \top$, $[A] \in I^\bot$ if and only if $I(A) = \bot$, and $[A] \not\in I^\top \cup I^\bot$ if and only if $I(A) = \cup$, where A is a ground atom.

An interpretation I is a *model* for a formula F, in symbols $I \models F$, if and only if I maps F to $\top$. As interpretations have to satisfy a given equational theory and the axioms of equality, so do models.

2.2 Fixed Point Theory

Let $\mathcal{S}$ be a set. A *(binary) relation* R on $\mathcal{S}$ is a subset of $\mathcal{S} \times \mathcal{S}$. xRy denotes $(x, y) \in R$.

A relation $\leq$ on $\mathcal{S}$ is a *partial order* if the following conditions hold:

- *reflexivity*: $(\forall x \in \mathcal{S})\,(x \leq x)$.
- *antisymmetry*: $(\forall x, y \in \mathcal{S})\,(x \leq y \wedge y \leq x \rightarrow x = y)$.
- *transitivity*: $(\forall x, y, z \in \mathcal{S})\,(x \leq y \wedge y \leq z \rightarrow x \leq z)$.

For example, $(2^{\mathcal{S}}, \subseteq)$ is a *partially ordered set.*

Let $(\mathcal{S}, \leq)$ be a partially ordered set.

- $a \in S$ is an *upper bound* of $\mathcal{X} \subseteq \mathcal{S}$ if for every $x \in \mathcal{X}$ we have $x \leq a$.
- $a \in \mathcal{S}$ is the *least upper bound* of $\mathcal{X} \subseteq \mathcal{S}$ if a is an upper bound of $\mathcal{X}$ and for every upper bound a' of $\mathcal{X}$ we find $a \leq a'$.
- $lub\,(\mathcal{X})$ denotes the least upper bound of $\mathcal{X}$ if it exists.
- $b \in \mathcal{S}$ is a *lower bound* of $\mathcal{X} \subseteq \mathcal{S}$ if for every $x \in \mathcal{X}$ we have $b \leq x$.
- $b \in \mathcal{S}$ is the *greatest lower bound* of $\mathcal{X} \subseteq \mathcal{S}$ if b is a lower bound of $\mathcal{X}$ and for every lower bound b' of $\mathcal{X}$ we find $b' \leq b$.
- $glb\,(\mathcal{X})$ denotes the greatest lower bound of $\mathcal{X}$ if it exists.
- A non-empty subset $\mathcal{X}$ of $\mathcal{S}$ is *directed* if for every $x, y \in \mathcal{X}$ there exists some $z \in \mathcal{X}$ such that $x \leq z$ and $y \leq z$.

A partially ordered set $\mathcal{C}$ is a *complete partial order* if $\mathcal{C}$ has a least element and for every directed subset $\mathcal{X}$ of $\mathcal{C}$ there exists $lub\,(\mathcal{X}) \in \mathcal{C}$ and $lub\,(\mathcal{X}) \in \mathcal{C}$.

Let $(\mathcal{S}, \leq)$ be a partially ordered set and $f : \mathcal{S} \rightarrow \mathcal{S}$ a mapping.

- f is *monotonic* if for every $x, y \in \mathcal{S}$ such that $x \leq y$ we find $f(x) \leq f(y)$.
- f is *continuous* if for every directed subset $\mathcal{X}$ of $\mathcal{S}$ we find

$$f(lub\,(\mathcal{X})) = lub\,(\{f(x) \mid x \in \mathcal{X}\}).$$

Proposition 1 *Every continuous mapping is montonic.*

Proof [**?**].

Theorem 2 *(Knaster-Tarski) Let $\mathcal{C}$ be a complete partial order and f a monotonic mapping on $\mathcal{C}$. Then, f has a least fixed point.*

Proof [13].

Proposition 3 *Let $\mathcal{C}$ be a complete partial order with least element $\perp$, f a monotonic mapping on $\mathcal{C}$, x the least fixed point of f, and*

$$\begin{array}{lcll} x_0 & = & \perp, & \\ x_\alpha & = & f(x_{\alpha-1}) & \textit{for every non-limit ordinal } \alpha > 0, \\ x_\alpha & = & \mathrm{lub}\,\{x_\beta \mid \beta < \alpha\} & \textit{for every limit ordinal } \alpha. \end{array}$$

Then, for some ordinal γ we find $x = x_\gamma$.

Proof [40].

Theorem 4 *(Kleene fixed point theorem) Let $\mathcal{C}$ be a complete partial order with least element $\perp$ and f a continuous mapping on $\mathcal{C}$. Then, the least fixed point of f is*

$$\mathrm{lub}\,(\{f^n(\perp) \mid n \geq 0\}).$$

Proof [13].

Lemma 5 *Let $\mathcal{X}$ be a directed set and $\mathcal{Y}$ be a finite subset of $\mathcal{X}$. Then, $\mathcal{X}$ contains an upper bound of $\mathcal{Y}$.*

Proof [?].

As an immediate consequence of Lemma 5 we obtain:

Corollary 6 *Any finite directed set contains its own least upper bound.*

Proposition 7 *Let $\mathcal{C}$ be a finite complete partial order and f a monotonic mapping on $\mathcal{C}$. Then, f is continuous.*

Proof [27].

From Proposition 7 and Theorem 4 we conclude:

Corollary 8 *Let $\mathcal{C}$ be a finite complete partial order with least element $\perp$ and f be a monotonic mapping on $\mathcal{C}$. Then, the least fixed point of f is*

$$\mathrm{lub}\,(\{f^n(\perp) \mid n \geq 0\}).$$

2.3 Metrics

A *metric* on a set $\mathcal{M}$ is a mapping $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ such that

1. $d(x, y) = 0$ if and only if $x = y$,
2. $d(x, y) = d(y, x)$,
3. $d(x, z) \leq d(x, y) + d(y, z)$.

$(\mathcal{M}, d)$ is called *metric space.*

A sequence $s_1, s_2, \ldots$ is *Cauchy* if for every $\varepsilon > 0$ there is an integer N such that for all $n, m \geq N$, $d(s_n, s_m) \leq \varepsilon$.

A sequence $s_1, s_2, \ldots$ *converges* if there is an s such that, for every $\varepsilon > 0$, there is an integer N such that for all $n \geq N$, $d(s_n, s) \leq \varepsilon$.

Let $(\mathcal{M}, d)$ be a metric space.

- $(\mathcal{M}, d)$ is *complete* if every Cauchy sequence converges.
- A mapping $f : \mathcal{M} \to \mathcal{M}$ is a *contraction* if for all $x, y \in \mathcal{M}$ there exists a $k \in \mathbb{R}$ with $0 < k < 1$ such that $d(f(x), f(y)) \leq k \cdot d(x, y)$.

Theorem 9 *(Banach Contraction Mapping Theorem) A contraction mapping f defined on a complete metric space $(\mathcal{M}, d)$ has a unique fixed point. The sequence y, $f(y)$, $f(f(y))$, $\ldots$ converges to this fixed point for any $y \in \mathcal{M}$.*

Proof [7].

2.4 The Fluent Calculus

The *fluent calculus* is a first-order calculus for reasoning about actions and causality. The original idea was published together with Josef Schneeberger in [35]. The name *fluent calculus* was coined later by Michael Thielscher in [65].

The basic idea underlying the fluent calculus is to consider states of the world as multisets of fluents. Multisets are reified and represented as terms in an equational logic program with the help of the AC1-theory (2.1). In fact, there is a one-to-one correspondence between a multiset[2] of the form

$$\dot{\{}t_1, \ldots, t_n\dot{\}}$$

and the AC1-term

$$t_1 \circ \ldots \circ t_n \circ 1,$$

where they AC1-symbols $\circ$ and 1 do not occur in the terms t_i, $1 \leq i \leq n$.

As example consider a scenario where a student is working late at the university: *She would like to eat a cookie (c) which she had bought earlier and would like to drink a lemonade (ℓ); checking her purse, she only has a dollar note (d) and a quater (q). In the basement of the*

[2] Multisets are denoted by putting a dot on top of the curly brackets.

building the university has set up a change machine, which allows to change a dollar note into four quaters, and a vending machine, which allows to buy a lemonade for three quaters.

The initial state can be descibed by the multiset

$$\dot{\{}c, d, q\dot{\}} \tag{2.3}$$

which is represented by the term

$$c \circ d \circ q \circ 1.$$

which is equal to

$$c \circ d \circ q$$

under AC1 (2.1). There are two actions, viz. exchanging a dollar note into four quaters and buying a lemonade for three quaters, which are represented by

$$\begin{aligned} \mathit{action}(d, \mathit{change}, q \circ q \circ q \circ q) &\leftarrow \top \\ \mathit{action}(d \circ d \circ d, \mathit{get}, \ell) &\leftarrow \top \end{aligned} \tag{2.4}$$

The first argument of *action* represents the *preconditions*, the second argument is the *name*, and the third argument represents the *immediate effects* of the action.

Causalities are expressed by the ternary relation *causes*, where

$$\mathit{causes}(s, p, s')$$

expresses that the execution of plan p transforms state s into state s' and a *plan* is a sequence of actions. The relation *causes* can be defined recursively on the structure of plans:

$$\begin{aligned} \mathit{causes}(X, [\,], X) &\leftarrow \top, \\ \mathit{causes}(X, [H|T], Y) &\leftarrow \mathit{action}(P, H, E) \wedge X \approx P \circ Z \wedge \mathit{causes}(E \circ Z, T, Y), \\ X \approx X &\leftarrow \top. \end{aligned} \tag{2.5}$$

The first clause expresses that the empty plan $[\,]$ causes no change of the current state X. The second clause expresses that a plan with initial action (or head) H and remaining sequence of actions (or tail) T transforms the current state X into state Y if there is an action with name H, preconditions P, and immediate effects E, which is applicable in X, its execution leads to the successor state $E \circ Z$, and the successor state is recursively transformed into Y by T. More precisely, H is applicable in X if we find a Z such that $X \approx P \circ Z$. The fluents occurring in P are consumed and deleted from X, whereas Z is equal to the remaining fluents occurring in X. In this way, Z is the solution to the *frame problem* [48, 49, 31]. Z can be computing by unifying the terms X and $P \circ Z$ modulo the equational theory (2.1) [62]. Formally, SLDE-resolution [30] is applied to solve the planning problem consisting of the clauses occurring in (2.4) and (2.5) and the initial query

$$\leftarrow \mathit{causes}(c \circ d \circ q, P, c \circ \ell \circ X)$$

asking whether there exists a plan P and fluents X such that the initial state (2.3) is transformed into a state where the student has a cookie, a lemonade, and X.

In the discussed scenario, the initial state (2.3) is transformed into

$$\dot{\{}c, q, q, q, q, q\dot{\}} \tag{2.6}$$

by applying the *change* action, which is further transformed into

$$\dot{\{}c, \ell, q, q\dot{\}} \tag{2.7}$$

by applying the *get* action. In the first step d is consumed and four quaters are added leading to (2.6). In the second step, three quarters are consumed and the lemonade ℓ is produced leading to (2.7).

It has been shown in [26] that the basic fluent calculus is equivalent to the multiplicative fragment of linear logic [47, 24] and the linear connection method [9]. In each of these calculi, actions are applied to a state by consuming its preconditions and producing its immediate effects.

There are many extensions of the basic fluent calculus, for example, to provide solutions to the ramification and the qualification problem [63, 64, 66]. In Section 4.6 we will be particularly concerned with the *ramification problem*, i.e., the problem to determine the indirect effects of an action. For example, if we are moving a box from one place to another, then anything that is in the box will go with it, although none of the content of the box is explicitely specified by the move action.

2.5 Connectionist Networks

Chapter 3

Theory

where we rigorously develop the theory of the Weak Completion Semantics.

3.1 Programs

A *(normal logic) program* $\mathcal{P}$ is a finite or countably infinite set of clauses of the form

$$A \leftarrow \mathit{Body}, \tag{3.1}$$

where the *head* A is an atom but not an equation and *Body* is either a non-empty conjunction of literals, $\top$, or $\bot$. Clauses are assumed to be universally closed. Clauses of the form $A \leftarrow \top$ and $A \leftarrow \bot$ are called *(positive) facts* and *(negative) assumptions*, respectively. All other clauses are called *rules.* A program is said to be a *datalog* program if and only if the terms occurring in this program are variables and constant symbols.

For example, consider the program

$$\begin{array}{rcl} e & \leftarrow & \top, \\ \ell & \leftarrow & e \wedge \neg ab_1, \\ ab_1 & \leftarrow & \bot. \end{array} \tag{3.2}$$

The first clause is a fact, the second a rule, and the last an assumption.

Let $\mathcal{P}$ be a program. The underlying *alphabet* consists precisely of the symbols occurring in $\mathcal{P}$. If $\mathcal{P}$ is a first-order program, then the alphabet must contain at least one constant symbol as, otherwise, the Herbrand universe would be empty.

A *ground instance* of a clause (3.1) is obtained from (3.1) by replacing each variable occurring in A and *Body* by a ground term. This replacement needs to be consistent in that multiple occurrences of the same variable are replaced by the same ground term. For example,

$$q(s(a)) \leftarrow q(a)$$

and

$$q(s(s(a))) \leftarrow q(s(a))$$

are ground instances of the rule

$$q(s(X)) \leftarrow q(X),$$

where q is a unary relation symbol, s a unary function symbol, a a constant symbol, and X a variable. The *ground instance* of a program $\mathcal{P}$ is obtained from $\mathcal{P}$ by replacing each clause occurring in $\mathcal{P}$ by the set of its ground instances. For example, let $\mathcal{P}$ consist of the clauses

$$\begin{array}{rcl} q(a) & \leftarrow & \top, \\ q(s(X)) & \leftarrow & q(X). \end{array}$$

Then, the ground instance of $\mathcal{P}$ is the following infinite set of clauses:

$$\begin{array}{rcl} q(a) & \leftarrow & \top, \\ q(s(a)) & \leftarrow & q(a), \\ q(s(s(a))) & \leftarrow & q(s(a)), \\ \vdots & \vdots & \vdots\,. \end{array}$$

Let $g\mathcal{P}$ denote the ground instance of $\mathcal{P}$. One should observe that the ground instance of a *propositional* program, i.e. a program in which only nullary relation symbols occur, is the program itself. For example, all programs considered in Chapter 1 like (3.2) are propositional and, in these cases, $g\mathcal{P} = \mathcal{P}$.

In the remainder of this section we assume that programs are ground. Let $\mathcal{P}$ be a ground program.

A (ground) atom A is *defined* in $\mathcal{P}$ iff $\mathcal{P}$ contains a clause of the form $A \leftarrow \mathit{Body}$; otherwise A is said to be *undefined*. The set of all atoms that are defined in $\mathcal{P}$ is denoted by $\mathit{def}\,\mathcal{P}$. For example,

$$\mathit{def}\,(3.2) = \{e,\ \ell,\ ab_1\}.$$

All relation symbols occurring in the program (3.2) are defined. Consider the program

$$\begin{array}{rcl} e & \leftarrow & \top, \\ \ell & \leftarrow & e \wedge \neg ab_1, \\ \ell & \leftarrow & t \wedge \neg ab_2, \\ ab_1 & \leftarrow & \bot, \\ ab_2 & \leftarrow & \bot. \end{array} \tag{3.3}$$

Then,

$$\mathit{def}\,(3.3) = \{e,\ \ell,\ ab_1,\ ab_2\},$$

whereas t is undefined. Likewise, consider the program

$$\begin{array}{rcl} e & \leftarrow & \top, \\ \ell & \leftarrow & e \wedge \neg ab_1, \\ \ell & \leftarrow & o \wedge \neg ab_3, \\ ab_1 & \leftarrow & \bot, \\ ab_1 & \leftarrow & \neg o, \\ ab_3 & \leftarrow & \bot, \\ ab_3 & \leftarrow & \neg e. \end{array} \tag{3.4}$$

Then,

$$\mathit{def}\,(3.4) = \{e,\ \ell,\ ab_1,\ ab_3\},$$

whereas o is undefined.

Let A be a ground atom. $\neg A$ is *assumed* in $\mathcal{P}$ if and only if $\mathcal{P}$ contains an assumption $A \leftarrow \bot$ and $\mathcal{P}$ does neither contain a fact $A \leftarrow \top$ nor a rule $A \leftarrow \mathit{Body}$. For example,

- $\neg ab_1$ is assumed in (3.2),
- $\neg ab_1$ and $\neg ab_2$ are assumed in (3.3),
- nothing is assumed in (3.4).

Consider the following transformation for a given ground program $\mathcal{P}$:

1. For all $A \in \mathit{def}\,\mathcal{P}$, replace all clauses of the form $A \leftarrow \mathit{Body}_1$, $A \leftarrow \mathit{Body}_2$, ... occurring in $\mathcal{P}$ by $A \leftarrow \mathit{Body}_1 \vee \mathit{Body}_2 \ldots \vee \ldots$.
2. Add $A \leftarrow \bot$ for all undefined ground atoms A occurring in $\mathcal{P}$.
3. Replace all occurrences of $\leftarrow$ by $\leftrightarrow$.

The resulting set of equivalences is called the *completion* of $\mathcal{P}$ following [12].[1] It is denoted by $c\mathcal{P}$. If the second step is ommitted then the resulting set of equivalences is called the *weak completion* of $\mathcal{P}$ following [34]. It is denoted by $wc\mathcal{P}$. For example,

- $wc(3.2) = c(3.2) = (1.8)$.
- $wc(3.3) = (1.14)$, whereas $c(3.3)$ contains all equivalences occurring in $wc(3.3)$ together with the equivalence
$$t \leftrightarrow \bot.$$

[1] Keith Leonhard Clark has used a different syntactic form in [12]. In classical two-valued logic his form is semantically equivalent to the one used herein. We will come back to this in Section 4.5.

- $wc(3.4) = (1.19)$, whereas $c(3.4)$ contains all equivalences occurring in $wc(3.4)$ together with the equivalence
$$o \leftrightarrow \bot.$$

- The programs, their weak completions, and the least models of the weak completions for the first six experiments of the suppression task are shown in Table 3.1.

As another example consider the program which consists of the clauses

$$\begin{array}{rcl} pa & \leftarrow & \top, \\ qb & \leftarrow & rb. \end{array} \tag{3.5}$$

As its completion we obtain

$$\begin{array}{rcl} pa & \leftrightarrow & \top, \\ pb & \leftrightarrow & \bot, \\ qa & \leftrightarrow & \bot, \\ qb & \leftrightarrow & rb, \\ ra & \leftrightarrow & \bot, \\ rb & \leftrightarrow & \bot, \end{array}$$

whereas its weak completion is

$$\begin{array}{rcl} pa & \leftrightarrow & \top, \\ qb & \leftrightarrow & rb. \end{array}$$

One should observe, that under the completion not only ra and rb are equivalent to false, but also pb and qa, whereas under the weak completion these atoms are all mapped to unknown. We will come back to this example in Section 4.5.

3.2 The Meaning of Programs

Throughout this section we consider a given logic program $\mathcal{P}$ and a given equational theory. Each equational theory defines a finest congruence relation $\equiv$ on the set of ground terms as discussed in Chapter 2. The equational theory may be empty, in which case $\equiv$ is syntactic equality.

Interpretations are mappings from the set of formulas into the set $\{\top, \cup, \bot\}$. They are represented by two sets $I^\top$ and $I^\bot$ consisting of the set of all ground atoms which are mapped to true and false, respectively. An interpretation I is a model for a program $\mathcal{P}$, in symbols $I \models \mathcal{P}$, if and only if I maps each clause occurring in $\mathcal{P}$ to true (see Chapter 2). Because clauses are universally closed, I maps a clause to true if and only if it maps each ground instance of the clause to true.

Consider the following program, its weak completion, and its completion:

$\mathcal{P}$	$wc\mathcal{P}$	$c\mathcal{P}$
$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow (e \wedge \neg ab_1)$	$\ell \leftrightarrow (e \wedge \neg ab_1)$
$\ell \leftarrow t \wedge \neg ab_2$	$\vee\ (t \wedge \neg ab_2)$	$\vee\ (t \wedge \neg ab_2)$
$e \leftarrow \top$	$e \leftrightarrow \top$	$e \leftrightarrow \top$
$ab_1 \leftarrow \bot$	$ab_1 \leftrightarrow \bot$	$ab_1 \leftrightarrow \bot$
$ab_2 \leftarrow \bot$	$ab_2 \leftrightarrow \bot$	$ab_2 \leftrightarrow \bot$
		$t \leftrightarrow \bot$

Then, we obtain:

I	$I(\mathcal{P})$	$I(wc\mathcal{P})$	$I(c\mathcal{P})$
$\langle\{e, ab_1\}, \emptyset\rangle$	$\top$	$\bot$	$\bot$
$\langle\{e, \ell\}, \{ab_1, ab_2\}\rangle$	$\top$	$\top$	U
$\langle\{e, \ell, t\}, \{ab_1, ab_2\}\rangle$	$\top$	$\top$	$\bot$
$\langle\{e, \ell\}, \{ab_1, ab_2, t\}\rangle$	$\top$	$\top$	$\top$

Remember that we interprete formulas under Łukasiewicz logic [46] (see Table 1.1). We intent to show that under this logic the *model intersection property* holds for programs as well as their weak completions. In other words, the intersection of all models for a program is a model for the program, where the intersection of two models $\langle I^\top, I^\bot\rangle$ and $\langle J^\top, J^\bot\rangle$ is defined as $\langle I^\top \cap J^\top, I^\bot \cap J^\top\rangle$. If the model intersection property holds then there is a least model. Thus, if we want to compute the logical consequences of programs or their weak completions then we need to consider only their least models.

In classical, two valued logic, the model intersection property holds for *definite* programs, i.e. programs in which the body of each clause is a conjunction of atoms or the constant $\top$ (see e.g. [44, 2]). In this case, the proof is quite straightforward by showing that the intersection of two models of a program is again a model of the program. Unfortunately, this is not the case if programs are interpreted under Łukasiewicz logic. As an example consider a (definite) program consisting of

$$\begin{aligned} p &\leftarrow q_1 \wedge r_1, \\ p &\leftarrow q_2 \wedge r_2, \end{aligned} \tag{3.6}$$

the empty equational theory, and the interpretations

$$\langle\emptyset, \{p, q_1, r_2\}\rangle$$

and

$$\langle\emptyset, \{p, q_2, r_1\}\rangle.$$

Both interpretations are models for the program as they map the bodies as well as the heads of the clauses to false. However, their intersection

$$\langle\emptyset, \{p\}\rangle$$

is not a model for the program as this interpretation maps the bodies to unknown and the heads to false and $\bot \leftarrow \mathsf{U} = \mathsf{U}$ in Łukasiewicz logic (see Table 1.1).

To prove the model intersection property under Łukasiewicz logic we proceed in two steps. Firstly, we show in Proposition 10 that if an interpretation $\langle I^\top, I^\bot\rangle$ is a model for a program then $\langle I^\top, \emptyset\rangle$ is a model as well. In other words, instead of mapping atoms to false, they may be mapped to unknown. Secondly, we consider only models of the form $\langle I^\top, \emptyset\rangle$ and show in Proposition 11 that the intersection of two such models is again a model.

Proposition 10 *Let I be an interpretation. If $I = \langle I^\top, I^\bot\rangle \models \mathcal{P}$ then $I' = \langle I^\top, \emptyset\rangle \models \mathcal{P}$.*

Proof Let $\mathcal{P}$ be a program and $I = \langle I^\top, I^\bot\rangle \models \mathcal{P}$, i.e. for all clauses $A \leftarrow \mathit{Body} \in g\mathcal{P}$ we find $I(A \leftarrow \mathit{Body}) = \top$. By definition of the Łukasiewicz implication, we have $I(A \leftarrow \mathit{Body}) = \top$ if and only if $I(A) \geq_t I(\mathit{Body})$ with respect to the truth ordering $\bot <_t \mathsf{U} <_t \top$. We consider all possible cases for $I(A)$ and show that $I' \models A \leftarrow \mathit{Body}$ by $I'(A) \geq_t I'(\mathit{Body})$:

1. If $I(A) = \top$ then $A \in I^\top$ and, because $I' = \langle I^\top, \emptyset\rangle$ we find $I'(A) = \top$ as well. In this case, $I'(A) \geq_t I'(\mathit{Body})$ holds for any *Body* because $I'(\mathit{Body}) \in \{\bot, \mathsf{U}, \top\}$ and $\top$ is the truth-maximal element occurring in $\{\bot, \mathsf{U}, \top\}$.

2. If $I(A) = \mathsf{U}$ then $A \notin I^\top \cup I^\bot$. Hence, $A \notin I^\top$ and because $I' = \langle I^\top, \emptyset\rangle$ we find $I'(A) = \mathsf{U}$ as well. From $I \models A \leftarrow \mathit{Body}$, i.e. $I(A \leftarrow \mathit{Body}) = \top$, we learn $I(\mathit{Body}) \leq_t I(A) = \mathsf{U}$ and, therefore, $I(\mathit{Body}) \in \{\bot, \mathsf{U}\}$. Hence, we find a conjunct L in *Body* such that $I(L) = \min_t\{I(L) \mid L \text{ is a conjunct in } \mathit{Body}\} = I(\mathit{Body}) \in \{\bot, \mathsf{U}\}$, where $\min_t$ denotes the minimal element with respect to the truth ordering $<_t$. We consider two cases for the literal L:

   (a) If L is an atom then $I(L) \in \{\bot, \mathsf{U}\}$ and we find
   $$I'(\mathit{Body}) = I'(L) = \mathsf{U} = I'(A)$$
   by the definition of $I'^\bot$, i.e. $I'^\bot = \emptyset$.

   (b) If L is a negated atom, i.e. $L = \neg B$ with B being an atom then because $I(L) = I(\neg B) \in \{\bot, \mathsf{U}\}$ we find $I(B) \in \{\mathsf{U}, \top\}$ and $I'(B) \in \{\mathsf{U}, \top\}$. Hence, $I'(\neg B) \in \{\bot, \mathsf{U}\}$.

   In both cases, we have $I'(A \leftarrow \mathit{Body}) = \top$ by $\mathsf{U} = I'(A) \geq_t I'(\mathit{Body})$.

3. If $I(A) = \bot$ then $I'(A) = \mathsf{U}$ by the definition $I'^\bot = \emptyset$. From $I \models A \leftarrow \mathit{Body}$, i.e. $I(A \leftarrow \mathit{Body}) = \top$, we learn
   $$I(\mathit{Body}) = \min_t\{I(L) \mid L \text{ is a conjunct in } \mathit{Body}\} = \bot.$$
   Hence, there is a conjunct L in *Body* such that
   $$I(L) = \min_t\{I(L) \mid L \text{ is a conjunct in } \mathit{Body}\} = I(\mathit{Body}) = \bot.$$
   We consider two cases for the literal L:

(a) If L is an atom then $I'(L) = \cup$ and we find $I'(Body) \leq_t I'(L) = \cup \leq_t I'(A)$.

(b) If L is a negated atom, i.e. $L = \neg B$ for an atom B, then $I(B) = \top = I'(B)$. Hence, $I'(Body) = I'(\neg B) = \bot \leq_t I'(A)$.

In both cases, we have $I'(A \leftarrow Body) = \top$ by $\cup = I'(A) \geq_t I'(Body)$. □

For example, consider the program (3.2), the empty equational theory, and the interpretation

$$\langle \{e, \ell\}, \{ab_1\} \rangle.$$

It is a model for the program because it maps each clause to true. But,

$$\langle \{e, \ell\}, \emptyset \rangle$$

is also a model for (3.2). Because ab_1 is mapped to unknown and $\cup \leftarrow \bot = \top$, the third clause occurring in (3.2) is mapped to true. Because the body $e \wedge \neg ab_1$ of the second clause is mapped to unknown, the head is mapped to true, and $\top \leftarrow \cup = \top$, the second clause is mapped to true as well. The head and the body of the first clause are mapped to true and, thus, the first clause is also mapped to true.

