Proceedings of the First Workshop on

Bridging the Gap between Human and Automated Reasoning

Bridging-2015

a CADE-25 workshop

Berlin, Germany, August 1st, 2015

Edited by Ulrich Furbach and Claudia Schon

Preface

This volume contains the papers presented at Bridging-15: Bridging the Gap between Human and Automated Reasoning held on August 1st, 2015 in Berlin in conjunction with CADE-25.

Human reasoning or the psychology of deduction is well researched in cognitive psychology and in cognitive science. There are a lot of findings which are based on experimental data about reasoning tasks, among others models for the Wason selection task or the suppression task discussed by Byrne and others. This research is supported also by brain researchers, who aim at localizing reasoning processes within the brain. Automated deduction, on the other hand, is mainly focusing on the automated proof search in logical calculi. And indeed there is tremendous success during the last decades. Recently a coupling of the areas of cognitive science and automated reasoning is addressed in several approaches. For example, there is increasing interest in modeling human reasoning within automated reasoning systems including modeling with answer set programming, deontic logic or abductive logic programming. There are also various approaches within AI research.

This workshop is intended to get an overview of existing approaches and makes a step towards a cooperation between computational logic and cognitive science.

In total, seven papers were submitted to the workshop. From these, five have been accepted for presentation. The papers present the following strands: logic programming approaches to model human reasoning; formalization of syllogisms in human reasoning; computational models for human reasoning; benchmarks for commonsense reasoning; interactive theorem proving.

Apart from the accepted papers, the workshop program includes one keynote presentation by Marco Ragni. He can be seen as a representative of interdisciplinary research – holding two PhDs, one in computer science and one in cognitive science. His talk on *Three-levels of Analysis: Connecting cognitive theories of reasoning with empirical results and cognitive modeling* certainly can be understood as a bridge connecting various disciplines.

Finally, the Bridging-15 organizers seize the opportunity to thank the Program Committee members for their most valuable comments on the submissions, the authors for inspiring papers, the audience for their interest in this workshop, the local organizers from the CADE-25 team, and the Workshops Chair.

We hope that in the years to come, Bridging will become a platform for dialogue and interaction for researchers in both cognitive science and automated reasoning and will effectively help to bridge the gap between human and automated reasoning.

July 8, 2015 — Ulrich Furbach
Koblenz — Claudia Schon

Table of Contents

Program Committee

Ulrich Furbach	Universität Koblenz-Landau
Steffen Hölldobler	Technische Universität Dresden
Gabriele Kern-Isberner	Technische Universität Dortmund
Markus Knauff	Universität Gießen
Kai-Uwe Kühnberger	Universität Osnabrück
Marco Ragni	Universität Freiburg
Claudia Schon	Universität Koblenz-Landau
Natarajan Shankar	SRI International
Keith Stenning	Edinburgh University
Frieder Stolzenburg	Hochschule Harz

Additional Reviewer

Pereira, Luís Moniz

Three-levels of Analysis: Connecting cognitive theories of reasoning with empirical results and cognitive modeling

Marco Ragni
Foundations of Artificial Intelligence lab
Technical Faculty, University of Freiburg

Summary

A recent increase in theories of human reasoning shows the need to evaluate these theories. While some properties like non-monotonicity can be evaluated theoretically other properties can be evaluated empirically only: For instance the ability to predict quantitative differences in error rates, reponse times, and physiological correlates like eye-movement and brain activations.

In this talk, I will focus on spatial reasoning and introduce a computational model implying a cognitive complexity measure. In a second step, this model will be evaluated on current empirical results. This raises the question of how well these cognitive models can predict human behavior and the associated cognitive difficulty. Moreover, there is the demand to examine these models on their cognitive adequacy for physiological correlates such as eye movements and brain activations. The goal is to have cognitive computational models that can form a bridge to psychology and neuroscience at the same time.

Weak Completion Semantics and its Applications in Human Reasoning

Steffen Hölldobler

International Center for Computational Logic,TU Dresden, 01062 Dresden, Germany
sh@iccl.tu-dresden.de

Abstract. I present a logic programming approach based on the weak completions semantics to model human reasoning tasks, and apply the approach to model the suppression task, the selection task as well as the belief-bias effect, to compute preferred mental models of spatial reasoning tasks and to evaluate indicative as well as counterfactual conditionals.

1 Introduction

Observing the performance of humans in cognitive tasks like the suppression [3] or the selection task [31] it is apparent that human reasoning cannot be adequately modeled by classical two-valued logic. Whereas there have been many approaches to develop a normative model for human reasoning which are not based on logic like the mental model theory [22] or probabilistic approaches [15], Keith Stenning and Michiel von Lambalgen have developed a logic-based approach [30] where, in a first step, they reason towards an appropriate representation of some aspects of the world as logic program and, in a second step, reason with respect to the least model of the program. Their approach is based on the three-valued (strong) Kripke-Kleene logic [23], is non-monotonic, and utilizes some form of completion as well as abduction. Most interestingly, the results developed within the fields of logic programming and computational logic within the last decades could not be immediately applied to adequately model human reasoning tasks but rather some modifications were needed. As a consequence, theorems, propositions and lemmas formally proven for a theory without these modification cannot be readily applied but their proofs must be adapted as well.

Unfortunately, some of the formal results stated in [30] are not correct. Somewhat surprisingly, we were able to show in [19] that the results do hold if the Kripke-Kleene logic is replaced by the three-valued Łukasiewicz logic [25]. We have called our approach *weak completion semantics* (WCS) because in the completion of a program, undefined relations are not identified with falsehood but rather are left *unknown*. Whereas our original emphasis was on obtaining formally correct results, WCS has been applied to many different human reasoning tasks in the meantime: the suppression task, the abstract as well as the social selection task, the belief-bias effect, the computation of preferred mental models in spational reasoning tasks as well as the evaluation of conditionals.

This paper gives an overview on WCS as well as its applications to human reasoning tasks.

2 Weak Completion Semantics

2.1 Logic Programs

We assume the reader to be familiar with logic programming, but we repeat basic notions and notations. A *(logic) program* is a finite set of (program) clauses of the form $A \leftarrow \top,\ A \leftarrow \bot$ or $A \leftarrow B_1 \wedge \ldots \wedge B_n,\ n > 0$ where A is an atom, B_i, $1 \leq i \leq n$, are literals and $\top$ and $\bot$ denote truth and falsehood, resp. A is called *head* and $\top$, $\bot$ as well as $B_1 \wedge \ldots \wedge B_n$ are called *body* of the corresponding clause. Clauses of the form $A \leftarrow \top$ and $A \leftarrow \bot$[1] are called *positive* and *negative facts*, resp. In this paper, $\mathcal{P}$ denotes a program, A a ground atom and F a formula. We assume that each non-propositional program contains at least one constant symbol. We also assume for each program that the underlying alphabet consists precisely of the symbols mentioned in the program, if not indicated differently. When writing sets of literals we omit curly brackets if a set has only one element.

$g\mathcal{P}$ denotes the set of all ground instances of clauses occurring in $\mathcal{P}$. A ground atom A is *defined* in $g\mathcal{P}$ iff $g\mathcal{P}$ contains a clause whose head is A; otherwise A is said to be *undefined*. $def(\mathcal{S}, \mathcal{P}) = \{A \leftarrow body \in g\mathcal{P} \mid A \in \mathcal{S} \vee \neg A \in \mathcal{S}\}$ is called *definition* of $\mathcal{S}$ in $\mathcal{P}$, where $\mathcal{S}$ is a set of ground literals. Such a set $\mathcal{S}$ is said to be *consistent* iff it does not contain a pair of complementary literals.

A *level mapping* for $\mathcal{P}$ is a function ℓ which assigns to each atom occurring in $g\mathcal{P}$ a natural number. Let $\ell(\neg A) = \ell(A)$. $\mathcal{P}$ is *acyclic* iff there exists a level mapping ℓ such that for each $A \leftarrow L_1 \wedge \ldots \wedge L_n \in g\mathcal{P}$ we find that $\ell(A) > \ell(L_i)$, $1 \leq i \leq n$.

2.2 Weak Completion

For a given $\mathcal{P}$, consider the following transformation: (1) For each defined atom A, replace all clauses of the form $A \leftarrow body_1, \ldots, A \leftarrow body_m$ occurring in $g\mathcal{P}$ by $A \leftarrow body_1 \vee \ldots \vee body_m$. (2) Replace all occurrences of $\leftarrow$ by $\leftrightarrow$. The obtained ground program is called *weak completion* of $\mathcal{P}$ or $wc\mathcal{P}$.[2]

2.3 Łukasiewicz Logic

An *interpretation* is a mapping from the set of formulas into the set of truth values. A *model* for F is an interpretation which maps F to *true*. We consider the three-valued Łukasiewicz (or Ł-) logic [25] (see Table 1) and represent each interpretation I by $\langle I^\top, I^\bot \rangle$, where $I^\top = \{A \mid I(A) = \top\}$, $I^\bot = \{A \mid I(A) = \bot\}$, $I^\top \cap I^\bot = \emptyset$, and each ground atom $A \notin I^\top \cup I^\bot$ is mapped to U. Hence, under the empty interpretation $\langle \emptyset, \emptyset \rangle$ all ground atoms are *unknown*. Let $\langle I^\top, I^\bot \rangle$ and $\langle J^\top, J^\bot \rangle$ be two interpretations. We define

$$\langle I^\top, I^\bot \rangle \subseteq \langle J^\top, J^\bot \rangle \text{ iff } I^\top \subseteq J^\top \text{ and } I^\bot \subseteq J^\bot,$$
$$\langle I^\top, I^\bot \rangle \cup \langle J^\top, J^\bot \rangle = \langle I^\top \cup J^\top, I^\bot \cup J^\bot \rangle.$$

[1] Under WCS a clause of the form $A \leftarrow \bot$ is turned into $A \leftrightarrow \bot$ provided that it is the only clause in the definition of A.

[2] Note that undefined atoms are not identified with $\bot$ as in the completion of $\mathcal{P}$ [5].

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	$\top$

Table 1. Truth tables for the Ł-semantics, where we have used $\top$, $\bot$ and U instead of *true*, *false* and *unknown*, resp., in order to shorten the presentation.

Theorem 1. *(Model Intersection Property) For each program $\mathcal{P}$, the intersection of all Ł-models of $\mathcal{P}$ is an Ł-model of $\mathcal{P}$.*

This result was formally proven in [19] for programs not containing negative facts, but it holds also for programs with negative facts.

2.4 A Semantic Operator

The following operator was introduced by Stenning and van Lambalgen [30], where they also showed that it admits a least fixed proint: $\Phi_{\mathcal{P}}(\langle I^{\top}, I^{\bot}\rangle) = \langle J^{\top}, J^{\bot}\rangle$, where

$$J^{\top} = \{A \mid A \leftarrow \mathit{body} \in g\mathcal{P} \text{ and } \mathit{body} \text{ is } \mathit{true} \text{ under } \langle I^{\top}, I^{\bot}\rangle\},$$
$$J^{\bot} = \{A \mid \mathit{def}(A,\mathcal{P}) \neq \emptyset \text{ and } \mathit{body} \text{ is } \mathit{false} \text{ under } \langle I^{\top}, I^{\bot}\rangle \text{ for all } A \leftarrow \mathit{body} \in \mathit{def}(A,\mathcal{P})\}.$$

The $\Phi_{\mathcal{P}}$ operator differs from the semantic operator defined by Fitting in [13] in the additional condition $\mathit{def}(A,\mathcal{P}) \neq \emptyset$ required in the definition of $J^{\bot}$. This condition states that A must be defined in order to be mapped to *false*, whereas in the (strong) Kripke-Kleene-semantics considered by Fitting an atom is mapped to *false* if it is undefined. This reflects precisely the difference between the weak completion and the completion semantics. The (strong) Kripke-Kleene-semantics was also applied in [30]. However, as shown in [19] this semantics is not only the cause for a technical bug in one theorem of [30], but it does also lead to a non-adequate model of some human reasoning tasks. Both, the technical bug as well as the non-adequate modeling, can be avoided by using WCS.

Theorem 2. *The least fixed point of $\Phi_{\mathcal{P}}$ is the least Ł-model of the weak completion of $\mathcal{P}$.* [19]

In the remainder of this paper, $\mathcal{M}_{\mathcal{P}}$ denotes the least Ł-model of $wc\mathcal{P}$.

2.5 Contraction

It was Fitting's idea [14] to apply metric methods to compute least fixed points of semantic operators and, in particular, he showed that for so-called *acceptable*[3]

[3] Please see [14] for a definition of acceptable programs. The class of acyclic programs is a proper subset of the class of acceptable programs.

programs the semantic operator defined in [13] is a contraction.[4] Consequently, Banach's contraction mapping theorem [2] can be applied to compute the least fixed point of the semantic operator.

As shown in [18], $\Phi_{\mathcal{P}}$ may not be a contraction if $\mathcal{P}$ is acceptable. But the following weaker result holds for programs not containing any cycles.

Theorem 3. *If $\mathcal{P}$ is an acyclic program, then $\Phi_{\mathcal{P}}$ is a contraction.* [18]

As a consequence, the computation of the least fixed point of $\Phi_{\mathcal{P}}$ can be initialized with an arbitrary interpretation.

2.6 A Connectionist Realization

Within the CORE-method [1, 17] semantic operators of logic programs are computed by feed-forward connectionist networks, where the input and the output layer represent interpretations. By connecting the output with the input layer, the networks are turned into recurrent ones and can now be applied to compute the least fixed points of the semantic operators.

Theorem 4. *For each datalog program $\mathcal{P}$ there exists a recurrent connectionist network which will converge to a stable state representing $\mathcal{M}_{\mathcal{P}}$ if initialized with the empty interpretation.*

The theorem was proven in [20] for propositional programs but extends to datalog programs. From the discussion in the previous paragraph we conclude that the network may be initialized by some interpretation if $\Phi_{\mathcal{P}}$ is a contraction.

2.7 Weak Completion Semantics

The *weak completion semantics* (WCS) is the approach to consider weakly completed logic programs and to reason with respect to the least Ł-models of these programs. We write $\mathcal{P} \models_{wcs} F$ iff formula F holds in $\mathcal{M}_{\mathcal{P}}$. WCS is non-monotonic.

2.8 Relation to Well-Founded Semantics

WCS is related to the well-founded semantics (WFS) as follows: Let $\mathcal{P}^+ = \mathcal{P} \setminus \{A \leftarrow \bot \mid A \leftarrow \bot \in \mathcal{P}\}$ and u be a new nullary relation symbol not occurring in $\mathcal{P}$. Furthermore, let $\mathcal{P}^* = \mathcal{P}^+ \cup \{B \leftarrow u \mid def(B, \mathcal{P}) = \emptyset\} \cup \{u \leftarrow \neg u\}$.

Theorem 5. *If $\mathcal{P}$ is a program which does not contain a positive loop, then $\mathcal{M}_{\mathcal{P}}$ and the well-founded model for $\mathcal{P}^*$ coincide.* [11]

[4] A mapping $f : \mathcal{M} \to \mathcal{M}$ on a metric space $(\mathcal{M}, d)$ is a *contraction* iff there exists a $k \in (0, 1)$ such that for all $x, y \in \mathcal{M}$ we find $d(f(x), f(y)) \leq k \times d(x, y)$.

2.9 Abduction

An *abductive framework* consists of a logic program $\mathcal{P}$, a set of *abducibles* $\mathcal{A}_\mathcal{P} = \{A \leftarrow \top \mid def(A,\mathcal{P}) = \emptyset\} \cup \{A \leftarrow \bot \mid def(A,\mathcal{P}) = \emptyset\}$, a set of *integrity constraints* $\mathcal{IC}$, i.e., expressions of the form $\bot \leftarrow B_1 \wedge \ldots \wedge B_n$, and the entailment relation $\models_{wcs}$; it is denoted by $\langle \mathcal{P}, \mathcal{A}_\mathcal{P}, \mathcal{IC}, \models_{wcs} \rangle$.

By Theorem 1, each program and, in particular, each finite set of positive and negative ground facts has an Ł-model. For the latter, this can be obtained by mapping all heads occurring in this set to *true*. Thus, in the following definition, explanations as well as the union of a program and an explanation are satisfiable.

An *observation* $\mathcal{O}$ is a set of ground literals; it is *explainable* in the framework $\langle \mathcal{P}, \mathcal{A}_\mathcal{P}, \mathcal{IC}, \models_{wcs} \rangle$ iff there exists a minimal $\mathcal{E} \subseteq \mathcal{A}_\mathcal{P}$ called *explanation* such that $\mathcal{M}_{\mathcal{P}\cup\mathcal{E}}$ satisfies $\mathcal{IC}$ and $\mathcal{P} \cup \mathcal{E} \models_{wcs} L$ for each $L \in \mathcal{O}$. F *follows creduluously from* $\mathcal{P}$ *and* $\mathcal{O}$ iff there exists an explanantion $\mathcal{E}$ such that $\mathcal{P} \cup \mathcal{E} \models_{wcs} F$. F *follows skeptically from* $\mathcal{P}$ *and* $\mathcal{O}$ iff for all explanantions $\mathcal{E}$ we find $\mathcal{P} \cup \mathcal{E} \models_{wcs} F$.

2.10 Revision

Let $\mathcal{S}$ be a finite and consistent set of ground literals in

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

where A denotes an atom. $rev(\mathcal{P},\mathcal{S})$ is called the *revision of* $\mathcal{P}$ *with respect to* $\mathcal{S}$. The following result was formally proven in [7].

Proposition 6. *1. rev is non-monotonic,*
i.e., there exist $\mathcal{P}$, $\mathcal{S}$ *and* F *such that* $\mathcal{P} \models_{wcs} F$ *and* $rev(\mathcal{P},\mathcal{S}) \not\models_{wcs} F$.
2. If $\mathcal{M}_\mathcal{P}(L) = U$ *for all* $L \in \mathcal{S}$, *then rev is monotonic.*
3. $\mathcal{M}_{rev(\mathcal{P},\mathcal{S})}(\mathcal{S}) = \top$.