As another example consider the program (3.6) and the empty equational theory. The empty interpretation $\langle \emptyset, \emptyset \rangle$ is a model for this program as it maps all bodies and all heads to unknown and $\cup \leftarrow \cup = \cup$ in Łukasiewicz logic (see Table 1.1).

As a third example consider the program consisting of the clauses

$$\begin{array}{rcl} qX & \leftarrow & \neg pX, \\ pa & \leftarrow & \top \end{array} \tag{3.7}$$

and the equational theory consisting of

$$a \approx b.$$

The interpretation

$$\langle \{[pa]\}, \{[qb]\} \rangle$$

is a model for (3.7). pa is mapped to true and, hence, the fact $pa \leftarrow \top$ is mapped to true. Because interpretations must satisfy the equational theory and $a \equiv b$, pb is also mapped to true. Consequently, all ground instances of the right-hand-side $\neg pX$ of the rule occurring in (3.7) are mapped to false. Because $[qb] = [qa]$ and qb is mapped to false, the rule is mapped to true. The interpretation

$$\langle \{[pa]\}, \emptyset \rangle$$

is a model for (3.7) as well. Under this interpretation, each instance of the body $\neg pX$ of the rule is again mapped to false, but each instance of the head qX is mapped to unknown. Under Łukasiewicz logic, $\cup \leftarrow \bot = \top$ (see Table 1.1).

Proposition 11 *If* $\langle I_1^\top, \emptyset\rangle \models \mathcal{P}$ *and* $\langle I_2^\top, \emptyset\rangle \models \mathcal{P}$ *then* $\langle I_1^\top \cap I_2^\top, \emptyset\rangle \models \mathcal{P}$*.*

Proof Let $\mathcal{P}$ be a program, $\langle I_1^\top, \emptyset\rangle \models \mathcal{P}$ and $\langle I_2^\top, \emptyset\rangle \models \mathcal{P}$, i.e. for all rules $A \leftarrow \mathit{Body} \in g\mathcal{P}$ we find $I_1(A) \geq_t I_1(\mathit{Body})$ and $I_2(A) \geq_t I_2(\mathit{Body})$. Let $I = \langle I_1^\top \cap I_2^\top, \emptyset\rangle$. Because its second component $I^\perp$ is the empty set we find $I(A) = \min_t(I_1(A), I_2(A)) \in \{\mathsf{U}, \top\}$ for all ground atoms A.

We have to show that $I \models A \leftarrow \mathit{Body}$ for every clause $A \leftarrow \mathit{Body} \in g\mathcal{P}$. We consider all possible cases for $I(A)$ and show $I(A \leftarrow \mathit{Body}) = \top$ by $I(A) \geq_t I(\mathit{Body})$:

1. If $I(A) = \top$ then for any Body we find $I(A) \geq_t I(\mathit{Body})$ because $I(A) = \top$ is the truth-maximal element in $\{\perp, \mathsf{U}, \top\}$.

2. If $I(A) = \min_t(I_1(A), I_2(A)) = \mathsf{U}$ then $I(\mathit{Body}) \leq_t \mathsf{U}$ by the following case analysis:

   (a) If $I_1(A) = \min_t(I_1(A), I_2(A)) = \mathsf{U}$, we find $I_1(\mathit{Body}) \leq_t \mathsf{U}$ because $I_1(A \leftarrow \mathit{Body}) = \top$. Therefore, $I(\mathit{Body}) = \min_t(I_1(\mathit{Body}), I_2(\mathit{Body})) \leq_t \mathsf{U}$.

   (b) If $I_2(A) = \min_t(I_1(A), I_2(A)) = \mathsf{U}$, we find $I_2(\mathit{Body}) \leq_t \mathsf{U}$ because $I_2(A \leftarrow \mathit{Body}) = \top$. Therefore, $I(\mathit{Body}) = \min_t(I_1(\mathit{Body}), I_2(\mathit{Body})) \leq_t \mathsf{U}$.

   In both cases, we have $I(\mathit{Body}) = \min_t(I_1(\mathit{Body}), I_2(\mathit{Body})) \leq_t \mathsf{U} = I(A)$. Therefore, $I(A \leftarrow \mathit{Body}) = \top$.

3. The case $I(A) = \perp$ is impossible because $I^\perp = \emptyset$.

Because the interpretation $I = \langle I_1^\top \cap I_2^\top, \emptyset\rangle$ is a model for each clause occurring in $g\mathcal{P}$, we conclude $I \models \mathcal{P}$. □

As example consider the empty equational theory and the program consisting of the rule

$$p \leftarrow q \wedge \neg r.$$

The interpretations

$$\langle \{p, q\}, \emptyset\rangle$$

and

$$\langle \{p, r\}, \emptyset\rangle$$

are models for this program. Likewise, their intersection

$$\langle \{p, q\} \cap \{p, r\}, \emptyset\rangle = \langle \{p\}, \emptyset\rangle$$

is a model.

Theorem 12 *The model intersection property holds for* $\mathcal{P}$*, i.e.* $\cap\{I \mid I \models \mathcal{P}\} \models \mathcal{P}$*.*

Proof. The claim follows immediately from Propositions 10 and 11. □

The *least model* of a program is the intersection of all models of the program. As examples consider the empty equational theory and the programs (3.2), (3.3), and (3.4). Their least model is

$$\langle \{e\}, \emptyset \rangle.$$

The interpreation

$$\langle \{[pa]\}, \emptyset \rangle = \langle \{[pb]\}, \emptyset \rangle$$

is the least model of program (3.7) and the equational theory $\{a \approx b\}$.

The empty interpretation is the least model of the program consisting only of the rule

$$p \leftarrow q \tag{3.8}$$

and the empty equational theory. One should observe that this does not hold if we interprete programs under the Kleene or the Fitting logic. In this case, the interpretations

$$\langle \{p, q\}, \emptyset \rangle$$

and

$$\langle \emptyset, \{p, q\} \rangle$$

are models for (3.8). However, their intersection is the empty interpretation and this is not a model for (3.8) as $\mathsf{U} \leftarrow \mathsf{U} = \mathsf{U}$ under Kleene and Fitting logic (see Tables 1.1). The model intersection property does not hold if we interprete programs under the Kleene or Fitting logic.

Theorem 13 *The model intersection property holds for wc$\mathcal{P}$ as well.*

We will prove this result in Section 3.3.

Several examples of this result have already been discussed in Chapter 1. The empty interpretation is the least model of the weak completion of the program (3.8). One should observe that the equivalence $p \leftrightarrow q$ has also a least model under Fitting logic, but not under Kleene logic.

Corollary 14 *If I is a model for the weak completion of a program $\mathcal{P}$ then I is a model for $\mathcal{P}$.*

Proof The result follows immediately from the observation that the law of equivalence, i.e. $F \leftrightarrow G$ is semantically equivalent to $(F \leftarrow G) \wedge (G \leftarrow F)$, holds under Łukasiewicz logic und the weak completion of a program is nothing but the addition of only-if-halves of definitions. □

One should observe that this result does not hold under Fitting logic. The empty interpretation is the least model for $p \leftrightarrow q$. However, it is not a model for $p \leftarrow q$.

Table 3.1 depicts the programs, their weak completions, and the least models of their weak completions for the first six experiments reported in [11] and discussed in Chapter 1.

Let $\mathcal{P}$ and $\mathcal{P}'$ be sets of formulas and F a formula. A logic is *monotonic* if and only if the following holds: if $\mathcal{P} \models F$ then $\mathcal{P} \cup \mathcal{P}' \models F$, where $\models$ is the entailment relation defined by the logic. For example, classical two-valued logic is monotonic.

A logic is *nonmonotonic* if and only if it is not monotonic, i.e. if we find sets of formulas $\mathcal{P}$ and $\mathcal{P}'$ and a formula F such that $\mathcal{P} \models F$ and $\mathcal{P} \cup \mathcal{P}' \not\models F$. In the Weak Completion Semantics we will reason with respect to the least model of a weakly completed program. Let

$$\mathcal{P} = \{ab \leftarrow \bot\}$$

Then, the least model of the weak completion of this program is

$$\langle \emptyset, \{ab\} \rangle$$

and the formula $\neg ab$ follows. If we add to $\mathcal{P}$ the set

$$\mathcal{P}' = \{ab \leftarrow \top\},$$

then the weak completion of $\mathcal{P} \cup \mathcal{P}'$ consists of the equivalence

$$ab \leftrightarrow \bot \vee \top.$$

Its least model is

$$\langle \{ab\}, \emptyset \rangle$$

from which we cannot conclude $\neg ab$ anymore. Rather, ab is now entailed. The example nicely illustrates a feature of the Weak Completion Semantics: negative assumptions will be overridden as soon as positive information becomes available.

3.3 Computing Least Models

In this section we address the question of how to compute the least model of the weak completion of a program. The idea goes back to [4], where Krzysztof R. Apt and Maarten H. van Emden showed that the least model of a definite logic program can be computed as the least fixed point of a semantic operator associated with the program. Given a program $\mathcal{P}$ and a classical two-valued interpretation I, the operator maps a ground atom A to true if there exists a clause of the form

$$A \leftarrow \mathit{Body} \tag{3.9}$$

occurring in $g\mathcal{P}$ such that $I(\mathit{Body}) = \top$. In other words, A is an *immediate consequence* of the given interpretation I and the program $\mathcal{P}$. The set of all immediate consequences is

Ex.	$\mathcal{P}$	$wc\mathcal{P}$	$\mathcal{M}_{\mathcal{P}}$
1	$e \leftarrow \top$	$e \leftrightarrow \top$	$\langle\{e,\ell\},\{ab_1\}\rangle$
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow e \wedge \neg ab_1$	
	$ab_1 \leftarrow \bot$	$ab_1 \leftrightarrow \bot$	
2	$e \leftarrow \top$	$e \leftrightarrow \top$	$\langle\{e,\ell\},\{ab_1,ab_2\}\rangle$
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow (e \wedge \neg ab_1) \vee (t \wedge \neg ab_2)$	
	$\ell \leftarrow t \wedge \neg ab_2$	$ab_1 \leftrightarrow \bot$	
	$ab_1 \leftarrow \bot$	$ab_2 \leftrightarrow \bot$	
	$ab_2 \leftarrow \bot$		
3	$e \leftarrow \top$	$e \leftrightarrow \top$	$\langle\{e\},\{ab_3\}\rangle$
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow (e \wedge \neg ab_1) \vee (o \wedge \neg ab_3)$	
	$\ell \leftarrow o \wedge \neg ab_3$	$ab_1 \leftrightarrow \bot \vee \neg o$	
	$ab_1 \leftarrow \bot$	$ab_3 \leftrightarrow \bot \vee \neg e$	
	$ab_3 \leftarrow \bot$		
	$ab_1 \leftarrow \neg o$		
	$ab_3 \leftarrow \neg e$		
4	$e \leftarrow \bot$	$e \leftrightarrow \bot$	$\langle\emptyset,\{e,\ell,ab_1\}\rangle$
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow e \wedge \neg ab_1$	
	$ab_1 \leftarrow \bot$	$ab_1 \leftrightarrow \bot$	
5	$e \leftarrow \bot$	$e \leftrightarrow \bot$	$\langle\emptyset,\{e,ab_1,ab_2\}\rangle$
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow (e \wedge \neg ab_1) \vee (t \wedge \neg ab_2)$	
	$\ell \leftarrow t \wedge \neg ab_2$	$ab_1 \leftrightarrow \bot$	
	$ab_1 \leftarrow \bot$	$ab_2 \leftrightarrow \bot$	
	$ab_2 \leftarrow \bot$		
6	$e \leftarrow \bot$	$e \leftrightarrow \bot$	$\langle\{ab_3\},\{e,\ell\}\rangle$
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow (e \wedge \neg ab_1) \vee (o \wedge \neg ab_3)$	
	$\ell \leftarrow o \wedge \neg ab_3$	$ab_1 \leftrightarrow \bot \vee \neg o$	
	$ab_1 \leftarrow \bot$	$ab_3 \leftrightarrow \bot \vee \neg e$	
	$ab_3 \leftarrow \bot$		
	$ab_1 \leftarrow \neg o$		
	$ab_3 \leftarrow \neg e$		

Table 3.1: Examples for programs, their weak completions, and their least models. In each of these cases the equational theory is empty.

again an interpretation and, hence, we can iterate the operator. Apt and van Emden showed that if we start this process with the empty interpretation then the operator admits a least fixed point which is equal to the least model of the program.

In [21], Melvin Fitting extended this idea to normal logic program by showing that completed logic programs admit a least model under the three-valued Fitting logic which can also be computed as the least fixed point of a semantic operator. Given a program $\mathcal{P}$ and an interpretation $I = \langle I^\top, I^\bot \rangle$, the Fitting operator computes positive immediate consequences as above. In addition, the operator maps a ground atom A to false if for all clauses of the form (3.9) occurring in $g\mathcal{P}$ we find that $I(\mathit{Body}) = \bot$. One should observe that if $g\mathcal{P}$ does not contain a clause of the form (3.9) then A is mapped to false. In other words, if A is undefined in $g\mathcal{P}$ then the Fitting operator maps A to false. This corresponds to the second step in the completion of a program: if A is undefined then $A \leftarrow \bot$ is added to the program, which in the third step of the completion process is turned into $A \leftrightarrow \bot$.

The Fitting operator was modified by Keith Stenning and Michiel van Lambalgen in [61] in that A is mapped to false if and only if there occurs a rule of the form (3.9) in $g\mathcal{P}$ and for all rules of this form we find that $I(\mathit{Body}) = \bot$. As shown in [40, 34] with this modification, the least fixed point of this operator is equal to the least model of the weak completion of the given program $\mathcal{P}$. In this book, we extend these results to logic programs with non-empty equational theories.

Let $\mathcal{P}$ be a program, $\mathcal{E}$ an equational theory, and I an interpretation. We define $\Phi_{\mathcal{P}}(I) = \langle J^\top, J^\bot \rangle$, where

$$\begin{array}{rcl} J^\top & = & \{[A] \mid \text{there exists } A \leftarrow \mathit{Body} \in g\mathcal{P} \text{ and } I(\mathit{Body}) = \top\}, \\ J^\bot & = & \{[A] \mid \text{there exists } A \leftarrow \mathit{Body} \in g\mathcal{P} \\ & & \qquad \text{and for all } A' \leftarrow \mathit{Body} \in g\mathcal{P} \text{ with } [A] = [A'] \text{ we find } I(\mathit{Body}) = \bot\}. \end{array}$$

In the remainder of this section let $\mathcal{P}$ be a program and $\mathcal{E}$ be an equational theory. Recall that $\mathcal{E}$ defines a finest congruence relation $\equiv$ on the set of ground terms (see Section 2.1). Let $\mathcal{X}$ be a set of interpretations. We define

$$\mathcal{X}^\top = \{I^\top \mid \langle I^\top, I^\bot \rangle \in \mathcal{X}\}$$

and

$$\mathcal{X}^\bot = \{I^\bot \mid \langle I^\top, I^\bot \rangle \in \mathcal{X}\}.$$

As an example consider the program consisting of

$$pX \leftarrow qX \tag{3.10}$$

and the equational theory consisting of

$$a \approx b. \tag{3.11}$$

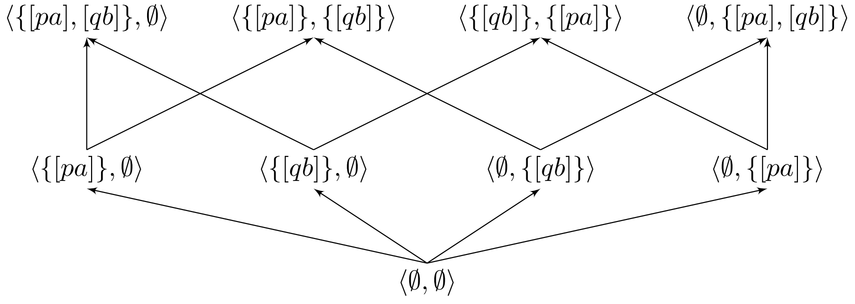


Figure 3.1: The Hasse diagram of the set of interpretations for the program (3.10) and the equational theory (3.11) with respect to the relation $\subseteq$.

The set of interpretations is shown in Figure 3.1. It is partially order with respect to the relation $\subseteq$. But it is not directed. For example, for the interpretations

$$\langle \{[pa], [qb]\}, \emptyset \rangle$$

and

$$\langle \{[pa]\}, \{[pb]\} \rangle$$

there is no interpretation which is a superset of both. However, the subset

$$\{\langle \{[pa], [qb]\}, \emptyset \rangle,\ \langle \{[pa]\}, \emptyset \rangle,\ \langle \{[qb]\}, \emptyset \rangle,\ \langle \emptyset, \emptyset \rangle\}$$

is directed. Now let $\mathcal{X}$ be this subset. Then,

$$\begin{array}{rcl} \mathcal{X}^{\top} & = & \{\{[pa], [qb]\},\ \{[pa]\},\ \{[qb]\},\ \emptyset\}, \\ \mathcal{X}^{\perp} & = & \{\emptyset,\ \emptyset,\ \emptyset,\ \emptyset\}, \\ \bigcup \mathcal{X}^{\top} & = & \{[pa], [qb]\}, \\ \bigcup \mathcal{X}^{\perp} & = & \emptyset \end{array}$$

and

$$\langle \bigcup \mathcal{X}^{\top}, \bigcup \mathcal{X}^{\perp} \rangle = \langle \{[pa], [qb]\}, \emptyset \rangle$$

is the least upper bound of $\mathcal{X}$. This is not a coincidence, but holds in general as the following proposition shows.

Proposition 15 *Let $\mathcal{X}$ be a directed set of interpretations. Then, the interpretation $I = \langle \bigcup \mathcal{X}^{\top}, \bigcup \mathcal{X}^{\perp} \rangle$ is the least upper bound of $\mathcal{X}$.*

Proof

1. Firstly, we show that $\langle\bigcup\mathcal{X}^\top,\bigcup\mathcal{X}^\perp\rangle$ is an interpretation.

   By definition, $\bigcup\mathcal{X}^\top$ and $\bigcup\mathcal{X}^\perp$ are unions of $\equiv$-congruence classes. It remains to show that $\bigcup\mathcal{X}^\top\cap\bigcup\mathcal{X}^\perp=\emptyset$. Assume we find $[A]$ with $[A]\in\bigcup X^\top\cap\bigcup X^\perp$. Then, there exist interpretations $J_1\in\mathcal{X}$ and $J_2\in\mathcal{X}$ such that $[A]\in J_1^\top$ and $[A]\in J_2^\perp$. Because the set $\mathcal{X}$ is directed, it contains a common upper bound $\mathcal{K}$ of J_1 and J_2, where $[A]\in\mathcal{K}^\top$ and $[A]\in\mathcal{K}^\perp$. Then, $\mathcal{K}^\top\cap\mathcal{K}^\perp\neq\emptyset$ and $\mathcal{K}$ is not an interpretation. This contradicts the precondition that $\mathcal{X}$ is a directed set of interpretations. Hence, the assumption $[A]\in\bigcup\mathcal{X}^\top\cap\bigcup\mathcal{X}^\perp$ is false and $\langle\bigcup X^\top,\bigcup X^\perp\rangle$ is an interpretation.

2. Secondly, we show that $\langle\bigcup\mathcal{X}^\top,\bigcup\mathcal{X}^\perp\rangle$ is an upper bound of $\mathcal{X}$, i.e. for all $J\in\mathcal{X}$ we find $J^\top\subseteq\bigcup\mathcal{X}^\top$ and $J^\perp\subseteq\bigcup\mathcal{X}^\perp$.

   (a) $J^\top\subseteq\bigcup\mathcal{X}^\top$ because for all $[A]\in J^\top$ we find $[A]\in\bigcup\{J^\top\mid J\in\mathcal{X}\}=\mathcal{X}^\top$.

   (b) $J^\perp\subseteq\bigcup\mathcal{X}^\perp$ because for all $[A]\in J^\perp$ we find $[A]\in\bigcup\{J^\perp\mid J\in\mathcal{X}\}=\mathcal{X}^\perp$.

3. Thirdly, it remains to show that $\langle\bigcup\mathcal{X}^\top,\bigcup\mathcal{X}^\perp\rangle$ is the least upper bound of $\mathcal{X}$, i.e. for every upper bound $\mathcal{J}$ of $\mathcal{X}$, we have $\bigcup\mathcal{X}^\top\subseteq\mathcal{J}^\top$ and $\bigcup\mathcal{X}^\perp\subseteq\mathcal{J}^\perp$.

   Assume that $\mathcal{X}$ has an upper bound $\mathcal{J}$ where $\bigcup\mathcal{X}^\top\not\subseteq\mathcal{J}^\top$ or $\bigcup\mathcal{X}^\perp\not\subseteq\mathcal{J}^\perp$.

   (a) Assume there is an $[A]\in\bigcup\mathcal{X}^\top$ such that $[A]\notin\mathcal{J}^\top$. By $[A]\in\bigcup\mathcal{X}^\top$, there is an interpretation $K\in\mathcal{X}$ where $[A]\in K^\top$. Because $[A]\notin\mathcal{J}^\top$, $\mathcal{J}$ is not an upper bound for $\mathcal{X}$.

   (b) Assume there is an $[A]\in\bigcup\mathcal{X}^\perp$ such that $[A]\notin\mathcal{J}^\perp$. By $[A]\in\bigcup\mathcal{X}^\perp$, there is an interpretation $K\in\mathcal{X}$ where $[A]\in K^\perp$. Because $[A]\notin\mathcal{J}^\perp$, $\mathcal{J}$ is not an upper bound for $\mathcal{X}$.

   In both cases, the assumption that the upper bound $\mathcal{J}$ of $\mathcal{X}$ differs from $\langle\bigcup\mathcal{X}^\top,\bigcup\mathcal{X}^\perp\rangle$ leads to a contradiction. Hence, $\langle\bigcup\mathcal{X}^\top,\bigcup\mathcal{X}^\perp\rangle$ is the least upper bound of $\mathcal{X}$. □

Corollary 16 *The set of all interpretations $\mathcal{I}$ is a complete partial order with respect to $\subseteq$.*

Proof. Reflexivity, antisymmetry and transitivity hold for $\subseteq$. The least element of $\mathcal{I}$ is $\langle\emptyset,\emptyset\rangle$. By Proposition 15, every directed subset of $\mathcal{I}$ has a least upper bound in $\mathcal{I}$. □

Figure 3.1 shows the set of all interpretations for the program (3.10) and the equational theory (3.11).

Proposition 17 *For each program $\mathcal{P}$ and equational theory $\mathcal{E}$ the mapping $\Phi_\mathcal{P}$ is monotonic.*

Proof Assume $I=\langle I^\top,I^\perp\rangle$ and $J=\langle J^\top,J^\perp\rangle$ are interpretations for $\mathcal{P}$ with $I^\top\subseteq J^\top$ and $I^\perp\subseteq J^\perp$. We show that $\Phi_\mathcal{P}(I)=I'=\langle I'^\top,I'^\perp\rangle\subseteq\langle J'^\top,J'^\perp\rangle=J'=\Phi_\mathcal{P}(J)$.

1. $I'^{\top} \subseteq J'^{\top}$: By the definition of $I' = \Phi_{\mathcal{P}}(I)$, we have $[A] \in I'^{\top}$ if and only if there is a clause $A \leftarrow \mathit{Body}$ in $g\mathcal{P}$ such that either

   (a) $\mathit{Body} = \top$ (i.e. $A \leftarrow \mathit{Body}$ is a fact) and, therefore, $J(\mathit{Body}) = \top$ and $[A] \in J'^{\top}$ or

   (b) $I(\mathit{Body}) = \min_t\{I(L) \mid L \text{ is a literal occurring in } \mathit{Body}\} = \top$. Then, for all conjuncts L occurring in Body we have one of the following cases:

      i. If $L = B$ for a ground atom B and $I(B) = \top$ then $[B] \in I^{\top} \subseteq J^{\top}$.

      ii. If $L = \neg B$ for a ground atom B and $I(B) = \bot$ then $[B] \in I^{\bot} \subseteq J^{\bot}$.

   In both cases, $J(\mathit{Body}) = \top$. By definition of $J' = \Phi_{\mathcal{P}}(J)$ we find $[A] \in J'^{\top}$.

2. $I'^{\bot} \subseteq J'^{\bot}$: By the definition of $\Phi_{\mathcal{P}}$ we have $[A] \in I'^{\bot}$ if and only if

   (a) there exists a clause $A \leftarrow \mathit{Body} \in g\mathcal{P}$ and

   (b) for all rules $A' \leftarrow \mathit{Body} \in g\mathcal{P}$ with $[A'] = [A]$ we have $I(\mathit{Body}) = \bot$.

   Hence, for all clauses $A_i \leftarrow \mathit{Body}_i \in g\mathcal{P}$ where $[A_i] = [A]$ we find $\mathit{Body}_i = \bot$ (i.e. the clause $A_i \leftarrow \mathit{Body}_i$ is a negative assumption) or

   $$I(\mathit{Body}_i) = \min_t\{I(L) \mid L \text{ is a literal occurring in } \mathit{Body}_i\} = \bot,$$

   i.e. there is a literal L occurring in Body_i such that $I(L) = \bot$. We have one of the following cases:

   (a) If $\mathit{Body}_i = \bot$ then $J(\mathit{Body}_i) = \bot$.

   (b) If $L = B$ for a ground atom B and $I(B) = \bot$ then $[B] \in I^{\bot} \subseteq J^{\bot}$.

   (c) If $L = \neg B$ for a ground atom B and $I(B) = \top$ then $[B] \in I^{\top} \subseteq J^{\top}$.

   In any of these cases, for all clauses $A_i \leftarrow \mathit{Body}_i \in g\mathcal{P}$ where $[A_i] = [A]$, we find that

   $$J(\mathit{Body}_i) = \min_t\{J(L) \mid L \text{ is a literal occurring in } \mathit{Body}\} = \bot$$

   and, therefore, $[A] \in J'^{\bot}$ by definition of $J' = \Phi_{\mathcal{P}}(J)$.

$\Phi_{\mathcal{P}}$ is monotonic because $I' \subseteq J'$, i.e. $I'^{\top} \subseteq J'^{\top}$ and $I'^{\bot} \subseteq J'^{\bot}$. □

Unfortunately, $\Phi_{\mathcal{P}}$ is generally not continuous. Consider the program $\mathcal{P}$ consisting of the clauses

$$\begin{array}{rcl} q(a) & \leftarrow & \top, \\ q(s(X)) & \leftarrow & q(X), \\ p & \leftarrow & \neg q(X), \end{array} \tag{3.12}$$

and the empty equational theory. The least fixed point of $\Phi_{\mathcal{P}}$ is

$$\langle\{[q(s^k(a))] \mid k \in \mathbb{N}\}, \{[p]\}\rangle$$

and is reached after iterating $\Phi_{\mathcal{P}}$ $\omega + 1$ times, where ω is the first limit ordinal. Hence, by Kleene's fixed point Theorem 4, $\Phi_{\mathcal{P}}$ is not continuous. One should observe that the Herbrand base contains infinitely many equivalence classes

$$[p],\ [q(a)],\ [q(s(a))],\ \ldots,$$

each of which has one element.

Likewise, consider the program $\mathcal{P}$ consisting of the clauses

$$\begin{array}{rcl} q(1) & \leftarrow & \top, \\ q(X \circ a) & \leftarrow & q(X), \\ p & \leftarrow & \neg q(X) \end{array} \tag{3.13}$$

and the AC1-theory (2.1) defined in Section 2.1. The least fixed point of $\Phi_{\mathcal{P}}$ is

$$\langle \{[q(1 \circ \overbrace{a \circ \ldots \circ a}^{k})] \mid k \in \mathbb{N}\}, \{[p]\}\rangle \tag{3.14}$$

and is reached after iterating $\Phi_{\mathcal{P}}$ $\omega + 1$ times. Again, $\Phi_{\mathcal{P}}$ is not continuous. One should observe that the Herbrand base contains infinitely many equivalence classes, viz.

$$[p],\ [q(1)],\ [q(a)],\ [q(a \circ a)],\ \ldots.$$

With the exception of $[p]$ each of these equivalence classes is infinite because

$$1 \equiv 1 \circ 1 \equiv 1 \circ 1 \circ 1 \equiv \ldots$$

and

$$a \equiv a \circ 1,$$

where $\equiv$ is the finest congruence relation on the set of ground terms defined by (2.1).