3 Applications

3.1 The Suppression Task

Ruth Byrne has shown in [3] that graduate students with no previous exposure to formal logic did suppress previously drawn conclusions when additional information became available. Table 2 shows the abbreviations that will be used in this subsection, whereas Table 3 gives an account of the findings of [3]. Interestingly, in some instances the previously drawn conclusions were valid (cases AE and ACE in Table 3) whereas in other instances the conclusions were invalid (cases AL and ABL in Table 3) with respect to classical two-valued logic.

Following [30] conditionals are encoded by licences for implications using *abnormality* predicates. In the case AE no abnormalities concerning the library are known. However, in the case ACE it becomes known that one can visit the library only if it is open and, thus, not being open becomes an abnormality for the first implication. Likewise, one may argue that there must be a reason for studying in the library. In the case ACE the only reason for studying in

A	*If she has an essay to finish, then she will study late in the library.*
B	*If she has a textbook to read, then she will study late in the library.*
C	*If the library stays open, she will study late in the library.*
E	*She has an essay to finish.*
$\overline{E}$	*She does not have an essay to finish.*
L	*She will study late in the library.*
$\overline{L}$	*She will not study late in the library.*

Table 2. The suppression task [3] and used abbreviations. Subjects received conditionals A, B or C and facts E, $\overline{E}$, L or $\overline{L}$ and had to draw inferences.

Cond.	Fact	Exp. Findings	Cond.	Fact	Exp. Findings
A	E	96% conclude L	A	L	53% conclude E
$A\ B$	E	96% conclude L	$A\ B$	L	16% conclude E
$A\ C$	E	38% conclude L	$A\ C$	L	55% conclude E
A	$\overline{E}$	46% conclude $\overline{L}$	A	$\overline{L}$	69% conclude $\overline{E}$
$A\ B$	$\overline{E}$	4% conclude $\overline{L}$	$A\ B$	$\overline{L}$	69% conclude $\overline{E}$
$A\ C$	$\overline{E}$	63% conclude $\overline{L}$	$A\ C$	$\overline{L}$	44% conclude $\overline{E}$

Table 3. The drawn conclusions in the experiment of Byrne. The different cases will be denoted by the word obtained by concatenating the conditionals and the fact like AE or AL for the cases in the first row of the table.

the library is to finish an essay and, consequently, not having to finish an essay becomes an abnormality for the second implication. Alltogether, for the cases AE and ACE we obtain the programs

$$\begin{aligned}\mathcal{P}_{AE} &= \{\ell \leftarrow e \wedge \neg ab_1,\ e \leftarrow \top,\ ab_1 \leftarrow \bot\},\\ \mathcal{P}_{ACE} &= \{\ell \leftarrow e \wedge \neg ab_1,\ e \leftarrow \top,\ ab_1 \leftarrow \neg o,\ \ell \leftarrow o \wedge \neg ab_2,\ ab_2 \leftarrow \neg e\}\end{aligned}$$

with $\mathcal{M}_{\mathcal{P}_{AE}} = \langle\{e, \ell\}, \{ab_1\}\rangle$ and $\mathcal{M}_{\mathcal{P}_{ACE}} = \langle\{e\}, \{ab_2\}\rangle$, where ℓ, e, o and ab denote that *she will study late in the library*, *she has an essay to finish*, *the library stays open* and *abnormality*, resp. Hence, $\mathcal{M}_{\mathcal{P}_{AE}}(\ell) = \top$ and $\mathcal{M}_{\mathcal{P}_{ACE}}(\ell) =$ U. Thus, WCS can model the suppression of a previously drawn conclusion.

For the examples in the second column of Table 3 abduction is needed. E.g., for the case ABL we obtain the program

$$\mathcal{P}_{AB} = \{\ell \leftarrow e \wedge \neg ab_1,\ ab_1 \leftarrow \bot,\ \ell \leftarrow t \wedge \neg ab_3,\ ab_3 \leftarrow \bot\}$$

with $\mathcal{M}_{\mathcal{P}_{AB}} = \langle\emptyset, \{ab_1, ab_3\}\rangle$, where t denotes that *she has a textbook to read*. The observation $\mathcal{O} = \ell$ can be explained by $\mathcal{E}_1 = \{e \leftarrow \top\}$ and $\mathcal{E}_2 = \{t \leftarrow \top\}$. In order to adequately model Byrne's selection task, we have to be skeptical as otherwise–being credoluous–we would conclude that *she has an essay to finish*.

A complete account of Byrne's selection task under WCS is given in [10, 21].

D	F	3	7
89%	16%	62%	25%

beer	coke	22yrs	16yrs
95%	0.025%	0.025%	80%

Table 4. The results of the abstract and social case of the selection task, where the first row gives the symbol(s) on the cards and the second row shows the percentage of participants which turned it.

$\mathcal{O}$	$\mathcal{E}$	$\mathcal{M}_{\mathcal{P}_{ac} \cup \mathcal{E}}$	turn
D	$\{D \leftarrow \top\}$	$\langle \{D, 3\}, ab_1 \rangle$	yes
F	$\{F \leftarrow \top\}$	$\langle F, ab_1 \rangle$	*no*
3	$\{D \leftarrow \top\}$	$\langle \{D, 3\}, ab_1 \rangle$	*yes*
7	$\{7 \leftarrow \top\}$	$\langle 7, ab_1 \rangle$	*no*

Table 5. The computational logic approach for the abstract case of the selection task.

3.2 The Selection Task

In the original (abstract) selection task [31] participants were given the conditional *if there is a D on one side of the card, then there is* 3 *on the other side* and four cards on a table showing the letters D and F as well as the numbers 3 and 7. Furthermore, they know that each card has a letter on one side and a number on the other side. Which cards must be turned to prove that the conditional holds?

Griggs and Cox [16] adapted the abstract task to a social case. Consider the conditional *if a person is drinking beer, then the person must be over 19 years of age* and again consider four cards, where one side shows the person's age and on the other side shows the person's drink: *beer*, *coke*, *22yrs* and *16yrs*. Which drinks and persons must be checked to prove that the conditional holds?

When confronted with both tasks, participants reacted quite differently as shown in Table 4. Moreover, if the conditionals are modeled as implications in classical two-valued logic, then some of the drawn conclusions are not valid.

The Abstract Case This case is artificial and there is no common sense knowledge about the conditional. Let D, F, 3, and 7 be propositional variables denoting that the corresponding symbol or number is on one side of a card. Following [24], we assume that the given conditional is viewed as a belief and represented as a clause in

$$\mathcal{P}_{ac} = \{3 \leftarrow D \wedge \neg ab_1,\ ab_1 \leftarrow \bot\},$$

where the negative fact was added as there are no known abnormalities. We obtain $\mathcal{M}_{\mathcal{P}_{ac}} = \langle \emptyset, ab_1 \rangle$ and find that this model does not explain any symbol on the cards. Let $\mathcal{A}_{ac} = \{D \leftarrow \top,\ D \leftarrow \bot,\ F \leftarrow \top,\ F \leftarrow \bot,\ 7 \leftarrow \top,\ 7 \leftarrow \bot\}$ in the abductive framework $\langle \mathcal{P}_{ac}, \mathcal{A}_{ac}, \emptyset, \models_{wcs} \rangle$. Table 5 shows the explanations for the cards with respect to this framework.

In case D was observed, the least model maps also 3 to $\top$. In order to be sure that this corresponds to the real situation, we need to check if 3 is *true*.

case	$\mathcal{P}_{sc}$	$\mathcal{M}_{\mathcal{P}_{sc}}$	$\models_{wcs} o \leftarrow b \wedge \neg ab_2$	*turn*
beer	$\{ab_2 \leftarrow \bot, b \leftarrow \top\}$	$\langle b, ab_2\rangle$	*no*	*yes*
coke	$\{ab_2 \leftarrow \bot, b \leftarrow \bot\}$	$\langle \emptyset, \{b, ab_2\}\rangle$	*yes*	*no*
22yrs	$\{ab_2 \leftarrow \bot, o \leftarrow \top\}$	$\langle o, ab_2\rangle$	*yes*	*no*
16yrs	$\{ab_2 \leftarrow \bot, o \leftarrow \bot\}$	$\langle \emptyset, \{o, ab_2\}\rangle$	*no*	*yes*

Table 6. The computational logic approach for the social case of the selection task.

Therefore, the card showing D is turned. Likewise, in case 3 is observed, D is also mapped to $\top$, which can only be confirmed if the card is turned.

The Social Case In this case most humans are quite familiar with the conditional as it is a standard law. They are also aware–it is common sense knowledge– that there are no exceptions or abnormalities. Let o represent a person being older than 19 years and b a person drinking beer. The conditional can be represented by $o \leftarrow b \wedge \neg ab_2$ and is viewed as a social constraint which must follow logically from the given facts. Table 6 shows the four different cases.

One should observe that in the case *16yrs* the least model of the weak completion of $\mathcal{P}_{sc}$, i.e. $\langle \emptyset, \{o, ab_2\}\rangle$, assigns U to b and, consequently, to both, $b \wedge \neg ab_2$ and $o \leftarrow b \wedge \neg ab_2$, as well. Overall, in the cases *beer* and *16yrs* the social constraint is not entailed by the least Ł-model of the weak completion of the program. Hence, we need to check these cases out and, hopefully, find that the beer drinker is older than 19 and that the 16 years old is not drinking beer.

A complete account of the selection task under WCS is given in [6].

3.3 The Belief-Bias Effect

Evans et. al. [12] made a psychological study showing possibly conflicting processes in human reasoning. Participants were confronted with syllogisms and had to decide whether they are logically valid. Consider the following syllogism:

No addictive things are inexpensive. (Premise1)
Some cigarettes are inexpensive. (Premise2)
Therefore, some addictive things are not cigarettes. (Conclusion)

The conclusion does not follow from the premises in classical logic: If there are inexpensive cigarettes but no addictive things, then the premises are *true*, but the conclusion is *false*. Nevertheless, most participants considered the syllogism to be valid. Evans et. al. explained the answers by an unduly influence of the participants' own beliefs.

Before we can model this line of reasoning under WCS, we need to tackle the problem that the head of a program clause must be an atom, whereas the conclusion of the rule *if something is inexpensive, then it is not addictive*[5] is a

[5] (Premise1) can be formalized in many syntactically different, but semantically equivalent ways in classical logic. We have selected a form which allows WCS to adequately model the belief-bias effect.

negated atom. If the relation symbol add is used to denote addiction, then this technical problem can be overcome by introducing a new relation symbol add', specifying by means of the clause

$$add(X) \leftarrow \neg add'(X) \tag{1}$$

that add' is the negation of add under WCS and requiring by means of the integrity constraint

$$\mathcal{IC}_{add} = \{\bot \leftarrow add(X) \wedge \neg add'(X)\}$$

that add and add' cannot be simultaneously true.

We can now encode (Premise1) following Stenning and van Lambalgen's idea to represent conditionals by licences for implications [30]:

$$add'(X) \leftarrow inex(X) \wedge \neg ab_1(X), \qquad ab_1(X) \leftarrow \bot. \tag{2}$$

As for (Premise2), Evans et. al. have argued that it includes two pieces of information. Firstly, there exists something, say a, which is a cigarette:

$$cig(a) \leftarrow \top. \tag{3}$$

Secondly, it contains the following belief that humans seem to have:

Cigarettes are inexpensive. (Bias1)

This belief implies (Premise2) and biases the process of reasoning towards a representation such that we obtain:

$$inex(X) \leftarrow cig(X) \wedge \neg ab_2(X), \qquad ab_2(X) \leftarrow \bot. \tag{4}$$

Additionally, it is assumed that there is a second piece of background knowledge, viz. it is commonly known that

Cigarettes are addictive, (Bias2)

which in the context of (1) and (2) can be specified by stating that cigarettes are abnormalities regarding add':

$$ab_1(X) \leftarrow cig(X). \tag{5}$$

Alltogether, let $\mathcal{P}_{add}$ be the program consisting of the clauses (1)-(5). Because (Conclusion) is about an object which is not necessarily a we need to add another constant, say b, to the alphabet underlying $\mathcal{P}_{add}$. We obtain

$$\mathcal{M}_{\mathcal{P}_{add}} = \langle \{cig(a), inex(a), ab_1(a), add(a)\}, \{ab_2(a), ab_2(b), add'(a)\}\rangle.$$

Turning to (Conclusion) we consider its first part as the observation $\mathcal{O} = add(b)$ which needs to be explained with respect to the abductive framework

$$\langle \mathcal{P}_{add}, \{cig(b) \leftarrow \top, cig(b) \leftarrow \bot\}, \mathcal{IC}_{add}, \models_{wcs}\rangle.$$

We find two minimal explanations $\mathcal{E}_\bot = \{cig(b) \leftarrow \bot\}$ and $\mathcal{E}_\top = \{cig(b) \leftarrow \top\}$ leading to the minimal models

$$\begin{aligned}\mathcal{M}_{\mathcal{P}_{add} \cup \mathcal{E}_\bot} = \langle\ & \{cig(a), inex(a), ab_1(a), add(a), add(b)\}, \\ & \{ab_2(a), ab_2(b), add'(a), cig(b), inex(b), ab_1(b), add'(b)\}\ \rangle, \\ \mathcal{M}_{\mathcal{P}_{add} \cup \mathcal{E}_\top} = \langle\ & \{cig(a), inex(a), ab_1(a), add(a), cig(b), inex(b), ab_1(b), add(b)\}, \\ & \{ab_2(a), ab_2(b), add'(a), add'(b)\}\ \rangle,\end{aligned}$$

respectively. Because under $\mathcal{M}_{\mathcal{P}_{add} \cup \mathcal{E}_\top}$ all known addictive objects (a and b) are cigarettes and under $\mathcal{M}_{\mathcal{P}_{add} \cup \mathcal{E}_\bot}$ the addictive object b is not a cigarette, (Conclusion) follows creduluously, but not skeptically.

On the other hand, the two explanations $\mathcal{E}_\bot$ and $\mathcal{E}_\top$ do not seem to be equally likely given (Premise1) and (Bias1). Rather, $\mathcal{E}_\bot$ seems to be the main explanation whereas $\mathcal{E}_\top$ seems to be the exceptional case. Pereira and Pinto [26] have introduced so-called *inspection points* which allow to distinguish between main and exceptional explanations in an abductive framework. Formally, they introduce a meta-predicate *inspect* and require that if $inspect(A) \leftarrow \top$ or $inspect(A) \leftarrow \bot$ are elements of an explanation $\mathcal{E}$ for some literal or observation L, then either $A \leftarrow \top$ or $A \leftarrow \bot$ must be in $\mathcal{E}$ as well and, moreover, $A \leftarrow \bot$ or $A \leftarrow \top$ must be elements of explanations for some literal or observation $L' \neq L$, where A is a ground atom.

With the help of inspection points, the program $\mathcal{P}_{add}$ can be rewritten to

$$\mathcal{P}'_{add} = (\mathcal{P}_{add} \setminus \{ab_1(X) \leftarrow cig(X)\}) \cup \{ab_1(X) \leftarrow inspect(cig(X))\}$$

and the explanation $\mathcal{O} = add(b)$ is to be explained with respect to the abductive framework $\langle \mathcal{P}'_{add}, \mathcal{A}'_{add}, \mathcal{IC}_{add}, \models_{wcs} \rangle$, where

$$\begin{aligned}\mathcal{A}'_{add} = \{\ & cig(b) \leftarrow \top,\ cig(b) \leftarrow \bot, \\ & inspect(cig(b)) \leftarrow \top,\ inspect(cig(b)) \leftarrow \bot, \\ & inspect(cig(a)) \leftarrow \top,\ inspect(cig(a)) \leftarrow \bot\ \}.\end{aligned}$$

Now, $\mathcal{E}_\bot$ is the only explanation for $add(b)$ and, hence, (Conclusion) follows skeptically in the revised approach.

More details about our model of the belief-bias effect and abduction using inspection points can be found in [27, 28].

3.4 Spatial Reasoning

Consider the following *spatial reasoning problem*. Suppose it is known that *a ferrari is left of a porsche*, *a beetle is right of the porsche*, *the porsche is left of a hummer*, and *the hummer is left of a dodge. Is the beetle left of the hummer?*

The *mental model theory* [22] is based on the idea that humans construct so-called *mental models*, which in case of a spatial reasoning problem is understood

as the presentation of the spatial arrangements between objects that correspond to the premises. In the example, there are three mental models:

ferrari porsche beetle hummer dodge
ferrari porsche hummer beetle dodge
ferrari porsche hummer dodge beetle

Hence, the answer to the above mentioned question depends on the construction of the mental models.

In the *preferred model theory* [29] it is assumed that humans do not construct all mental models, but rather a single, *preferred* one, and that reasoning is performed with respect to the preferred mental model. The preferred mental model is believed to be constructed by considering the premises one by one in the order of their occurrence and to place objects directly next to each other or, if this impossible, in the next available space. For the example, the preferred mental model is constructed as follows:

ferrari porsche
ferrari porsche beetle
ferrari porsche beetle hummer
ferrari porsche beetle hummer dodge

Hence, according to the preferred model theory, *the beetle is left of the hummer*.

In [8] we have specified a logic program $\mathcal{P}$ taking into account the premises of a spatial reasoning problem such that $\mathcal{M}_\mathcal{P}$ corresponds to the preferred mental model. Moreover, within the computation of $\mathcal{M}_\mathcal{P}$ as the least fixed point of $\Phi_\mathcal{P}$, the preferred mental model is constructed step by step as in [29].