Because $\Phi_{\mathcal{P}}$ is montonic, the least fixed point of $\Phi_{\mathcal{P}}$ can be obtained by iterating $\Phi_{\mathcal{P}}$ starting with the empty interpretation. But, if $\Phi_{\mathcal{P}}$ is not continuous then we may have to iterate beyond the first limit ordinal ω. So, we are searching for conditions where this is not necessary, viz., for cases, where $\Phi_{\mathcal{P}}$ is continuous.

Proposition 18 *For each finite propositional program $\mathcal{P}$ the mapping $\Phi_{\mathcal{P}}$ is continuous.*

Proof Because the set of all propositional variables in a finite propositional program $\mathcal{P}$ is finite and we only have finitely many truth values, the set $\mathcal{I}$ of all interpretations is finite. By Corollary 16, $\mathcal{I}$ is a complete partial order with respect to the relation $\subseteq$. By Proposition 17, $\Phi_{\mathcal{P}}$ is monotonic on $\mathcal{I}$. Moreover, $\mathcal{I}$ is finite and, thus, by Proposition 7, $\Phi_{\mathcal{P}}$ is continuous. □

Proposition 19 *If the Herbrand base for a program $\mathcal{P}$ and a set of equations $\mathcal{E}$ is finite then the mapping $\Phi_{\mathcal{P}}$ is continuous.*

Proof Let $\mathcal{P}$ be a program and $\mathcal{E}$ be a set of equations such that the Herbrand base as well as $\mathcal{E}$ is finite. The result follows immediately from Proposition 18 and the fact that there is a bijection between the Herbrand base and an equally large set of propositional atoms. □

As an example consider the equational theory consisting of the equation

$$a \approx c$$

and the program $\mathcal{P}$ consisting of the clauses

$$\begin{array}{rcl} qa & \leftarrow & \top, \\ qb & \leftarrow & \top, \\ pX & \leftarrow & qX. \end{array}$$

In this case, the Herbrand base is

$$\{[qa],\ [qb],\ [pa],\ [pb]\}$$

with $[qa] = [qc]$ and $[pa] = [pc]$. Let $r_1 - r_4$ be four propositional variables and define the bijection

$$[qa] \Leftrightarrow r_1,\ [qb] \Leftrightarrow r_2,\ [pa] \Leftrightarrow r_3,\ [pc] \Leftrightarrow r_4.$$

If this bijection is applied to each element of an equivalence class, then the propositional program consisting of the clauses

$$\begin{array}{rcl} r_1 & \leftarrow & \top, \\ r_2 & \leftarrow & \top, \\ r_3 & \leftarrow & r_1, \\ r_4 & \leftarrow & r_2 \end{array}$$

is equivalent to $g\mathcal{P}$. One should observe that the ground instances $pa \leftarrow qa$ and $pc \leftarrow qc$ are both mapped onto $r_3 \leftarrow r_1$.

Unfortunately, this result is insufficient for using the fluent calculus in general as the fluent calculus utilizes a binary function symbol $\circ$ in order to represent multisets of fluents. As shown in (3.14), $\circ$ may be used to define infinitely many equivalence classes in the Herbrand base of a program. However, in the fluent calculus the function symbol $\circ$ is only used to represent multisets. If we consider only finite multisets in the same way as Bart Selman, Hector Joseph Levesque, and David G. Mitchell consider only finite plans in [59] then the Herbrand base for a given program is finite and, consequently, Proposition 19 applies. Likewise, if the initial state is finite and there is no action whose application leads to an increase of the number of fluents occurring in a state then the Herbrand base of such a program can also be restricted to a finite set. In particular, in the context of human reasoning episodes such restrictions appear to be quite reasonable. In the trolley problems discussed in Section 4.6

the largest multiset has size six, the initial states are always finite, and there is no action which increases the number of fluents occurring in a state.

On the other hand, if we consider finite datalog programs and finite sets of equations between constants then the Herbrand base is also finite.

We proceed to show that for a given program $\mathcal{P}$ and a given set of equations $\mathcal{E}$, the least model of $wc\mathcal{P}$ and the least fixed point of $\Phi_{\mathcal{P}}$ coincide.

Lemma 20 *Let $\mathcal{P}$ be a program, $\mathcal{E}$ an equational theory, J the least fixed point of $\Phi_{\mathcal{P}}$, and I a model of $wc\mathcal{P}$. Then, for every ground atom A the following holds:*

1. *If $J(A) = \top$ then $I(A) = \top$.*
2. *If $J(A) = \bot$ then $I(A) = \bot$.*

Proof Let J be the least fixed point of $\Phi_{\mathcal{P}}$. It can be computed by iterating $\Phi_{\mathcal{P}}$ starting from the empty interpretation as follows:

$$J_0 = \langle \emptyset, \emptyset \rangle, \tag{3.15}$$
$$J_\alpha = \Phi_{\mathcal{P}}(J_{\alpha-1}) \text{ for every non-limit ordinal } \alpha > 0, \tag{3.16}$$
$$J_\alpha = \bigcup_{\beta<\alpha} J_\beta \text{ for every limit ordinal } \alpha. \tag{3.17}$$

Then, there must be some ordinal $\alpha_{\mathcal{P}}$ such that $J = J_{\alpha_{\mathcal{P}}}$. We will prove by transfinite induction that for every ordinal α and every ground atom A the following holds:

1. If $J_\alpha(A) = \top$ then $I(A) = \top$.
2. If $J_\alpha(A) = \bot$ then $I(A) = \bot$.

With this result, the claim will follow from Propositions 3 and 17.

Turning to the induction proof, we consider three cases: the base case when the ordinal $\alpha = 0$ and two inductive cases, one for non-limit ordinals and the other for limit ordinals:

1. Let $\alpha = 0$. Then, by (3.15) we find $J_\alpha = \langle \emptyset, \emptyset \rangle$. Because there is no atom such that $J_\alpha(A) = \top$ or $J_\alpha(A) = \bot$, the claim follows trivially.
2. Let $\alpha > 0$ be a non-limit ordinal. By the inductive hypothesis we find for every ground atom B that:

   $$\text{If } J_{\alpha-1}(B) = \top, \text{ then } I(B) = \top. \tag{3.18}$$
   $$\text{If } J_{\alpha-1}(B) = \bot, \text{ then } I(B) = \bot. \tag{3.19}$$

   Moreover, by (3.16) we find $J_\alpha = \Phi_{\mathcal{P}}(J_{\alpha-1})$ and we distinguish two two cases:

(a) If $J_\alpha(A) = \top$ then according to the definition of $\Phi_\mathcal{P}$ there must be some clause $A' \leftarrow \mathit{Body}_i$ in $g\mathcal{P}$ with $[A'] = [A]$ such that $J_{\alpha-1}(\mathit{Body}_i) = \top$. We distinguish two cases with respect to the form of Body_i.

i. If
$$\mathit{Body}_i = B_1 \wedge B_2 \wedge \cdots \wedge B_k \wedge \neg B_{k+1} \wedge \neg B_{k+2} \wedge \cdots \wedge \neg B_m,$$
where each B_j, $1 \leq j \leq m$, is a ground atom. Then, for each s with $1 \leq s \leq k$ we have $J_{\alpha-1}(B_s) = \top$ and for each t with $k < t \leq m$ we have $J_{\alpha-1}(B_t) = \bot$. Using the induction hypothesis and, in particular, (3.18) and (3.19) we learn that for each s with $1 \leq s \leq k$ we have $I(B_s) = \top$ and for each t with $k < t \leq m$ we have $I(B_t) = \bot$. Hence, $I(\mathit{Body}_i) = \top$.

ii. If
$$\mathit{Body}_i = \top,$$
then $I(\mathit{Body}_i) = I(\top) = \top$.

In either case, $I(\mathit{Body}_i) = \top$. Furthermore, in $wc\mathcal{P}$ there is a formula of the form $A' \leftrightarrow F$, where F is a disjunction with Body_i as one of the disjuncts. Thus, we have $I(F) = \top$ and also $I(A' \leftrightarrow F) = \top$ because I is a model of $wc\mathcal{P}$. This implies $I(A') = \top$. Because $[A] = [A']$ and I satisfies $\mathcal{E}$, $I(A) = \top$ holds as well.

(b) If $J_\alpha(A) = \bot$ then according to the definition of $\Phi_\mathcal{P}$ there must be a clause of the form $A \leftarrow \mathit{Body}$ in $g\mathcal{P}$ and all clauses of the form $A' \leftarrow \mathit{Body}_i$ in $g\mathcal{P}$ with $[A'] = [A]$ we find $J_{\alpha-1}(\mathit{Body}_i) = \bot$. Pick an arbitrary but fixed j. Again, we distinguish two cases with respect to the form of Body_j.

i. If
$$\mathit{Body}_j = B_1 \wedge B_2 \wedge \cdots \wedge B_k \wedge \neg B_{k+1} \wedge \neg B_{k+2} \wedge \cdots \wedge \neg B_m,$$
where B_l, $1 \leq l \leq m$, are ground atoms then we have to consider two cases:

A. There is some s with $1 \leq s \leq k$ such that $J_{\alpha-1}(B_s) = \bot$. Then, by (3.19) we find $I(B_s) = \bot$ and, hence, $I(\mathit{Body}_j) = \bot$.

B. There is some t with $k < t \leq m$ such that $J_{\alpha-1}(B_t) = \top$. Then, by (3.18) we obtain $I(B_t) = \top$ and, hence, $I(\mathit{Body}_j) = \bot$.

ii. If
$$\mathit{Body}_j = \bot,$$
then $I(\mathit{Body}_j) = \bot$.

In either case we have $I(\mathit{Body}_j) = \bot$. Because j was arbitrarily chosen, we can conclude that for every i we have $I(\mathit{Body}_i) = \bot$. Furthermore, in $wc\mathcal{P}$ there is a formula of the form $A' \leftrightarrow F$ with $F = \mathit{Body}_1 \vee \mathit{Body}_2 \vee \ldots$. So we have $I(F) = \bot$. Because I is a model of $wc\mathcal{P}$ we find $I(A' \leftrightarrow F) = \top$. This implies $I(A') = \bot$. Because $[A] = [A']$ and I satisfies $\mathcal{E}$, we conclude $I(A) = \bot$.

3) Let α be a limit ordinal. By the induction hypothesis we find for every ground atom B and every ordinal $\beta < \alpha$ that:

$$\text{If } J_\beta(B) = \top, \text{ then } I(B) = \top. \tag{3.20}$$

$$\text{If } J_\beta(B) = \bot, \text{ then } I(B) = \bot. \tag{3.21}$$

Moreover, by (3.17) we have $J_\alpha = \bigcup_{\beta<\alpha} J_\beta$. There are again two cases to consider:

(a) If $J_\alpha(A) = \top$ then there is some ordinal $\beta < \alpha$ such that $J_\beta(A) = \top$. By the induction hypothesis (3.20) we have $I(A) = \top$.

(b) If $J_\alpha(A) = \bot$ then there is some ordinal $\beta < \alpha$ such that $J_\beta(A) = \bot$. By the induction hypothesis (3.21) we have $I(A) = \bot$. □

Lemma 21 *If $\mathcal{P}$ is a program, $\mathcal{E}$ an equational theory, and J a fixed point of $\Phi_\mathcal{P}$ then J is a model of $wc\mathcal{P}$.*

Proof By the definition of $\Phi_\mathcal{P}$ an interpretation $I = \langle I^\top, I^\bot\rangle$ is a fixed point of $\Phi_\mathcal{P}$ if and only if

$$\begin{array}{rcl} I^\top & = & \{[A] \mid \text{there exists } A \leftarrow \mathit{Body} \in g\mathcal{P} \text{ and } I(\mathit{Body}) = \top\}, \\ I^\bot & = & \{[A] \mid \text{there exists } A \leftarrow \mathit{Body} \in g\mathcal{P} \\ & & \quad \text{and for all } A' \leftarrow \mathit{Body} \in g\mathcal{P} \text{ with } [A] = [A'] \text{ we find } I(\mathit{Body}) = \bot\}. \end{array}$$

We show that for every equivalence $A \leftrightarrow F$ occurring in $wc\mathcal{P}$ we have $I(A \leftrightarrow F) = \top$, i.e. $I(A) = I(F)$. We distinguish three cases:

1. If $[A] \in I^\top$ then there is a clause $A \leftarrow \mathit{Body} \in g\mathcal{P}$ such that $I(\mathit{Body}) = \top$. Hence, for each equivalence $A \leftrightarrow F \in wc\mathcal{P}$, where $F = \mathit{Body} \vee F'$ for a (possibly empty) disjunction F', we have $I(F) = I(\mathit{Body} \vee F') = \max_t(I(\mathit{Body}), I(F')) = \top$. Hence, $I(A) = I(\mathit{Body} \vee F') = I(F)$ and, therefore, $I(A \leftrightarrow F) = \top$.

2. If $[A] \in I^\bot$ then there is a rule $A \leftarrow \mathit{Body} \in g\mathcal{P}$ and for all rules $A' \leftarrow \mathit{Body} \in g\mathcal{P}$ with $[A'] = [A]$ we have $I(\mathit{Body}) = \bot = I(A)$. Hence, for each equivalence $A' \leftrightarrow F \in wc\mathcal{P}$ with $[A'] = [A]$ we have $I(F) = \bot$ and, therefore, $I(A' \leftrightarrow F) = \top$.

3. If $[A] \notin I^\top \cup I^\bot$ then there are two possibilities:

   (a) There is no rule $A' \leftarrow \mathit{Body} \in g\mathcal{P}$ with $[A'] = [A]$ and, therefore, there is no equivalence $A' \leftrightarrow F \in wc\mathcal{P}$.

   (b) There are rules $A'_i \leftarrow \mathit{Body}_i \in g\mathcal{P}$ for $i \in \{1, \ldots, n\}$ with $[A'_i] = [A]$ and $I(A'_i) = \cup$, but neither $I(\mathit{Body}_i) = \bot$ for all $i \in \{1, \ldots, n\}$ nor there is an $i \in \{1, \ldots, n\}$ such that $I(\mathit{Body}_i) = \top$. Hence,

   $$I(\bigvee_{i \in \{1,\ldots,n\}} \mathit{Body}_i) = \max_t(\{I(\mathit{Body}_i) \mid i \in \{1, \ldots, n\}\}) = \cup$$

   and, therefore, $I(A' \leftrightarrow \bigvee_{i\in\{1,\ldots,n\}} \mathit{Body}_i) = \top$.

Hence, for each equivalence of the form $A' \leftrightarrow F$ occurring in $wc\mathcal{P}$ with $[A'] = [A]$ we have $I(A' \leftrightarrow F) = \top$ and, therefore, I is a model of $wc\mathcal{P}$. □

Proposition 22 *If J is the least fixed point of $\Phi_{\mathcal{P}}$ then J is a minimal model of $wc\mathcal{P}$.*

Proof By Lemma 21, the least fixed point J of $\Phi_{\mathcal{P}}$ is a model of $wc\mathcal{P}$. By Lemma 20, for every model I of $wc\mathcal{P}$ we have $J^{\top} \subseteq I^{\top}$ and $J^{\bot} \subseteq I^{\bot}$, i.e. $J \subseteq I$. Hence, J is a minimal model of $wc\mathcal{P}$. □

Proposition 23 *If I is a minimal model of $wc\mathcal{P}$ then I is the least fixed point of $\Phi_{\mathcal{P}}$.*

Proof Let I be a minimal model of $wc\mathcal{P}$ and J the least fixed point of $\Phi_{\mathcal{P}}$. From Lemma 20 we learn that $J^{\top} \subseteq I^{\top}$ and $J^{\bot} \subseteq I^{\bot}$. From Proposition 22 we learn that J is a minimal model of $wc\mathcal{P}$. But then $I = J$ because, otherwise, we have a conflict with the minimality of I. □

Proof of Theorem 13. The claim that $wc\mathcal{P}$ has a least model follows from Propositions 22 and 23 and the fact that the least fixed point of $\Phi_{\mathcal{P}}$ is unique. □

Theorem 24 *I is the least fixed point of $\Phi_{\mathcal{P}}$ if and only if I is the least model of $wc\mathcal{P}$.*

Proof The claim follows immediately from Propositions 22 and 23 and Theorem 13. □

Table 3.1 shows the computation of the least fixed point for the first six examples of the suppression task.

Let $\mathcal{P}$ be a program and $\mathcal{E}$ an equational theory. Furthermore, let $\mathcal{M}_{\mathcal{P}}$ denote the least fixed point of $\Phi_{\mathcal{P}}$. By the previous theorem, it is equal to the least model of the weak completion of $\mathcal{P}$. Remember that $\mathcal{M}_{\mathcal{P}}$ satisfies $\mathcal{E}$ as well. A formula F *follows from $\mathcal{P}$ under the Weak Completion Semantics*, in symbols $\mathcal{P} \models_{wcs} F$, if and only if $\mathcal{M}_{\mathcal{P}}$ maps F to true.

In order to compute logical consequences of the weak completion of a program we need to construct its least model. If the semantic operator is continuous, then this can be done as follows. Starting with the empty interpretation, the semantic operator is iteratively applied until a fixed point is reached:

$$\begin{array}{rcll} \Phi_{\mathcal{P}}\uparrow 0 & = & \langle \emptyset, \emptyset \rangle, & \\ \Phi_{\mathcal{P}}\uparrow (i+1) & = & \Phi_{\mathcal{P}}(\Phi_{\mathcal{P}}\uparrow i) & \text{for } i \geq 0. \end{array}$$

But several questions remain: What shall we do in case the semantic operator cannot be shown to be continuous? More precisely, what shall we do in case the preconditions of the Propositions 18 or 19 are not met?

Ex.	$\mathcal{P}$	$wc\mathcal{P}$	$\Phi_{\mathcal{P}}$	$I^{\top}$	$I^{\bot}$
1	$e \leftarrow \top$	$e \leftrightarrow \top$	1	e	ab_1
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow e \wedge \neg ab_1$	2	ℓ	
	$ab_1 \leftarrow \bot$	$ab_1 \leftrightarrow \bot$			
2	$e \leftarrow \top$	$e \leftrightarrow \top$	1	e	ab_1
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow (e \wedge \neg ab_1) \vee (t \wedge \neg ab_2)$			ab_2
	$\ell \leftarrow t \wedge \neg ab_2$	$ab_1 \leftrightarrow \bot$	2	ℓ	
	$ab_1 \leftarrow \bot$	$ab_2 \leftrightarrow \bot$			
	$ab_2 \leftarrow \bot$				
3	$e \leftarrow \top$	$e \leftrightarrow \top$	1	e	
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow (e \wedge \neg ab_1) \vee (o \wedge \neg ab_3)$	2		ab_3
	$\ell \leftarrow o \wedge \neg ab_3$	$ab_1 \leftrightarrow \bot \vee \neg o$			
	$ab_1 \leftarrow \bot$	$ab_3 \leftrightarrow \bot \vee \neg e$			
	$ab_3 \leftarrow \bot$				
	$ab_1 \leftarrow \neg o$				
	$ab_3 \leftarrow \neg e$				
4	$e \leftarrow \bot$	$e \leftrightarrow \bot$	1		e
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow e \wedge \neg ab_1$			ab_1
	$ab_1 \leftarrow \bot$	$ab_1 \leftrightarrow \bot$	2		ℓ
5	$e \leftarrow \bot$	$e \leftrightarrow \bot$	1		e
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow (e \wedge \neg ab_1) \vee (t \wedge \neg ab_2)$			ab_1
	$\ell \leftarrow t \wedge \neg ab_2$	$ab_1 \leftrightarrow \bot$			ab_2
	$ab_1 \leftarrow \bot$	$ab_2 \leftrightarrow \bot$			
	$ab_2 \leftarrow \bot$				
6	$e \leftarrow \bot$	$e \leftrightarrow \bot$	1		e
	$\ell \leftarrow e \wedge \neg ab_1$	$\ell \leftrightarrow (e \wedge \neg ab_1) \vee (o \wedge \neg ab_3)$	2	ab_3	
	$\ell \leftarrow o \wedge \neg ab_3$	$ab_1 \leftrightarrow \bot \vee \neg o$	3		ℓ
	$ab_1 \leftarrow \bot$	$ab_3 \leftrightarrow \bot \vee \neg e$			
	$ab_3 \leftarrow \bot$				
	$ab_1 \leftarrow \neg o$				
	$ab_3 \leftarrow \neg e$				

Table 3.2: Examples for the computation of the least model of the weak completions of programs. All iterations of $\Phi_{\mathcal{P}}$ start with the empty interpretation $\langle \emptyset, \emptyset \rangle$. In the last two columns the atoms added to $I^{\top}$ and $I^{\bot}$ in the ith iteration are depicted, where $i = 1, 2, 3$. The iteration stops as soon as no new atoms are added, in which case the least fixed point is reached.

Do we always have to start the construction of the least fixed point with the empty interpretation? This requirement seems to be rather strong in case of human reasoning episodes. As an example consider the program $\mathcal{P}$ consisting of the following rules:

$$\begin{array}{lcl} p & \leftarrow & q, \\ q & \leftarrow & p, \end{array}$$

where p and q may denote that *it is cold* and *Lucy is wearing a sweater*, respectively. The empty interpretation is the least fixed point of the weak completion of this program and we obtain:

$$\Phi_{\mathcal{P}} \uparrow 1 = \Phi_{\mathcal{P}} \uparrow 0 = \langle \emptyset, \emptyset \rangle.$$

However, if we start the iteration with the interpretation

$$\langle \{p\}, \emptyset \rangle$$

then

$$\Phi_{\mathcal{P}}(\langle \{p\}, \emptyset \rangle) = \langle \{q\}, \emptyset \rangle$$

and

$$\Phi_{\mathcal{P}}(\langle \{q\}, \emptyset \rangle) = \langle \{p\}, \emptyset \rangle.$$

Although $\Phi_{\mathcal{P}}$ has a fixed point, we will never reach it this way.

3.4 Semantic Operators as Contraction Mappings

We may not only use Kleene's Fixed Point Theorem 4 to compute the least fixed point of our semantic operator, but Banach's Contraction Mapping Theorem 9. This does not require the semantic operator to be continuous, but it must be a contraction mapping on a complete metric space instead. But what kind of metric space shall we consider and under which condition will the semantic operator be a contraction mapping? In the context of logic programs these questions were first considered by Melvin Fitting in [22]. They were extended to the three-valued Łukasiewicz logic by Carroline D. P. Kencana Ramli in [40, 33]. In this book, they will be further extended to allow for equational theories.

A *level mapping* for a program $\mathcal{P}$ is a function *lvl* from the set of ground atoms to $\mathbb{N}$ such that $\mathit{lvl}(A) = \mathit{lvl}(B)$ for each ground atom B with $[A] = [B]$. In other words, if two ground atoms A and B are in the same equivalence class with respect to an equational theory $\mathcal{E}$ then they must have the same level. Each level mapping is extended to a mapping from the set of ground literals to $\mathbb{N}$ by $\mathit{lvl}(\neg A) = \mathit{lvl}(A)$ for each ground atom A.

Let *lvl* be a level mapping for the program $\mathcal{P}$. $\mathcal{P}$ is *acyclic with respect to lvl* if and only if for every rule $A \leftarrow L_1 \wedge \ldots \wedge L_n$ occurring in $g\mathcal{P}$ we find $\mathit{lvl}(A) > \mathit{lvl}(L_i)$ for all $1 \leq i \leq n$. $\mathcal{P}$ is *acyclic* if and only if $\mathcal{P}$ is acyclic with respect to some level mapping.

Although problem to determine whether a given program is acyclic is undecidable [3], we may try to find appropriate level mappings.

As an example consider the progam $\mathcal{P}$ with clauses

$$\begin{array}{lcl} p & \leftarrow & r \wedge q, \\ q & \leftarrow & r \wedge p. \end{array}$$

This program is not acyclic as by the first clause $lvl(p) > lvl(q)$ and by the second clause $lvl(q) > lvl(p)$. In fact, $\Phi_{\mathcal{P}}$ has two fixed points, viz. the empty interpretation $\langle \emptyset, \emptyset \rangle$ and $\langle \emptyset, \{p, q\} \rangle$. By Banach Contraction Mapping Theorem 9, $\Phi_{\mathcal{P}}$ cannot be a contraction.

As another example consider the program $\mathcal{P}$ with clauses

$$\begin{array}{lcl} p & \leftarrow & q \wedge r, \\ q & \leftarrow & \neg r, \\ r & \leftarrow & \top. \end{array} \tag{3.22}$$

This program is acyclic with respect to the following level mapping:

A	$lvl(A)$
r	0
q	1
p	2

Suppose we iterate the semantic operator $\Phi_{(4.38)}$ starting with the interpretation $\langle \{q, r\}, \{p\} \rangle$:

$\Phi_{(4.38)}$	$I^{\top}$	$I^{\perp}$
$\uparrow 1$	p r	q
$\uparrow 2$	r	q p

where $\Phi_{\mathcal{P}} \uparrow 1$ is the interpretation obtained by applying $\Phi_{\mathcal{P}}$ to the given interpretation and

$$\Phi_{\mathcal{P}} \uparrow (i+1) = \Phi_{\mathcal{P}}(\Phi_{\mathcal{P}} \uparrow i)$$

for all $i > 0$. One should observe that $\Phi_{(4.38)} \uparrow 3 = \Phi_{(4.38)} \uparrow 2$.

Likewise, if we start with the interpretation $\langle \{p\}, \emptyset \rangle$ we obtain:

$\Phi_{(4.38)}$	$I^{\top}$	$I^{\perp}$
$\uparrow 1$	r	
$\uparrow 2$	r	q
$\uparrow 3$	r	q p

In each case the iteration terminates with the interpretation

$$\langle \{r\}, \{q, p\} \rangle$$

which is the least model of the weak completion of the given program. In the sequel, we will show that this is not a coincidence, but holds in general for acylic programs.

Proposition 25 *Let $\mathcal{P}$ be a program, $\mathcal{E}$ an equational theory, lvl a level mapping for $\mathcal{P}$, $\mathcal{I}$ the set of interpretations for $\mathcal{P}$, and $I,\ J \in \mathcal{I}$. The function $d_{lvl} : \mathcal{I} \times \mathcal{I} \to \mathbb{R}$ defined as*

$$d_{lvl}(I,J) = \begin{cases} \frac{1}{2^n} & I \neq J \textit{ and} \\ & I(A) = J(A) \neq \mathsf{U} \textit{ for all } A \textit{ with } lvl(A) < n \textit{ and} \\ & I(A) \neq J(A) \textit{ or } I(A) = I(J) = \mathsf{U} \textit{ for some } A \textit{ with } lvl(A) = n, \\ 0 & \textit{otherwise,} \end{cases}$$

is a metric.

Proof We have to show that $d_{lvl}(I,J) = 0$ if and only if $I = J$, d_{lvl} is symmetric, and the triangle inequality holds. Let $I,\ J$, and K be interpretations.

1. $d_{lvl}(I,J) = 0$ if and only if $I = J$: If $d_{lvl}(I,J) = 0$ then I and J coincide on all ground atoms and, hence, are equal. Conversely, if I and J are equal then for all ground atoms A we find $I(A) = J(A)$ and, by definition, $d_{lvl}(I,J) = 0$.

2. $d_{lvl}(I,J) = d_{lvl}(J,I)$: This follows immediately as the definition of d_{lvl} is symmetric.