3.5 Conditionals

Conditionals are statements of the form *if condition then consequence*. In this paper we distinguish between indicative and subjunctive (or counterfactual) conditionals. *Indicative conditionals* are conditionals whose condition is either *true* or *unknown*; the consequence is asserted to be *true* if the condition is *true*. On the contrary, the condition of a *subjunctive* or *counterfactual conditional* is either *false* or *unknown*; in the counterfactual circumstance of the condition being *true*, the consequence is asserted to be *true*.[6] We assume that the condition and the consequence of a conditional are finite and consistent sets of literals.

Conditionals are evaluated with respect to some background information specified as a program and a set of integrity constraints. More specifically, as the weak completion of each program admits a least Ł-model, conditionals are evaluated under the least Ł-model of a program. In the reminder of this section let $\mathcal{P}$ be a program, $\mathcal{IC}$ be a finite set of integrity constraints, and $\mathcal{M}_\mathcal{P}$ be the least Ł-model of $wc\mathcal{P}$ such that $\mathcal{M}_\mathcal{P}$ satisfies $\mathcal{IC}$.

[6] In the literature the case of a condition being *unknown* is usually not explicitely considered; there also seems to be no standard definition for indicative and counterfactual conditionals.

In this setting we propose to evaluate a conditional $cond(\mathcal{C}, \mathcal{D})$ as follows, where $\mathcal{C}$ and $\mathcal{D}$ are finite and consistent sets of literals:

1. If $\mathcal{M}_\mathcal{P}(\mathcal{C}) = \top$ and $\mathcal{M}_\mathcal{P}(\mathcal{D}) = \top$, then $cond(\mathcal{C}, \mathcal{D})$ is *true*.
2. If $\mathcal{M}_\mathcal{P}(\mathcal{C}) = \top$ and $\mathcal{M}_\mathcal{P}(\mathcal{D}) = \bot$, then $cond(\mathcal{C}, \mathcal{D})$ is *false*.
3. If $\mathcal{M}_\mathcal{P}(\mathcal{C}) = \top$ and $\mathcal{M}_\mathcal{P}(\mathcal{D}) = \mathrm{U}$, then $cond(\mathcal{C}, \mathcal{D})$ is *unknown*.
4. If $\mathcal{M}_\mathcal{P}(\mathcal{C}) = \bot$, then evaluate $cond(\mathcal{C}, \mathcal{D})$ with respect to $\mathcal{M}_{rev(\mathcal{P},\mathcal{S})}$, where $\mathcal{S} = \{L \in \mathcal{C} \mid \mathcal{M}_\mathcal{P}(L) = \bot\}$.
5. If $\mathcal{M}_\mathcal{P}(\mathcal{C}) = \mathrm{U}$, then evaluate $cond(\mathcal{C}, \mathcal{D})$ with respect to $\mathcal{M}_{\mathcal{P}'}$, where
   - $\mathcal{P}' = rev(\mathcal{P}, \mathcal{S}) \cup \mathcal{E}$,
   - $\mathcal{S}$ is a smallest subset of $\mathcal{C}$ and $\mathcal{E} \subseteq \mathcal{A}_{rev(\mathcal{P},\mathcal{S})}$ is a minimal explanation for $\mathcal{C} \setminus \mathcal{S}$ such that $\mathcal{M}_{\mathcal{P}'}(\mathcal{C}) = \top$.

In words, if the condition of a conditional is *true*, then the conditional is an indicative one and is evaluated as implication in Ł-logic. If the condition is *false*, then the conditional is a counterfactual conditional. In this case, i.e., in case 4, non-monotonic revision is applied to the program in order to reverse the truth value of those literals, which are mapped to *false*.

The main novel contribution concerns the final case 5. If the condition $\mathcal{C}$ of a conditional is *unknown*, then we propose to split $\mathcal{C}$ into two disjoint subsets $\mathcal{S}$ and $\mathcal{C} \setminus \mathcal{S}$, where the former is treated by revision and the latter by abduction. In case $\mathcal{C}$ contains some literals which are *true* and some which are *unknown* under $\mathcal{M}_\mathcal{P}$, then the former will be part of $\mathcal{C} \setminus \mathcal{S}$ because the empty explanation explains them. As we assume $\mathcal{S}$ to be minimal this approach is called *minimal revision followed by abduction* (MRFA). Furthermore, because revision as well as abduction are only applied to literals which are assigned to *unknown*, case 5 is monotonic.

As an example consider the *forest fire scenario* taken from [4]: The conditional $cond(\neg dl, \neg ff)$, *if there had not been so many dry leaves on the forest floor, then the forest fire would not have occurred*, is to be evaluated with respect to

$$\mathcal{P}_{ff} = \{ff \leftarrow l \wedge \neg ab_1,\ l \leftarrow \top,\ ab_1 \leftarrow \neg dl,\ dl \leftarrow \top\},$$

which states that *lightning* (l) *causes a forest fire* (ff) *if nothing abnormal* (ab_1)*, is taking place, lightning happened, the absence of dry leaves* (dl) *is an abnormality, and dry leaves are present*. We obtain $\mathcal{M}_{\mathcal{P}_{ff}} = \langle\{dl, l, ff\}, \{ab_1\}\rangle$ and find that the condition $\neg dl$ is *false*. Hence, we are dealing with a counterfactual conditional. Following Step 4 we obtain $\mathcal{S} = \{\neg dl\}$,

$$rev(\mathcal{P}_{ff}, \neg dl) = \{ff \leftarrow l \wedge \neg ab_1,\ l \leftarrow \top,\ ab_1 \leftarrow \neg dl,\ dl \leftarrow \bot\}$$

and $\mathcal{M}_{rev(\mathcal{P}_{ff},\neg dl)} = \langle\{l, ab_1\}, \{dl, ff\}\rangle$. Because ff is mapped to *false* under this model, the conditional is *true*.

Let us extend the example by adding *arson* (a) *causes a forest fire*:

$$\mathcal{P}_{ffa} = \mathcal{P}_{ff} \cup \{ff \leftarrow a \wedge \neg ab_2,\ ab_2 \leftarrow \bot\}.$$

We find $\mathcal{M}_{\mathcal{P}_{ffa}} = \langle\{dl, l, ff\}, \{ab_1, ab_2\}\rangle$ and $\mathcal{M}_{rev(\mathcal{P}_{ffa},\neg dl)} = \langle\{l, ab_1\}, \{dl, ab_2\}\rangle$. Under this model ff is *unknown* and, consequently, $cond(\neg dl, \neg ff)$ is *unknown* as well.

As final example consider $\mathcal{P}_{\mathit{ffa}}$ and the conditional $cond(\{\mathit{ff}, \neg dl\}, a)$: *if a forest fire occurred and there had not been so many dry leaves on the forest floor, then arson must have caused the fire*. Because the condition $\{\mathit{ff}, \neg dl\}$ is *false* under $\mathcal{M}_{\mathcal{P}_{\mathit{ffa}}}$ we follow Step 4 and obtain $\mathcal{S} = \{\neg dl\}$,

$$rev(\mathcal{P}_{\mathit{ffa}}, \neg dl) = (\mathcal{P}_{\mathit{ffa}} \setminus \{dl \leftarrow \top\}) \cup \{dl \leftarrow \bot\}$$

and $\mathcal{M}_{rev(\mathcal{P}_{\mathit{ffa}}, \neg dl)} = \langle \{l, ab_1\}, \{dl, ab_2\} \rangle$. One should observe that ff as well as the condition $\{\mathit{ff}, \neg dl\}$ are *unknown* under this model. Hence, we follow Step 5, consider the abductive framework

$$\langle rev(\mathcal{P}_{\mathit{ffa}}, \neg dl), \{a \leftarrow \top, a \leftarrow \bot\}, \emptyset, \models_{wcs} \rangle$$

and learn that $\{\mathit{ff}, \neg dl\}$ can be explained by $\{a \leftarrow \top\}$. Hence, by MRFA we obtain as final program $rev(\mathcal{P}_{\mathit{ffa}}, \neg dl) \cup \{a \leftarrow \top\}$ and find

$$\mathcal{M}_{rev(\mathcal{P}_{\mathit{ffa}}, \neg dl) \cup \{a \leftarrow \top\}} = \langle \{l, ab_1, \mathit{ff}, a\}, \{dl, ab_2\} \rangle.$$

Because a is mapped to *true* under this model, the conditional is *true* as well.

More details about the evaluation of conditionals under WCS can be found in [7, 9].

4 Conclusion

I have presented the weak completion semantics (WCS) and have demonstrated how various human reasoning tasks can be adequately modeled under WCS. To the best of my knowledge, WCS is the computational logic based approach which can handle most human reasoning tasks within a single framework. For example, [30] discusses only the selection task in detail and mentions the selection task, whereas [24] discusses the selection task in detail and mentions the suppression task.

But there are many open questions. I only claim that conditionals are adequately evaluated as shown in Section 3.5; this claim must be thoroughly tested. We may also consider scenarios, where abduction needs to be applied to satisfy the consequent of a conditional. The connectionist model reported in 2.6 does not yet include abduction and we are unaware of any connectionist realization of sceptical abduction.

Acknowledgements I like to thank *Michiel van Lambalgen* for the discussions at the ICCL summer school 2008 which initialized this research. *Carroline Dewi Puspa Kencana Ramli* wrote an outstandings master's thesis in which she developed the formal framework of the WCS including the connectionist realization; she has received the EMCL best master's thesis award 2009. The relationship between WCS and WFS was established jointly with *Emmanuelle-Anna Dietz* and *Christoph Wernhard*. Abduction was added to the framework with the help of *Emma*, *Christoph* and *Tobias Philipp*. The ideas underlying the revision operator were developed jointly with *Emma* and *Luís Moniz Pereira*.

The suppression task was the running example throughout the development of WCS involving *Carroline*, *Tobias*, *Christoph*, *Emma* and *Marco Ragni*. The solution for the selection task was developed with *Emma* and *Marco*. The approach to model spatial reasoning problems is a revised version of the ideas first developed by Raphael Höps in his bachelor thesis under the supervision of *Emma*; many thanks to *Marco* who introduced us to this problem. *Emma* and *Luís* proposed the solution for the belief bias effect. The procedure to evaluate conditionals is the result of many discussions with *Emma*, *Luís* and *Bob Kowalski*. Finally, I like to thank the referees of the paper for many helpful comments.

References

1. S. Bader and S. Hölldobler. The Core method: Connectionist model generation. In S. Kollias, A. Stafylopatis, W.Duch, and E. Ojaet, editors, *Proceedings of the 16th International Conference on Artificial Neural Networks (ICANN)*, volume 4132 of *Lecture Notes in Computer Science*, pages 1–13. Springer-Verlag, 2006.
2. S. Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fund. Math.*, 3:133–181, 1922.
3. R. Byrne. Suppressing valid inferences with conditionals. *Cognition*, 31:61–83, 1989.
4. R. M. J. Byrne. *The Rational Imagination: How People Create Alternatives to Reality*. MIT Press, Cambridge, MA, USA, 2007.
5. K. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Databases*, pages 293–322. Plenum, New York, 1978.
6. 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`.
7. 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*. Springer, 2015.
8. E.-A. Dietz, S. Hölldobler, and R. Höps. A computational logic approach to human spatial reasoning. Technical Report KRR-2015-02, TU Dresden, International Center for Computational Logic, 2015.
9. E.-A. Dietz, S. Hölldobler, and L. M. Pereira. On indicative conditionals. In S. Hölldobler and Y. Liang, editors, *Proceedings of the First International Workshop on Semantic Technologies*, volume 1339 of *CEUR Workshop Proceedings*, pages 19–30. CEUR-WS.org, 2015. `http://ceur-ws.org/Vol-1339/`.
10. 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.
11. E.-A. Dietz, S. Hölldobler, and C. Wernhard. Modelling the suppression task under weak completion and well-founded semantics. *Journal of Applied Non-Classical Logics*, 24:61–85, 2014.
12. J. Evans, J. Barston, and P. Pollard. On the conflict between logic and belief in syllogistic reasoning. *Memory & Cognition*, 11:295–306, 1983.

13. M. Fitting. A Kripke–Kleene semantics for logic programs. *Journal of Logic Programming*, 2(4):295–312, 1985.
14. M. Fitting. Metric methods – three examples and a theorem. *Journal of Logic Programming*, 21(3):113–127, 1994.
15. G. Gigerenzer and D. Murray. *Cognition ad Intuitive Statistics*. Hillsdale, NJ: Erlbaum, 1987.
16. R. Griggs and J. Cox. The elusive thematic materials effect in the wason selection task. *British Journal of Psychology*, 73:407–420, 1982.
17. S. Hölldobler and Y. Kalinke. Towards a new massively parallel computational model for logic programming. In *Proceedings of the ECAI94 Workshop on Combining Symbolic and Connectionist Processing*, pages 68–77. ECCAI, 1994.
18. S. Hölldobler and C. D. P. Kencana Ramli. Contraction properties of a semantic operator for human reasoning. In L. Li and K. K. Yen, editors, *Proceedings of the Fifth International Conference on Information*, pages 228–231. International Information Institute, 2009.
19. 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.
20. S. Hölldobler and C. D. P. Kencana Ramli. Logics and networks for human reasoning. In C. Alippi, M. M. Polycarpou, C. G. Panayiotou, and G. Ellinasetal, editors, *Artificial Neural Networks – ICANN*, volume 5769 of *Lecture Notes in Computer Science*, pages 85–94. Springer-Verlag Berlin Heidelberg, 2009.
21. S. Hölldobler, T. Philipp, and C. Wernhard. An abductive model for human reasoning. In *Proc. Tenth International Symposium on Logical Formalizations of Commonsense Reasoning*, 2011. `commonsensereasoning.org/2011/proceedings.html`.
22. P. Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge University Press, Cambridge, 1983.
23. S. Kleene. *Introduction to Metamathematics*. North-Holland, 1952.
24. R. Kowalski. *Computational Logic and Human Thinking: How to be Artificially Intelligent*. Cambridge University Press, 2011.
25. 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.
26. J. Pereira and A. Pinto. Inspecting side-effects of abduction in logic programming. In M. Balduccini and T. 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.
27. L. M. Pereira, E.-A. Dietz, and S. Hölldobler. An abductive reasoning approach to the belief-bias effect. In C. Baral, G. D. 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.
28. 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, 2014. Cambridge University Press.
29. M. Ragni and M. Knauff. A theory and a computational model of spatial reasoning. *Psychological Review*, 120:561–588, 2013.
30. K. Stenning and M. van Lambalgen. *Human Reasoning and Cognitive Science*. MIT Press, 2008.
31. P. Wason. Reasoning about a rule. *The Quarterly Journal of Experimental Psychology*, 20:273–281, 1968.

A Computational Logic Approach to Syllogisms in Human Reasoning

Emmanuelle-Anna Dietz
dietz@iccl.tu-dresden.de

International Center for Computational Logic, TU Dresden, 01062 Dresden, Germany

Abstract. Psychological experiments on syllogistic reasoning have shown that participants did not always deduce the classical logically valid conclusions. In particular, the results show that they had difficulties to reason with syllogistic statements that contradicted their own beliefs. This paper discusses syllogisms in human reasoning and proposes a formalization under the weak completion semantics.

1 Introduction

Evans, Barston and Pollard [10] made a psychological study about deductive reasoning, which demonstrated possibly conflicting processes in human reasoning. Participants were presented different syllogisms, for which they had to decide whether these were (classical) logically valid. Consider S_{vit}:

Premise 1	*No nutritional things are inexpensive.*
Premise 2	*Some vitamin tablets are inexpensive.*
Conclusion	*Therefore, some vitamin tablets are not nutritional.*

The conclusion necessarily follows from the premises. However, approximately half of the participants said that this syllogism was not logically valid. They were explicitly asked to logically validate or invalidate various syllogisms. Table 1 gives four examples of syllogisms, which have been tested in [10]. If participants judged that "the conclusion necessarily follows from the statements in the passage, [you]" they "should answer 'yes,' otherwise 'no'." The last column shows the percentage of the participants that believed the syllogism to be valid. Evans, Barston and Pollard asserted that the participants were influenced by their own beliefs, their so-called belief bias, where we distinguish between the negative and the positive belief bias [11]. The negative belief bias, i.e., when a support for the unbelievable conclusion is suppressed, happens for 56% of the participants in S_{vit}. A positive belief bias, i.e., when the acceptance for the believable conclusion is raised, happens for 71% of the participants in S_{cig}. As pointed out in [14], Wilkins [32] already observed that syllogisms, which conflict with our beliefs are more difficult to solve. People reflectively read the instructions and understand well that they are required to reason logically from the premises to the conclusion. However, the results show that their intuitions are stronger and deliver a tendency to say 'yes' or 'no' depending on whether it

	Type	Case	%
S_{dog}	valid and believable	*No police dogs are vicious.* *Some highly trained dogs are vicious.* *Therefore, some highly trained dogs are not police dogs.*	89
S_{vit}	valid and unbelievable	*No nutritional things are inexpensive.* *Some vitamin tablets are inexpensive.* *Therefore, some vitamin tablets are not nutritional.*	56
S_{rich}	invalid and unbelievable	*No millionaires are hard workers.* *Some rich people are hard workers.* *Therefore, some millionaires are not rich people.*	10
S_{cig}	invalid and believable	*No addictive things are inexpensive.* *Some cigarettes are inexpensive.* *Therefore, some addictive things are not cigarettes.*	71

Table 1. Examples of four kinds of syllogisms. The percentages are summarized results over three experiments and show the rate that the conclusion is accepted to be valid [10].

is believable [9]. Various theories have tried to explain this phenomenon. Some conclusions can be explained by converting the premises [2] or by assuming that the atmosphere of the premises influences the acceptance for the conclusion [33]. Johnson-Laird and Byrne [20] proposed the mental model theory [19], which additionally supposes the search for counterexamples when validating the conclusion. These theories have been partly rejected or claimed to be incomplete. Evans et al. [10, 12] proposed a theory, which is sometimes referred to as the selective scrutiny model [1, 14]. First, humans heuristically accept any syllogism having a believable conclusion, and only check on the logic if the conclusion contradicts their belief. Adler and Rips [1] claim that this behavior is rational because it efficiently maintains our beliefs, except in case if there is any evidence to change them. It results in an adaptive process, for which we only make an effort towards a logical evaluation when the conclusion is unbelievable. It would take a lot of effort if we would constantly verify them even though there is no reason to question them. As people intend to keep their beliefs as consistent as possible, they invest more effort in examining statements that contradict them, than the ones that comply with them. However, this theory cannot fully explain all classical logical errors in the reasoning process. Yet another approach, the selective processing model [8], accounts only for a single preferred model. If the conclusion is neutral or believable, humans attempt to construct a model that supports it. Otherwise, they attempt to construct a model, which rejects it.