3. $d_{lvl}(I,J) \leq d_{lvl}(I,K) + d_{lvl}(K,J)$: If $d_{lvl}(I,K) = 0$ then I and K are equal and can be interchanged in the definition of the metric d_{lvl}. We obtain
$$d_{lvl}(I,J) = d_{lvl}(K,J) = 0 + d_{lvl}(K,J) = d_{lvl}(I,K) + d_{lvl}(K,J).$$
Likewise, if $d_{lvl}(K,J) = 0$ we obtain
$$d_{lvl}(I,J) = d_{lvl}(I,K) = d_{lvl}(I,K) = 0 = d_{lvl}(I,K) + d_{lvl}(K,J).$$
It remains to consider the case where $d_{lvl}(I,K) \neq 0$ and $d_{lvl}(K,J) \neq 0$. Then, we find $m,\ k \in \mathbb{N}$ such that $d_{lvl}(I,K) = \frac{1}{2^m}$ and $d_{lvl}(K,J) = \frac{1}{2^k}$. Without loss of generality we may assume that $m \leq k$. We will show that $d_{lvl}(I,J) \leq \frac{1}{2^m}$. Consider some atom A with $lvl(A) < m$. Then, we have $I(A) = K(A) \neq \mathsf{U}$ and $K(A) = J(A) \neq \mathsf{U}$. Consequently, $I(A) = J(A) \neq \mathsf{U}$ and we are done. □

As an example consider the program $\mathcal{P}$ with clauses

$$\begin{array}{rcl} even(0) & \leftarrow & \top, \\ even(s(X)) & \leftarrow & \neg even(X). \end{array}$$

Let

$$\begin{array}{rcl} I & = & \langle \{even(s^k(0)) \mid k \text{ is even}\}, \{even(s^k(0)) \mid k \text{ is odd}\}\rangle, \\ J & = & \langle \{even(s^k(0)) \mid k \text{ is even}\}, \emptyset\rangle, \end{array}$$

and

$$lvl(even(s^k(0))) = k.$$

Then, $g\mathcal{P}$ is infinite, $\mathcal{P}$ is acyclic, and

$$d_{lvl}(I,J) = \frac{1}{2}.$$

Proposition 26 *Let $\mathcal{P}$ be a program, $\mathcal{E}$ and equational theory, lvl a level mapping for $\mathcal{P}$, and $\mathcal{I}$ the set of interpretations for $\mathcal{P}$. Then, $(\mathcal{I}, d_{lvl})$ is a complete metric space.*

Proof We have to show that every Cauchy sequence of interpretations converges. Suppose $(I_k \mid k \geq 1)$ is a Cauchy sequence of interpretations. Then, for every $n \in \mathbb{N}$ there is an $K \in \mathbb{N}$ such that for alle $k_1,\ k_2 \geq K$ we find

$$d_{lvl}(I_{k_1}, I_{k_2}) \leq \frac{1}{2^{n+1}}.$$

Let K_n be the least such K for every $n \in \mathbb{N}$. Hence,

$$K_{n_1} \leq K_{n_2} \text{ for any } n_1,\ n_2 \in \mathbb{N} \text{ with } n_1 \leq n_2.$$

Let the limit interpretation I be defined such that for any atom A we have $I(A) = I_{K_\ell}(A)$, where $\ell = lvl(A)$.

We have to prove that for every $\epsilon > 0$ there is some $K \in \mathbb{N}$ such that for any $k \geq K$ we have $d_{lvl}(I, I_k) \leq \epsilon$. We choose $\epsilon > 0$ and let $n \in \mathbb{N}$ be such that $\frac{1}{2^{n+1}} \leq \epsilon$. We will prove $d_{lvl}(I, I_k) \leq \frac{1}{2^{n+1}}$ for any $k \geq K_n$ from which the claim will follow.

Consider an atom A with $lvl(A) = \ell \leq n$. Then, $K_\ell \leq K_n$ and, hence, by the definition of K_ℓ we have $d_{lvl}(I_{K_\ell}, I_{K_n}) \leq \frac{1}{2^{\ell+1}}$. Consequently, by the definition of d_{lvl} we have

$$I(A) = I_{K_\ell}(A) = I_{K_n}(A).$$

Furthermore, for any $k \geq K_n$ we have $d_{lvl}(I_{K_n}, I_k) \leq \frac{1}{2^{n+1}}$. So we obtain

$$I(A) = I_{K_n}(A) = I_k(A)$$

and, therefore, also

$$d_{lvl}(I, I_k) \leq \frac{1}{2^{n+1}}$$

which completes the proof. □

Theorem 27 *Let $\mathcal{P}$ be a program, $\mathcal{E}$ and equational theory, lvl a level mapping for $\mathcal{P}$, and $\mathcal{I}$ the set of interpretations for $\mathcal{P}$. If $\mathcal{P}$ is acyclic with respect to lvl then $\Phi_{\mathcal{P}}$ is a contraction on the metric space $(\mathcal{I}, d_{lvl})$.*

We will prove the more general Theorem 30 in Subsection 4.5.2.

Corollary 28 *If a program $\mathcal{P}$ is acyclic then $\Phi_{\mathcal{P}}$ has a unique fixed point which can be computed by iterating $\Phi_{\mathcal{P}}$ up to ω times starting with any interpretation.*

Proof The result follows from Theorems 27 and 9. □

One should observe that Theorem 24 applies here. The unique fixed point of the semantic operator $\Phi_{\mathcal{P}}$ is the least fixed point of $\Phi_{\mathcal{P}}$ and the least model of $wc\mathcal{P}$.

$\mathcal{P}$	$\mathcal{A}_{\mathcal{P}}$
$\ell \leftarrow e \wedge \neg ab_1$	$e \leftarrow \top$
$ab_1 \leftarrow \bot$	$e \leftarrow \bot$
	$ab_1 \leftarrow \top$
$\ell \leftarrow e \wedge \neg ab_1$	$e \leftarrow \top$
$\ell \leftarrow t \wedge \neg ab_2$	$e \leftarrow \bot$
$ab_1 \leftarrow \bot$	$t \leftarrow \top$
$ab_2 \leftarrow \bot$	$t \leftarrow \bot$
	$ab_1 \leftarrow \top$
	$ab_2 \leftarrow \top$
$\ell \leftarrow e \wedge \neg ab_1$	$e \leftarrow \top$
$\ell \leftarrow o \wedge \neg ab_3$	$e \leftarrow \bot$
$ab_1 \leftarrow \bot$	$o \leftarrow \top$
$ab_3 \leftarrow \bot$	$o \leftarrow \bot$
$ab_1 \leftarrow \neg o$	
$ab_3 \leftarrow \neg e$	

Table 3.3: Examples for programs and their sets of abducibles. In each of these cases the equational theory is empty.

3.5 Abduction

An *integrity constraint* is an expression of the form

$$\mathsf{U} \leftarrow L_1 \wedge \ldots \wedge L_n,$$

where each L_i, $1 \leq i \leq n$, is a literal. In the sequel, $\mathcal{IC}$ denotes a finite set of integrity constraints.

Let I be an interpretation. I *satisfies* $\mathcal{IC}$ if and only if for each $\mathsf{U} \leftarrow \mathit{Body}$ occurring in $\mathcal{IC}$ we find $I(\mathit{Body}) \in \{\mathsf{U}, \bot\}$.

Let $\mathcal{P}$ be a ground program. The *set of abducibles* is

$$\begin{array}{rcl} \mathcal{A}_{\mathcal{P}} & = & \{A \leftarrow \top \mid A \text{ is defined in } \mathcal{P}\} \\ & \cup & \{A \leftarrow \bot \mid A \text{ is defined in } \mathcal{P}\} \\ & \cup & \{A \leftarrow \top \mid \neg A \text{ is assumed in } \mathcal{P}\}. \end{array}$$

Hence, all undefined ground atoms are abducibles as positive ground facts or negative assumptions. Moreover, if $\neg A$ is assumed then the positive fact $A \leftarrow \top$ is also an abducible, which can be abduced to defeat the negative assumption. In Table 3.3 the programs for the experiments 7-12 of the suppression task are shown together with their sets of abducibles.

An *abductive framework* $\langle \mathcal{P}, \mathcal{A}, \mathcal{IC}, \models_{wcs} \rangle$ consists of

1. a program $\mathcal{P}$,
2. a set of abducibles $\mathcal{A} \subseteq \mathcal{A}_{\mathcal{P}}$,
3. a set $\mathcal{IC}$ of integrity constraints,
4. the entailment relation $\models_{wcs}$.

In the suppression task discussed in Chapter 1 we were using the abductive framework

$$\langle \mathcal{P}, \{A \leftarrow \top \mid A \text{ is defined in } \mathcal{P}\} \cup \{A \leftarrow \bot \mid A \text{ is defined in } \mathcal{P}\}, \emptyset, \models_{wcs} \rangle. \tag{3.23}$$

An *observation* $\mathcal{O}$ is a set of ground literals.

In the experiment $7 - 9$ of the suppression task $\mathcal{O} = \{\ell\}$, i.e. we are observing that *she is studying late in the library*, whereas in the experiments $10 - 12$ of the suppression task $\mathcal{O} = \{\neg \ell\}$.

An observation $\mathcal{O}$ is *explainable* in the abductive framework $\langle \mathcal{P}, \mathcal{A}, \mathcal{IC}, \models_{wcs} \rangle$ if and only if there exist $\mathcal{X} \subseteq \mathcal{A}$ called *explanation* such that

1. $\mathcal{M}_{\mathcal{P} \cup \mathcal{X}} \models_{wcs} L$ for all $L \in \mathcal{O}$ and
2. $\mathcal{M}_{\mathcal{P} \cup \mathcal{X}}$ satisfies $\mathcal{IC}$.

Sometimes explanations are required to be minimal, where an explanation is *minimal* if it cannot be subsumed by another explanation.

Under the *Weak Completion Semantics* each program $\mathcal{P}$ has a least model (Theorem 12). For each explanation $\mathcal{X}$, $\mathcal{P} \cup \mathcal{X}$ is a program and, hence, $\mathcal{P} \cup \mathcal{X}$ has a least model.

Table 3.4 shows the programs, abducibles, observations, and minimal explanations used in experiments $7 - 12$ of the suppression task.

Let $\langle \mathcal{P}, \mathcal{A}, \mathcal{IC}, \models_{wcs} \rangle$ be an abductive framework, $\mathcal{O}$ an observation, and F a formula.

- *F follows credulously* from $\mathcal{P}$ and $\mathcal{O}$ if and only if there exists an explanation $\mathcal{X}$ for $\mathcal{O}$ such that $\mathcal{P} \cup \mathcal{X} \models_{wcs} F$.
- *F follows skeptically* from $\mathcal{P}$ and $\mathcal{O}$ if and only if for all explanations $\mathcal{X}$ for $\mathcal{O}$ we find $\mathcal{P} \cup \mathcal{X} \models_{wcs} F$.

Returning to the experiments presented in Table 3.4 we find:

- In experiments 7 and 9, e follows credulously as well as skeptically from the program and the obervation.

$Ex.$	$\mathcal{P}$	$\mathcal{A}$	$\mathcal{O}$	$\mathcal{X}$	
7	$\ell \leftarrow e \wedge \neg ab_1$	$e \leftarrow \top$	ℓ	$e \leftarrow \top$	
	$ab_1 \leftarrow \bot$	$e \leftarrow \bot$			
8	$\ell \leftarrow e \wedge \neg ab_1$	$e \leftarrow \top$	ℓ	$e \leftarrow \top$	$t \leftarrow \top$
	$\ell \leftarrow t \wedge \neg ab_2$	$e \leftarrow \bot$			
	$ab_1 \leftarrow \bot$	$t \leftarrow \top$			
	$ab_2 \leftarrow \bot$	$t \leftarrow \bot$			
9	$\ell \leftarrow e \wedge \neg ab_1$	$e \leftarrow \top$	ℓ	$e \leftarrow \top$	
	$\ell \leftarrow o \wedge \neg ab_3$	$e \leftarrow \bot$		$o \leftarrow \top$	
	$ab_1 \leftarrow \bot$	$o \leftarrow \top$			
	$ab_3 \leftarrow \bot$	$o \leftarrow \bot$			
	$ab_1 \leftarrow \neg o$				
	$ab_3 \leftarrow \neg e$				
10	$\ell \leftarrow e \wedge \neg ab_1$	$e \leftarrow \top$	$\neg\ell$	$e \leftarrow \bot$	
	$ab_1 \leftarrow \bot$	$e \leftarrow \bot$			
11	$\ell \leftarrow e \wedge \neg ab_1$	$e \leftarrow \top$	$\neg\ell$	$e \leftarrow \bot$	
	$\ell \leftarrow t \wedge \neg ab_2$	$e \leftarrow \bot$		$t \leftarrow \bot$	
	$ab_1 \leftarrow \bot$	$t \leftarrow \top$			
	$ab_2 \leftarrow \bot$	$t \leftarrow \bot$			
12	$\ell \leftarrow e \wedge \neg ab_1$	$e \leftarrow \top$	$\neg\ell$	$e \leftarrow \bot$	$o \leftarrow \bot$
	$\ell \leftarrow o \wedge \neg ab_3$	$e \leftarrow \bot$			
	$ab_1 \leftarrow \bot$	$o \leftarrow \top$			
	$ab_3 \leftarrow \bot$	$o \leftarrow \bot$			
	$ab_1 \leftarrow \neg o$				
	$ab_3 \leftarrow \neg e$				

Table 3.4: The programs, abducibles, observations, and minimal explanations used in experiments 7 – 12 of the suppression task. One should observe that only undefined atoms can be abduced in the abductive framework (3.23).

- In experiment 8, both, e and t, follow credulously, but neither e, t, $\neg e$ nor $\neg t$ follows skeptically. However, $e \vee t$ follows skeptically.
- In experiments 10 and 11, $\neg e$ follows skeptically because the program and the explanation are weakly completed whereby $e \leftarrow \bot$ becomes $e \leftrightarrow \bot$.
- In experiment 12, both, $\neg e$ and $\neg o$, follow credulously, but neither e, t, $\neg e$ nor $\neg t$ follow skeptically. However, $\neg e \vee \neg o$ follows skeptically.

Comparing to the answers provided by the participants in the experiments of the suppression task, experiments 8 and 12 show that humans seem to reason skeptically.

Chapter 4

Applications and Extensions

where we discuss various applications like the treatment of conditionals, the selection task, syllogistic reasoning, and contextual abduction. It will become necessary to extend the basic theory, but all extensions wil be conservative.

4.1 Conditionals

Conditionals are statements of the form

if antecedent then consequence.

Conditionals are categorized in many different ways (see e.g. [51] for an overview). For the time being we will just consider indicative and subjunctive conditionals.

Indicative conditionals are conditionals, whose antecedent may or may not be true, whose consequence may or may not be true, but the consequence is asserted to be true if the antecedent is true.

Subjunctive conditionals or *counterfactuals* are conditionals whose antecedent is false, whose consequence may or may not be true, but in the counterfactual circumstance of the antecedent being true, the consequence is asserted to be true.

In the sequel, let

$$\mathcal{C} \Rightarrow \mathcal{D}$$

be a conditional, where the *antecedent* $\mathcal{C}$ and the *consequent* $\mathcal{D}$ are finite and consistent sets of ground literals and a set of literals is *consistent* if and only if it does not contain an atom and its negation.

Conditionals will be evaluated with respect to some background knowledge, which consists of a program $\mathcal{P}$, the empty equational theory $\mathcal{E}$, and a set $\mathcal{IC}$ of integrity constraints. We will assume that $\mathcal{P}$ is either a finite propositional or a finite datalog program. Hence, by Propositions 18 or 19, respectively, the semantic operator $\Phi_{\mathcal{P}}$ is continuous and, consequently, by Theorem 4 its least fixed point can be computed by iterating $\Phi_{\mathcal{P}}$ starting with the empty interpretation. As usual $\mathcal{M}_{\mathcal{P}}$ denotes the least fixed point of $\Phi_{\mathcal{P}}$. Moreover, we assume that $\mathcal{M}_{\mathcal{P}}$ evaluates the integrity constraints $\mathcal{IC}$ to true.

This section is based on results first published in [14, 15].

4.1.1 Revision

Given some background knowledge $\mathcal{P}$ and $\mathcal{IC}$, a conditional $\mathcal{C} \Rightarrow \mathcal{D}$ is a counterfactual if $\mathcal{M}_{\mathcal{P}}$ maps $\mathcal{C}$ to false. This holds if there is at least one literal occurring in $\mathcal{C}$ such that this literal is mapped to false by $\mathcal{M}_{\mathcal{P}}$. In the general case, each element of a subset $\mathcal{S}$ of $\mathcal{C}$ may be mapped to false by $\mathcal{M}_{\mathcal{P}}$.

Hence, in order to evaluate the counterfactual, we must revise the background knowlege with respect to a consistent set $\mathcal{S}$ of ground literals, where we may assume that each literal occurring in $\mathcal{S}$ is mapped to false by $\mathcal{M}_{\mathcal{P}}$. Let $\mathcal{P}$ be a ground program and

$$\mathit{clauses}(\mathcal{S}, \mathcal{P}) = \{A \leftarrow \mathit{Body} \in \mathcal{P} \mid A \in \mathcal{S} \vee \neg A \in \mathcal{S}\},$$

i.e. $\mathit{clauses}(\mathcal{S}, \mathcal{P})$ consists of the definitions for all atoms in $\mathcal{P}$ which occur positively or negatively in $\mathcal{S}$.

The *revision of $\mathcal{P}$ with respect to $\mathcal{S}$* is defined as

$$\mathit{rev}(\mathcal{P}, \mathcal{S}) = (\mathcal{P} \setminus \mathit{clauses}(\mathcal{S}, \mathcal{P})) \cup \{A \leftarrow \top \mid A \in \mathcal{S}\} \cup \{A \leftarrow \bot \mid \neg A \in \mathcal{S}\}.$$

This definition is very straightforward. The clauses which are responsible for mapping $\mathcal{S}$ to false are removed, facts and assumptions which will map $\mathcal{S}$ to true are added.

Proposition 29

1. *rev is non-monotonic.*
2. *If $\mathcal{M}_{\mathcal{P}}(L) = \mathsf{U}$ for all $L \in \mathcal{S}$ then rev is monotonic, i.e. $\mathcal{M}_{\mathcal{P}} \subseteq \mathcal{M}_{\mathit{rev}(\mathcal{P},\mathcal{S})}$.*
3. *$\mathcal{M}_{\mathit{rev}(\mathcal{P},\mathcal{S})}(\mathcal{S}) = \top$.*

Proof

1. We have to find $\mathcal{P}$, $\mathcal{S}$, and F such that $\mathcal{P} \models_{wcs} F$ and $\mathit{rev}(\mathcal{P}, \mathcal{S}) \not\models_{wcs} F$. Let $\mathcal{P} = \{a \leftarrow \top\}$, $\mathcal{S} = \{\neg a\}$ and $F = a$. We find $\mathcal{M}_{\mathcal{P}} = \langle \{a\}, \emptyset \rangle$, $\mathit{rev}(\mathcal{P}, \mathcal{S}) = \{a \leftarrow \bot\}$, $\mathcal{M}_{\mathit{rev}(\mathcal{P},\mathcal{S})} = \langle \emptyset, \{a\} \rangle$, $\mathcal{P} \models_{wcs} a$, and $\mathit{rev}(\mathcal{P}, \mathcal{S}) \not\models_{wcs} a$.
2. $\mathcal{M}_{\mathcal{P}}$ and $\mathcal{M}_{\mathit{rev}(\mathcal{P},\mathcal{S})}$ can be computed by iterating $\Phi_{\mathcal{P}}$ and $\Phi_{\mathit{rev}(\mathcal{P},\mathcal{S})}$, respectively. By induction on n we can show that for all $n \in \mathbb{N}$ the relationship $\Phi_{\mathcal{P}} \uparrow n \subseteq \Phi_{\mathit{rev}(\mathcal{P},\mathcal{S})} \uparrow n$ holds. In case $n = 0$ we find

$$\Phi_{\mathcal{P}} \uparrow 0 = \langle \emptyset, \emptyset \rangle = \Phi_{\mathit{rev}(\mathcal{P},\mathcal{S})} \uparrow 0.$$

We assume that the result holds for n and turn to the induction step:

$$\Phi_{\mathcal{P}}\!\uparrow\!(n+1) = \Phi_{\mathcal{P}}(\Phi_{\mathcal{P}}\!\uparrow\! n) = \langle I^{\top}, I^{\bot}\rangle, \tag{4.1}$$

where

$$\begin{array}{rcl} I^{\top} & = & \{A \mid \text{there exists } A \leftarrow \mathit{Body} \in g\mathcal{P} \text{ and } \Phi_{\mathcal{P}}\!\uparrow\! n(\mathit{Body}) = \top\} \\ I^{\bot} & = & \{A \mid \text{there exists } A \leftarrow \mathit{Body} \in g\mathcal{P} \\ & & \qquad \text{and for all } A \leftarrow \mathit{Body} \in g\mathcal{P} \text{ we find } \Phi_{\mathcal{P}}\!\uparrow\! n(\mathit{Body}) = \bot\} \end{array}$$

As $\mathcal{M}_{\mathcal{P}}(L) = \mathsf{U}$ for all $L \in \mathcal{S}$, we find that $\mathit{atom}\, L$ is neither in $I^{\top}$ nor in $I^{\bot}$, where $\mathit{atom}\, L = L$ if L is an atom and $\mathit{atom}\, L = A$ if $L = \neg A$. By the definition of revision, however, $\mathit{atom}\, L$ is either in $J^{\top}$ or in $J^{\bot}$, where

$$\begin{array}{rcl} J^{\top} & = & \{A \mid \text{there exists } A \leftarrow \mathit{Body} \in g\, \mathit{rev}(\mathcal{P},\mathcal{S}) \text{ and } \Phi_{\mathit{rev}(\mathcal{P},\mathcal{S})}\!\uparrow\! n(\mathit{Body}) = \top\} \\ J^{\bot} & = & \{A \mid \text{there exists } A \leftarrow \mathit{Body} \in g\, \mathit{rev}(\mathcal{P},\mathcal{S}) \\ & & \qquad \text{and for all } A \leftarrow \mathit{Body} \in g\, \mathit{rev}(\mathcal{P},\mathcal{S}) \text{ we find } \Phi_{\mathit{rev}(\mathcal{P},\mathcal{S})}\!\uparrow\! n(\mathit{Body}) = \bot\} \end{array}$$

As $\mathcal{P}$ and $\mathit{rev}(\mathcal{P},\mathcal{S})$ contain identical definitions for atoms not occurring in $\mathcal{S}$ we conclude by the induction hypothesis that $I^{\top} \subseteq J^{\top}$, $I^{\bot} \subseteq J^{\bot}$ and

$$\langle I^{\top}, I^{\bot}\rangle \subseteq \langle J^{\top}, J^{\bot}\rangle = \Phi_{\mathit{rev}(\mathcal{P},\mathcal{S})}(\Phi_{\mathit{rev}(\mathcal{P},\mathcal{S})}\!\uparrow\! n) = \Phi_{\mathit{rev}(\mathcal{P},\mathcal{S})}\!\uparrow\!(n+1). \tag{4.2}$$

The result follows by combining (4.1) and (4.2) and the induction theorem.

3. Follows immediately from the definition of revision and the monotonicity of $\Phi_{\mathcal{P}}$. □

4.1.2 Minimal Revision Followed by Abduction

We suggest to evaluate conditionals $\mathcal{C} \Rightarrow \mathcal{D}$ with respect to the background knowledge $\mathcal{P}$, $\mathcal{E}$ and $\mathcal{IC}$ as follows:

$$\begin{array}{ll} \text{If } \mathcal{M}_{\mathcal{P}}(\mathcal{C}) = \top & \text{then the value of } \mathcal{C} \Rightarrow \mathcal{D} \text{ is } \mathcal{M}_{\mathcal{P}}(\mathcal{D}). \\ \text{If } \mathcal{M}_{\mathcal{P}}(\mathcal{C}) = \bot & \text{then evaluate } \mathcal{C} \Rightarrow \mathcal{D} \text{ with respect to } \mathcal{M}_{\mathit{rev}(\mathcal{P},\mathcal{S})}, \text{ where} \\ & \quad \mathcal{S} = \{L \in \mathcal{C} \mid \mathcal{M}_{\mathcal{P}}(L) = \bot\}. \\ \text{If } \mathcal{M}_{\mathcal{P}}(\mathcal{C}) = \mathsf{U} & \text{then evaluate } \mathcal{C} \Rightarrow \mathcal{D} \text{ with respect to } \mathcal{M}_{\mathcal{P}'}, \text{ where} \\ & \quad \mathcal{P}' = \mathit{rev}(\mathcal{P},\mathcal{S}) \cup \mathcal{X}, \\ & \quad \mathcal{S} \text{ is a smallest subset of } \mathcal{C}, \\ & \quad \mathcal{X} \subseteq \mathcal{A}_{\mathit{rev}(\mathcal{P},\mathcal{S})} \text{ is an explanation for } \mathcal{C} \setminus \mathcal{S} \\ & \quad \text{such that } \mathcal{P}' \models_{wcs} \mathcal{C} \text{ and } \mathcal{M}_{\mathcal{P}'} \text{ satisfies } \mathcal{IC}. \end{array}$$

This procedure is called *minimal revision followed by abduction* or *MRFA*. Recall that skeptical reasoning is applied under the *Weak Completion Semantics.* This applies to MRFA as well. If in the case $\mathcal{M}_{\mathcal{P}}(\mathcal{C}) = \mathsf{U}$ there are several $\mathcal{P}'$ then the evaluation of the conditional $\mathcal{C} \Rightarrow \mathcal{D}$ must be skeptical. Such examples are discussed in Section 4.1.3.

The case, where the antecendent $\mathcal{C}$ of the conditional $\mathcal{C} \Rightarrow \mathcal{D}$ is mapped to true under $\mathcal{M}_{\mathcal{P}}$, is the easiest. The conditional is an indicative one. Hence, we evaluate the conclusion $\mathcal{D}$ under $\mathcal{M}_{\mathcal{P}}$. The value of the conditional is simply the value of $\mathcal{D}$ under $\mathcal{M}_{\mathcal{P}}$.

If the antecedent $\mathcal{C}$ is false under $\mathcal{M}_{\mathcal{P}}$ then the conditional is a counterfactual. Hence, we revise the background knowledge $\mathcal{P}$ with respect to the literals occurring in $\mathcal{C}$ which are mapped to false under $\mathcal{M}_{\mathcal{P}}$. Thereafter, the conditional is evaluated with respect to the revised background knowledge. One should observe that the antecedent $\mathcal{C}$ is no longer false under the revised background knowledge but may be true or unknown.

The most interesting case is when the antecedent $\mathcal{C}$ is unknown under $\mathcal{M}_{\mathcal{P}}$. We are unaware of any paper where this case has been studied. In the *Weak Completion Semantics* we can apply abduction and/or revision. As we will show in Subsection 4.1.5 there are cases, which can not be solved with abduction alone. Hence, we need revision. On the other hand, revision is so powerful, that it can solve all cases, but in a very straightforward and direct way. We propose to limit revision as much as possible. This is the reason for requiring that $\mathcal{S}$ is a minimal subset of $\mathcal{C}$; all other elements of $\mathcal{S}$ shall be explained by abduction.

4.1.3 The Suppression Task Revisited

The procedure *minimal revision followed by abduction* is an extension of the procedure applied in Chapters 1 and 3 to model the selection task. But the selections task needs to be slightly reformulated. Instead of adding a positive fact or a negative assumption to the background knowledge and asking for a certain goal to hold, the fact and the assumption become the antecedent of a conditional, whose consequent is the goal. Let us illustrate this with the first experiment. Now, the background knowledge is conditional (1.2), i.e.

if she has an essay to write then she will study late in the library.