As summarized in [14], there are several stages in which a belief bias can take place. First, beliefs can influence our interpretation of the premises. Second, in case a statement contradicts our belief, we might search for alternative models and check whether the conclusion is plausible.

Stenning and van Lambalgen [30] explain why certain aspects influence the interpretations made by humans when evaluating syllogisms and discuss this in the context of mental models. They propose to model human reasoning in a

two step procedure. First, human reasoning should be modeled towards an adequate representation. Second, human reasoning should be adequately modeled with respect to this representation. In our context, the first step is about the representational part, that is, which our beliefs influence the interpretation of the premises. The second step is about the procedural part, that is, whether we search for alternative models and whether the conclusion is plausible.
After we have specified some preliminaries, we explain in Section 3 how the just discussed four cases of the syllogistic reasoning task can be represented in logic programs. Based on this representation, Section 4 discusses how beliefs and background knowledge influences the reasoning process and shows that the results can be modeled by computing the least models of the weak completion.

2 Preliminaries

The general notation, which we will use in the paper, is based on [15, 22].

2.1 Logic Programs

We restrict ourselves to datalog programs, i.e., the set of terms consists only of constants and variables. A *logic program* $\mathcal{P}$ is a finite set of clauses of the form

$$A \leftarrow L_1 \wedge \ldots \wedge L_n, \tag{1}$$

where $n \geq 0$ with finite n. A is an atom and L_i, $1 \leq i \leq n$, are literals. A is called *head* of the clause and the subformula to the right of the implication sign is called *body* of the clause. If the clause contains variables, then they are implicitly universally quantified within the scope of the entire clause. A clause that does not contain variables, is called a *ground* clause. In case $n = 0$, the clause is a *positive fact* and denoted as

$$A \leftarrow \top.$$

A *negative fact* is denoted as

$$A \leftarrow \bot,$$

where *true* ,$\top$, and *false*, $\bot$, are *truth-value constants*. The notion of falsehood appears counterintuitive at first sight, but programs will be interpreted under their (weak) completion where we replace the implication by the equivalence sign. We assume a fixed set of constants, denoted by $\mathsf{CONSTANTS}$, which is nonempty and finite. $\mathsf{constants}(\mathcal{P})$ denotes the set of all constants occurring in $\mathcal{P}$. If not stated otherwise, we assume that $\mathsf{CONSTANTS} = \mathsf{constants}(\mathcal{P})$.
$\mathsf{g}\,\mathcal{P}$ denotes ground $\mathcal{P}$, which means that $\mathcal{P}$ contains exactly all the ground clauses with respect to the alphabet. $\mathsf{atoms}(\mathcal{P})$ denotes the set of all atoms occurring in $\mathcal{P}$. If atom A is not the head of any clause in $\mathcal{P}$, then A is *undefined* in $\mathcal{P}$. The set of all atoms that are undefined in $\mathcal{P}$, is denoted by $\mathsf{undef}(\mathcal{P})$.

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	$\bot$

$\leftarrow_{\text{Ł}}$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	U	$\top$	$\top$
$\bot$	$\bot$	U	$\top$

$\leftrightarrow_{\text{Ł}}$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	$\top$	U
$\bot$	$\bot$	U	$\top$

Table 2. $\top$, $\bot$, and U denote *true*, *false*, and *unknown*, respectively.

2.2 Three-Valued Łukasiewicz Semantics

We consider the three-valued Łukasiewicz Semantics [23], for which the corresponding truth values are $\top$, $\bot$ and U, which mean *true*, *false* and *unknown*, respectively. A *three-valued interpretation* I is a mapping from formulas to a set of truth values $\{\top, \bot, \mathsf{U}\}$. The truth value of a given formula under I is determined according to the truth tables in Table 2. We represent an interpretation as a pair $I = \langle I^\top, I^\bot \rangle$ of disjoint sets of atoms where $I^\top$ is the set of all atoms that are mapped to $\top$ by I, and $I^\bot$ is the set of all atoms that are mapped to $\bot$ by I. Atoms, which do not occur in $I^\top \cup I^\bot$, are mapped to U. Let $I = \langle I^\top, I^\bot \rangle$ and $J = \langle J^\top, J^\bot \rangle$ be two interpretations: $I \subseteq J$ iff $I^\top \subseteq J^\top$ and $I^\bot \subseteq J^\bot$. $I(F) = \top$ means that a formula F is mapped to true under I. $\mathcal{M}$ is a *model* of $\mathsf{g}\,\mathcal{P}$ if it is an interpretation, which maps each clause occurring in $\mathsf{g}\,\mathcal{P}$ to $\top$. I is the *least model* of $\mathsf{g}\,\mathcal{P}$ iff for any other model J of $\mathsf{g}\,\mathcal{P}$ it holds that $I \subseteq J$.

2.3 Reasoning with Respect to Least Models

Consider following transformation for $\mathsf{g}\,\mathcal{P}$:

1. Replace all clauses in $\mathsf{g}\,\mathcal{P}$ with the same head $A \leftarrow \mathit{Body}_1, A \leftarrow \mathit{Body}_2, \ldots$ by the single expression $A \leftarrow \mathit{Body}_1 \vee \mathit{Body}_2, \vee \ldots$.
2. If $A \in \mathsf{undef}(\mathsf{g}\,\mathcal{P})$, then add $A \leftarrow \bot$.
3. Replace all occurrences of $\leftarrow$ by $\leftrightarrow$.

The resulting set of equivalences is called the *completion* of $\mathsf{g}\,\mathcal{P}$ [3]. If Step 2 is omitted, then the resulting set is called the *weak completion* of $\mathsf{g}\,\mathcal{P}$ ($\mathsf{wc}\,\mathsf{g}\,\mathcal{P}$). In contrast to completed programs, the model intersection property holds for weakly completed programs [17]. This guarantees the existence of a least model for every program. Stenning and van Lambalgen [30] devised such an operator, which has been generalized for first-order programs by [16]: Let I be an interpretation in $\Phi_{\mathsf{SvL},\mathcal{P}}(I) = \langle J^\top, J^\bot \rangle$, where

$$
\begin{aligned}
J^\top &= \{A \mid \text{there exists a clause } A \leftarrow \mathit{Body} \in \mathsf{g}\,\mathcal{P} \text{ with } I(\mathit{Body}) = \top\},\\
J^\bot &= \{A \mid \text{there exists a clause } A \leftarrow \mathit{Body} \in \mathsf{g}\,\mathcal{P} \text{ and}\\
&\qquad \text{for all clauses } A \leftarrow \mathit{Body} \in \mathsf{g}\,\mathcal{P} \text{ we find } I(\mathit{Body}) = \bot\}.
\end{aligned}
$$

As shown in [16] the least fixed point of $\Phi_{SvL,\mathcal{P}}$ is identical to the least model of the weak completion of $\mathsf{g}\,\mathcal{P}$ under three-valued Łukasiewicz semantics. In the

following, we will denote the least model of the weak completion of a given program $\mathcal{P}$ by $\mathsf{lm}_{\text{L}}\mathsf{wc}\,\mathsf{g}\,\mathcal{P}$. From $I = \langle\emptyset, \emptyset\rangle$, $\mathsf{lm}_{\text{L}}\mathsf{wc}\,\mathsf{g}\,\mathcal{P}$ is computed by iterating $\Phi_{SvL,\mathcal{P}}$. Given a program $\mathcal{P}$ and a formula F, $\mathcal{P} \models_{\text{L}}^{\mathsf{lmwc}} F$ iff $\mathsf{lm}_{\text{L}}\mathsf{wc}\,\mathsf{g}\,\mathcal{P}(F) = \top$ for formula F. Notice that Φ_{SvL} differs in a subtle way from the well-known Fitting operator Φ_F, introduced in [13]: The definition of Φ_F is like that of Φ_{SvL}, except that in the specification of $J^{\bot}$ the first line "there exists a clause $A \leftarrow \mathit{Body} \in \mathsf{g}\,\mathcal{P}$ and" is dropped. The least fixed point of $\Phi_{F,\mathcal{P}}$ corresponds to the least model of the completion of $\mathsf{g}\,\mathcal{P}$. If an atom A is undefined in $\mathsf{g}\,\mathcal{P}$, then, for arbitrary interpretations I it holds that $A \in J^{\bot}$ in $\Phi_{F,\mathcal{P}}(I) = \langle J^{\top}, J^{\bot}\rangle$, whereas if Φ_{SvL} is applied instead of Φ_F, this does not hold for any interpretation I.
The correspondence between weak completion semantics and well-founded semantics [31] for tight programs, i.e. those without positive cycles, is shown in [6].

2.4 Integrity Constraints

A set of *integrity constraints* $\mathcal{IC}$ comprises clauses of the form $\bot \leftarrow \mathit{Body}$, where *Body* is a conjunction of literals. Under three-valued semantics, there are several ways on how to understand integrity constraints [21], two of them being the *theoremhood view* and the *consistency view*. Consider $\mathcal{IC}$:

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

The theoremhood view requires that a model only satisfies the set of integrity constraints if for all its clauses, *Body* is false under this model. In the example, this is only the case if p is true or if q is false in the model. In the consistency view, the set of integrity constraints is satisfied by the model if *Body* is unknown or false in it. Here, a model satisfies $\mathcal{IC}$ already if either p or q is unknown.
Given $\mathcal{P}$ and set $\mathcal{IC}$, $\mathcal{P}$ *satisfies* $\mathcal{IC}$ iff there exists I, which is a model for $\mathsf{g}\,\mathcal{P}$, and for each $\bot \leftarrow \mathit{Body} \in \mathcal{IC}$, we find that $I(\mathit{Body}) \in \{\bot, \mathsf{U}\}$.

2.5 Abduction

We extend two-valued abduction [21] for three-valued semantics. The set of abducibles $\mathcal{A}_{\mathcal{P}}$ may not only contain positive but can also contain negative facts:

$$\{A \leftarrow \top \mid A \in \mathsf{undef}(\mathcal{P})\} \cup \{A \leftarrow \bot \mid A \in \mathsf{undef}(\mathcal{P})\}.$$

Let $\langle \mathcal{P}, \mathcal{A}_{\mathcal{P}}, \mathcal{IC}, \models_{\text{L}}^{\mathsf{lmwc}}\rangle$ be an abductive framework, $\mathcal{E} \subset \mathcal{A}_{\mathcal{P}}$ and observation $\mathcal{O}$ a non-empty set of literals.

$\mathcal{O}$ is *explained by* $\mathcal{E}$ *given* $\mathcal{P}$ *and* $\mathcal{IC}$ iff
$\mathcal{P} \not\models_{\text{L}}^{\mathsf{lmwc}} \mathcal{O}$, $\mathcal{P} \cup \mathcal{E} \models_{\text{L}}^{\mathsf{lmwc}} \mathcal{O}$ and $\mathsf{lm}_{\text{L}}\mathsf{wc}\,\mathsf{g}\,(\mathcal{P} \cup \mathcal{E})$ *satisfies* $\mathcal{IC}$.
$\mathcal{O}$ is *explained given* $\mathcal{P}$ *and* $\mathcal{IC}$ iff
there exists an $\mathcal{E}$ such that $\mathcal{O}$ is explained by $\mathcal{E}$ given $\mathcal{P}$ and $\mathcal{IC}$.

We assume that explanations are minimal, that means, there is no other explanation $\mathcal{E}' \subset \mathcal{E}$ for $\mathcal{O}$. In case abducibles are not abduced as positive or negative facts, they stay unknown in the least model of the weak completion. We distinguish between skeptical and credulous abduction as follows:

F follows skeptically from $\mathcal{P}$, $\mathcal{IC}$ and $\mathcal{O}$ iff $\mathcal{O}$ can be explained given $\mathcal{P}$ and $\mathcal{IC}$, and for all minimal $\mathcal{E}$ for $\mathcal{O}$, given $\mathcal{P}$ and $\mathcal{IC}$, it holds that $\mathcal{P} \cup \mathcal{E} \models_{\text{L}}^{\mathsf{lmwc}} F$.

F follows credulously from $\mathcal{P}$, $\mathcal{IC}$ and $\mathcal{O}$ iff there exists a minimal $\mathcal{E}$ for $\mathcal{O}$, given $\mathcal{P}$ and $\mathcal{IC}$, and it holds that $\mathcal{P} \cup \mathcal{E} \models_{\text{L}}^{\mathsf{lmwc}} F$.

3 Reasoning Towards an Appropriate Logical Form

Let us specify the syllogisms from the introduction in logic programs. We first discuss a technical aspect that allows us to encode the negative consequences of the premises. Section 3.2 covers the representational part and show how the beliefs, which might influence the interpretation of the premises, are encoded.

3.1 Positive Encoding of Negative Consequences

The first premise of S_{dog} is

No police dogs are vicious.

and is equivalent to

If something is vicious, then it is not a police dog.
and *If something is a police dog, then it is not vicious.*

The consequences in both inferences are the negation of *it is a police dog* and the negation of *it is vicious*, respectively. As the weak completion semantics does not allow negative heads in clauses, we cannot represent these inferences in a logic program straightaway. For every negative conclusion $\neg p(X)$ we introduce an auxiliary formula $p'(X)$ together with the clause $p(X) \leftarrow \neg p'(X)$. We obtain the following preliminary representation of the first premise of S_{dog} wrt *vicious*:[1]

$$\mathit{police\_dog}'(X) \leftarrow \mathit{vicious}(X), \qquad \mathit{police\_dog}(X) \leftarrow \neg \mathit{police\_dog}'(X),$$

where $\mathit{police\_dog}(X)$, $\mathit{police\_dog}'(X)$, and $\mathit{vicious}(X)$ denote that X is a police dog, X is not a police dog, and X is vicious, respectively. A model $I = \langle I^{\top}, I^{\bot} \rangle$ that contains both $\mathit{police\_dog}(X)$ and $\mathit{police\_dog}'(X)$ in $I^{\top}$ should be invalidated. This condition can be represented by the integrity constraint

$$\mathcal{IC}_{\mathit{police\_dog}} = \{\bot \leftarrow \mathit{police\_dog}(X) \wedge \mathit{police\_dog}'(X)\},$$

and is to be understood as discussed in Section 2.4. For the following examples, whenever there exists a $p(X)$ and its $p'(X)$ counterpart in $\mathcal{P}$, we implicitly assume $\mathcal{IC}_p = \{\bot \leftarrow p(X) \wedge p'(X)\}$.

[1] In the following we will only encode one of the inferences.

3.2 Abnormality Predicates and Background Knowledge

Newstead and Griggs [25] have shown, that the universal quantifiers in natural language are often understood as fuzzy quantifiers, which allow exceptions. In some circumstances *for all* is understood as *for almost all*. They argue that the statement *all Germans are hardworking* seems to permit exceptions and is understood as a generalization about all Germans and not a statement, which is true for each one.
This fuzzy interpretation of quantifiers seems to be in line with Stenning and van Lambalgen's suggestion to implement conditionals by default licenses for implications [29, 30]. They propose to introduce abnormality predicates, which should be added to the antecedent of the implication, where the abnormality predicate is initially assumed to be false. Consider again PREMISE 1 in S_{dog}, which can be understood as

If something is vicious and not abnormal (in that respect),
then it is not a police dog.
Nothing (by default) is abnormal (regarding the previous sentence).

This information together with the previously introduced clauses for PREMISE 1 in S_{dog} can now be encoded as:

$$\begin{aligned} \mathit{police\_dog}'(X) &\leftarrow \mathit{vicious}(X) \wedge \neg \mathit{ab}_{dog'}(X), \\ \mathit{police\_dog}(X) &\leftarrow \neg \mathit{police\_dog}'(X), \\ \mathit{ab}_{dog'}(X) &\leftarrow \bot. \end{aligned}$$

$\mathbf{S}_{dog}$ PREMISE 2 states that there are some highly trained dogs that are vicious. This statement presupposes that there actually exists something, let us say a new reserved (Skolem) constant a, for which the following is true:

$$\mathit{highly\_trained}(a) \leftarrow \top \qquad \text{and} \qquad \mathit{vicious}(a) \leftarrow \top.$$

$\mathcal{P}_{dog}$ represents the first two premises of S_{dog}:

$$\begin{aligned} \mathit{police\_dog}'(X) &\leftarrow \mathit{vicious}(X) \wedge \neg \mathit{ab}_{dog'}(X), \\ \mathit{police\_dog}(X) &\leftarrow \neg \mathit{police\_dog}'(X), \\ \mathit{ab}_{dog'}(X) &\leftarrow \bot, \\ \mathit{highly\_trained}(a) &\leftarrow \top, \\ \mathit{vicious}(a) &\leftarrow \top. \end{aligned}$$

We encode the first two premises of the other syllogisms similarly.

$\mathbf{S}_{vit}$ PREMISE 2 states that there are some vitamin tablets, which are inexpensive. We presuppose that there exists something, a, for which these facts are true:

$$\mathit{vitamin}(a) \leftarrow \top \qquad \text{and} \qquad \mathit{inex}(a) \leftarrow \top.$$

Additionally, it is commonly known that

The purpose of vitamin tablets is to aid nutrition.

This belief and the clause representing Premise 1 leads to

If something is a vitamin tablet, then it is abnormal
(regarding Premise 1 *of* S_{vit}*).*

The program $\mathcal{P}_{vit}$ represents Premise 1 and Premise 2 together with the background knowledge:

$$\begin{aligned} nutritional'(X) &\leftarrow inex(X) \wedge \neg ab(X), \\ nutritional(X) &\leftarrow \neg nutritional'(X), \\ ab(X) &\leftarrow \bot, \\ ab(X) &\leftarrow vitamin(X), \\ vitamin(a) &\leftarrow \top, \\ inex(a) &\leftarrow \top. \end{aligned}$$

$nutritional(X)$, $nutritional'(X)$ denote X is nutritional, not nutritional, resp.

$\mathbf{S}_{rich}$ Premise 2 states that there are some hard workers who are rich. We presuppose that there is someone, let us say, a, for which these facts are true:

$$hard\_worker(a) \leftarrow \top \qquad \text{and} \qquad rich(a) \leftarrow \top.$$

$\mathcal{P}_{rich}$ represents Premise 1 and Premise 2 of S_{rich}:

$$\begin{aligned} mil'(X) &\leftarrow hard\_worker(X) \wedge \neg ab(X), \\ mil(X) &\leftarrow \neg mil'(X), \\ ab(X) &\leftarrow \bot, \\ rich(a) &\leftarrow \top, \\ hard\_worker(a) &\leftarrow \top. \end{aligned}$$

$mil(X)$ and $mil'(X)$ denote X is a millionaire and not a millionaire, resp.

$\mathbf{S}_{cig}$ Premise 2 states that there are some cigarettes, which are inexpensive. Again, we presuppose that there is something, a, for which these facts are true:

$$cig(a) \leftarrow \top \qquad \text{and} \qquad inex(a) \leftarrow \top.$$

Additionally, it is commonly known that

Cigarettes are addictive.

This belief and the clause representing Premise 1 leads to

If something is a cigarette, then it is abnormal
(regarding Premise 1 *of* S_{cig}*).*

As discussed by Evans et al. [10], humans seem to have a background knowledge or belief, which might provide the motivation on whether to validate a syllogism. A direct representation of Premise 2 is

There exists a cigarette, which is inexpensive. (1)

Additionally, in the context of Premise 1, we assume that

Compared to other addictive things, cigarettes are inexpensive. (2)

which implies (1) and biases the reasoning towards a representation. Note that (2) only implies (1) because we understand quantifiers with existential import, i.e., *for all* implies *there exists*. This is a reasonable assumption when modeling human reasoning, as in natural language we normally do not quantify over things that don't exist. Furthermore, Stenning and an Lambalgen [30] have shown that humans require existential import for the conditional to be true.
The belief bias represented by (2), together with the idea to represent conditionals by a normal default permission for implication, leads to the conditional

If something is a cigarette and not abnormal, then it is inexpensive. (3)
Nothing (as a rule) is abnormal (regarding (3)*).*

$\mathcal{P}_{cig}$ represents the first two premises and the background knowledge in S_{cig} as follows:

$$\begin{aligned} \mathit{addictive}'(X) &\leftarrow \mathit{inex}(X) \wedge \neg \mathit{ab}_{\mathit{add}'}(X), \\ \mathit{addictive}(X) &\leftarrow \neg \mathit{addictive}'(X), \\ \mathit{ab}_{\mathit{add}'}(X) &\leftarrow \bot, \\ \mathit{ab}_{\mathit{add}'}(X) &\leftarrow \mathit{cig}(X), \\ \mathit{inex}(X) &\leftarrow \mathit{cig}(X) \wedge \neg \mathit{ab}_{\mathit{inex}}(X), \\ \mathit{ab}_{\mathit{inex}}(X) &\leftarrow \bot, \\ \mathit{cig}(a) &\leftarrow \top, \\ \mathit{inex}(a) &\leftarrow \top, \end{aligned}$$

$\mathit{addictive}(X)$ and $\mathit{addictive}'(X)$ denote X is addictive and not addictive, resp.

4 Reasoning with Respect to Least Models

This section deals with Stenning and van Lambalgen's second step, and discusses where a possible belief bias during the reasoning procedure can influence the result. We show how to compute the least model for each case and discuss whether it represents the participants' conclusions shown in the introduction.

4.1 Valid Arguments

$\mathcal{P}_{dog}$ represents S_{dog}. Its weak completion, $\mathsf{wc}\,\mathsf{g}\,\mathcal{P}_{dog}$, is:

$$\begin{aligned} \mathit{police\_dog}'(a) &\leftrightarrow \mathit{vicious}(a) \wedge \neg \mathit{ab}_{\mathit{dog}'}(a), \\ \mathit{police\_dog}(a) &\leftrightarrow \neg \mathit{police\_dog}'(a), \\ \mathit{ab}_{\mathit{dog}'}(a) &\leftrightarrow \bot, \\ \\ \mathit{highly\_trained}(a) &\leftrightarrow \top, \\ \mathit{vicious}(a) &\leftrightarrow \top. \end{aligned}$$

Its least model is:

$$\langle \{\mathit{highly\_trained}(a), \mathit{vicious}(a), \mathit{police\_dog}'(a)\}, \{\mathit{police\_dog}(a), \mathit{ab}_{\mathit{dog}'}(a)\}\rangle.$$

This model entails the Conclusion of S_{dog}, *some highly trained dogs are not police dogs*. According to [10], S_{dog} is logically valid and psychologically believable. No conflict arises neither at the psychological nor at the logical level, and the majority concludes that this syllogism holds, which complies with the least model of $\mathsf{wc}\,\mathsf{g}\,\mathcal{P}_{dog}$.

The psychological results of the second syllogism, S_{vit}, indicate that there seems to be two kinds of participants each taking a different interpretation of the statements. The group, which validated the syllogism, was not influenced by the bias with respect to nutritional things. Accordingly, the logic program that represents their view, corresponds to $\mathcal{P}_{vit} \setminus \{\mathit{ab}(X) \leftarrow \mathit{vitamin}(X)\}$. The weak completion of $\mathsf{g}\,\mathcal{P}_{vit} \setminus \{\mathit{ab}(a) \leftarrow \mathit{vitamin}(a)\}$ is:

$$\begin{aligned} \mathit{nutritional}'(a) &\leftrightarrow \mathit{inex}(a) \wedge \neg \mathit{ab}(a), \\ \mathit{nutritional}(a) &\leftrightarrow \neg \mathit{nutritional}'(a), \\ \mathit{ab}(a) &\leftrightarrow \bot, \\ \\ \mathit{vitamin}(a) &\leftrightarrow \top, \\ \mathit{inex}(a) &\leftrightarrow \top. \end{aligned}$$

The corresponding least model is:

$$\langle \{\mathit{vitamin}(a), \mathit{inex}(a), \mathit{nutritional}'(a)\}, \{\mathit{nutritional}(a), \mathit{ab}(a)\}\rangle,$$

which entails the conclusion, that *some vitamin tables are not nutritional*, and indeed we can conclude that this syllogism is valid.

The other interpretation, where participants' chose not to validate the syllogism, is the group who has apparently been influenced by their belief. Their interpretation of S_{vit} is represented by $\mathcal{P}_{vit}$. Its weak completion, $\mathsf{wc}\,\mathsf{g}\,\mathcal{P}_{vit}$, is:

$$\begin{aligned} \mathit{nutritional}'(a) &\leftrightarrow \mathit{inex}(a) \wedge \neg \mathit{ab}(a), \\ \mathit{nutritional}(a) &\leftrightarrow \neg \mathit{nutritional}'(a), \\ \mathit{ab}(a) &\leftrightarrow \bot \vee \mathit{vitamin}(a), \\ \mathit{vitamin}(a) &\leftrightarrow \top, \\ \mathit{inex}(a) &\leftrightarrow \top. \end{aligned}$$

Its least model is:

$$\langle\{\mathit{vitamin}(a), \mathit{inex}(a), \mathit{nutritional}(a), \mathit{ab}(a)\}, \{\mathit{nutritional}'(a)\}\rangle.$$

The Conclusion of S_{vit} is not entailed. According to [10], S_{vit} is logically valid but psychologically unbelievable. There arises a conflict at the psychological level because we generally assume that the purpose of vitamin tablets is to aid nutrition. The participants who have been influenced by this belief concluded that the syllogism does not hold, which complies with the least model of $\mathsf{lm}_{\text{L}}\mathsf{wc}\,\mathsf{g}\,\mathcal{P}_{vit}$.

4.2 Invalid Arguments

The third and the fourth cases of the syllogistic reasoning task cannot be modeled straightforwardly as the first two cases. We assume that the belief has an influence on the procedural part, that is, the reasoning process is biased. We can model this by abduction, which has been explained in Section 2.5.

$\mathcal{P}_{rich}$ represents S_{rich}. Its weak completion, $\mathsf{wc}\,\mathsf{g}\,\mathcal{P}_{rich}$, is:

$$\begin{aligned} \mathit{mil}'(a) &\leftrightarrow \mathit{hard\_worker}(a) \wedge \neg \mathit{ab}(a),\\ \mathit{mil}(a) &\leftrightarrow \neg \mathit{mil}'(a),\\ \mathit{ab}(a) &\leftrightarrow \bot,\\ \\ \mathit{rich}(a) &\leftrightarrow \top,\\ \mathit{hard\_worker}(a) &\leftrightarrow \top. \end{aligned}$$

Its least model is:

$$\langle\{\mathit{hard\_worker}(a), \mathit{rich}(a), \mathit{mil}'(a)\}, \{\mathit{ab}(a), \mathit{mil}(a)\}\rangle,$$

and states nothing about the Conclusion, *some millionaires are not rich people*. Actually, the Conclusion in S_{rich} states something, which contradicts Premise 2, and thus needs to be about something that cannot be the previously introduced constant a. According to our background knowledge, we know that millionaires exist. Let us formulate this as an observation, let's say about b: $\mathcal{O} = \{\mathit{mil}(b)\}$. If we want to allow to suppose truth or falsity of something about b with respect to $\mathcal{P}_{rich}$, say about the truth of $\mathit{hard\_worker}(b)$, we can no longer assume that $\mathsf{CONSTANTS} = \mathsf{constants}(\mathcal{P}_{rich})$, because $\mathcal{A}_{\mathsf{g}\,\mathcal{P}_{rich}}$ would not contain any facts about b. Therefore, we specify that the new set of constants in consideration is $\mathsf{CONSTANTS} = \{a, b\}$. $\mathsf{g}\,\mathcal{P}_{rich}$ with respect to $\mathsf{CONSTANTS}$ contains additionally three more clauses:

$$\begin{aligned} \mathit{mil}'(b) &\leftarrow \mathit{hard\_worker}(b) \wedge \neg \mathit{ab}(b),\\ \mathit{mil}(b) &\leftarrow \neg \mathit{mil}'(b),\\ \mathit{ab}(b) &\leftarrow \bot. \end{aligned}$$

The set of abducibles, $\mathcal{A}_{\mathsf{g}\,\mathcal{P}_{rich}}$, contains the following clauses:

$$\mathit{hard\_worker}(b) \leftarrow \top, \qquad \mathit{hard\_worker}(b) \leftarrow \bot.$$

$\mathcal{E} = \{hard\_worker(b) \leftarrow \bot\}$ is the only explanation for $\mathcal{O}$. $\mathsf{wc\,g}\,(\mathcal{P}_{rich} \cup \mathcal{E})$ contains:

$$\begin{aligned} mil'(b) &\leftrightarrow hard\_worker(b) \wedge \neg ab(b),\\ mil(b) &\leftrightarrow \neg mil'(b),\\ ab(b) &\leftrightarrow \bot,\\ \\ hard\_worker(b) &\leftrightarrow \bot. \end{aligned}$$

Its least model, where $\mathsf{lm_L wc\,g}\,(\mathcal{P}_{rich} \cup \mathcal{E}) = \langle I^\top, I^\bot \rangle$, contains:

$$\begin{aligned} I^\top &= \{mil(b)\},\\ I^\bot &= \{ab(b), mil'(b), hard\_worker(b)\}. \end{aligned}$$

As this model does not confirm the Conclusion it does not validate S_{rich}. According to [10] this case is quite easy to solve, because it is neither logically valid nor believable. Almost no one validated S_{rich}, which complies with the least model of $\mathsf{wc\,g}\,(\mathcal{P}_{rich} \cup \mathcal{E})$.

$\mathcal{P}_{cig}$ represents S_{cig}. Its weak completion, $\mathsf{wc\,g}\,\mathcal{P}_{cig}$, is:

$$\begin{aligned} addictive'(a) &\leftrightarrow inex(a) \wedge \neg ab_{add'}(a),\\ addictive(a) &\leftrightarrow \neg addictive'(a),\\ ab_{add'}(a) &\leftrightarrow \bot \vee cig(a),\\ \\ cig(a) &\leftrightarrow \top,\\ \\ inex(a) &\leftrightarrow (cig(a) \wedge \neg ab_{inex}(a)) \vee \top,\\ ab_{inex}(a) &\leftrightarrow \bot. \end{aligned}$$

Its least model of the weak completion is:

$$\langle \{cig(a), inex(a), \mathbf{addictive(a)}, ab_{add'}(a)\}, \{addictive'(a), ab_{inex}(a)\}\rangle,$$

which, similarly to the previous case, does not state anything about the Conclusion, *some addictive things are not cigarettes*. Again, the Conclusion of S_{cig} is something, which cannot be about a. According to our background knowledge, we know that addictive things exist. Let us formulate this again as an observation, say about b: $\mathcal{O} = \{addictive(b)\}$, which needs to be explained. In order to generate an explanation for $\mathcal{O}$, let us define $\mathsf{CONSTANTS} = \{a, b\}$. $\mathsf{g}\,\mathcal{P}_{rich}$ with respect to $\mathsf{CONSTANTS}$ now additionally contains five more clauses:

$$\begin{aligned} addictive'(b) &\leftarrow inex(b) \wedge \neg ab_{add'}(b),\\ addictive(b) &\leftarrow \neg addictive'(b),\\ ab_{add'}(b) &\leftarrow \bot,\\ ab_{add'}(b) &\leftarrow cig(b),\\ \\ inex(b) &\leftarrow cig(b) \wedge \neg ab_{inex}(b),\\ ab_{inex}(b) &\leftarrow \bot. \end{aligned}$$

Given $\mathsf{g}\,\mathcal{P}_{cig}$, the set of abducibles, $\mathcal{A}_{\mathsf{g}\,\mathcal{P}_{cig}}$, contains the following clauses:

$$cig(b) \leftarrow \top, \qquad cig(b) \leftarrow \bot.$$

$\mathcal{O}$ is true if $addictive'(b)$ is false, which is false if $inex(b)$ is false or $ab_{add'}(b)$ is true. $inex(b)$ is false if $cig(b)$ is false and $ab_{add'}(b)$ is true if $cig(b)$ is true. For $\mathcal{O}$ we have two minimal explanations, $\mathcal{E}_\bot = \{cig(b) \leftarrow \bot\}$ and $\mathcal{E}_\top = \{cig(b) \leftarrow \top\}$. The weak completion of $\mathsf{g}\,(\mathcal{P}_{cig} \cup \mathcal{E}_\bot)$ contains:

$$\begin{aligned} addictive'(b) &\leftrightarrow inex(b) \wedge \neg ab_{add'}(b),\\ addictive(b) &\leftrightarrow \neg addictive'(b),\\ ab_{add'}(b) &\leftrightarrow \bot \vee cig(b),\\ \\ inex(b) &\leftrightarrow cig(b) \wedge \neg ab_{inex}(b),\\ ab_{inex}(b) &\leftrightarrow \bot,\\ \\ cig(b) &\leftrightarrow \bot. \end{aligned}$$

Its least model, where $\mathsf{lm}_{\text{L}}\mathsf{wc}\,\mathsf{g}\,(\mathcal{P}_{cig} \cup \mathcal{E}_\bot) = \langle I^\top, I^\bot \rangle$ contains:

$$\begin{aligned} I^\top &= \{addictive(b)\},\\ I^\bot &= \{cig(b), inex(b), ab_{add'}(b), ab_{inex}(b)\}, \end{aligned}$$

which entails the Conclusion of S_{cig}. As $\mathcal{E}_\top$ is yet another explanation for $\mathcal{O}$, the Conclusion, that b is not a cigarette, only follows credulously. S_{cig} is logically invalid but psychologically believable and therefore causes a conflict [10]: S_{cig} does not follow logically from the premises; however, people are biased and search for a model, which confirms their beliefs. Therefore, the majority concluded that this syllogism holds, which complies with the least model of $\mathsf{wc}\,\mathsf{g}\,(\mathcal{P}_{add} \cup \mathcal{E}_\bot)$.
In [26,27], we show an extension of this case, where the conclusion follows skeptically. With help of meta predicates, we specify that the first premise describes the usual and the second premise describes the exceptional case. That is, an inexpensive cigarette is meant to be the exception not the rule, in the context of things that are addictive and expensive.

5 Conclusion

The weak completion semantics has shown to successfully model various human reasoning episodes [4,5,7,18,26,27]. This paper presents yet another human reasoning task modeled under the weak completion semantics. As in our previous formalizations, we follow Stenning and van Lambalgen's two step approach. We motivate our assumptions based on results from Psychology, where syllogisms in human reasoning have been investigated extensively in the past decades.
As has been shown in the previous formalizations, the advantage of the weak completion semantics over other logic programming approaches, is, that undefined atoms stay unknown, instead of becoming false. The syllogistic reasoning tasks, which have been discussed in the literature so far, have never accounted to give the option 'I don't know' to the participants. As has been discussed in [24], participants who say that no valid conclusion follows, might have problems to actually find a conclusion easily and possibly mean that they simply do not know.