It is encoded as the program consisting of

$$\begin{array}{rcl} \ell & \leftarrow & e \wedge \neg ab_1, \\ ab_1 & \leftarrow & \bot, \end{array} \tag{4.3}$$

whose least model is

$$\mathcal{M}_{(4.3)} = \langle \emptyset, \{ab_1\} \rangle.$$

The set of integrity constraints is empty. The question is how to evaluate conditional

$$e \Rightarrow \ell \tag{4.4}$$

with respect to program (4.3). The antecedent e is unknown under $\mathcal{M}_{(4.3)}$. Because e is undefined in (4.3), the minimal and skeptical explanation

$$\{e \leftarrow \top\}$$

can be abduced and added to the program. We are now back to the original encoding of the first case of the selection task and obtain

$$\mathcal{M}_{(4.3) \cup \{e \leftarrow \top\}} = \langle \{e, \ell\}, \{ab_1\} \rangle.$$

According to the procedure *minimal revision followed by abduction*, the value of conditional (4.4) is

$$\mathcal{M}_{(4.3)\cup\{e\leftarrow\top\}}(\ell) = \top.$$

Experiments $2-6$ of the suppression task are solved in the same way.

In experiments $7-12$ of the suppression task abduction was already applied in Chapters 1 and 3 (see Table 3.4). This corresponds exactly to the procedure *minimal revision followed by abduction*. Recall that in experiments 8 and 12 there were two minimal explanations and we had to reason skeptically in order to adequately model the experimental data. This applies to the procedure *minimal revision followed by abduction* as well. If in the case $\mathcal{M}_{\mathcal{P}}(\mathcal{C}) = \mathsf{U}$ there are several $\mathcal{P}'$, then skeptical reasoning needs to be applied.

We will proceed by discussing various scenarios in order to illustrate minimal revision followed by abduction.

4.1.4 The Shooting of Kennedy

The following example is discussed in [1]. Given the background knowledge:

> *If Oswald shot then the president was killed. If somebody else shot then the president was killed. Oswald shot.*

Evaluate the following conditionals:

1. *If Oswald had not shot then someone else would have.*
2. *If Kennedy was killed and Oswald did not shoot then someone else did.*

The background knowledge is encoded using the clauses

$$\begin{array}{rcl} k & \leftarrow & os \wedge \neg ab_{os}, \\ ab_{os} & \leftarrow & \bot, \\ k & \leftarrow & ses \wedge \neg ab_{ses}, \\ ab_{ses} & \leftarrow & \bot, \\ os & \leftarrow & \top. \end{array} \tag{4.5}$$

Its least model is

$$\mathcal{M}_{(4.5)} = \langle\{os, k\}, \{ab_{os}, ab_{ses}\}\rangle.$$

The set $\mathcal{IC}$ of integrity constraints is empty.

To evaluate the first conditional $os \Rightarrow ses$[1] we learn that

$$\mathcal{M}_{(4.5)}(\neg os) = \bot.$$

[1] We will omit curly brackets if $\mathcal{C}$ or $\mathcal{D}$ are singleton sets.

Hence, the conditional is a counterfactual and we revise the background knowledge by replacing $os \leftarrow \top$ by $os \leftarrow \bot$ to obtain

$$\mathcal{M}_{rev((4.5),\{\neg os\})} = \langle \emptyset, \{os, ab_{os}, ab_{ses}\}\rangle.$$

Consequently, the antecedent $\neg os$ of the conditional $\neg os \Rightarrow ses$ is true, whereas its consequent ses is unknown and, thus, the value of the conditional is unknown. It is even unknown whether Kennedy was killed.

If, however, it was observed that Kennedy was killed then this corresponds to the second conditional $\{k, \neg os\} \Rightarrow ses$. In this case, we learn

$$\mathcal{M}_{(4.5)}(k \wedge \neg os) = \bot$$

with $\neg os$ being the only literal occurring in $\{k, \neg os\}$ which is mapped to false under $\mathcal{M}_{(4.5)}$. Consequently, we revise the program as before and evaluate the conditional under $\mathcal{M}_{rev((4.5),\{\neg os\})}$. Now, we learn that its antecendent is unknown because k is mapped to unknown. But k can be explained. With

$$\mathcal{A}_{rev((4.5),\{\neg os\})} = \{ses \leftarrow \top, ses \leftarrow \bot, ab_{os} \leftarrow \top, ab_{ses} \leftarrow \top\}$$

we learn that

$$\mathcal{X} = \{ses \leftarrow \top\}$$

is the only minimal explanation for k. We obtain

$$\mathcal{M}_{rev((4.5),\{\neg os\}) \cup \{ses \leftarrow \top\}} = \langle \{ses, k\}, \{os, ab_{os}, ab_{ses}\}\rangle.$$

Evaluating the conditional $\{k, \neg os\} \Rightarrow ses$ under this model we notice that its antecendent as well as its consequent are true, and so is the conditional.

4.1.5 The Firing Squad

Judae Pearl discusses the following example in [54]:

> *If the court orders an execution then the captain will give the signal upon which riflemen a and b will shoot the prisoner; consequently, the prisoner will be dead. It is assumed that the court's decision is unknown, that both riflemen are accurate, alert and law-abiding, and that the prisoner is unlikely to die from any other causes.*

The reader is asked to evaluate the following conditionals:

1. *If the prisoner is not dead then the captain did not signal.*
2. *If rifleman a shot then rifleman b shot as well.*
3. *If rifleman a did not shoot then the prisoner is not dead.*

4. *If the captain gave no signal and rifleman a decides to shoot then the court did not order an execution.*

The background knowledge is encoded using the clauses

$$\begin{array}{rcl} \mathit{signal} & \leftarrow & \mathit{execution} \wedge \neg \mathit{ab}_1 \\ \mathit{ab}_1 & \leftarrow & \bot \\ \mathit{rifleman}_a & \leftarrow & \mathit{signal} \wedge \neg \mathit{ab}_2 \\ \mathit{ab}_2 & \leftarrow & \bot \\ \mathit{rifleman}_b & \leftarrow & \mathit{signal} \wedge \neg \mathit{ab}_3 \\ \mathit{ab}_3 & \leftarrow & \bot \\ \mathit{dead} & \leftarrow & \mathit{rifleman}_a \wedge \neg \mathit{ab}_4 \\ \mathit{ab}_4 & \leftarrow & \bot \\ \mathit{dead} & \leftarrow & \mathit{rifleman}_b \wedge \neg \mathit{ab}_5 \\ \mathit{ab}_5 & \leftarrow & \bot \\ \mathit{alive} & \leftarrow & \neg \mathit{dead} \wedge \neg \mathit{ab}_6 \\ \mathit{ab}_6 & \leftarrow & \bot \end{array} \tag{4.6}$$

and we obtain

$$\mathcal{M}_{(4.6)} = \langle \emptyset, \{\mathit{ab}_i \mid 1 \leq i \leq 6\} \rangle.$$

The set $\mathcal{IC}$ of integrity constraints is empty. The abnormalities will not play a role in evaluating the conditionals mentioned above. But they may be used to model exceptions like, for example, that the firing pin of a rifle broken and this may cause the rifle to malfunktion if a rifleman is pulling the trigger. The abnormalities may also be used to invalidate the rules of the background knowledge, but there is no reason for questioning the rules. On the other hand, it is explicitely stated that the *court's decision is unknown*, which translates into the undefinete atom *execution* and the set of abducibles

$$\{\mathit{execution} \leftarrow \top, \mathit{execution} \leftarrow \bot\}.$$

The fact

$$\mathit{execution} \leftarrow \top \tag{4.7}$$

explains the observation

$$\{\mathit{signal},\ \mathit{rifleman}_a,\ \mathit{rifleman}_b,\ \mathit{dead},\ \neg \mathit{alive}\},$$

whereas the assumption

$$\mathit{execution} \leftarrow \bot \tag{4.8}$$

explains the obervation

$$\{\neg \mathit{signal},\ \neg \mathit{rifleman}_a,\ \neg \mathit{rifleman}_b,\ \neg \mathit{dead},\ \mathit{alive}\}$$

On the other hand, the observation

$$\{\neg \mathit{signal}, \mathit{rifleman}_a\} \tag{4.9}$$

cannot be explained at all. One may be tempted to consider the union of (4.7) and (4.8) as a possible explanation. However, under the weak completion semantics this union is translated into

$$\mathit{execution} \leftrightarrow \top \vee \bot$$

which is semantically equivalent to

$$\mathit{execution} \leftrightarrow \top.$$

The assumption (4.8) is overridden and can no longer be applied to explain that *the captain did not signal*.

The first conditional to be evaluated is

$$\neg \mathit{dead} \Rightarrow \neg \mathit{signal}.$$

Its antecedent $\neg \mathit{dead}$ is unknown under $\mathcal{M}_{(4.6)}$. Adding the explanation (4.8) to the program (4.6) we obtain

$$\mathcal{M}_{(4.6)\cup(4.8)} = \langle\{\mathit{alive}\}, \{\mathit{signal},\ \mathit{rifleman}_a,\ \mathit{rifleman}_b,\ \mathit{dead}\} \cup \{ab_i \mid 1 \leq i \leq 6\}\rangle$$

and the conditional is evaluated to true.

The second conditional is

$$\mathit{rifleman}_a \Rightarrow \mathit{rifleman}_b.$$

Its antecenent $\mathit{rifleman}_a$ is unknown under $\mathcal{M}_{(4.6)}$. Adding the explanation (4.7) to the program (4.6) we obtain

$$\mathcal{M}_{(4.6)\cup(4.7)} = \langle\{\mathit{signal},\ \mathit{rifleman}_a,\ \mathit{rifleman}_b,\ \mathit{dead}\}, \{\mathit{alive}\} \cup \{ab_i \mid 1 \leq i \leq 6\}\rangle$$

and the conditional is evaluated to true.

The third conditional is

$$\neg \mathit{rifleman}_a \Rightarrow \neg \mathit{dead}.$$

Its antecedent $\neg \mathit{rifleman}_a$ is unknown under $\mathcal{M}_{(4.6)}$. As in the first case we add the explanation (4.8) to the program and evaluate the conditional to true.

The forth conditional is

$$\{\neg \mathit{signal}, \mathit{rifleman}_a\} \Rightarrow \neg \mathit{execution}.$$

Its antecedent $\{\neg \mathit{signal}, \mathit{rifleman}_a\}$ is unknown under $\mathcal{M}_{(4.6)}$ and, as discussed above, cannot be explained. Hence, we must apply revision. There are two candidates for minimal revision, viz.

$$\{\mathit{rifleman}_a\} \tag{4.10}$$

and

$$\{\neg \mathit{signal}\} \tag{4.11}$$

Considering the first option, we revise program (4.6) with respect to (4.10) by deleting the rule defining *rifleman*$_a$ and adding the fact $\mathit{rifleman}_a \leftarrow \top$. We obtain

$$\mathcal{M}_{rev((4.6),\mathit{rifleman}_a)} = \langle \{\mathit{rifleman}_a, \mathit{dead}\}, \{\mathit{alive}\} \cup \{ab_i \mid 1 \leq i \leq 6\}\rangle.$$

The second antecedent $\neg \mathit{signal}$ is still unknown under this model. But now we can apply abduction. The explanation (4.8) explains $\neg \mathit{signal}$ and we obtain

$$\begin{aligned}&\mathcal{M}_{rev((4.6),\mathit{rifleman}_a)\cup(4.8)}\\ =\ &\langle \{\mathit{rifleman}_a, \mathit{dead}\}, \{\mathit{alive}, \mathit{execution}, \mathit{signal}, \mathit{rifleman}_b\} \cup \{ab_i \mid 1 \leq i \leq 6\}\rangle.\end{aligned}$$

The conditional is true.

Considering the second option, we revise program (4.6) with respect to (4.11) by deleting the rule defining *signal* and adding the assumption $\mathit{signal} \leftarrow \bot$. We obtain

$$\mathcal{M}_{rev((4.6),\neg \mathit{signal})} = \langle \{\mathit{alive}\}, \{\mathit{signal}, \mathit{rifleman}_a, \mathit{rifleman}_b, \mathit{dead}\} \cup \{ab_i \mid 1 \leq i \leq 6\}\rangle.$$

The second antecedent *rifleman*$_a$ is false under this model and, hence, we would have to revise the program again, but this time with respect to *rifleman*$_a$. But then, the revision is no longer minimal. Hence, the second option is disgarded.

The first option can be nicely illustrated by considering the dependency graph of the program (4.6). The *depends on* relation is the transitive closure of the following relation: Given a ground program $\mathcal{P}$, atom A *depends on* atom B if $\mathcal{P}$ contains a rule of the form $A \leftarrow \mathit{Body}$ and B occurs (positively or negatively) in *Body*.

Fig. 4.1 shows the dependency graph of the program (4.6). Revision cuts the dependencies from a particular node and assigns true or false to the node. Abduction assigns true or false to the node marked *execution*. The revision step can be understood analogously to Pearl's interventions in his *Do-Calculus* [54], where the antecedent node is isolated from its parent nodes in the network and imposed to be true or false.

4.1.6 The Forest Fire

This example is taken from [10].

> *Lightning causes a forest fire. Lightning happened. Dry leaves are usually present.*

The reader is asked to evaluate the conditional

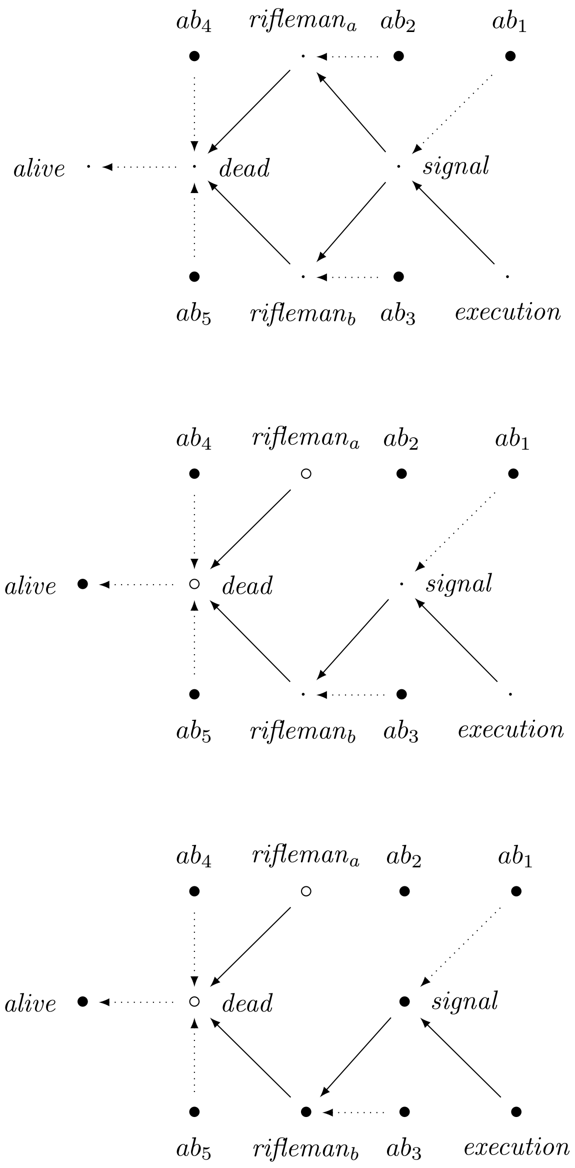


Figure 4.1: (Top) The dependency graph of program (4.6). Positive dependencies are depicted by solid arrows, negative dependencies by dotted arrows. $\bullet$, $\cdot$, and $\circ$ denote nodes, which are mapped to $\bot$, $\cup$ and $\top$ by $\mathcal{M}_{(4.6)}$, respectively. The leaf node marked *execution* is undefined, whereas all other nodes are defined. (Middle) The dependency graph of $rev((4.6), rifleman_a)$: $rifleman_a$ does not depend on *signal* and ab_2 anymore and is mapped to true. (Bottom) The dependency graph of $rev((4.6), rifleman_a) \cup \{execution \leftarrow \bot\}$.

> *If there had not been so many dry leaves on the forest floor then the forest fire would not have occurred*

The background knowledge can be encoded in a program consisting of the following clauses:

$$\begin{array}{rcl} \mathit{forestfire} & \leftarrow & \mathit{lightning} \wedge \neg ab_\ell, \\ \mathit{lightning} & \leftarrow & \top, \\ ab_\ell & \leftarrow & \neg \mathit{dryleaves}, \\ \mathit{dryleaves} & \leftarrow & \top. \end{array} \tag{4.12}$$

The least model of this program is

$$\mathcal{M}_{(4.12)} = \langle \{\mathit{dryleaves}, \mathit{lightning}, \mathit{forestfire}\}, \{ab_\ell\} \rangle.$$

The set $\mathcal{IC}$ of integrity constraints is empty.

To evaluate the conditional

$$\neg \mathit{dryleaves} \Rightarrow \neg \mathit{forestfire}$$

we notice that its antecedent $\neg \mathit{dryleaves}$ is mapped to false by $\mathcal{M}_{(4.12)}$. Hence, the conditional is a counterfactual. Reversing the program (4.12) by $\neg \mathit{dryleaves}$ we obtain

$$\mathcal{M}_{rev((4.12),\mathit{dryleaves})} = \langle \{\mathit{lightning}, ab_\ell\}, \{\mathit{dryleaves}, \mathit{forestfire}\} \rangle.$$

Because *forestfire* is mapped to false now, the counterfactual is mapped to true.

In [56] the background knowledge was extended by

> *Arson may cause a forest fire.*

This additional knowledge can be encoded by

$$\begin{array}{rcl} \mathit{forestfire} & \leftarrow & \mathit{arson} \wedge \neg ab_a \\ ab_a & \leftarrow & \bot. \end{array} \tag{4.13}$$

Now we obtain

$$\mathcal{M}_{(4.12)\cup(4.13)} = \langle \{\mathit{dryleaves}, \mathit{lightning}, \mathit{forestfire}\}, \{ab_\ell, ab_a\} \rangle,$$

which maps the antecedent $\neg \mathit{dryleaves}$ of the conditional

$$\neg \mathit{dryleaves} \Rightarrow \neg \mathit{forestfire}$$

to false. Reversing the extended program by $\neg \mathit{dryleaves}$ we learn

$$\mathcal{M}_{rev((4.12)\cup((4.13),\mathit{dryleaves})} = \langle \{\mathit{lightning}, ab_\ell\}, \{\mathit{dryleaves}, ab_a\} \rangle.$$

In this case, *forestfire* and, consequently, the counterfactual are evaluated to unknown. As arson may have caused the forest fire, the absence of dry leaves is not sufficient to guarantee that there will be no forest fire.

4.1.7 Relevance

4.2 Obligation versus Factual Conditionals

The example is taken from [45, 19] with minor modifications.

If it rains then the roofs are wet and she takes her umbrella.

The logic program representing this background knowledge consists of the following clauses:

$$\begin{array}{rcl} \mathit{wet\_roofs} & \leftarrow & \mathit{rain} \wedge \neg \mathit{ab}_w, \\ \mathit{ab}_w & \leftarrow & \bot, \\ \mathit{umbrella} & \leftarrow & \mathit{rain} \wedge \neg \mathit{ab}_u, \\ \mathit{ab}_u & \leftarrow & \bot. \end{array} \tag{4.14}$$

The least model of program (4.14) is

$$\mathcal{M}_{(4.14)} = \langle \emptyset, \{\mathit{ab}_w, \mathit{ab}_u\} \rangle.$$

The set of abducibles is

$$\{\mathit{rain} \leftarrow \top,\ \mathit{rain} \leftarrow \bot\}.$$

The set of integrity constraints is empty. We are going to evaluate several conditionals with respect to this background knowledge.

1. *If the roofs are not wet then it did not rain.*
2. *If she did not take her umbrella then it did not rain.*
3. *If the roofs are wet then it did rain.*
4. *If she took her umbrella then it did rain.*

In the first two conditionals, the consequent of the background knowledge is denied, whereas in the last two conditionals, the consequent of the background knowledge is confirmed.

4.2.1 Evaluation under MRFA

We start by applying the procedure *minimal revision followed by abduction.*

1. Consider the conditional

   $$\neg \mathit{wet\_roofs} \Rightarrow \neg \mathit{rain}.$$

   It is evaluated to true. Because

   $$\mathcal{M}_{(4.14)}(\neg \mathit{wet\_roofs}) = \mathsf{U}$$

the third case applies. We find a minimal and skeptical explanation

$$\{rain \leftarrow \bot\}$$

for $\neg wet\_roofs$. Thus, the evaluation of the conditional continues with

$$\mathcal{M}_{(4.14)\cup\{rain\leftarrow\bot\}} = \langle\emptyset, \{rain, wet\_roofs, umbrella, ab_w, ab_u\}\rangle.$$

Because

$$\mathcal{M}_{(4.14)\cup\{rain\leftarrow\bot\}}(\neg wet\_roofs) = \top$$

the first case applies and the conditional is evaluated to

$$\mathcal{M}_{(4.14)\cup\{rain\leftarrow\bot\}}(\neg rain) = \top.$$

2. Consider the conditional

   $$\neg umbrella \Rightarrow \neg rain.$$

   It is also evaluated to true following the same steps as in the case of the first example.

3. Consider the conditional

   $$wet\_roofs \Rightarrow rain.$$

   It is also evaluated to true following the same steps as in the case of the first example except that the minimal and skeptical explanation

   $$\{rain \leftarrow \top\}$$

   is abduced to explain the antecedent.

4. Consider the conditional

   $$umbrella \Rightarrow rain.$$

   It is also evaluated to true following the same steps as in the case of the third example.

The examples are summarized in Table 4.1. All conditionals are mapped to true. This is a bit surprising. It is not clear that given the information that *the roofs are not wet*, humans would conclude *it did not rain* in the same way as they would conclude *it did not rain* in case the given information was *she did not take her umbrella*. The same holds for the affirmation of the consequent. Given the information that *she took her umbrella*, it is again not clear that humans would conclude *it did rain* in the same way as they would conclude *it did rain* given the information that the *roofs are wet*.

It appears that the two conditionals of the background knowledge should be semantically interpreted in two different ways. Such a semantic interpretation will be developed in the remainder of this section.

$\mathcal{C}$	$\mathcal{D}$	MRFA steps	semantic MRFA steps
$\neg wet\_roofs$	$\neg rain$	3: $\mathcal{S} = \emptyset$ $\mathcal{X} = \{rain \leftarrow \bot\}$ 1: true	3: $\mathcal{S} = \emptyset$ $\mathcal{X} = \{rain \leftarrow \bot\}$ 1: true
$\neg umbrella$	$\neg rain$	3: $\mathcal{S} = \emptyset$ $\mathcal{X} = \{rain \leftarrow \bot\}$ 1: true	3: $\mathcal{S} = \emptyset$ $\mathcal{X}_1 = \{rain \leftarrow \bot\}$ $\mathcal{X}_2 = \{ab_u \leftarrow \top\}$ 1: unknown
wet_roofs	$rain$	3: $\mathcal{S} = \emptyset$ $\mathcal{X} = \{rain \leftarrow \top\}$ 1: true	3. $\mathcal{S} = \emptyset$ $\mathcal{X} = \{rain \leftarrow \top\}$ 1: true
$umbrella$	$rain$	3: $\mathcal{S} = \emptyset$ $\mathcal{X} = \{rain \leftarrow \top\}$ 1: true	3. $\mathcal{S} = \emptyset$ $\mathcal{X}_1 = \{rain \leftarrow \top\}$ $\mathcal{X}_2 = \{umbrella \leftarrow \top\}$ 1: unknown

Table 4.1: Evaluating four examples under the procedures *minimal revision followed by abduction* and its semantic version. 1 and 3 refer to the first and the third case of the procedure.

4.2.2 Semantics of Conditionals

Obligation and Factual Conditionals

Consider the conditionals

$$\textit{if it rains then the roofs are wet} \tag{4.15}$$

and

$$\textit{if it rains then she takes her umbrella.} \tag{4.16}$$

The consequence of the first conditional is obligatory. We cannot easily imagine a case, where the antecedent is true and the consequence is not. On the other hand, we can easily imagine a situation, where the antecedent of the second conditional is true and the consequence is not. The consequence of the second conditional is not obligatory. We will call the first conditional a *obligation conditional* and the second one a *factual conditional*.

As explained in [10], a conditional whose consequence is denied is more likely to be evaluated to true if it is an obligation conditional. This happens because for this type of conditional there is a forbidden or unlikely possibility where antecedent and not consequent happen together and, in this case, where the consequent is known to be false, it cannot be the case that the antecedent is true as, otherwise, the forbidden possibility is violated. Thus, not antecedent is concluded. Because in the case of a factual conditional this forbidden possibility does not exist, a conditional whose consequence is denied should be evaluated as unknown.

There are many other obligation conditionals:

If the roofs are not wet then it did not rain.
If a german tourist wants to enter Russia then he needs a visa.
If there is no light then plants will not grow.

Likewise, there are many other factual conditionals:

If she did not take her umbrella then it did not rain.
If the sun is shining all day then I will water my garden in the evening.
If Carl is not doing his homework then he will fail the exam.

Subjects may classify conditionals as obligation or factual conditionals. This is an informal and pragmatic classification. It depends on the background knowledge and experience of a subject as well as on the context in which a conditional is stated.

Necessary and Sufficient Antecedents

The antecedent of conditional (4.15) is *necessary.* The consequent cannot be true unless the antecedent is true. The antecedent of conditional (4.16) does not appear to be necessary. There are many different reasons for taking an umbrella like, for example, that the sun is shining. The antecedent of conditional (4.4) is *sufficient.*

Subjects may classify antecedents as necessary or sufficient. The classification is informal and pragmatic. It depends on the background knowledge and experience of a subject as well as on the context in which the condition is stated.

4.2.3 Representing Obligation and Factual Conditionals

Obligation and factual conditionals are represented by programs as before. Thus, the program (4.14) remains unchanged. However, the semantics of conditionals is taken into consideration when modifying the set of abducibles for a given program $\mathcal{P}$:

$$\begin{aligned}
\mathcal{A}_\mathcal{P} &= \{A \leftarrow \top \mid A \text{ is undefined in } \mathcal{P}\} && (4.17)\\
&\cup \{A \leftarrow \bot \mid A \text{ is undefined in } \mathcal{P}\} && (4.18)\\
&\cup \{A \leftarrow \top \mid A \text{ is head of a conditional with sufficient antecedent in } \mathcal{P}\} && (4.19)\\
&\cup \{ab \leftarrow \top \mid ab \text{ occurs in the body of a factual conditional in } \mathcal{P}\} && (4.20)
\end{aligned}$$

The sets (4.17) and (4.18) are the usual facts and assumptions for the undefined atoms occurring in the program $\mathcal{P}$, respectively. The set (4.19) contains facts for the heads of conditionals with sufficient antecedent occurring in $\mathcal{P}$. If an antecedent of a conditional is sufficient then there may be other reasons for establishing the conclusion of the conditional. The set (4.60) contains facts for the abnormalities occurring in factual conditionals. The antecedent of a factual conditional may be true, yet the conclusion of the conditional may still not hold. Adding a fact for the abnormality occurring in the body of the representation of a factual conditional will force this abnormality to become true and its negation to become false. Hence, the body of the clause will be false.

4.2.4 Evaluation under Semantic MRFA

We may evaluate the conditional 1.-4. introduced at the beginning of Section 4.2 using the modified set of abducibels specified in Section 4.2.3. For program (4.14) we obtain the set

$$\mathcal{A}_{(4.14)} = \{\mathit{rain} \leftarrow \top,\ \mathit{rain} \leftarrow \bot,\ \mathit{ab}_u \leftarrow \top,\ \mathit{umbrella} \leftarrow \top\}.$$

1. The evaluation of the conditional

   $$\neg \mathit{wet\_roofs} \Rightarrow \neg \mathit{rain}$$

   remains the same because the explanation for the antecedent $\neg \mathit{wet\_roofs}$ is unique. Thus, the consequence $\neg \mathit{rain}$ is a skeptical consequence from program (4.14) and the observation $\neg \mathit{wet\_roofs}$. The conditional is true.