They also point to [28], who suggest that, if a conclusion is stated as being not valid, this could just simply mean that the reasoning process is exhausted. An experimental study, which would allow the participants to distinguish between 'I don't know' and 'not valid', might possibly give us more insights about their reasoning processes and identify where exactly the belief bias takes effect.

6 Acknowledgements

Many thanks to Steffen Hölldobler and Luís Moniz Pereira for valuable feedback.

References

1. J. Adler and L. Rips. *Reasoning: Studies of Human Inference and Its Foundations.* Cambridge University Press, 2008.
2. L. J. Chapman and J. P. Chapman. Atmosphere effect re-examined. *Journal of Experimental Psychology*, 58(3):220–6, 1959.
3. K. L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, volume 1, pages 293–322. Plenum Press, New York, NY, 1978.
4. 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, Austin, TX, 2012.
5. E.-A. Dietz, S. Hölldobler, and M. Ragni. A computational logic approach to the abstract and the social case of the selection task. In *11th International Symposium on Logical Formalizations of Commonsense Reasoning*, 2013.
6. E.-A. Dietz, S. Hölldobler, and C. Wernhard. Modeling the suppression task under weak completion and well-founded semantics. *Journal of Applied Non-Classical Logics*, 2013.
7. E.-A. Dietz, S. Hölldobler, and C. Wernhard. Modeling the suppression task under weak completion and well-founded semantics. *Journal of Applied Non-Classsical Logics*, 24(1–2):61–85, 2014.
8. J. Evans. Thinking and believing. *Mental models in reasoning*, 2000.
9. J. Evans. Biases in deductive reasoning. In R. Pohl, editor, *Cognitive Illusions: A Handbook on Fallacies and Biases in Thinking, Judgement and Memory.* Psychology Press, 2012.
10. J. Evans, J. L. Barston, and P. Pollard. On the conflict between logic and belief in syllogistic reasoning. *Memory & Cognition*, 11(3):295–306, 1983.
11. J. Evans, S. Handley, and C. Harper. Necessity, possibility and belief: A study of syllogistic reasoning. *Quarterly Journal of Experimental Psychology*, 54(3):935–958, 2001.
12. J. S. Evans. *Bias in human reasoning - causes and consequences.* Essays in cognitive psychology. Lawrence Erlbaum, 1989.
13. M. Fitting. A Kripke-Kleene semantics for logic programs. *Journal of Logic Programming*, 2(4):295–312, 1985.
14. A. Garnham and J. Oakhill. *Thinking and Reasoning.* Wiley, 1994.
15. S. Hölldobler. *Logik und Logikprogrammierung 1: Grundlagen.* Kolleg Synchron. Synchron, 2009.

16. S. Hölldobler and C. D. Kencana Ramli. Logic programs under three-valued Łukasiewicz semantics. In P. M. Hill and D. S. Warren, editors, *Logic Programming, 25th International Conference, ICLP 2009*, volume 5649 of *Lecture Notes in Computer Science*, pages 464–478, Heidelberg, 2009. Springer.
17. S. Hölldobler and C. D. Kencana Ramli. Logics and networks for human reasoning. In C. Alippi, M. M. Polycarpou, C. G. Panayiotou, and G. Ellinas, editors, *International Conference on Artificial Neural Networks, ICANN 2009, Part II*, volume 5769 of *Lecture Notes in Computer Science*, pages 85–94, Heidelberg, 2009. Springer.
18. S. Hölldobler, T. Philipp, and C. Wernhard. An abductive model for human reasoning. In *Logical Formalizations of Commonsense Reasoning, Papers from the AAAI 2011 Spring Symposium*, AAAI Spring Symposium Series Technical Reports, pages 135–138, Cambridge, MA, 2011. AAAI Press.
19. P. N. Johnson-Laird. *Mental models: towards a cognitive science of language, inference, and consciousness*. Harvard University Press, Cambridge, MA, 1983.
20. P. N. Johnson-Laird and R. M. Byrne. Deduction. 1991.
21. A. C. Kakas, R. A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1993.
22. J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag New York, Inc., New York, NY, USA, 1984.
23. J. Łukasiewicz. O logice trójwartościowej. *Ruch Filozoficzny*, 5:169–171, 1920. English translation: On three-valued logic. In: Łukasiewicz J. and Borkowski L. (ed.). (1990). *Selected Works*, Amsterdam: North Holland, pp. 87–88.
24. S. Newstead, S. Handley, and E. Buck. Falsifying mental models: Testing the predictions of theories of syllogistic reasoning. *Memory & Cognition*, 27(2):344–354, 1999.
25. S. E. Newstead and R. A. Griggs. Fuzzy quantifiers as an explanation of set inclusion performance. *Psychological Research*, 46(4):377–388, 1984.
26. L. M. Pereira, E.-A. Dietz, and S. Hölldobler. A computational logic approach to the belief bias effect. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning*, 2014.
27. L. M. Pereira, E.-A. Dietz, and S. Hölldobler. Contextual abductive reasoning with side-effects. volume 14, pages 633–648, 2014.
28. T. A. Polk and A. Newell. Deduction as verbal reasoning. *Psychological Review*, 102(3):533–566, 1995.
29. K. Stenning and M. van Lambalgen. Semantic interpretation as computation in nonmonotonic logic: The real meaning of the suppression task. *Cognitive Science*, 6(29):916–960, 2005.
30. K. Stenning and M. van Lambalgen. *Human Reasoning and Cognitive Science*. A Bradford Book. MIT Press, Cambridge, MA, 2008.
31. A. Van Gelder, K. A. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):619–649, 1991.
32. M. Wilkins. he effect of changed material on the ability to do formal syllogistic reasoning. 16(102), 1928.
33. R. S. Woodworth and S. B. Sells. An atmosphere effect in formal syllogistic reasoning. *Journal of Experimental Psychology*, 18(4):451–60, 1935.

There Is No One Logic to Model Human Reasoning: the Case from Interpretation

Alexandra Varga[1], Keith Stenning[2], Laura Martignon[3]
[1] University of Giessen, Alexandra.Varga@psychol.uni-giessen.de
[2] University of Edinburgh, K.Stenning@ed.ac.uk
[3] Pädagogische Hochschule Ludwigsburg, martignon@ph-ludwigsburg.de

Abstract. The paper discusses a multiple-logics proposal for cognitive modelling of reasoning processes. It describes a staged view of human reasoning which takes interpretation seriously, and provides a non-technical introduction to a logic fit for modelling interpretative processes – Logic Programming. It summarises some results of the multiple-logics approach obtained with modelling psychological data, and with empirical tests of a combined use of reasoning strategies by human subjects. It draws some interim conclusions, and proposes avenues for future research.

1 Introduction

We are interested in computational models for human reasoning at the performance level. Cognitive modelling amounts to the use of some formalism in order to provide a productive description of cognitive phenomena. "Productive" has an explanatorily-oriented, twofold meaning: on the one hand, the description helps a better understanding of the phenomena, and second, it can be used to generate empirical predictions aiming to refine the theory that backs the model. By 'performance model' we imply that the formalism is actually used by real human agents in real reasoning contexts, wittingly or not. The reasoning process at the psychological level is an instantiation of the formal model. The 'wittingly or not' specification points to the need to include those forms of reasoning which are merely implicit, or below-awareness. A model of such reasoning processes involved in, e.g., understanding an utterance in one's native language, amounts to expressing these unwitting processes and subsequent behaviors 'as if' they were the result of computations expressed in a formal language.

We propose that the highest level of explanatory productivity, or information gain, can be achieved by a multiple-logics approach to cognitive modelling. In brief, this is so because of the complex differences between different kinds of reasoning which cannot be adequately captured by the formal properties of a single system. A multiple-logics approach is mandated because an all-purpose logic of human reasoning conflicts with the many things that humans may use reasoning for [1], e.g., to prove beyond reasonable doubt that the accused is guilty of the crime, to make the child understand the moral behind the story of the Ant and the Grasshopper. This would remain so even if all of the many formal candidates could be reconstructed in a single highly expressive logical system, because its use in human reasoning would be too resource-demanding; in other words, computational efficiency is an opportunity cost of expressive power. Performance models should at all points keep the balance.

Cognitive modelling from a multiple-logics perspective is also sanctioned by the history of psychological research. For instance, the withdrawal of previously validly derived conclusions

when new information is added to the premise set [6], does not afford description in terms of a monotonic formalism such as classical logic. Everyday reasoning is most often non-monotonic. However monotonicity can be triggered by, e.g., by task instructions that create a dispute setting [1]. The bottom line is that different forms of reasoning, meant to achieve different goals, should be modelled in a formalism that bears the context-dependent properties of the inferences.

The main purpose of the current paper is to review the 'bridging potential' of a multiple-logics approach. The roadmap is as follows. We start in Section 2 by introducing the distinction between two kinds of reasoning, interpretation and further reasoning from that interpretation. We introduce the working example of a formalism, namely Logic Programming, and emphasise its application to interpretative processes. The remainder of the paper develops the argument based on taking interpretation seriously. Section 3 describes in detail a case of pre-linguistic implicit reasoning and summarises the modelling work in [35]. It shows how the logical and psychological aspects of reasoning can be integrated. Section 4 exemplifies the multiple-logics approach by describing the use of Logic Programming and fast and frugal heuristics for better understanding subjects' reasoning processes; we hemphasize the consequential methodological advantage of theoretical unification of the fields of reasoning and of judgement and decision-making. We end with some suggestions for further development of the multiple-logics approach, based on collaborative modelling among different systems.

2 The Proposed View of Reasoning and an Example of Formal Implementation

We are mostly concerned with everyday reasoning, i.e., the processes involved in habitual activities such as conversations, disputes, stories, demonstrations, etc. Stenning and van Lambalgen [32] set forth two kinds of processes: reasoning *to* an interpretation of the context, and reasoning *from* that interpretation.

Language processing is perhaps the clearest instantiation of the two reasoning stages. When speakers ask their interlocutors a question, they must first process the string of words in the context (linguistic and extra-linguistic) and produce an interpretation or model of it; in order to achieve the default purpose of communication fast and efficiently, these computations are aimed at the one model intended by the speakers. Because of this assumption that the right interpretation is in terms of what "(s)he must have meant to ask", the interpretative process is a paradigmatic case of credulous or cooperative reasoning. But this is only the beginning of the story. Should the first interpretation be unsatisfactory, e.g., being asked by one's life-time partner the question "How old are you?", hearers might resort to compensatory mechanisms, e.g., taking into account metaphorical meanings. Once a model is available the interlocutors can start to compute what they believe to be the contextually appropriate answer – this is reasoning *from* the interpretation. The reasoning path is not linear, e.g., additional utterances usually require model updates or re-computations of the initial discourse model.

The focus of cooperative interpretation on constructing a minimal contextual model can be described as the use of closed-world assumptions to frame the inferential scope [32, 34]. The basic format is the assumption for reasoning about abnormalities (CWA), which prescribes that,

if there is no positive information that a given event must occur, one may assume it does not occur. These 'given events' are abnormalities with respect to the smooth, habitual running of a process; for example, a metaphorical interpretation is abnormal with respect to the literal one, and thus disregarded in minimal model construction. A conditional abnormality list is attached to each conditional; the list should be viewed as at the back of reasoners' minds [35]. That is, abnormalities are reasoned about only when evidence arrives (otherwise the assumption would be self-defeating). CWAs require construction of a minimal interpretation based only on what that is derivable from explicitly mentioned information. This is why they 'frame' [25] reasoning to manageable dimensions. Interpretation with CWAs is thus a plausible candidate to model the reasoning of agents with limited memory and computational resources in real-time.

The CWA is captured by all three parameters of Logic Programming – LP (syntactic, semantic, and definition of validity), a computational logic designed for automated planning [20]; it is the formal system that we use to instantiate our proposal. We view the utilization of such a formalism to model human inferences as a contribution to the bridge that this workshop seeks to build. Its cognitive plausibility has been shown from a variety of perspectives: it has been used to construct a formal semantics of tense [34], it helped understanding the formal structure of various cognitive tasks (e.g., Wason's task, the suppression task, the false belief task – dealt with in [32]), which in turn led to fine-grained experimental predictions (see [2] for a review).

Whereas an extensional formal approach deals with sets of items and with relations between those, an intensional one deals with characteristics and constitutive properties of the items in these classes. Relatedly, Logic Programming is an intensional formalism because its completion semantics is not directly truth-functional. We adopt the formal description of the logic set forth in [32, 34].

The CWA provides the notion of valid inference in LP, as truth preserving inferences in minimal models where nothing abnormal is the case. Relatedly, the LP conditional is represented as $p \ \& \sim ab \rightarrow q$ – "If p and nothing abnormal is the case, then q". Closed-world reasoning manifests itself in that, unless positive evidence (i.e., either explicit mentioning, or facts inferable from the database with the LP syntactic rules), the negation of the abnormality conjunct holds true. The syntactic expression of closed-world reasoning is the derivation rule of negation-as-failure – NAF. If a fact can be represented as the consequence of falsum $\perp$, thus it cannot be derived by backwards reasoning from program clauses, its negation is assumed true and the fact is thereby eliminated from the derivation. When resolving the query q given a program with clauses $p \ \& \sim ab \rightarrow q$ and $\perp \rightarrow ab$, q reduces to $p \ \& \sim ab$, from which p is derived by means of NAF. Use of negation-as-failure in derivations means that derivation checks if a query can be made true in a minimal model of the program. A minimal model is a 'closed world' in the sense that facts not forced to occur by inferences over the program clauses using the LP syntactic rules are assumed not to occur. The system's three-valued Kleene semantics (procedural in nature) warrants the construction of a unique minimal model, which is the only

interpretation of concern of the current reasoning input[1]. Minimal models are provided by a semantic restriction of logic program clauses, called completion. It is obtained by introducing disjunction between all the bodies (antecedents) with the same head (consequent) in a program, and substituting implication with equivalence between the disjunctive body and the head.

The use of CWAs in interpretation is only the beginning of the intensional, or meaning-directed part of reasoning. Computations of a minimal preferred interpretation have been described at the psychological level in [32] as an interaction between the knowledge base of long-term memory and incoming input (e.g., new discourse statements, or new observations), in search for relevant information. Novel input may override the assumption and lead to subsequent model extensions by inclusion of the encountered abnormalities. This is a constitutively difficult task because at any give point, the vast majority of the long-term memory knowledge base is irrelevant. The Kleene semantics models this phenomenon by setting propositions to value U (undecided), which can develop to either T or F as a result of further inferences. The extensions of minimal models are also minimal. LP reasoning is thus inherently non-monotonic. Because of this it aligns with both the efficiency and the flexibility of everyday reasoning.

Let us relate this to the empirical sciences of human reasoning. What is most missing in the literature is detailed consideration of a positive account of the mental processes of interpretation, and of the interplay of the two forms of reasoning. In psychological experiments, when subjects are presented with the premises of a syllogism, they must first make sense of the information presented in order to be able to perform the inferences they are asked for. Reasoning to an interpretation must be acknowledged at face value by cognitive scientists when operationalizing theories into testable hypothesis, when deciding on the standards for response evaluation, when interpreting the empirical data, and obviously, when setting forth computational models for better understanding the cognitive phenomena. Despite a long period of utter neglect[2], recent work in the psychology of reasoning has started to acknowledge the role of interpretation, e.g., [18, 29]. This is a salutary new direction which calls for development of its consequences in modelling; consequently we argue that intensional formalisms are a necessary (though certainly not sufficient) ingredient of models for reasoning.

3 Logic for Modelling Implicit Reasoning

[1] Hölldobler and Kencana Ramli [15] criticised the Kleene semantics used in [35] by reference to modelling the suppression task [6]; these authors propose using the Lukasiewicz semantics instead. A technical rejoinder is available in the Appendix. Here we wish to emphasize that Byrne's task calls for a cooperative interpretation of the experimental material. The syntactic restrictions on LP conditionals on the other hand, e.g., non-iterability, allow completion to succeed in providing a minimal model as a pre-fixed point in a cooperative context, where epistemic trust is justified.

[2] A notable exception here is [17].

In a series of seminal studies with the head-touch task [12, 19], pre-linguistic infants have been shown to engage in selective imitative learning. We first introduce the experiment. After showing behavioral signs of being cold and wrapping a scarf around her shoulders, an adult demonstrates to 14-month-olds an unfamiliar head touch as a new means to activate a light-box. Half the infants see that the demonstrator's hands are occupied holding the scarf while executing the head action (Hands-Occupied condition – HO), the other half observe her acting with hands visibly free after having knotted the scarf (Hands-Free condition – HF). After a one-week delay subjects are given the chance to act upon the light-box themselves. They all attempt to light-up the lamp; however reenactment of the observed novel means action with the head is selective: 69% of the infants in the HF, and only 21% in the HO. More, [19] have shown that selectivity is contingent on a communicative action demonstration. This involves that throughout the demonstration session the experimenter behaves prosocially towards the infant, using both verbal and non-verbal communicative-referential cues. When the action was presented in a communicative context, the previous results were replicated. However, when the novel action is performed aloof, without infant-directed gaze or speech, the reenactment rate is always below chance level, and there is no significant difference between the HO and HF conditions. Gergely and his colleagues propose that infants' selectivity is underlain by a normative understanding of human actions with respect to goals. That is, infants learn some means actions but not others depending on the interpretation in terms of goals (teleological) afforded by the observed context.

The model set forth in [35] adopts this inferential perspective from the standpoint of multi-level teleology, i.e., a broad representation of goals that covers a whole range from physical goals (e.g., turning on a light-box) to higher-order intentions and meta-goals (e.g., the adult's teaching intention, infants' intentions to understand and to learn what is new and relevant)[3]. The inferential engine is constraint logic programming (CLP). The model gives voice to infants' interpretation of observations and to planning their own actions in the test phase. This voice is spelled out in the language of the event calculus [32] – 14-month-olds' observations and relevant bits of causal knowledge are represented as event calculus program clauses, e.g., *Initially*(communication) – agent exhibits infant-directed communicative behaviour, *Terminates*(contact, light-activity, t_k[4]) – contact is the culminating point of the light-box directed activity. Their teleological processing is called for and guided by the epistemic goals to understand and to learn, represented as integrity constraints [21, 34]. CLP allows to express higher-order goals as integrity constraints. These are peculiar conditional clauses which impose local (contextual) norms on the computations; they are universally quantified (but see footnote 6). For instance, IF ?*Initialy*(communication) succeeds THEN ?*HoldsAt*(teach$_f$, t) succeeds[5]

[3] Multi-level teleology is based on Kowalski's [21] distinction between achievement physical goals, and maintenance goals.