2. The evaluation of the conditional

   $$\neg \mathit{umbrella} \Rightarrow \neg \mathit{rain}$$

   changes. Because of the modified set of abducibles $\mathcal{A}_{(4.14)}$ there are now two minimal explanations for the antecedent $\neg \mathit{umbrella}$:

   $$\mathcal{X}_1 = \{\mathit{rain} \leftarrow \bot\}$$

   as before and

   $$\mathcal{X}_2 = \{\mathit{ab}_u \leftarrow \top\}.$$

   Whereas

   $$(4.14) \cup \mathcal{X}_1 \models_{wcs} \neg \mathit{rain}$$

   we find that

   $$(4.14) \cup \mathcal{X}_2 \not\models_{wcs} \neg \mathit{rain}.$$

   Reasoning skeptically the conditional is unknown.

3. The evaluation of the conditional

   $$\mathit{wet\_roofs} \Rightarrow \mathit{rain}$$

   remains the same. The conditional is true following the same steps as before.

4. The evaluation of the conditional

$$\mathit{umbrella} \Rightarrow \mathit{rain}$$

changes. Because of the modified set of abducibles $\mathcal{A}_{(4.14)}$ there are now two minimal explanations for the antecedent $\neg \mathit{umbrella}$:

$$\mathcal{X}_1 = \{\mathit{rain} \leftarrow \top\}$$

as before and

$$\mathcal{X}_2 = \{\mathit{umbrella} \leftarrow \top\}.$$

Whereas

$$(4.14) \cup \mathcal{X}_1 \models_{wcs} \mathit{rain}$$

we find that

$$(4.14) \cup \mathcal{X}_2 \not\models_{wcs} \mathit{rain}.$$

Reasoning skeptically the conditional is unknown.

The evaluation of the four examples under *semantic minimal revision followed by abduction* is summarized in Table 4.1. Although we are unaware of any experimental data to support the following hypothesis, we strongly believe that the semantic version of MRFA models human reasoning much better than the original version.

4.3 The Selection Task

The selection task is yet another famous psychological experiment which has been repeated many times leading to similar results. In its original, *abstract* version [67], participants were told that cards had letters on one side and numbers on the other side. In Table 4.2(a) four cards are depicted, showing the letters d and f as well as the numbers 3 and 7. Then, participants were given the conditional

$$\textit{if there is a d on one side of a card then there is a 3 on the other side.} \tag{4.21}$$

Finally, the participants were asked which cards must be turned to prove that the conditional holds.

From a classical logic point of view, the conditional can be represented by the implication

$$d \rightarrow 3.$$

In this case, the cards showing d (modus ponens) and 7 (modus tollens) must be turned. As repeated experiments have shown consistently, the majority of the participants correctly selected the card showing d. However, they failed to select the card showing 7 and incorrectly selected the card showing 3. In other words, the overall correctness of the answers for the abstract selection task if modeled by a classical, two-valued implication is pretty bad.

d	f	3	7
89%	16%	62%	25%

(a)

beer	coke	22	16
95%	0,025%	0,025%	80%

(b)

Table 4.2: (a) the abstract and (b) the social selection task.

Griggs and Cox [25] presented a version of the selection task which is structurally isomorphic, but in a *social* context. Participants were told that cards are showing liquids on one side and the age of the person drinking the liquid on the other side. In Table 4.2(b) four cards are depicted, showing (alcoholic) *beer* and *coke* as well as the numbers *22* and *16*. Then, the participants were given the conditional

if a person is drinking beer then the person must be over 19 years of age. (4.22)

Again, participants were asked which cards must be turned to prove that the conditional holds. Participants consistently solved this task correctly by turning the cards showing *beer* and *16*.

One explaination for the difference between the two cases can be found in [43], namely that people consider the conditional in the abstract case as a belief. The participants perceived the task to examine whether the conditional is either true or false. On the other hand, in the social case, people consider the conditional as a social constraint which ought to be true. Participants intuitively aim at preventing the violation of such a constraint, which is normally done by observing whether the state of the world complies with the rule.

In [17] a computational logic approach using the *Weak Completion Semantics* has been proposed to model the two cases of the selection task. The approach presented there does not distinguish between the two conditionals, but instead models the different interpretations outside of the logical framework.

The different results for the abstract and the social case of the selection task with the same structure confirms that the semantics of the conditionals is relevant for their evaluation. Taking Bob Kowalski's [43] explanation and the semantics presented in Section 4.2.2 into account, we understand

- the conditional in the abstract case as a factual conditional with necessary antecedent

and

- the conditional in the social case as an obligation conditional with sufficient antecedent.

In the following, we will model the two cases within one logical framework by distinguishing the abstract and the social case with respect to the classification of the conditionals.

4.3.1 Modeling the Abstract Case

The background knowledge of this case is given by the following program

$$\begin{array}{rcl} \mathit{3} & \leftarrow & d \wedge \neg ab_a, \\ ab_a & \leftarrow & \bot. \end{array} \tag{4.23}$$

As the conditional was classified as a factual conditional with necessary antecedent, its set of abducibles is

$$\{d \leftarrow \top,\ d \leftarrow \bot,\ ab_a \leftarrow \top\}.$$

Observing a card, the decision by the participants is modeled by abductively explaining the given observation $\mathcal{O}$, computing the least model $\mathcal{M}_{(4.23)\cup\mathcal{X}}$ of the weak completion of the program (4.23) and the explanation $\mathcal{X}$, and reasoning skeptically to decide whether the card must be turned. In particular, a card is turned if and only if

1. $\mathit{3}$ and d follow skeptically from program (4.23) and $\mathcal{O}$ or
2. for all explanations $\mathcal{X}$ explaining the observation $\neg\mathit{3}$ we find $d \leftarrow \bot \in \mathcal{X}$.

$\mathcal{O}$	d		$\neg d$		$\mathit{3}$		$\neg\mathit{3}$			
$\mathcal{X}$	$d \leftarrow \top$		$d \leftarrow \bot$		$d \leftarrow \top$		$d \leftarrow \bot$		$ab_a \leftarrow \top$	
$\Phi_{(4.23)\cup\mathcal{X}}$	$I^\top$	$I^\bot$	$I^\top$	$I^\bot$	$I^\top$	$I^\bot$	$I^\top$	$I^\bot$	$I^\top$	$I^\bot$
0										
1	d	ab_a		d ab_a	d	ab_a		d ab_a	ab_a	
2	$\mathit{3}$			$\mathit{3}$	$\mathit{3}$			$\mathit{3}$		$\mathit{3}$
decision	turn		no turn		turn		no turn			
	89%		16%		62%		25%			

- If $\mathcal{O} = \{d\}$ then the only explanation is $\{d \leftarrow \top\}$ and we obtain

  $$\mathcal{M}_{(4.23) \cup \{d \leftarrow \top\}} = \langle \{d, \mathit{3}\}, \{ab_a\} \rangle$$

  As both, d and $\mathit{3}$ are mapped to true, the card must be turned in order to verify that a $\mathit{3}$ is on the other side. This corresponds to modus ponens.

- If $\mathcal{O} = \{\neg d\}$, i.e. if, for example, an f is on the one side of a card, then the only explanation is $\{d \leftarrow \bot\}$ and we obtain

  $$\mathcal{M}_{(4.23) \cup \{d \leftarrow \bot\}} = \langle \emptyset, \{d, ab_a, \mathit{3}\} \rangle$$

  There is no need to turn the card as the antecedent d of the conditional (4.21) is false and the conditional (4.21) is true.

- If $\mathcal{O} = \{\mathit{3}\}$ then the only explanation is $\{d \leftarrow \top\}$ and we obtain

  $$\mathcal{M}_{(4.23) \cup \{d \leftarrow \top\}} = \langle \{d, \mathit{3}\}, \{ab_a\} \rangle$$

  As both, d and $\mathit{3}$ are mapped to true, the card must be turned in order to verify that a d is on the other side. This corresponds to the interpretation of d being a necessary antecedent.

- If $\mathcal{O} = \{\neg \mathit{3}\}$, i.e. if, for example, a $\mathit{7}$ is on the one side of a card, then there are two minimal explanations, viz. $\{d \leftarrow \bot\}$ and $\{ab_a \leftarrow \top\}$, and we obtain

  $$\mathcal{M}_{(4.23) \cup \{d \leftarrow \bot\}} = \langle \emptyset, \{d, ab_a, \mathit{3}\} \rangle$$

  and

  $$\mathcal{M}_{(4.23) \cup \{ab_a \leftarrow \top\}} = \langle \{ab_a\}, \{\mathit{3}\} \rangle.$$

  Reasoning skeptically we conclude that $\mathit{3}$ is false, whereas d and ab_a are unknown. There is no need to turn the card. One should observe that a creduluous reasoner would have turned the card as the first explanation $\{d \leftarrow \bot\}$ leads to a least model where d and $\mathit{3}$ are mapped to false. Given the observation ¬3, the creduluous reasoner must verify that there is no d on the other side of the card.

4.3.2 Modeling the Social Case

The background knowledge of this case is given by the following program

$$\begin{array}{rcl} o & \leftarrow & b \wedge \neg ab_s, \\ ab_s & \leftarrow & \bot, \end{array} \tag{4.24}$$

where b denotes *beer* and o denotes that the person is *old enough to drink beer*, i.e. older than 18. As the conditional was classified as an obligation conditional with sufficient antecedent, its set of abducibles is

$$\{b \leftarrow \top,\ b \leftarrow \bot,\ o \leftarrow \top\}.$$

Observing a card, the decision by the participants is again modeled by abductively explaining the given observation $\mathcal{O}$, computing the least model $\mathcal{M}_{(4.24)\cup\mathcal{X}}$ of the weak completion of the program (4.24) and the explanation $\mathcal{X}$, and reasoning skeptically to decide whether the card must be turned. In particular, a card is turned if and only if

1. o and b follow skeptically from program (4.24) and $\mathcal{O}$ or
2. for all explanations $\mathcal{X}$ explaining the observation $\neg o$ we find $b \leftarrow \bot \in \mathcal{X}$.

$\mathcal{O}$	b		$\neg b$		o				$\neg o$	
$\mathcal{X}$	$b \leftarrow \top$		$b \leftarrow \bot$		$o \leftarrow \top$		$b \leftarrow \top$		$b \leftarrow \bot$	
$\Phi_{(4.24)\cup\mathcal{X}}$	$I^\top$	$I^\bot$	$I^\top$	$I^\bot$	$I^\top$	$I^\bot$	$I^\top$	$I^\bot$	$I^\top$	$I^\bot$
0										
1	b	ab_s		b ab_s	o	ab_s	b	ab_s		b ab_s
2	o			o			o			o
decision	turn		no turn		no turn				turn	
	95%		0,025%		0,025%				80%	

- If $\mathcal{O} = \{b\}$ then the only explanation is $\{b \leftarrow \top\}$ and we obtain

$$\mathcal{M}_{(4.24)\cup\{b\leftarrow\top\}} = \langle\{b, o\}, \{ab_s\}\rangle$$

  As both, b and o are mapped to true, the card must be turned in order to verify that the person drinking beer is old enough.

- If $\mathcal{O} = \{\neg b\}$, i.e. if, for example, a coke is shown on the one side of a card, then the only explanation is $\{b \leftarrow \bot\}$ and we obtain

$$\mathcal{M}_{(4.24)\cup\{b\leftarrow\bot\}} = \langle\emptyset, \{b, ab_s, o\}\rangle$$

  There is no need to turn the card as the antecedent b of the conditional (4.22) is false and the conditional (4.22) is true.

- If $\mathcal{O} = \{o\}$, i.e. if, for example, a *22* is on the one side of a card, then there are two minimal explanations, viz. $\{o \leftarrow \top\}$ and $\{b \leftarrow \top\}$, and we obtain

$$\mathcal{M}_{(4.24)\cup\{o\leftarrow\top\}} = \langle\{o\}, \{ab_s\}\rangle$$

  and

$$\mathcal{M}_{(4.24)\cup\{b\leftarrow\top\}} = \langle\{b, o\}, \{ab_s\}\rangle.$$

  Reasoning skeptically we conclude that o is true and ab_a is false. There is no need to turn the card. One should observe that a creduluous reasoner would have to turn the card. The second explanation $\{b \leftarrow \top\}$ leads to a least model where b and o are mapped to true.

- If $\mathcal{O} = \{\neg o\}$ then the only explanation is $\{b \leftarrow \bot\}$ and we obtain

  $$\mathcal{M}_{(4.24)\cup\{b\leftarrow\bot\}} = \langle \emptyset, \{b, ab_s, o\}\rangle$$

  As $\neg o$ is observed we must verify that there is no *beer* is on the other side.

The conditions for turning a card in the abstract and the social case of the selection task are structurally identical. They can be generalized. Suppose we are considering cards with a symbol of class x on one side and a symbol of class y on the other side. Given the conditional

if x then y

the background knowledge is encoded using the clauses

$$\begin{array}{rcl} y & \leftarrow & x \wedge \neg ab, \\ ab & \leftarrow & \bot. \end{array}$$

The set of abducibles is determined based on the semantic interpretation of the conditional. A card is *turned* if and only if

1. x and y follow skeptically from the program and the observation or
2. for all explanations $\mathcal{X}$ explaining the observation $\neg y$ we find $x \leftarrow \bot \in \mathcal{X}$.

4.4 Syllogistic Reasoning

4.5 Contextual Abduction

Consider the following szenario:

Usually birds fly. Tweety and Jerry are birds.

This can be represented by a (data logic) program consising of the clauses

$$\begin{array}{rcl} \mathit{fly}\,X & \leftarrow & \mathit{bird}\,X \wedge \neg ab_{\mathit{fly}}\,X, \\ ab_{\mathit{fly}}\,X & \leftarrow & \bot, \\ \mathit{bird}\,\mathit{tweety} & \leftarrow & \top, \\ \mathit{bird}\,\mathit{jerry} & \leftarrow & \top. \end{array} \tag{4.25}$$

The least model of the weak completion of this program is

$$\langle\{\mathit{bird}\,\mathit{tweety},\ \mathit{bird}\,\mathit{jerry},\ \mathit{fly}\,\mathit{tweety},\ \mathit{fly}\,\mathit{jerry}\}, \{ab_{\mathit{fly}}\,\mathit{tweety},\ ab_{\mathit{fly}}\,\mathit{jerry}\}\rangle.$$

Its set of abducibles is

$$\{ab_{fly}\ tweety \leftarrow \top,\ ab_{fly}\ jerry \leftarrow \top\}.$$

If we observe that *Tweety does not fly* then there are two explanations, viz.

$$\begin{array}{rcl} \mathcal{X}_1 & = & \{ab_{fly}\ tweety \leftarrow \top\}, \\ \mathcal{X}_2 & = & \{ab_{fly}\ tweety \leftarrow \top,\ ab_{fly}\ jerry \leftarrow \top\} \end{array}$$

with $\mathcal{X}_1$ being the minimal one. If we add $\mathcal{X}_1$ to the program (4.25) and weakly complete the extended program then we obtain the least model

$$\langle\{bird\ tweety,\ bird\ jerry,\ fly\ jerry,\ ab_{fly}\ tweety\}, \{fly\ tweety,\ ab_{fly}\ jerry\}\rangle,$$

which entails the observation.

Now consider the extended scenario

Usually birds fly, but kiwis and penguins do not. Tweety and Jerry are birds.

This can be represented by a program consising of the clauses

$$\begin{array}{rcl} fly\ X & \leftarrow & bird\ X \wedge \neg ab_{fly}\ X, \\ ab_{fly}\ X & \leftarrow & kiwi\ X, \\ ab_{fly}\ X & \leftarrow & penguin\ X, \\ bird\ tweety & \leftarrow & \top, \\ bird\ jerry & \leftarrow & \top. \end{array} \tag{4.26}$$

The least model of the weak completion of this program is

$$\langle\{bird\ tweety,\ bird\ jerry\}, \emptyset\rangle.$$

Its set of abducibles consists of

$$\begin{array}{rclrcl} kiwi\ tweety & \leftarrow & \top, & kiwi\ tweety & \leftarrow & \bot, \\ kiwi\ jerry & \leftarrow & \top, & kiwi\ jerry & \leftarrow & \bot, \\ penguin\ tweety & \leftarrow & \top, & penguin\ tweety & \leftarrow & \bot, \\ penguin\ jerry & \leftarrow & \top, & penguin\ jerry & \leftarrow & \bot. \end{array}$$

If we observe that *Jerry does fly* then the minimal explanation

$$\mathcal{X} = \{kiwi\ jerry \leftarrow \bot,\ kiwi\ penguin \leftarrow \bot\}$$

explains this observation. Thus, in order to explain the observation we need to consider all known exceptions before we can conclude that *Jerry does fly.* In this case, we need to assume that *Jerry is not a kiwi* and *Jerry is not a penguin.*

There are several problems with this approach. Firstly, to the best of our knowledge there are currently 41 known classes of birds which do not fly. Secondly, there may be classes of non-flying birds which we are unaware of. Hence, it is unlikely that humans consider all known exceptions before concluding that *Jerry does fly.*

L	$ctxt\,L$
$\top$	$\top$
$\bot$	$\bot$
U	$\bot$

Table 4.3: The truth table for $ctxt(L)$.

4.5.1 The Context Operator

Luís Moniz Pereira and Alexandre Miguel Pinto have introduced *inspection points* of the form $inspect\,L$, where L is a literal [57]. Inspection points are treated as meta-predicates belonging to a special case of abducibles: $inspect\,L$ can only be abduced to explain some observation in case L is abduced to explain some given observation. More formally, $\mathcal{X}$ is an explanation if for each $inspect\,L \in \mathcal{X}$ we find that $L \in \mathcal{X}$. That is, $inspect\,L$ is only accepted in the context of L.

Inspired by the idea underlying inspection points, we introduce a new truth-functional operator *ctxt* (called *context*), whose meaning is specified in Table 4.3. With the help of *ctxt*, preferences on explanations–among other things–can be syntactically specified. These preferences are context-dependent.

The interpretation of *ctxt* can be understood as a mapping from three-valuedness to two-valuedness. It is one possible way to capture negation as failure under the *Weak Completion Semantics*. The original idea of negation as failure [12] is to derive the negation of a ground atom A in case we fail to derive A, where the meaning of derivation failure depends on the semantics. Negation as failure does not exist under the *Weak Completion Semantics*, quite the contrary is the case. Consider a program consisting of the clauses

$$\begin{array}{rcl} p & \leftarrow & q, \\ p & \leftarrow & \bot. \end{array} \tag{4.27}$$

Its weak completion is

$$\{p \leftrightarrow q \vee \bot\},$$

which is semantically equivalent to $\{p \leftrightarrow q\}$. The least model of $\{p \leftrightarrow q\}$ is the empty interpretation $\langle\emptyset, \emptyset\rangle$. Both, p and q are unknown, whereas they would be false if negation as failure had been adopted. The assumption $p \leftarrow \bot$ has been overridden by the rule $p \leftarrow q$ and does not have any effect at all. On the other hand, $ctxt\,L = \bot$ if L is unknown.

As another example consider the program consisting of the clauses

$$\begin{array}{rcl} p\,X & \leftarrow & X \approx a, \\ q\,X & \leftarrow & X \approx b \wedge rb, \\ X \approx X & \leftarrow & \top, \\ a \approx b & \leftarrow & \bot, \\ b \approx a & \leftarrow & \bot. \end{array} \tag{4.28}$$

As the weak completion of this program we obtain

$$\begin{array}{rcl} p\,a & \leftrightarrow & a \approx a, \\ p\,b & \leftrightarrow & b \approx a, \\ q\,a & \leftrightarrow & a \approx b \wedge r\,b, \\ q\,b & \leftrightarrow & b \approx b \wedge r\,b, \\ a \approx a & \leftrightarrow & \top, \\ b \approx b & \leftrightarrow & \top, \\ a \approx b & \leftrightarrow & \bot, \\ b \approx a & \leftrightarrow & \bot, \end{array}$$

whose least model is

$$\langle \{a \approx a,\ b \approx b,\ p\,a\}, \{a \approx b,\ b \approx a,\ p\,b,\ q\,a\} \rangle. \tag{4.29}$$

Compared to the weak completion of program (3.5) we find that pb and qa are mapped to false, whereas ra and rb are still mapped to unknown. The main reasons for this difference are the explicitly assumed inequalities $a \not\approx b$[2] and $b \not\approx a$ and the different form of specifying the definition for pa and qb, viz. the from used by Keith Leonhard Clark in [12].

To specify inequalities like $a \not\approx b$ or $b \not\approx a$ seems possible if these inequalities are known and if there are not too many. However, if we are modelling a company with 1000 employes, we probably do not want to record explicitly that any two of these employees are different. In many scenarios we may not even have the knowledge. For example, who knows the 41 different species of non-flying birds in the world? In such cases we would like to find a way to somehow *jump to the conclusion* that two syntactically different constants denote different object, or that two employes with different names are different persons, or that a bird is flying. Of course, we should be willing to override such *default conclusions*. This is possible if we use the context operator and replace program (4.28) by:

$$\begin{array}{rcl} p\,X & \leftarrow & \mathit{ctxt}\, X \approx a, \\ q\,X & \leftarrow & \mathit{ctxt}\, X \approx b \wedge r\,b, \\ X \approx X & \leftarrow & \top. \end{array} \tag{4.30}$$

As the weak completion of (4.30) we obtain

$$\begin{array}{rcl} p\,a & \leftrightarrow & \mathit{ctxt}\, a \approx a, \\ p\,b & \leftrightarrow & \mathit{ctxt}\, b \approx a, \\ q\,a & \leftrightarrow & \mathit{ctxt}\, a \approx b \wedge r\,b, \\ q\,b & \leftrightarrow & \mathit{ctxt}\, b \approx b \wedge r\,b, \\ a \approx a & \leftrightarrow & \top, \\ b \approx b & \leftrightarrow & \top, \end{array}$$

whose least model is

$$\langle \{a \approx a,\ b \approx b,\ p\,a\}, \{p\,b,\ q\,a\} \rangle. \tag{4.31}$$

[2] We prefer to write $\neg a \approx b$ in the more common form $a \not\approx b$.

Comparing (4.29) and (4.31) we observe that the models are identical except for the inequalities $a \not\approx b$ and $b \not\approx a$. They are unknown under the model (4.31).

Program (4.30) is a so-called *contextual program* which will be formally introduced in the next subsection.

4.5.2 Contextual Programs

Remember that literals are atoms or negated atoms. Let L be a literal. A *contextual literal* is an expression of the form $\mathit{ctxt}\,L$ or $\neg\mathit{ctxt}\,L$. A *contextual rule* is an expression of the form $A \leftarrow \mathit{Body}$, where A is an atom and Body is a finite conjunction of literals and contextual literals containing at least one contextual literal. A *contextual program* is a set of rules, contextual rules, facts, and assumptions containing at least one contextual rule.

In other words, a set of clauses is a contextual program if and only if it contains an occurrence of the context operator. Otherwise, it is just a program. We are quite carefully distinguishing between programs and contextual programs as they have very different properties. This will become clear in a moment.

As an example consider the contextual program consisting of the clauses

$$\begin{array}{ccl} p & \leftarrow & \mathit{ctxt}\,q, \\ p & \leftarrow & \bot. \end{array} \tag{4.32}$$

Its weak completion consists of the equivalence

$$p \leftrightarrow \mathit{ctxt}\,q \vee \bot. \tag{4.33}$$

The empty interpretation $\langle \emptyset, \emptyset \rangle$ is not a model for (4.33): If q is unknown then $\mathit{ctxt}\,q$ is false and, consequently, the right-hand side of (4.33) is false, whereas its left-hand side p is unknown. On the other hand,

$$\langle \emptyset, \{p\} \rangle \tag{4.34}$$

is a model for (4.33). This model is also computed by the semantic operator $\Phi_{(4.32)}$ in one iteration starting with the empty interpretation. It is a minimal model, but not the least one.

$$\langle \{p, q\}, \emptyset \rangle \tag{4.35}$$

is another minimal model. However, (4.35) cannot be computed by the semantic operator. On the contrary, if start the iteration of the semantic operator with (4.35) we obtain:

$\Phi_{(4.32)}$	$I^\top$	$I^\bot$
$\uparrow 1$	p	
$\uparrow 2$		p

We will call (4.34) a *supported* model as it is a fixed point of the semantic operator. A least model does not exist in this example.

But fixed points do not always exist. As an example consider the contextual program which consists only of the contextual rule

$$p \leftarrow \mathit{ctxt}\,\neg p \tag{4.36}$$

If we iterate the semantic operator starting with the empty interpretation then we obtain:

$\Phi_{(4.36)}$	$I^\top$	$I^\bot$
$\uparrow 1$		p
$\uparrow 2$	p	
$\uparrow 3$		p
$\vdots$	$\vdots$	$\vdots$

There is no fixed point. The operator is no longer monotonic and, hence, the Knaster-Tarski Theorem 24 is no longer applicable.

The contextual program (4.36) has two models, viz. the empty interpretation and the interpretation $\langle \{p\}, \emptyset \rangle$. However, the weak completion of (4.36), viz. the equivalence

$$p \leftrightarrow \mathit{ctxt}\,\neg p$$

has no model at all. It is unsatisfiable. One should remember that this cannot happen with ordinary programs. If a program does not contain an occurrence of the context operator then its weak completion has a least model by Theorem 13. For example, the empty interpretation is the least model of $p \leftrightarrow \neg p$.

Returning to the Tweety scenario, suppose we consider the program consisting of the clauses

$$\begin{array}{rcl} \mathit{fly}\,X & \leftarrow & \mathit{bird}\,X \wedge \neg \mathit{ab}_{\mathit{fly}}\,X, \\ \mathit{ab}_{\mathit{fly}}\,X & \leftarrow & \mathit{ctxt}\,\mathit{kiwi}\,X, \\ \mathit{ab}_{\mathit{fly}}\,X & \leftarrow & \mathit{ctxt}\,\mathit{penguin}\,X, \\ \mathit{bird}\,\mathit{tweety} & \leftarrow & \top, \\ \mathit{bird}\,\mathit{jerry} & \leftarrow & \top. \end{array} \tag{4.37}$$

instead of (4.26), where we have revised the definition of $\mathit{ab}_{\mathit{fly}}$ by introducing the context operator in the bodies of the rules. If we iterate the semantic operator starting with the empty interpretation we obtain:

$\Phi_{(4.37)}$	$I^\top$	$I^\bot$
$\uparrow 1$	*bird tweety* *bird jerry*	$\mathit{ab}_{\mathit{fly}}$ *tweety* $\mathit{ab}_{\mathit{fly}}$ *jerry*
$\uparrow 2$	*bird tweety* *bird jerry* *fly tweety* *fly jerry*	$\mathit{ab}_{\mathit{fly}}$ *tweety* $\mathit{ab}_{\mathit{fly}}$ *jerry*

Thus, we are *reasoning by default* that *Tweety and Jerry are flying* without even considering the exceptional cases of birds which do not fly. However, if we are additionally told that *Tweety is a penguin* then we obtain the program consisting of the clauses

$$\begin{aligned} \mathit{fly}\ X &\leftarrow \mathit{bird}\ X \wedge \neg \mathit{ab}_{\mathit{fly}}\ X, \\ \mathit{ab}_{\mathit{fly}}\ X &\leftarrow \mathit{ctxt}\ \mathit{kiwi}\ X, \\ \mathit{ab}_{\mathit{fly}}\ X &\leftarrow \mathit{ctxt}\ \mathit{penguin}\ X, \\ \mathit{bird}\ \mathit{tweety} &\leftarrow \top, \\ \mathit{bird}\ \mathit{jerry} &\leftarrow \top, \\ \mathit{penguin}\ \mathit{tweety} &\leftarrow \top. \end{aligned} \tag{4.38}$$

Iterating the semantic operator starting with the empty interpretation we obtain:

$\Phi_{(4.38)}$	$I^{\top}$	$I^{\bot}$
$\uparrow 1$	*bird tweety* *bird jerry* *penguin tweety*	$\mathit{ab}_{\mathit{fly}}$ *tweety* $\mathit{ab}_{\mathit{fly}}$ *jerry*
$\uparrow 2$	*bird tweety* *bird jerry* *penguin tweety* $\mathit{ab}_{\mathit{fly}}$ *tweety* *fly tweety* *fly jerry*	$\mathit{ab}_{\mathit{fly}}$ *jerry*
$\uparrow 3$	*bird tweety* *bird jerry* *penguin tweety* $\mathit{ab}_{\mathit{fly}}$ *tweety* *fly jerry*	$\mathit{ab}_{\mathit{fly}}$ *jerry* *fly tweety*

The example shows again that the semantic operator is no longer monotonic. Whereas $\mathit{ab}_{\mathit{fly}}$ *tweety* is false after the first iteration (the context operator is applied to unknown in all instantiations of X), it becomes true after the second iteration. This holds as *penguin tweety* was mapped to true in the first iteration. Likewise, *fly tweety* was true after the second iteration, but is false after the third iteration. After the third iteration, a fixed point has been reached. *Jerry is flying* but *Tweety is not.*

The example also shows that in the first iteration of the semantic operator we jump to the conclusion that *Tweety is not abnormal*, which is revised in the second iteration after we learn that *Tweety is a penguin.* Likewise, in the second iteration we jump to the conclusion that *Tweety is flying*, which again has to be revised in the third iteration.

Although the semantic operator is no longer monotonic for contextual program, it may have a fixed point under certain conditions. In particular, we will show that if the contextual program is acylcic, then such a fixed point exists and can be computed.

In order to do so, we have to extend the notion of a level mapping. Let L be a literal.

$$lvl(ctxt\, L) = lvl(\neg ctxt\, L) = lvl(L).$$

A contextual program $\mathcal{P}$ is *acyclic with respect to the level mapping lvl* if and only if for each rule $A \leftarrow Body$ occurring in $\mathcal{P}$ and each (normal or contextual) literal L occurring in $Body$ we find $lvl(A) > lvl(L)$. With these extensions, Proposition 25 still applies ensuring that d_{lvl} is a metric. Furthermore, Proposition 26 still applies ensuring that $(\mathcal{I}, d_{lvl})$ is a complete metric space.

We can now state a more general result than Theorem 27:

Theorem 30 *Let $\mathcal{P}$ be a contextual program, $\mathcal{E}$ and equational theory, lvl a level mapping for $\mathcal{P}$, and $\mathcal{I}$ the set of interpretations for $\mathcal{P}$. If $\mathcal{P}$ is acyclic with respect to lvl then $\Phi_{\mathcal{P}}$ is a contraction on the metric space $(\mathcal{I}, d_{lvl})$.*

Proof We will show

$$d_{lvl}(\Phi_{\mathcal{P}}(I), \Phi_{\mathcal{P}}(J)) \leq \frac{1}{2} d_{lvl}(I, J).$$

If $I = J$ then $\Phi_{\mathcal{P}}(I) = \Phi_{\mathcal{P}}(J)$ and, consequently,

$$d_{lvl}(\Phi_{\mathcal{P}}(I), \Phi_{\mathcal{P}}(J)) = d_{lvl}(I, J) = 0$$

and we are done.

If $I \neq J$ then we find $n \in \mathbb{N}$ such that $d_{lvl}(I, J) \leq \frac{1}{2^n}$. We will show that

$$d_{lvl}(\Phi_{\mathcal{P}}(I), \Phi_{\mathcal{P}}(J)) \leq \frac{1}{2^{n+1}},$$

i.e. for all ground atoms A with $lvl(A) \leq n$ we have $\Phi_{\mathcal{P}}(I)(A) = \Phi_{\mathcal{P}}(J)(A)$. Let's take some A with $lvl(A) \leq n$ and let $\mathcal{P}_A$ be the set of all clauses occurring in $g\mathcal{P}$ whose head is A. Because $\mathcal{P}$ is acyclic, for any clause $A \leftarrow L_1 \wedge \ldots \wedge L_m$ occurring in $\mathcal{P}_A$ we find for all $1 \leq i \leq m$ that $lvl(L_i) < lvl(A) \leq n$, where L_i is either a literal or a contextual literal. From $d_{lvl}(I, J) \leq \frac{1}{2^n}$ we conclude for all $1 \leq i \leq m$ that $I(L_i) = J(L_i)$. Therefore, I and J interprete identically all bodies of clauses in the definition of A. Consequently,

$$\Phi_{\mathcal{P}}(I)(A) = \Phi_{\mathcal{P}}(J)(A)$$

as desired. □

Proof of Theorem 27 This follows immediately from Theorem 30 by considering programs which are not contextual. □.

Corrollary 28 extends to contextual programs as well:

Corollary 31 *If a contextual program $\mathcal{P}$ is acyclic then $\Phi_{\mathcal{P}}$ has a unique fixed point which can be computed by iterating $\Phi_{\mathcal{P}}$ up to ω times starting with any interpretation.*

Proof The result follows from Theorems 30 and 9. □

However, as shown in the beginning of this subsection, we cannot conclude that the fixed point of the semantic operator is the least model of the weak completion of the contextual program.

Proposition 32 *If a contextual program $\mathcal{P}$ is acyclic then the unique fixed point of $\Phi_{\mathcal{P}}$ is a model of $wc\mathcal{P}$.*

Proof Let $I = \langle I^\top, I^\bot \rangle$ be the least fixed point of $\Phi_{\mathcal{P}}$ and $A \leftrightarrow F \in wc\mathcal{P}$, where F is the disjunction of all bodies of clauses in the definition of A. We distinguish three cases:

1. If $I(A) = \top$ then $A \in I^\top$. Hence, we find a clause $A \leftarrow \mathit{Body} \in g\mathcal{P}$ such that $I(\mathit{Body}) = \top$. As *Body* is one of the disjuncts occurring in F, $I(F) = \top$, and, consequently, $I(A \leftrightarrow F) = \top$.

2. If $I(A) = \bot$ then $A \in I^\bot$. Hence, we find a clause $A \leftarrow \mathit{Body} \in g\mathcal{P}$ and for all clauses of the form $A \leftarrow \mathit{Body} \in g\mathcal{P}$ we find $I(\mathit{Body}) = \bot$. Because F is the disjunction of all bodies we conclude $I(F) = \bot$. Consequently, $I(A \leftarrow F) = \top$.

3. If $I(A) = \mathsf{U}$ then $A \not\in I^\top \cup I^\bot$. Because $A \leftrightarrow F$ occurs in $wc\mathcal{P}$, we find a clause of the form $A \leftarrow \mathit{Body} \in g\mathcal{P}$. Because $A \not\in I^\top$, we conclude that for each clause of the form $A \leftarrow \mathit{Body} \in g\mathcal{P}$ it cannot be the case that $I(\mathit{Body}) = \top$. Because $A \not\in I^\bot$, we must find a clause of the form $A \leftarrow \mathit{Body} \in g\mathcal{P}$ such that $I(\mathit{Body}) \neq \bot$. Together with $I(\mathit{Body}) \neq \top$ we learn that $I(\mathit{Body}) = \mathsf{U}$. Because F is the disjunction of all bodies in the definition of A we conclude $I(F) = \mathsf{U}$ and, consequently, $I(A \leftarrow F) = \top$. □

Open Problem Is the unique fixed point of $\Phi_{\mathcal{P}}$ a minimal model of $wc\mathcal{P}$?

4.5.3 Contextual Abduction

4.6 Ethical Decision Problems

We will consider different trolley problems [23]: the bystander, the footbridge, the loop, the loop-push, the man-in-front, and the collapse-bridgecase. All cases are taken from [58] with some minor adaptations.

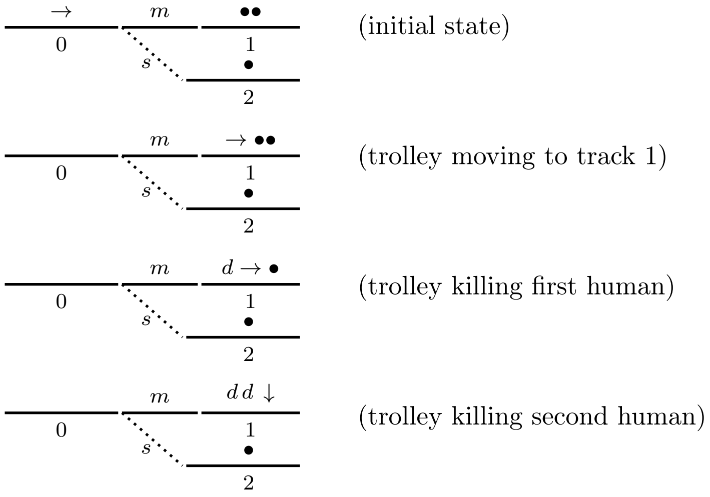

→
m
●●
0
1
s
●
2
(initial state)
m
→ ●●
0
1
s
●
2
(trolley moving to track 1)
m
d → ●
0
1
s
●
2
(trolley killing first human)
m
d d ↓
0
1
s
●
2
(trolley killing second human)

Figure 4.2: The bystander case (initial state) and its ramifications if Hank decides to do nothing, where ↓ denotes that no further action is applicable.

4.6.1 The Bystander

A trolley whose conductor has fainted is headed towards two people walking on the main track.[3] *The banks of the track are so steep that these two people will not be able to get off the track in time. Hank is standing next to a switch which can turn the trolley onto a side track, thereby preventing it from killing the two people. However, there is a man standing on the side track. Hank can change the switch, killing him. Or he can refrain from doing so, letting the two die. Is it morally permissible for Hank to change the switch?*

The case is illustrated in Figure 4.2 (initial state). The tracks are divided into segments 0, 1, and 2, the arrow represents that the trolley t is moving forward and that the track is clear (c), the switch is in position m (main) but can be changed into position s (side), and a bullet above a track segment represents a human (h) on this track. t, c, and h may be indexed to denote the track to which they apply. In addition, we need a fluent d denoting a dead human.

We choose to represent a state by a pair of multisets consisting of the casualties in its second element and all other fluents in its first element. Multisets are represented by so-called *fluent terms* in the fluent calculus, i.e. the initial state of the bystander case is the pair

$$(t_0 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2, 1) \tag{4.39}$$

[3] Note that in the original trolley problem, five people are on the main track. For the sake of simplicity, we assume that only two people are on the main track.

of fluent terms. The casualties are represented in the second element of (4.39) by the constant 1 encoding the empty multiset. Initially, there are no casualties, but casualties will play a special role when preferring one action over another as will be discussed later in this section. The first element of (4.39) encodes the multiset

$$\dot{\{}t_0, c_0, m, h_1, h_1, h_2\dot{\}}.$$

There are two kinds of actions, the ones which can be performed by Hank (the direct actions *donothing* and *change*), and the actions which are performed by the trolley (the indirect actions *downhill* and *kill*). We will represent the actions by the trolley explicitly with the help of a five-place relation symbol *action* specifying the preconditions, the name, and the immediate effects of an action. As a state is represented by two multisets, the preconditions and the immediate effects have also two parts:

$$\begin{aligned} \mathit{action}(t_0 \circ c_0 \circ m, 1, \mathit{downhill}, t_1 \circ c_0 \circ m, 1) &\leftarrow \top &(4.40)\\ \mathit{action}(t_0 \circ c_0 \circ s, 1, \mathit{downhill}, t_2 \circ c_0 \circ s, 1) &\leftarrow \top &(4.41)\end{aligned}$$

$$\begin{aligned} \mathit{action}(t_1 \circ h_1, 1, \mathit{kill}, t_1, d) &\leftarrow \top &(4.42)\\ \mathit{action}(t_2 \circ h_2, 1, \mathit{kill}, t_2, d) &\leftarrow \top &(4.43)\end{aligned}$$

If the trolley is on track 0, this track is clear, and the switch is in position m then it will run downhill onto track 1 whereas track 0 remains clear and the switch will remain in position m; if, however, the switch is in position s, the trolley will run downhill onto track 2. If the trolley is on either track 1 or 2 and there is a human on this track, it will kill the human leading to a casualty.

In the original version of the fluent calculus, causality is expressed by the ternary predicate *causes* stating that the execution of a plan transfers an initial into a goal state (see Section 2.4). Its base case is of the form

$$\mathit{causes}(X, [\,], X),$$

i.e. an empty plan does not change any state X. Generating models bottom up using a semantic operator one has to consider all ground instances of this atom. This set is usually too large to consider it as a base case for human reasoning episodes. The solution presented herein overcomes this problem in that we only have a small number of base cases depending on the number of options an agent like Hank may consider.

In fact, we are not going to solve planning problems like whether there exists a plan such that its execution transforms the intial state (4.39) into a goal state meeting certain constraints. Rather we want to compare the outcomes, i.e. the indirect effects, of the actions Hank can possibly perform. In other words, we want to compare the ramifications of either doing nothing or throwing the switch in the bystander scenario.

To this end, we will use a binary relation symbol *ramify* whose first argument is the name of an action and whose second and third arguments are the state obtained when executing the action. The possible actions of Hank are the base cases in the definition of *ramify*:

$$\begin{aligned} \mathit{ramify}(\mathit{donothing}, t_0 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2, 1) &\leftarrow \top && (4.44)\\ \mathit{ramify}(\mathit{change}, t_0 \circ c_0 \circ s \circ h_1 \circ h_1 \circ h_2, 1) &\leftarrow \top && (4.45)\end{aligned}$$

Further actions can be applied to the second argument of *ramify* given the actions specified in (4.40)

$$\begin{aligned} \mathit{ramify}(A, E_1 \circ Z_1, E_2 \circ Z_2) \leftarrow\ & \mathit{action}(P_1, P_2, A', E_1, E_2) \wedge && (4.46)\\ & \mathit{ramify}(A, P_1 \circ Z_1, P_2 \circ Z_2) \wedge \\ & \neg ab_{\mathit{ramify}}\, A'. \end{aligned}$$

It checks whether in a given state $(P_1 \circ Z_1, P_2 \circ Z_2)$ an action A' is applicable, which is the case if the preconditions (P_1, P_2) are contained in the given state. If this holds then the action is executed leading to the successor state $(E_1 \circ Z_1, E_2 \circ Z_2)$, where (E_1, E_2) are the direct effects of the action A'. In other words, if an action is applied then its preconditions are consumed and its direct effects are produced. Such an action application is considered to be a ramification [66] with respect to the initial, direct action performed by Hank. Hence, the first argument A of *ramify* is not changed.

The execution of an action is also conditioned by $\neg ab_{\mathit{ramify}}\, A'$, where ab_{ramify} is an abnormality predicate. Such abnormalities were introduced in [61] to represent conditionals as licenses for inference. In this example, there is nothing abnormal known with respect to the actions *downhill* and *kill* and, consequently, the assumptions

$$\begin{aligned} ab_{\mathit{ramify}}\, \mathit{downhill} &\leftarrow \bot, && (4.47)\\ ab_{\mathit{ramify}}\, \mathit{kill} &\leftarrow \bot && (4.48)\end{aligned}$$

are added to the program (4.46). But we can imagine situations, where the trolley will only cross the switch if the switch is not broken. If the switch is broken, the trolley may derail. Such a scenario can be modeled using the techniques presented in Sections 3.5 and 4.5.

Let

$$\mathcal{P}_0 = \{(4.40), (4.41), (4.42), (4.43), (4.46), (4.47), (4.48)\}$$

and consider the AC1-equational theory (2.1). Hank has the choice to do nothing or to change the switch. The indirect effects of his decision are computed as ramifications in the fluent calculus [66].

If Hank is doing nothing then let

$$\mathcal{P}_1 = \mathcal{P}_0 \cup \{(4.44)\}$$

The least model of the weak completion of $\mathcal{P}_1$ – which is equal to the least fixed point of $\Phi_{\mathcal{P}_1}$ – is computed by iterating $\Phi_{\mathcal{P}_1}$ starting with the empty interpretation $\langle\emptyset, \emptyset\rangle$. The following equivalence classes will be mapped to true in subsequent steps of the iteration:

$$\begin{aligned} &[\mathit{ramify}(\mathit{donothing}, t_0 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2, 1)],\\ &[\mathit{ramify}(\mathit{donothing}, t_1 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2, 1)],\\ &[\mathit{ramify}(\mathit{donothing}, t_1 \circ c_0 \circ m \circ h_1 \circ h_2, d)],\\ &[\mathit{ramify}(\mathit{donothing}, t_1 \circ c_0 \circ m \circ h_2, d \circ d)]. \end{aligned}$$

$\Phi_{\mathcal{P}'_1}$	$I^\top$	$I^\bot$
1	$[\mathit{ramify}(\mathit{donothing}, t_0 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2, 1)]$ $[\mathit{action}(t_0 \circ c_0 \circ m, 1, \mathit{downhill}, t_1 \circ c_0 \circ m, 1)]$ $[\mathit{action}(t_0 \circ c_0 \circ s, 1, \mathit{downhill}, t_2 \circ c_0 \circ s, 1)]$ $[\mathit{action}(t_1 \circ h_1, 1, \mathit{kill}, t_1, d)]$ $[\mathit{action}(t_2 \circ h_2, 1, \mathit{kill}, t_2, d)]$	$[\mathit{ab}_{\mathit{ramify}}\ \mathit{downhill}]$ $[\mathit{ab}_{\mathit{ramify}}\ \mathit{kill}]$
2	$[\mathit{ramify}(\mathit{donothing}, t_1 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2, 1)]$ $[\mathit{aa}(\mathit{donothing}, t_0 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2, 1)]$	
3	$[\mathit{ramify}(\mathit{donothing}, t_1 \circ c_0 \circ m \circ h_1 \circ h_2, d)]$ $[\mathit{aa}(\mathit{donothing}, t_1 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2, 1)]$	
4	$[\mathit{ramify}(\mathit{donothing}, t_1 \circ c_0 \circ m \circ h_2, d \circ d)]$ $[\mathit{aa}(\mathit{donothing}, t_1 \circ c_0 \circ m \circ h_1 \circ h_2, d)]$	

Table 4.4: The computation of the least model of $wc\mathcal{P}'_1$, i.e. the program obtained if Hank is doing nothing. In each step, only the atoms are listed which are newly added.

They correspond precisely to the four states shown in Figure 4.2. No further action is applicable to the elements of the final congruence class. The two people on the main track will be killed.

But one problem remains: We need to identify the nstance of the *ramify* predicate to which no further action is applicable. To this end we specify

$$\begin{array}{rcl} \mathit{aa}(A, P_1 \circ Z_1, P_2 \circ Z_2) & \leftarrow & \mathit{action}(P_1, P_2, A', E_1, E_2)\ \wedge \\ & & \mathit{ramify}(A, P_1 \circ Z_1, P_2 \circ Z_2)\ \wedge \\ & & \neg \mathit{ab}_{\mathit{ramify}}\ A'. \end{array} \tag{4.49}$$

Informally, $aa(A, X_1, X_2)$ is true if there is an action A' which is applicable in the state (X_1, X_2). Comparing (4.56) and (4.46) we find that the bodies of these rules are identical. Thus, whenever a truth value is assigned to the head of (4.46), the same truth value will be assigned to the corresponding head of (4.56). Formally, let

$$\mathcal{P}'_1 = \mathcal{P}_1 \cup \{(4.56)\}.$$

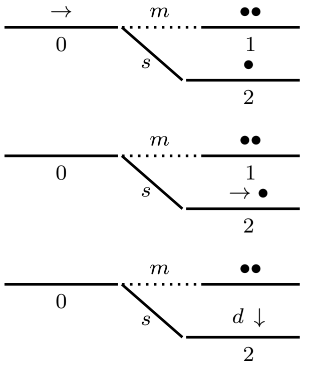
The computation of least fixed point of $\Phi_{\mathcal{P}'_1}$ is shown in Table 4.4.

Let us return to Hank's choices. If Hank is changing the switch then let

$$\mathcal{P}'_2 = \mathcal{P}_0 \cup \{(4.45),\ (4.56)\}.$$

The least fixed point of $\Phi_{\mathcal{P}'_2}$ contains

$$[\mathit{ramify}(\mathit{change}, t_2 \circ c_0 \circ s \circ h_1 \circ h_1, d)].$$



(initial state)

(trolley moving to track 2)

(trolley killing human on track 2)

Figure 4.3: The bystander case (initial state) and its ramifications if Hank decides to change the switch. One should observe that now the switch points to the side track.

The two people on the main track will be saved but the person on the side track will be killed. This case is illustrated in Figure 4.3.

The two cases can be compared by means of a *prefer* rule:

$$\begin{aligned} \mathit{prefer}(A_1, A_2) &\leftarrow \mathit{ramify}(A_1, Z_1, D_1) \ \wedge \\ & \neg \mathit{ctxt}\,\mathit{aa}(A_1, Z_1, D_1) \ \wedge \\ & \mathit{ramify}(A_2, Z_2, D_1 \circ d \circ D_2) \ \wedge \\ & \neg \mathit{ctxt}\,\mathit{aa}(A_2, Z_2, D_1 \circ d \circ D_2) \ \wedge \\ & \neg \mathit{ab}_{\mathit{prefer}}\, A_1, \\ \mathit{ab}_{\mathit{prefer}}\, \mathit{change} &\leftarrow \bot, \\ \mathit{ab}_{\mathit{prefer}}\, \mathit{donothing} &\leftarrow \bot. \end{aligned} \tag{4.50}$$

prefer only compares states, to which no further action is applicable. In the bystander case these are the states

$$(t_1 \circ c_0 \circ m \circ h_2, d \circ d)$$

and

$$(t_2 \circ c_0 \circ s \circ h_1 \circ h_1, d)$$

They can be identified in the least fixed points of $\Phi_{\mathcal{P}_1'}$ and $\Phi_{\mathcal{P}_2'}$ because there is no corresponding tuple of the *aa* relation. The *ctxt* operator will map these unknowabilities to false and the negations thereof will be mapped to true. Comparing D_1 and $D_1 \circ d \circ D_2$, action A_2 leads to at least one more dead person than action A_1. Hence, A_1 is preferred over A_2 if nothing abnormal is known about A_1.

Under an *utilitarian* point of view [8], the *change* action is preferable to the *donothing* action as it will kill fewer humans. On the other hand, we know that a purely utilitarian

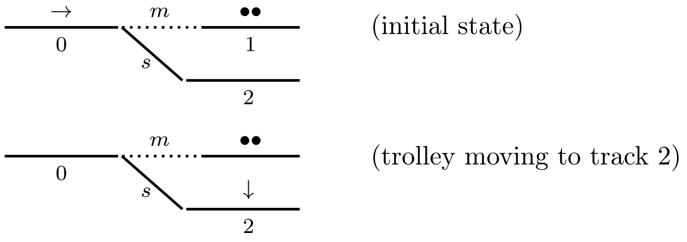


Figure 4.4: The bystander case (initial state) and its ramifications if Hank is considering the counterfactual.

view is impossible in case of human casualties. Hank may ask himself: *Would I still save the humans on the main track if there were no human on the side track and I changed the switch?* This is a counterfactual because its antecedent is false in the current state. But we can easily deal with it by starting a new computation with the additional fact

$$\mathit{ramify}(\mathit{change}, t_0 \circ c_0 \circ s \circ h_1 \circ h_1 \circ c_2, 1) \;\leftarrow\; \top. \tag{4.51}$$

Comparing (2) and the second fact in the program (4.44), h_2 has been replaced by c_2. There is no human on track 2 anymore and, hence, this track is clear. This is a minimal change necessary to satisfy the precondition of the counterfactual. In this case, the least model of the extended program will contain

$$[\mathit{ramify}(\mathit{change}, t_0 \circ c_0 \circ s \circ h_1 \circ h_1 \circ c_2, 1)].$$

This case is illustrated in Figure 4.4. Using

$$\begin{array}{rcl} \mathit{perm\_double}\ \mathit{change} & \leftarrow & \mathit{prefer}(\mathit{change}, \mathit{donothing}) \ \wedge \\ & & \mathit{ramify}(\mathit{change}, t_2 \circ c_0 \circ s \circ h_1 \circ h_1 \circ c_2, 1) \ \wedge \\ & & \neg \mathit{ctxt}\,\mathit{aa}(\mathit{change}, t_2 \circ c_0 \circ s \circ h_1 \circ h_1 \circ c_2, 1) \ \wedge \\ & & \neg \mathit{ab}_{\mathit{perm\_double}}\ \mathit{change}, \\ \mathit{ab}_{\mathit{perm\_double}}\ \mathit{change} & \leftarrow & \bot \end{array} \tag{4.52}$$

allows Hank to conclude that changing the switch is permissible within the doctrine of double effect [5].

4.6.2 The Footbridge

The case is similar to the bystander case except that instead of the switch a footbridge is crossing the main track. Ian is standing on the footbridge next to a heavy human, which he can throw on the track in the path of the trolley to stop it. Is it morally permissible for Ian to throw the human down?

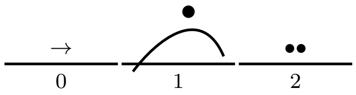


Figure 4.5: The footbridge case.

This case is illustrated in Figure 4.5. The track is again segmented. We use b_1 to denote that there is a heavy human on the footbridge crossing segment 1 of the track. Ian has two possibilities: *donothing* and *throw*. They are represented as the base cases in the definition of *ramify*:

$$\begin{array}{rcl} \mathit{ramify}(\mathit{donothing}, t_0 \circ c_0 \circ c_1 \circ b_1 \circ h_2 \circ h_2, 1) & \leftarrow & \top, \\ \mathit{ramify}(\mathit{throw}, t_0 \circ c_0 \circ h_2 \circ h_2, d) & \leftarrow & \top. \end{array} \tag{4.53}$$

One should observe that in the case of *donothing* track 1 is clear (c_1), whereas this does not hold if Ian has decided to throw down the heavy human. In the latter case, a dead body is blocking track 1.

As in the footbridge case, one is tempted to reason that the *throw* action is preferable to the *donothing* action as it will kill fewer humans. But throwing down a heavy human involves an intentional direct kill, and intentional kills are not allowed under the doctrine of double effect. This can be modeled with the help of the abnormality predicate ab_{prefer} by adding

$$\begin{array}{rcl} ab_{prefer}\ \mathit{throw} & \leftarrow & \bot, \\ ab_{prefer}\ \mathit{throw} & \leftarrow & \mathit{intent\_direct\_kill}\ \mathit{throw}, \\ \mathit{intent\_direct\_kill}\ \mathit{throw} & \leftarrow & \top \end{array} \tag{4.54}$$

to the program (4.50) Hence, throwing down the heavy human is not preferred and, thus, not permissible. The example demonstrates again the way abnormalities are used in the *Weak Completion Semantics*. If nothing is known then a negative assumption about the abnormality is made by the first clause occurring in (4.50). This assumption can be overridden once additional knowledge becomes available. In this case we learn that an intentional direct kill overrides the negative assumption, which is expressed in the second clause occurring in (4.50). Moreover, from the specification of the *throw* action we can derive that the killing of the heavy human was intentional as it is a direct effect of this action, which leads to the third clause occurring in (4.50).

4.6.3 The Loop

The case is similar to the bystander case. Ned is standing next to a switch which he can change that will temporarily turn the trolley onto a loop side track. There is a heavy human on the side track. If the trolley hits the heavy human then this will slow down the trolley, giving the two people on the main track sufficient time to escape. But it will kill the heavy human. Is it morally permissible for Ned to throw the switch?

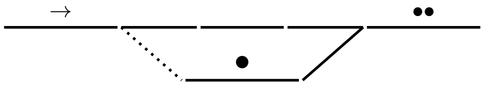


Figure 4.6: The loop case.

This case is illustrated in Figure 4.6. Ned can reason that if he does nothing then the humans on the main track will be killed. Likewise, if he changes the switch then the humans on the main track will be saved whereas the human on the side track will be killed. But the counterfactual *if there were no human on the side and he changes the switch then he would still save the humans on the main track* will be false. Hence, according to the doctrine of double effect changing the switch is not permissible. However, the doctrine of triple effect [39] allows to distinguish between direct and indirect intentional kills such that the *change* action becomes permissible under the doctrine of triple effects.