[4] t_k is a temporal constant.

[5] Note that the semantics of the conditional in integrity constraints is an unsettled issue [21]. [36] adopted a classical semantics.

expresses the assignment of a pedagogical intention to the observed agent conditional on her infant directed communicative behavior. When the antecedent is made true by the environment, i.e., in the communicative conditions, the young reasoner must act such that the goal expressed in the consequent becomes true. "teachf" is a parameterised fluent, i.e., a variable that must be specialized to a constant in the course of resolution. Infants' propensity for teleological understanding has been represented as an unconditional integrity constraint, namely ?*Happens*(x,t), *Initiates*(x,f(x),t), gx = f(x) succeeds. It demands assigning a concrete goal to an observed instrumental behaviour, i.e. finding a value for the Skolem function [6] *f(x)*. The requirement succeeds makes an existential claim with respect to a physical goal, i.e. there is such a state as *g*, which is a function *f(x)* of an action *x*.

Contextual interpretation amounts to finding the means – ends structure. Given the program clause *Initially*(communication) in the communicative condition, infants assign the adult the pedagogical intention expressed in the consequent of the constraint; further computations must unify parameter *f* with a concrete observed fluent, which is deemed to count as new and relevant information. Infants goal assignment to the agent's object-directed activity is done by resolving the unconditional constraint mentioned above. A successful unification is sought by specializing the function *f(x)* to a constant fluent from the narrative of events, given an evaluation of the causal relations available in the contextual causal model. The model shows how backward derivations from the constraint output the solution that the state *light-on* is the goal of contacting the light-box with the head, which is the culminating point of the observed activity. This represents infants' teleological conjecture, expected to render the action context understandable.

Interpretation is then subserved by a plan simulation algorithm – infants verify the goal conjecture by considering what they themselves would have done in order to achieve the goal *light-on*. This view of inferential plan simulation, and not merely motor simulation as traditionally construed, e.g., [26], is one of the main innovations brought about by this use of CLP for modelling. In the HO condition the mismatch between infants' closed-world plan calling for default hand contact, and observation of head contact is resolved by reasoning that the adult must use her hands for another goal, i.e., to hold the scarf in order not to be cold. The situation is fully understandable, hence infants specialize parameter *f* in ?*HoldsAt*(teachf ,t) to the object's newly inferred function, *light-on*.

The HO simulationist explanation does not work in the HF condition – the adult's free hands are not required to fulfill any different goal, so why it is that she does not use them to activate the object? Infants then integrate the adult's previously assigned pedagogical intentions in the explanatory attempt. Assigning a pedagogical intention to the reliable adult's otherwise incomprehensible head action renders it worth learning. Although touching a light box with the head in order to light it up may not be the most efficient action for the physical goal, the model proposes that it is considered efficient (and thereby reenacted) with respect to the adult's

[6] This is needed to handle the combination of universal and existential quantification – the existentially quantified variable within the scope of a universal quantifier is replaced with the value of a function of the universally quantified variable.

intention to share knowledge and the infant's corresponding intention to learn.

In the test phase, upon re-encountering the light-box, infants plan their actions. The integrity constraint that guides their computations is ?*HoldsAt*(learn$_f$,t), *Happens*(f',t) succeeds; it corresponds to the adult's pedagogical intention, and it expresses a 'learning by doing' kind of requirement. The outcome of interpretation, i.e., the means - ends structure of observations and the corresponding specialization of parameter *f*, modulate the constraint resolution. It sets up the physical goals that infants act upon in the test phase – either learn the new object's function in HO (upon specialization of *f* to *light-on*), or also learn how to activate it in HF (upon specialization of *f* to *contact*$_{head}$). These goals are reduced to basic actions through the CLP resolution rule of backwards reasoning, which prescribes infants' observed behaviour. In the HF condition thus, infants act upon two goals, learning the function and learning the means. The former goal is reduced to default hand actions (as required by closed-world reasoning), whereas the latter – to the novel head action. This explains infants' performance of both hand and head actions. Reenactment of the head action can be described as 'behavioural abduction', a continuation in behavioural terms of the unsatisfactory explanatory reasoning.

The CLP model of observational imitative learning corroborates developmentalists' argument that infants' acquisition of practical knowledge from observation of adult agents is an instance of instrumental rationality. It does so by providing a concrete example of pre-linguistic reasoning to an interpretation, and of planning from the inferred means – ends structure of the situation. A logic is thus shown to be helpful in formalizing a quasi-automatic kind of reasoning, very different from the traditional understandings whereby playing chess, or proving mathematical theorems are the paradigmatic cases of reasoning. More research is needed in modelling other instances of fast and automatic reasoning processes, evidence of which is on the rise, e.g., [8].

4 A Joint Enterprise of Logic Programming and Heuristics for Reasoning and Decision-Making

We now show how a combined use of LP and its meta-analysis extension for counting can provide an account of causal reasoning. Martignon et al.'s [24] replication of Cummins's [7] seminal results is an empirical proof that subjects' judgments expressed in heuristic terms predict their confidence in conditional inferences. The authors propose that the use of fast and frugal heuristics is thus a method of reasoning to interpretations.

In the context of the ABC group, heuristics have inherited Einstein's meaning [11]. That is, they are fast and frugal algorithms that "make us smart" *because* of their simplicity and not *in spite* of it [13]. In the field of judgement and decision-making they are specified as simple linear models for combining cues in tasks like comparison, estimation or categorization. There is extensive empirical evidence of their use, e.g., [5, 30]. Typical examples of heuristics are Take The Best – a linear model with non-compensatory weights, Tallying – a linear model with all weights equal to 1, or WADD – the weighted additive heuristic [27] whose weights are the cues validities (or 'diagnosticities').

Martignon et al. [24] set forth an analogy between the use of heuristics for combining cues in

decision-making, and people's reasoning with defeaters. Consider for instance the causal conditional "If the brake was depressed then the car slowed down"; defeaters are cases when although the brake is depressed, the car does not slow down, e.g., the brake is broken. [9] showed that the more defeaters people generate, the less likely they are to endorse the conclusion of Modus Ponens. Martignon and colleagues recognized that it is precisely the Tallying heuristic on a profile of defeaters that is used for *combining* them in further inferences. This same heuristic is used for comparison decisions. In the typical comparison task analysed by [13], subjects must decide which of two German cities has a larger population, based on cues like "city *A* has a soccer team in the Bundesliga and city *B* does not", etc. When cues are abundant, subjects tend to tally them to make the comparison, and when cues are scarce, they rely on Take The Best, i.e., use the first cue that discriminates the cities and choose the one with the highest value [23].

So far cue ranking has been modeled in a Bayesian framework. Such ranking assumes that for each cue, e.g., having a soccer team in the Bundesliga, its validity is given by the probability that a city with a soccer team is larger than one without – a cue is valid when probability is larger than 0.5. This probabilistic computation has always been seen as cumbersome in the theory of fast and frugal heuristics [10], leading to serious doubts that probabilities can provide realistic performance models. LP on the other hand offers a simpler way for ranking cues. It is easy to see that a broken brake, for instance, can be represented as an abnormality in the LP representation of the conditional as $p \,\&\, \sim ab \rightarrow q$. The simpler way for ranking cues thus amounts to counting abnormalities for the conditional "If city *A* has a soccer team in the Bundesliga and city *B* does not, then city *A* is larger than city *B*". Here defeaters tallying will provide a good approximation of the conditional validity without complex probabilistic computations. In a similar vein, [24] have showed that other heuristics, like Best Cue [16] or WADD effectively predict subjects' confidence in the causal strength of the conditional. The crucial message is that LP can solve one aspect of modelling the use of heuristics in decision-making that has been criticized by other authors, namely relying on a Bayesian computation of cue validities [10]. This is so because LP facilitates heuristic selection compared with previously proposed modelling frameworks [22]. Ultimately, Martignon and colleagues [24] argue that LP may give a computational model of how the interpretations necessary for further probabilistic reasoning are arrived at.

It is a fascinating result that precisely the same heuristics that function so well for cue combination in judgment and decision-making are excellent for defeater combination in conditional reasoning. Because LP can easily model an interpretation of causal conditionals taking into account defeaters, and of the conditional expression of typical cues for decision-making, it provides a unified framework for the fields of (causal) reasoning, and of judgment and decision-making. This aligns with recent similar 'unificationist' approaches in the new paradigm of psychology of reasoning, e.g., [4].

5 Conclusions: Wrapping-up and Further-on

Despite the fact that gaps such as the one that gives the theme of the workshop are not easy to

see in the raw data of the psychology of reasoning lab, to begin with however, their possibility must be acknowledged in order to allow for bridging. We started by presenting interpretation as an intrinsic, *sine qua non* stage of reasoning; this acknowledgement constrains realistic modelling endeavours to take it into account. We reviewed evidence that an approach to modelling which does take intensionality seriously by use of an expressive yet simple (at most linear on the name of nodes) formalism contributes to the theoretical integration of reasoning with judgement and decision-making. We also presented a computational model of pre-linguistic reasoning based on data from developmental psychology, and mentioned some consequences of this result for the ongoing debate with respect to dual-process theories of cognition.

With respect to future prospects for modelling applications of Logic Programming, we highlight the need for hypotheses of different domains where interpretation via minimal model construction may be adequate, and model that in terms of formalisms with minimal model semantics. The methodological implication of the multiple-logics proposal is a research program where modellers, given the properties of a particular formalism, hypothesise what kind of reasoning task it might model, and collaborate with experimenters to test those predictions; or observe properties of a reasoning task, hypothesise an appropriate formalisation, and test its empirical generalisations. With respect to LP, for instance, we propose that minimal model construction accurately models people's cooperative interpretation of conditionals uttered in a conversation setting [36]; investigations concerning other cases of cooperative reasoning, e.g., joint planning, joint intentionality, are current work in progress.

Throughout the paper we used LP to instantiate the multiple-logic proposal. Some other examples of applying non-deductive logics to human reasoning are Diderik Batens's program of adaptive logics [3], or Fariba Sadri's review of work on intention recognition [31]. It is noteworthy that both are essentially multiple-logic approaches. Consequently, last and most importantly, we wish to encourage pursuit of a multiple-system approach in research concerned with human reasoning. Our concrete suggestion concerns research on combining a logic that might appropriately model interpretation under computational constraints, i.e., in realistic cases of reasoning, with other formalisms such as probability [9]. One envisaged result is an alleviation of the problem of the priors, e.g., [28], by means of an intensional perspective offered by logics of interpretation. Such endeavour would bridge the gap between logical and AI systems for engineered reasoning, on the one hand, and empirical human reasoning research.

Appendix

In Chapter 7 of Stenning and van Lambalgen's *Human Reasoning and Cognitive Science* definite logic programs are used to represent non-monotonic reasoning with conditionals. The main technical tool is the interpretation of conditionals via the immediate consequence operator: the semantics is procedural, not declarative. This is because in a cooperative setting the truth of a conditional is not an issue, only what can be inferred from the conditional. This has consequences for what is meant by 'model of a program'. One may interpret the '$\rightarrow$' in program clauses truth-functionally, and say that $\mathcal{M} \models_3 \varphi \rightarrow q$ (where $\mathcal{M}$ is a 3-valued model) if the truth value of $\varphi \rightarrow q$ equals 1. Truth-functionality is not appropriate, since it would license nested occurrences of '$\rightarrow$', whereas nesting is not allowed by the syntax of logic programs, and hardly ever occur in natural language. Furthermore in this setting conditionals are never false, but apparent counterexamples are absorbed as 'abnormalities'. It follows that the expression 'model of a program P' cannot be given its literal meaning; its different sense is outlined below.

Let us start with the simpler case of positive programs. Recall that a positive logic program has clauses of the form $p_1 \wedge \ldots \wedge p_n \rightarrow q$, where the p_i, q are proposition letters and the antecedent (also called the body of the clause) may be empty. Models of a positive logic program P are given by the fixed points of a monotone operator:

Definition 1. *The operator T_P associated to a positive logic program P transforms a valuation $\mathcal{M}$ (viewed as a function $\mathcal{M} : L \longrightarrow \{0,1\}$, where L is the set of proposition letters) into a model $T_P(\mathcal{M})$ according to the following stipulations: if v is a proposition letter,*

1. $T_P(\mathcal{M})(v) = 1$ *if there exists a set of proposition letters C, true on $\mathcal{M}$, such that $\bigwedge C \rightarrow v \in P$*
2. $T_P(\mathcal{M})(v) = 0$ *otherwise.*

Definition 2. *An ordering $\subseteq$ on (two-valued) models is given by: $\mathcal{M} \subseteq \mathcal{N}$ if all proposition letters true in $\mathcal{M}$ are true in $\mathcal{N}$.*

Lemma 1. *If P is a positive logic program, T_P is monotone in the sense that $\mathcal{M} \subseteq \mathcal{N}$ implies $T_P(\mathcal{M}) \subseteq T_P(\mathcal{N})$.*

Now consider the completion *comp*(P).

Definition 3. *Let $\mathcal{M}$ be a valuation. $\mathcal{M}$ is a* model of P *if $\mathcal{M} \models$ comp(P).*

Again it is easy to see that program clauses are not interpreted as truth functional implications, but rather as closure conditions on a model. This idea is best expressed using the operator T_P.

Lemma 2. *Suppose $\mathcal{M} \models$ comp(P). Then $T_P(\mathcal{M}) \subseteq \mathcal{M}$.*

PROOF. Application of T_P results in changing the truth value of atoms for which there is no immediate ground in the program P from 1 to 0. □

Definition 4. *A model $\mathcal{M}$ such that $T_P(\mathcal{M}) \subseteq \mathcal{M}$ is called a* pre-fixpoint *of T_P. It is* fixpoint *if $T_P(\mathcal{M}) = \mathcal{M}$.*

Let us next investigate the relation between completion, pre-fixpoints and fixpoints.

Lemma 3. *(Knaster-Tarski) A monotone operator defined on a directed complete partial order with bottom element (dcpo) has a least fixed point.*

In the simple situation considered (no negation), a model of the completion is a fixpoint of T_P and conversely, but this will no longer be true once negation is taken into account. Models of the completion *comp*(P) figure mostly when studying semantic consequences of the program P, therefore the following theorem provides all one needs:

Theorem 1. *Let P be a positive program, then there exists a fixpoint $T_P(\mathcal{M}) = \mathcal{M}$ such that for every positive formula*[1] *F:*

$$\textit{comp}(P) \models F \Longleftrightarrow \mathcal{M} \models F.$$

PROOF. $\Leftarrow$ Choose a model $\mathcal{K} \models \textit{comp}(P)$. The set of models $\{\mathcal{B} \mid \mathcal{B} \leq \mathcal{K}\}$ is a dcpo, hence T_P has a least fixed point $\mathcal{M} \subseteq \mathcal{K}$ here. Indeed, if $\mathbf{0}$ denotes the bottom element of the dcpo, then $\mathbf{0} \subseteq \mathcal{K}$ implies $T_P(\mathbf{0}) \subseteq T_P(\mathcal{K}) \subseteq \mathcal{K}$, whence it follows that the least fixpoint of T_P is a submodel of any $\mathcal{K} \models \textit{comp}(P)$. By hypothesis $\mathcal{M} \models F$. Since F is positive and $\mathcal{M} \subseteq \mathcal{K}$, $\mathcal{K} \models F$, whence *comp*$(P) \models F$.
$\Rightarrow$ Since $\mathcal{M}$ is the least fixpoint of T_P, $\mathcal{M} \models \textit{comp}(P)$, whence $\mathcal{M} \models F$. □

Definition 5. *A model $\mathcal{K} \models \textit{comp}(P)$ is called* minimal *if there is no $\mathcal{N}$ which is a proper submodel of $\mathcal{K}$ (i.e. makes fewer atoms true).*

Lemma 4. *The least fixpoint of T_P is the unique minimal model of comp(P).*

PROOF. Let $\mathcal{M}$ be the least fixpoint of T_P (which is obviously minimal). Let $\mathcal{K} \models \textit{comp}(P)$ be another minimal model. Then since the bottom element $\mathbf{0} \subseteq \mathcal{K}$ and hence $T_P(\mathbf{0}) \subseteq T_P(\mathcal{K}) \subseteq \mathcal{K}$, it follows that $\mathcal{M} \subseteq \mathcal{K}$, which by minimality implies $\mathcal{M} \subseteq \mathcal{K}$.
□

A 'minimal model of the program P' actually refers to the minimal model of the completion of P. Again, the difference is that to specify a model for P, one would need a declarative semantics for the arrow of logic programming, whereas no such thing is required in defining a model for the completion of P.

The needed logic programs must allow negation in the body of a clause, since the natural language conditional 'p implies q' is represented by the clause $p \wedge \neg ab \rightarrow q$. As observed above, extending the definition of the operator T_P with the classical definition of negation would destroy its monotonicity, necessary for the incremental approach to the least fixpoint. The pursued solution is to replace the classical two-valued logic by Kleene's strong three-valued logic, for which see figure 2.2. in Chapter 2. The equivalence $\leftrightarrow$ is defined by assigning 1 to $\varphi \leftrightarrow \psi$ if φ,ψ have the same truth value (in $\{u, 0, 1\}$) , and 0 otherwise.

We show how to construct models for definite programs, as fixed points of a three-valued consequence operator $\mathcal{T}_P^3$. We will drop the superscript when there is no danger of confusing it with its two-valued relative defined above.