This example can also be modeled under the *Weak Completion Semantics.* Because killing a human is not a direct effect of the *change* action we may add

$$\begin{aligned} \mathit{ab}_{\mathit{prefer}}\ \mathit{change} &\leftarrow \mathit{intent\_direct\_kill}\ \mathit{change}, \\ \mathit{intent\_direct\_kill}\ \mathit{change} &\leftarrow \bot \end{aligned} \tag{4.55}$$

to the program (4.50). Consequently, the *change* action will be preferred over the *donothing* action. A properly revised definition for permissibility will allow Ned to conclude that changing the switch is permissible under the doctrine of triple effect:

$$\begin{aligned} \mathit{perm\_triple}\ \mathit{change} &\leftarrow \mathit{prefer}(\mathit{change}, \mathit{donothing})\ \wedge \\ &\qquad \neg\, \mathit{intent\_direct\_kill}\ \mathit{change} \\ &\qquad \neg\, \mathit{ab}_{\mathit{perm\_triple}}\ \mathit{change}, \\ \mathit{ab}_{\mathit{perm\_triple}}\ \mathit{change} &\leftarrow \bot \end{aligned} \tag{4.56}$$

4.6.4 The Loop and a Push

The loop-push case is a variant of the loop case in that *besides changing the switch, a heavy human has to be pushed on the looping side track in order to save the humans on the main track* (see Figure 4.7). Thus, a direct intentional kill is needed to stop the trolley and, consequently, neither the doctrine of double effect nor the doctrine of triple effect permit the *change* action.

4.6.5 Man in Front

The man-in-front case is another variant of the loop case in that *a heavy object is blocking the sidetrack behind the heavy human. If the trolley hits the heavy object, it will stop* (see Figure 4.8). Hence, the killing of the heavy human is no longer intended in order to save

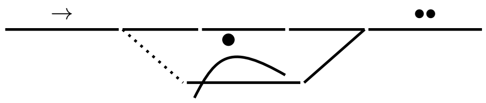


Figure 4.7: The loop-push case.

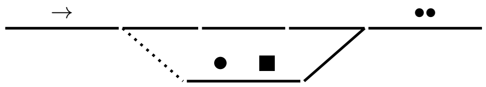


Figure 4.8: The loop-push case.

the humans on the main track and the *chance* action is permissible under the doctrines of double and triple effect.

4.6.6 The Collapsing Bridge

The collapse-bridge case is a variant of the footbridge case. *Instead of throwing the heavy human from the bridge, the bridge is collapsed in its entirety. This places the heavy human and the debris of he bridge on the track, effectively stopping the trolley* (see Figure 4.9). Hence, the killing of the heavy human is not intentional and the collapse of the bridge becomes permissible under the doctrines of double and triple effect.

4.6.7 Summary

Table 4.5 gives a summary according to which view which action is permissible for each case.

4.6.8 Fluent Matching Problems

Let us discuss the computation of the least fixed point of the semantic operator $\Phi_{\mathcal{P}'_1}$ shown in Table 4.4 in more detail.

$$\Phi_{\mathcal{P}'_1}(\langle\emptyset,\emptyset\rangle) = \Phi_{\mathcal{P}'_1}\!\uparrow 1 = \langle I_1^\top, I_1^\bot\rangle,$$

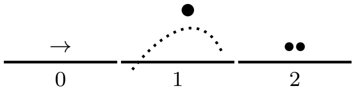


Figure 4.9: The collapse-bridge case.

	Bystander	Loop	Footbridge	Loop-Push	Man-in-Front	Collapse
Double Effect	*change*	-	-	-	*change*	*collapse*
Triple Effect	*change*	*change*	-	-	*change*	*collapse*
Utilitarianism	*change*	*change*	*throw*	*change throw*	*change*	*collapse*

Table 4.5: The six cases and the permissible actions according to the different views.

where

$$\begin{aligned} I_1^\top = \{ \; & [\mathit{ramify}(\mathit{donothing}, t_0 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2, 1)], \\ & [\mathit{action}(t_0 \circ c_0 \circ m, 1, \mathit{downhill}, t_1 \circ c_0 \circ m, 1)], \\ & [\mathit{action}(t_0 \circ c_0 \circ s, 1, \mathit{downhill}, t_2 \circ c_0 \circ s, 1)], \\ & [\mathit{action}(t_1 \circ h_1, 1, \mathit{kill}, t_1, d)], \\ & [\mathit{action}(t_2 \circ h_2, 1, \mathit{kill}, t_2, d)] \; \}, \\ I_1^\bot = \{ \; & [\mathit{ab}_{\mathit{ramify}} \; \mathit{downhill}], \\ & [\mathit{ab}_{\mathit{ramify}} \; \mathit{kill}] \; \}. \end{aligned}$$

Considering the body of (4.46) we find that both possible ground instances of $\mathit{ab}_{\mathit{ramify}}\, A'$, viz. $\mathit{ab}_{\mathit{ramify}}\, \mathit{downhill}$ and $\mathit{ab}_{\mathit{ramify}}\, \mathit{kill}$, are false under $\Phi_{\mathcal{P}_1'} \uparrow 1$ and their negations are true under $\Phi_{\mathcal{P}_1'} \uparrow 1$. The only ground instance of

$$\mathit{ramify}(A, P_1 \circ Z_1, P_2 \circ Z_2) \tag{4.57}$$

being true under $\Phi_{\mathcal{P}_1'} \uparrow 1$ is

$$\mathit{ramify}(\mathit{donothing}, t_0 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2, 1). \tag{4.58}$$

Hence, we are searching for a ground instance of

$$\mathit{action}(P_1, P_2, A', E_1, E_2)$$

being true under $\Phi_{\mathcal{P}_1'} \uparrow 1$ such that

- the ground instance of P_1 is contained in $t_0 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2$ and
- the ground instance of P_2 is contained in 1.

There are four candidates in $\Phi_{\mathcal{P}_1'} \uparrow 1$. The only possible ground instance of an action meeting the conditions is

$$\mathit{action}(t_0 \circ c_0 \circ m, 1, \mathit{downhill}, t_1 \circ c_0 \circ m, 1). \tag{4.59}$$

Comparing the second arguments of (4.57) and (4.58) with the first argument of (4.59) we find that

$$P_1 = t_0 \circ c_0 \circ m \quad \text{and} \quad Z_1 = h_1 \circ h_1 \circ h_2.$$

Likewise, comparing the third arguments of (4.57) and (4.58) with the second argument of (4.59) we find that

$$P_2 = 1 \quad \text{and} \quad Z_2 = 1.$$

Combining Z_1 with the fourth argument of (4.59) and, likewise, combining Z_2 with the fifth argument of (4.59) we learn that

$$\mathit{ramify}(\mathit{donothing}, t_1 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2, 1)$$

must be true under $\Phi_{\mathcal{P}'_1}\!\uparrow\!2$.

Likewise, we can compute that

$$[\mathit{ramify}(\mathit{donothing}, t_1 \circ c_0 \circ m \circ h_1 \circ h_2, d)]$$

must be true under $\Phi_{\mathcal{P}'_1}\!\uparrow\!3$ and

$$[\mathit{ramify}(\mathit{donothing}, t_1 \circ c_0 \circ m \circ h_2, d \circ d)]$$

must be true under $\Phi_{\mathcal{P}'_1}\!\uparrow\!4$.

Hence, in order to compute the semantic operator we have to solve AC1-matching problems of the form

$$t_0 \circ c_0 \circ m \circ Z_1 =_{AC1} t_0 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2 \quad \text{and} \quad 1 \circ Z_2 =_{AC1} 1$$

or of the form

$$t_0 \circ c_0 \circ s \circ Z_1 =_{AC1} t_0 \circ c_0 \circ m \circ h_1 \circ h_1 \circ h_2 \quad \text{and} \quad 1 \circ Z_2 =_{AC1} 1.$$

Whereas the latter has no solution, the former does. In general, we need to solve so-called *fluent matching problems* of the form

$$s \circ Z =_{AC1} t \tag{4.60}$$

where s and t are ground fluent terms and Z is a variable.

Such problems have been considered in [62, 36], where s was a fluent term. It was shown that fluent matching is decidable, finitary, and there always exists a minimal and complete set of matchers. The fluent matching algorithm presented in [62, 36] can be easily adapted to the fact that s is ground:

```
1   If $s =_{AC1} 1$ then return $\{Z \mapsto t\}$.
2   Don't care non-deterministically select a fluent $u$ occurring in $s$
    and remove $u$ from $s$.
3   If $u$ occurs in $t$ then delete $u$ from $t$ and goto 1,
    else stop with failure.
```

Hence, with s being a ground fluent term, fluent matching becomes unitary.

Using the correspondence between fluent terms and multisets, let $\mathcal{S}$ and $\mathcal{T}$ be the multisets corresponding to the fluent terms s and t. Then, the fluent matching problem (4.60) has a solution iff

$$\mathcal{S} \dot{\subseteq} \mathcal{T}.$$

If (4.60) has a solution then Z is mapped onto the fluent term corresponding to

$$\mathcal{T} \dot{\setminus} \mathcal{S}.$$

4.6.9 On the Adequateness of the Approach

Do humans reason with $AC1$-matchers in the limited form described in the previous section? Obviously, the multisets should not be large as there is compelling evidence that humans cannot deal with many different objects at one time [50]. In the trolley problems discussed in this paper the maximal number of fluents was six. Even if we increase the number of humans on the main track to five as in the original version of the bystander case [23], the size of the multisets becomes only nine. Moreover, the actions did not increase the number of fluents in that the number of immediate effects was always equal to the number of preconditions.

In Germany small children in the Kindergarden are asked to solve puzzles of the following form. Given several fruits like, for example, four apples and three peas, they are asked how many pieces are left after they would give some, say, two apples and one pea, away. The puzzles are presented in pictures. In most cases, the children are crossing out the pieces given away and, afterwards, are counting the remaining ones. In other words, they seem to solve exactly the AC1-matching problems discussed in the previous section. But to the best of our knowledge, there are almost no experimental data on how humans deal with multisets (see e.g. [53, 28]). Hence, we hypothesize that humans can solve such matching problems although we must be careful as the ethical decision problems considered herein are more abstract than the puzzles solved by the children and it is well-known that humans solve less abstract problems differently than abstract ones (see e.g. [51, 67, 25]). Thus, the hypothesis must be experimentally tested.

Chapter 5

A Connectionist Realization

where we develop a connectionist model for the Weak Completion Semantics.

5.1 Representing Interpretations

5.2 Computung Semantic Operators

5.3 Computing Possible Explanations

5.4 Computing Skeptical Abduction

Chapter 6

Outlook

where we discuss some open problems.

6.1 Reasoning Towards a Program

6.2 Bounded Skeptical Abduction

6.3 Sequence Matters

6.4 Counterfactuals with Unknown Antecedents

Let us consider a variant of the forest fire example discussed in Section 4.1.6 where the background knowledge has been slightly changed:

> *If it is not raining then lightning may cause a forest fire. Lightning happened. If it is not raining then the leaves are dry. The absence of dry leaves is an abnormality.*

The background knowledge can be encoded in a program consisting of the following clauses:

$$\begin{array}{rcl} \mathit{forestfire} & \leftarrow & \mathit{lightning} \wedge \neg \mathit{rain} \wedge \neg \mathit{ab}_\ell, \\ \mathit{ab}_\ell & \leftarrow & \neg \mathit{dryleaves}, \\ \mathit{lightning} & \leftarrow & \top, \\ \mathit{dryleaves} & \leftarrow & \neg \mathit{rain}. \end{array} \tag{6.1}$$

The least model of this program is

$$\mathcal{M}_{(6.1)} = \langle \{\mathit{lightning}\}, \emptyset \rangle.$$

The atom *rain* is undefined and, hence, the set of abducibles is

$$\{\mathit{rain} \leftarrow \top, \mathit{rain} \leftarrow \bot\}.$$

Consider again the conditional

> *if there had not been so many dry leaves on the forest floor then the forest fire would not have occurred*

or

$$\neg \mathit{dryleaves} \Rightarrow \neg \mathit{forestfire}.$$

In addition, suppose we were told (or a speech analysis tool suggests) that the conditional is a counterfactual. Its antecedent $\neg \mathit{dryleaves}$ is unknown under $\mathcal{M}_{(6.1)}$.

There are at least two different ways to proceed in the evaluation of a counterfactual $\mathcal{C} \Rightarrow \mathcal{D}$ whose antecedent is unknown with respect to the background knowledge:

1. We can apply the procedure *minimal revision followed by abduction* to map $\mathcal{C}$ to true and evaluate the counterfactual with respect to the modified program. One should observe that the modified program is a monotonic extension of the original program.
2. We can apply the procedure *minimal revision followed by abduction* to map $\mathcal{C}$ to false and, thereafter, (non-monotonically) revise the modified program to force $\mathcal{C}$ to be true.

Let us investigate the outcome given the scenario mentioned above. For the first option, $\neg \mathit{dryleaves}$ is explained by the minimal explanation

$$\mathit{rain} \leftarrow \top. \tag{6.2}$$

Adding the explanation to the program (6.1) we obtain

$$\mathcal{M}_{(6.1)\cup(6.2)} = \langle \{\mathit{lightning}, \mathit{rain}, \mathit{ab}_\ell\}, \{\mathit{forestfire}, \mathit{dryleaves}\} \rangle.$$

Under this interpretation the counterfactual is true.

For the second option we need to map $\neg \mathit{dryleaves}$ to false in a first step. This will happen if we abduce

$$\mathit{rain} \leftarrow \bot. \tag{6.3}$$

Adding this explanation to the program (6.1) we obtain

$$\mathcal{M}_{(6.1)\cup(6.3)} = \langle \{\mathit{lightning}, \mathit{dryleaves}, \mathit{forestfire}\}, \{\mathit{ab}_\ell, \mathit{rain}\} \rangle.$$

Under this interpretation the antecedent $\neg \mathit{dryleaves}$ of the counterfactual is false and we revise the program (6.1) with respect to $\neg \mathit{dryleaves}$ to obtain

$$\mathcal{M}_{\mathit{rev}((6.1)\cup(6.3),\neg \mathit{dryleaves})} = \langle \{\mathit{lightning}, \mathit{ab}_\ell\}, \{\mathit{forestfire}, \mathit{dryleaves}, \mathit{rain}\} \rangle.$$

Under this interpretation the counterfactual is still true, but the model of the revised program differs from $\mathcal{M}_{(6.1)\cup(6.2)}$ in the value for the atom *rain*. Hence, the conditional

$$\neg \mathit{dryleaves} \Rightarrow \mathit{rain}$$

will be true under the first option but false under the second one.

How shall we evaluate counterfactuals with unknown antecedent?

6.5 Unknown Conclusions

In the forest fire example discussed in Section 4.1.6 we have evaluated the conditional

> *If there had not been so many dry leaves on the forest floor then the forest fire would not have occurred*

with respect to the background knowledge that lightning as well as arson may cause forest fire. In the final case considered, the conditional was evaluated to unknown as *forestfire* was unknown.

We could apply abduction in case of unknown consequences of conditionals. Given the program (4.12) ∪ (4.13) the only undefined atom is *arson*. Assuming that the background knowledge was correct, the set of abducibles is

$$\{\mathit{arson} \leftarrow \top, \mathit{arson} \leftarrow \bot\}.$$

The only minimal explanation for $\neg$*forestfire* is

$$\mathit{arson} \leftarrow \bot.$$

Adding this explanation to the program will make the counterfactual true under the condition that *no arson has taken place.*

This corresponds to evaluating the extended conditional

> *If there had not been so many dry leaves on the forest floor and no arson has taken place then the forest fire would not have occurred*

under the procedure *minimal revision followed by abduction* discussed in Section 4.1.2.

How shall we evaluate conditionals with unknown conclusion?

6.6 The Moral Machine Experiment

[6]

6.7 The Need for Experimental Data

In [15] seperate inference rules for abduction and revision were defined. These rules were applied in cases, when the antecedent of a conditional was unknown. Moreover, the rules could be applied in any order. As example the firing squad case of Section 4.1.5 was discussed. In particular, it was shown that depending on the order in which these rules are applied the conditional

> *if the captain gave no signal and rifleman a decides to shoot then the court did not order an execution*

may be either true, false, or unknown. Unfortunately, Judae Pearl, who discussed this example in [54] did not report any experimental data to see how humans evaluate this and the other conditionals discussed in Section 4.1.5. But only experiments could tell us, in which direction the *Weak Completion Semantics* shall be developed. For the time being we hypothesize that the procedure *minimal revision followed by abduction* discussed in Section 4.1.2 shall be applied. But this needs to be tested.

6.8 Connectionist Reasoning and Learning

Bibliography

[1] E. W. Adams. Subjunctive and indicative conditionals. *Foundations of Language*, 6(1):89–94, 1970.

[2] K. R. Apt. *From Logic to Logic Programming.* Prentice Hall, London, 1997.

[3] K. R. Apt and M. Bezem. Acyclic programs (extended abstract). In *Proceedings of the International Conference on Logic Programming*, pages 617–633, 1990.

[4] K. R. Apt and M. H. van Emden. Contributions to the theory of logic programming. *Journal of the ACM*, 29:841–862, 1982.

[5] T. Aquinas. Summa Theologica II-II, q. 64, art. 7, âĂIJOf KillingâĂİ. In W. P. Baumgarth and R.J. Regan, editors, *On Law, Morality, and Politics*, pages 226–227. Hackett Publishing Co., Indianapolis, 1988.

[6] E. Awad, S. Dsouzza, R. Kim, J. Schulz, J. Henrich, A. Shariff, J.-F. Bonnefon, and I. Rahwan. The moral machine experiment. *Nature*, 563:59–64, 2018.

[7] S. Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fund. Math.*, 3:133–181, 1922.

[8] J. Bentham. *An Introduction to the Principles of Morals and Legislation.* Dover Publications Inc., 2009.

[9] W. Bibel. A deductive solution for plan generation. *New Generation Computing*, 4:115–132, 1986.

[10] R. M. J. Byrne. *The Rational Imagination: How People Create Alternatives to Reality.* MIT Press, Cambridge, MA, USA, 2005.

[11] R. M. J. Byrne. Suppressing valid inferences with conditionals. *Cognition*, 31:61–83, 1989.

[12] K. L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Databases*, pages 293–322. Plenum, New York, 1978.

[13] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order.* Cambridge University Press, 2nd edition, 2002.

[14] E.-A. Dietz and S. Hölldobler. A new computational logic approach to reason with conditionals. In F. Calimeri, G. Ianni, and M. Truszczynski, editors, *Logic Programming and Nonmonotonic Reasoning, 13th International Conference, LPNMR*, volume 9345 of *Lecture Notes in Artificial Intelligence*, pages 265–278. Springer, 2015.

[15] E.-A. Dietz, S. Hölldobler, and L. M. Pereira. On conditionals. In G. Gottlob, G. Sutcliffe, and A. Voronkov, editors, *Global Conference on Artificial Intelligence*, volume 36 of *Epic Series in Computing*, pages 79–92. EasyChair, 2015.

[16] E.-A. Dietz, S. Hölldobler, and M. Ragni. A computational logic approach to the suppression task. In N. Miyake, D. Peebles, and R. P. Cooper, editors, *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, pages 1500–1505. Cognitive Science Society, 2012.

[17] E.-A. Dietz, S. Hölldobler, and M. Ragni. A computational logic approach to the abstract and the social case of the selection task. In *Proceedings Eleventh International Symposium on Logical Formalizations of Commonsense Reasoning*, 2013. `commonsensereasoning.org/2013/proceedings.html`.

[18] E.-A. Dietz Saldanha, S. Hölldobler, C. D. P. Kencana Ramli, and L. Palacios Medinacelli. A core method for the weak completion semantics with skeptical abduction. *Journal of Artificial Intelligence Research*, 63:51–86, 2018.

[19] E.-A. Dietz Saldanha, S. Hölldobler, and I. Lourêdo Rocha. Obligation versus factual conditionals under the weak completion semantics. In S. Hölldobler, A. Malikov, and C. Wernhard, editors, *Proceedings of the Second Young Scientists' International Workshop on Trends in Information Processing*, volume 1837, pages 55–64. CEUR-WS.org, 2017. `http://ceur-ws.org/Vol-1837/`.

[20] K. Dieussaert, W. Schaeken, W. Schroyen, and G. d'Ydewalle. Strategies during complex conditional inferences. *Thinking and Reasoning*, 6(2):125–161, 2000.

[21] M. Fitting. A Kripke–Kleene semantics for logic programs. *Journal of Logic Programming*, 2(4):295–312, 1985.

[22] M. Fitting. Metric methods – three examples and a theorem. *Journal of Logic Programming*, 21(3):113–127, 1994.

[23] P. Foot. The problem of abortion and the doctrine of double effect. *Oxford Review*, 5, 1967.

[24] J. Y. Girard. Linear logic. *Journal of Theoretical Computer Science*, 50(1):1 – 102, 1987.

[25] R.A. Griggs and J.R. Cox. The elusive thematic materials effect in the Wason selection task. *British Journal of Psychology*, 73:407–420, 1982.

[26] G. Große, S. Hölldobler, and J. Schneeberger. Linear deductive planning. *Journal of Logic and Computation*, 6(2):233–262, 1996.

[27] C. A. Gunter and D. S. Scott. Semantic domains. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B)*, pages 633–674. North-Holland, 1990.

[28] G. S. Halford. *Children's Understanding.* Psychology Press, New York, NY, 1993.

[29] C. Hartshorn and P. Weiss, editors. *Collected Papers of Charles Sanders Peirce*, volume 2. Harvard University Press, 1932.

[30] S. Hölldobler. *Foundations of Equational Logic Programming*, volume 353 of *Lecture Notes in Artificial Intelligence.* Springer-Verlag, Berlin, 1989.

[31] S. Hölldobler. On deductive planning and the frame problem. In A. Voronkov, editor, *Proceedings of the Conference on Logic Programming and Automated Reasoning*, pages 13–29. Springer-Verlag, LNCS, 1992.

[32] S. Hölldobler. Ethical decision making under the weak completion semantics. In C. Schon, editor, *Proceedings of the Workshop on Bridging the Gap between Human and Automated Reasoning*, volume 2261, pages 1–5. CEUR-WS.org, 2018. `http://ceur-ws.org/Vol-2261/`.

[33] S. Hölldobler and C. D. P. Kencana Ramli. Contraction properties of a semantic operator for human reasoning. In Lei Li and K. K. Yen, editors, *Proceedings of the Fifth International Conference on Information*, pages 228–231. International Information Institute, 2009.

[34] S. Hölldobler and C. D. P. Kencana Ramli. Logic programs under three-valued Łukasiewicz's semantics. In P. M. Hill and D. S. Warren, editors, *Logic Programming*, volume 5649 of *Lecture Notes in Computer Science*, pages 464–478. Springer-Verlag Berlin Heidelberg, 2009.

[35] S. Hölldobler and J. Schneeberger. A new deductive approach to planning. *New Generation Computing*, 8:225–244, 1990.

[36] S. Hölldobler, J. Schneeberger, and M. Thielscher. AC1–unification/matching in linear logic programming. In F. Baader, J. Siekmann, and W. Snyder, editors, *Proceedings of the Sixth International Workshop on Unification.* BUCS Tech Report 93-004, Boston University, Computer Science Department, 1993.

[37] J. Jaffar, J-L. Lassez, and M. J. Maher. A theory of complete logic programs with equality. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 175–184. ICOT, 1984.

[38] A. C. Kakas, R. A. Kowalski, and F. Toni. Abductive Logic Programming. *Journal of Logic and Computation*, 2(6):719–770, 1993.

[39] F. M. Kamm. *Intricate Ethics: Rights, Responsibilities, and Permissible Harm.* Oxford University Press, Oxford, 2006.

[40] C. D. P. Kencana Ramli. Logic programs and three-valued consequence operators. Master's thesis, International Center for Computational Logic, TU Dresden, 2009.

[41] S. Khemlani and P. N. Johnson-Laird. Theories of the syllogism: A meta-analysis. *Psychological Bulletin*, 138(3):427–457, 2012.

[42] S. C. Kleene. *Introduction to Metamathematics.* North-Holland, 1952.

[43] R.A. Kowalski. *Computational Logic and Human Thinking: How to be Artificially Intelligent.* Cambridge University Press, 2011.

[44] J. W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, 1984.

[45] I. Lourêdo Rocha. Obligation versus factual conditionals under the weak completion semantics. Project Work, ICCL, TUD, 2017.

[46] J. Łukasiewicz. O logice trójwartościowej. *Ruch Filozoficzny*, 5:169–171, 1920. English translation: On Three-Valued Logic. In: *Jan Łukasiewicz Selected Works.* (L. Borkowski, ed.), North Holland, 87-88, 1990.

[47] M. Masseron, C. Tollu, and J. Vauzielles. Generating plans in linear logic. In *Foundations of Software Technology and Theoretical Computer Science*, volume 472 of *Lecture Notes in Computer Science*, pages 63–75. Springer Berlin, Heidelberg, 1990.

[48] J. McCarthy. Situations and actions and causal laws. Stanford Artificial Intelligence Project: Memo 2, 1963.

[49] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of Artificial Intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463 – 502. Edinburgh University Press, 1969.

[50] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63(2):81–97, 1956.

[51] R. S. Nickerson. *Conditional Reasoning.* Oxford University Press, 2015.

[52] A. Oliviera da Costa, E.-A. Dietz Saldanha, S. Hölldobler, and M. Ragni. A computational logic approach to human syllogistic reasoning. In G. Gunzelmann, A. Howes, T. Tenbrink, and E. J. Davelaar, editors, *Proceedings of the 39th Annual Conference of the Cognitive Science Society*, pages 883–888, Austin, TX, 2017. Cognitive Science Society.

[53] D. N. Osherson. *Logical Abilities in Children*, volume 1. Routledge, London, 1974.

[54] J. Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, New York, USA, 2000.

[55] L. M. Pereira, E.-A. Dietz, and S. Hölldobler. An abductive reasoning approach to the belief-bias effect. In C. Baral, G. De Giacomo, and T. Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 14th International Conference*, pages 653–656, Cambridge, MA, 2014. AAAI Press.

[56] L. M. Pereira, E.-A. Dietz, and S. Hölldobler. Contextual abductive reasoning with side-effects. In I. Niemelä, editor, *Theory and Practice of Logic Programming (TPLP)*, volume 14, pages 633–648, Cambridge, UK, 2014. Cambridge University Press.

[57] L. M. Pereira and A. M. Pinto. Inspecting side-effects of abduction in logic programming. In M. Balduccini and T. C. Son, editors, *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning: Essays in Honour of Michael Gelfond*, volume 6565 of *Lecture Notes in Artificial Intelligence*, pages 148–163. Springer, 2011.

[58] L. M. Pereira and A. Saptawijaya. *Programming Machine Ethics.* Springer, Berlin, Heidelberg, 2016.

[59] B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 440–446, Menlo Park, 1992. AAAI Press.

[60] K. Stenning and M. van Lambalgen. Semantic interpretation as computation in nonmonotonic logic: The real meaning of the suppression task. *Cognitive Science*, 29(919-960), 2005.

[61] K. Stenning and M. van Lambalgen. *Human Reasoning and Cognitive Science.* MIT Press, 2008.

[62] M. Thielscher. AC1-Unifikation in der linearen logischen Programmierung. Master's thesis, Intellektik, Informatik, TH Darmstadt, 1992.

[63] M. Thielscher. Computing ramification by postprocessing. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1994–2000, 1995.

[64] M. Thielscher. Causality and the qualification problem. In L. C. Aiello, J. Doyle, and S. C. Shapiro, editors, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, Cambridge, MA, 1996. Morgan Kaufmann.

[65] M. Thielscher. Introduction to the fluent calculus. *Electronic Transactions on Artificial Intelligence*, 2(3-4):179–192, 1998.

[66] M. Thielscher. Controlling semi-automatic systems with FLUX (extended abstract). In C. Palamidessi, editor, *Logic Programming*, volume 2916 of *Lecture Notes in Computer Science*, pages 515–516. Springer, Berlin, Heidelberg, 2003.

[67] P. C. Wason. Reasoning about a rule. *The Quarterly Journal of Experimental Psychology*, 20:273–281, 1968.

Index