[1] A formula containing only $\vee$, $\wedge$.

Definition 6. *A three-valued model is an assignment of the truth values $u, 0, 1$ to the set of proposition letters. If the assignment does not use the value u, the model is called* two-valued*. If $\mathcal{M}, \mathcal{N}$ are models, the relation $\mathcal{M} \leq \mathcal{N}$ means that the truth value of a proposition letter p in $\mathcal{M}$ is less than or equal to the truth value of p in $\mathcal{N}$ in the canonical ordering on $u, 0, 1$.*

Lemma 5. *Let F a formula not containing $\leftrightarrow$, with connectives interpreted using strong Kleene 3-valued logic; in particular $\rightarrow$ is defined using $\neg$ and $\vee$. Let $\mathcal{M} \leq \mathcal{N}$, then $truth_{\mathcal{M}}(F) \leq truth_{\mathcal{N}}(F)$.*

Definition 7. *Let P be a program.*

a. The operator $\mathcal{T}_P$ applied to formulas constructed using only $\neg$, $\wedge$ and $\vee$ is determined by the strong Kleene truth tables.
b. Given a three-valued model $\mathcal{M}$, $T_P(\mathcal{M})$ is the model determined by
  (a) $T_P(\mathcal{M})(q) = 1$ iff there is a clause $\varphi \rightarrow q$ such that $\mathcal{M} \models \varphi$
  (b) $T_P(\mathcal{M})(q) = 0$ iff there is a clause $\varphi \rightarrow q$ in P and for all such clauses, $\mathcal{M} \models \neg\varphi$
  (c) $T_P(\mathcal{M})(q) = u$ otherwise

The preceding definition ensures that unrestricted negation as failure applies only to proposition letters q which occur in a formula $\bot \rightarrow q$; other proposition letters about which there is no information at all may remain undecided. This will be useful later, when the operation of negation as failure is applied restrictively to ab only. Once a literal has been assigned value 0 or 1 by $\mathcal{T}_P^3$, it retains that value at all stages of the construction; if it has been assigned value u, that value may mutate into 0 or 1 at a later stage.

Lemma 6. *If P is a definite logic program, T_P is monotone in the sense that $\mathcal{M} \leq \mathcal{N}$ implies $T_P(\mathcal{M}) \leq T_P(\mathcal{N})$.*

Lemma 7. *Let P be a definite program.*

1. *The operator $\mathcal{T}_P^3$ has a least fixpoint, obtained by starting from the model $\mathcal{M}_0$ in which all proposition letters have the value u. By abuse of language, the least fixpoint of $\mathcal{T}_P^3$ will be called the minimal model of P.*
2. *There exists a fixpoint $T_P^3(\mathcal{M}) = \mathcal{M}$ such that for every formula F not containing $\leftrightarrow$:*
$$comp(P) \models F \Longleftrightarrow \mathcal{M} \models F;$$
for $\mathcal{M}$ we may take the least fixpoint of $\mathcal{T}_P^3$.

PROOF OF (2). The argument is similar to that in the proof of theorem 1.
$\Leftarrow$ Choose a model $\mathcal{K}$ with $\mathcal{K} \models comp(P)$. We have $T_P^3(\mathcal{K}) \leq \mathcal{K}$:
(i) suppose r is assigned 1 by $T_P^3(\mathcal{K})$, then there exists a program clause $\theta \rightarrow r$ in P such that $\mathcal{K}$ assigns 1 to θ. Since $\mathcal{K} \models comp(P)$, in particular $\mathcal{K} \models r \leftrightarrow Def(r)$, and since $\theta \rightarrow Def(r)$, it follows that r is true on $\mathcal{K}$.
(ii) suppose r is assigned 0 by $T_P^3(\mathcal{K})$, then there exists a program clause $\theta \rightarrow r$ in P and for all such clauses, $\mathcal{K}$ assigns 0 to their bodies. It follows that $Def(r)$ is assigned

0 by $\mathcal{K}$, hence the same holds for r.
(iii) if r has value u in $T_P^3(\mathcal{K})$, this means neither (i) nor (ii) applies and there exists no program clause $\theta \rightarrow r$ in P with θ either 0 or 1. It follows that θ must have value u, hence r as well.
Note that we may have $T_P^3(\mathcal{K}) < \mathcal{K}$, for instance in case $P = \{q \rightarrow r\}$ and $\mathcal{K} \models$ *comp*(P), $\mathcal{K}$ makes r, q false, then $T_P^3(\mathcal{K})$ makes q undecided.
The set of models $\{\mathcal{B} \mid \mathcal{B} \leq \mathcal{K}\}$ is a dcpo, hence T_P^3 has a least fixpoint $\mathcal{M} \subseteq \mathcal{K}$ here. Indeed, if $\mathbf{0}$ denotes the bottom element of the dcpo, then $\mathbf{0} \leq \mathcal{K}$ implies $T_P^3(\mathbf{0}) \leq T_P^3(\mathcal{K}) \leq \mathcal{K}$, whence it follows that the least fixpoint of T_P^3 is a submodel of any $\mathcal{K}$ such that $\mathcal{K} \models$ *comp*(P). By hypothesis $\mathcal{M} \models F$. Since F is monotone and $\mathcal{M} \leq \mathcal{K}$, $\mathcal{K} \models F$, whence *comp*$(P) \models F$.
$\Rightarrow$ Since $\mathcal{M}$ is the least fixpoint of T_P^3, $\mathcal{M} \models$ *comp*(P), whence $\mathcal{M} \models F$. □

One step in the proof deserves special mention

Lemma 8. *For any model $\mathcal{K}$ with $\mathcal{K} \models$ comp(P) one has $T_P^3(\mathcal{K}) \leq \mathcal{K}$. In other words, a model of the completion is a pre-fixpoint of the consequence operator.*

A final remark regarding Lemma 4(3) in Chapter 7 of *Human Reasoning and Cognitive Science* is that it inadvertently stated that every model for the completion is a fixpoint. This doesn't affect the cognitive applications however, which are couched in terms of least fixpoints; and as we have seen entailment is determined by the least fixpoint.

References

1. Achourioti, T., Fugard, A., Stenning, K.: The empirical study of norms is just what we are missing. Frontiers in Cognitive Science 5, 1159, doi: 10.3389/fpsyg.2014.01159 (2014)
2. Baggio, G., van Lambalgen, M., Hagoort, P.: Logic as Marr's computational level: Four case studies. Topics in Cognitive Science (2014)
3. Batens, D.: A universal logic approach to adaptive logics. Logica Universalis 1, 221–242 (2007)
4. Bonnefon, J. F.: A theory of utility conditionals: paralogical reasoning from decision-theoretic leakage. Psychological Review 116, 888 – 907 (2009)
5. Bröder, A.: Decision making with the adaptive toolbox. Influence of environmental structure, intelligence and working memory load. Journal of Experimental Psychology: Learning, Memory and Cognition 29, 601-625 (2003)
6. Byrne, R. J: Suppressing valid inferences with conditionals. Cognition 3, 61 – 83 (1989)
7. Cummins, D. D.: Naïve theories and causal deduction. Memory and Cognition 23, 646 – 658 (1995)
8. Day, S. B., Gentner, D.: Nonintentional analogical inference in text comprehension. Memory and Cognition 35, 39 – 49 (2007)
9. Demolombe, R., Fernandez, A. M. O.: Intention recognition in the situation calculus and probability theory frameworks. In: Proceedings of Computational Logic in Multi-agent Systems (CLIMA) (2006)
10. Dougherty, M. R., Franco-Watkins, A. M., Thomas, R.: Psychological plausibility of the theory of probabilistic mental models and the fast and frugal heuristics. Psychological Review 115, 199 – 211 (2008)
11. Einstein, A.: Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt. Annalen der Physik 4, 132-147 (1905)
12. Gergely, G., Bekkering, H., Király, I.: Rational imitation in preverbal infants. Nature 415, 755 (2002)
13. Gigerenzer, G., Todd, P. M., the ABC research group: Simple heuristics that make us smart. Oxford University Press, New York (1999)
14. Hertwig, R., Benz, B., Krauss, S.: The conjunction fallacy and the many meanings of and. Cognition 108, 740 – 753 (2008)
15. Hölldobler, S., Kencana Ramli, C. D. P.: Logic Programs under Three-Valued Lukasiewicz's Semantics. In: P.M. Hill and D.S. Warren (eds.), Logic Programming, LNCS series, vol. 5649, pp. 464 – 478, Springer-Verlag Berlin Heidelberg (2009)
16. Holte, R. C.: Very simple rules perform well on mostly used data sets. Machine Learning 11, 63 – 91 (1993)
17. Henle, M.: On the relation between logic and thinking. Psychological Review 69, 366 – 378 (1962)
18. Hertwig, R., Benz, B., Krauss, S.: The conjunction fallacy and the many meanings of and. Cognition 108, 740 – 753 (2008)
19. Király, I., Csibra, G., Gergely, G.: Beyond rational imitation: Learning arbitrary means actions from communicative demonstrations. Journal of Experimental Child Psychology 116, 471 – 486 (2013)

20. Kowalski, R.: The Early Years of Logic Programming. In: Commun. ACM, vol. 31, pp. 38 – 43 (1988)
21. Kowalski, R.: Computational Logic and Human Thinking: How to be Artificially Intelligent. Cambridge University Press, New York Cambridge University Press, New York (2011)
22. Marevski, J., Schooler, L. J.: Cognitive niches: an ecological model of strategy selection. Psychological Review 118, 393 – 437 (2001)
23. Martignon, L., Hoffrage, U.: Fast, frugal and fit: Simple heuristics for pair comparison. Theory and Decision 52, 29 – 71 (2002)
24. Martignon, L., Stenning, K., van Lambalgen, M.: Qualitative models of reasoning and judgment: theoretical analysis and experimental exploration (in prep)
25. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of Artificial Intelligence. In: Machine Intelligence, pp. 463-502, Edinburgh University Press (1969)
26. Meltzoff, A. N.: The 'like me' framework for recognizing and becoming an intentional agent. Acta Psychologica 124, 106 – 128 (2007)
27. Payne, J. W., Bettman, J. R., Johnson, E. J.: The adaptive decision maker. Cambridge University Presss, Cambridge (1993)
28. Pearl, J.: Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufman Publishers, San Francisco (2014)
29. Politzer, G.: Premise interpretation in conditional reasoning. In: Hardmann, D., Macchi, L. (eds.), Thinking: Psychological Perspectives on Reasoning, Judgment, and Decision Making, pp. 79 – 93 (2003)
30. Rieskamp, J., Otto, P. E.: SSL: A theory of how people learn to select strategies. Journal of Experimental Psychology: General 135, 207 – 236 (2006)
31. Sadri, F.: Logic-based approaches to intention recognition. In: Handbook of Research on Ambient Intelligence: Trends and Perspectives, pp. 346 – 375, Citeseer (2010)
32. Stenning, K., van Lambalgen, M.: Human Reasoning and Cognitive Sciences. MIT University Press, Cambridge, MA (2008)
33. Stenning, K., Varga, A.: Many logics for the many things that people do in reasoning. In: International Handbook of Thinking and Reasoning, Ball, L., Thompson, V. (eds.), Psychology Press, London (under review)
34. van Lambalgen, M., Hamm, F.: The Proper Treatment of Events. Blackwell Publishing, Malden (2005)
35. Varga, A.: A Formal Model of Infants' Acquisition of Practical Knowledge from Observation. PhD thesis, Central European University, Budapest (2013)
36. Varga, A., Gazzo Castañeda, E.: Yes, you may Affirm the Consequent – In a conversation setting (in prep)

Tackling Benchmark Problems of Commonsense Reasoning

Ulrich Furbach[1], Andrew S. Gordon[2], and Claudia Schon[1]*

[1] Universität Koblenz-Landau, {uli,schon}@uni-koblenz.de
[2] University of Southern California, gordon@ict.usc.edu

Abstract. There is increasing interest in the field of automated commonsense reasoning to find real world benchmarks to challenge and to further develop reasoning systems. One interesting example is the Triangle Choice of Plausible Alternatives (Triangle-COPA), which is a set of problems presented in first-order logic. The setting of these problems stems from the famous Heider-Simmel film used in early experiments in social psychology. This paper illustrates with two logical approaches—abductive logic programming and deonitc logic—how these problems can be solved. Furthermore, we propose an idea of how to use background knowledge to support the reasoning process.

1 Introduction

In his influential 1958 paper, entitled "Programs with Common Sense" [14], John McCarthy set in motion his research agenda for Artificial Intelligence. He proposed the use of logic and deduction to overcome the difficult challenges of commonsense reasoning. His own pursuits led him later introduce the logic of circumscription [15], to handle the non-monotonic nature of human inference. In the intervening decades, numerous other approaches have been proposed by different researchers, e.g. based on probability theory or on argumentation frameworks. Progress on varied approaches was recently demonstrated, in dramatic fashion, in the success of IBM's Watson system in the Jeopardy challenge [3]. Subsequently, there has been considerable effort to investigate the varied techniques of the Watson system as a new programming paradigm, *cognitive computing*, and apply these techniques to diverse research and commercial problems, including eHealth, cancer research, and even supporting culinary chefs.

Although the Jeopardy challenge served to demonstrate the potential of new technologies, it does not provide the most appropriate benchmark problems for testing and evaluating individual research methods and approaches. Watson's success required a large engineering team, integrating technologies across many different fields of computer science. Logic-based approaches to commonsense reasoning may increasingly play a role in future cognitive programming applications, but the Jeopardy challenge is too ambitious as a tool for benchmarking

* Work supported by DFG FU 263/15-1 'Ratiolog,' and by the U.S. Office of Naval Research, grant N00014-13-1-0286.

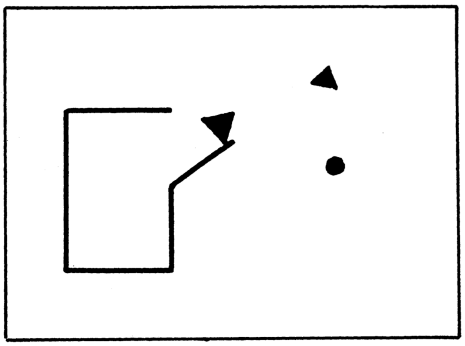

Fig. 1: One frame of the film used by Heider and Simmel in their study.

progress in this area. Over the years, logic-based approaches have been slow to move beyond the ubiquitous Tweety and Emu example problems to demonstrate their usefulness, although specialized benchmarking suites are increasingly being used in sub-disciplines of automated reasoning, e.g. in first-order theorem proving, answer set programming, and SAT solving. Recently, new sets of benchmark problems have been proposed for commonsense reasoning, such as the Winograd Schema Challenge [12] and the Choice Of Plausible Alternatives challenge [21]. Both of these challenges, however, require substantial capabilities for handling natural language (English), which complicates their use by researchers hoping to focus specifically on logic-based reasoning approaches.

The Triangle-COPA challenge[1] [13] provides a suite of one hundred logic-based commonsense reasoning problems, and was developed specifically for the purpose of advancing new logical reasoning approaches. Based on an influential psychology experiment from the 1940's, Triangle-COPA serves as a useful tool for studying the differences between human and logical reasoning. In the sections that follow, we describe the Triangle-COPA challenge problems and demonstrate that they can be solved using very different approaches to automated logical reasoning—first using a probabilistic form of logical abduction, and second using deontic logic—and discuss the challenges of authoring or acquiring the necessary background knowledge.

2 The Triangle-COPA Benchmarks

In an early and influential study of human social perception, psychologists Fritz Heider and Marianne Simmel [7] presented subjects with a short animated film depicting the movements of two triangles and a circle in and around a box with a hinged opening (Figure 1). Asked what they saw in the film, subjects each responded with similar narratives that anthropomorphized the moving shapes as intentional characters with beliefs, goals, and emotions. The simplicity of the film

[1] Available at https://github.com/asgordon/TriangleCOPA/

was in sharp contrast with the richness of the subjects' narratives, highlighting the role of knowledge and personal experience in the process of interpretation. Heider [6] later argued that the interpretation of intentional behavior was driven by commonsense theories of psychology and sociology, and was the basis of human social interaction.

How could we build a software system that was capable of interpreting the Heider-Simmel film in the same manner as the study's subjects? Researchers in artificial intelligence and cognitive science have sought to construct such a system. Thibadeau [23] takes a symbolic approach, representing the coordinates of each object in each frame of original film, which are matched to defined action schemas, such as opening the door or going outside the box. Pautler et al. [18] follows a related approach, beginning with object trajectory information from an animated recreation of the Heider-Simmel film. An incremental chart parsing algorithm with a hand-authored action grammar is then applied to recognize character actions as well as their intentions.

These earlier attempts highlight several problems for the use of the original Heider-Simmel film as a challenge problem by automated reasoning researchers. First, any system must overcome the difficult challenge of recognizing actions in the visual scenes, e.g. by first extracting quantitative trajectory information from the image data. Contemporary gesture recognition methods may be suitable for this task, using models trained on copious amounts of annotated examples. However, the effort involved in apply these techniques shifts research attention away from the central automated reasoning task of interpretation. Second, the original Heider-Simmel film provides a compelling input as a challenge problem, but the correct output is unspecified. Precisely because the input is "open to interpretation" is it difficult to compare the relative performance of two competing approaches, or even of the same approach as it develops over time.

The Triangle Choice of Plausible Alternatives (Triangle-COPA) set of one hundred challenge problems is a recent attempt to overcome these two problems with the original Heider-Simmel movie [13]. Each of the one hundred questions in this problem set describes, in English and in first order logic, a short sequence of events involving the characters of the original Heider-Simmel film: two triangles and a circle moving around a box with a hinged opening. This description ends with a question that requires the interpretation of the action sequence, and provides a choice of two possible answers, also in both English and logical form. The task is to select which of the two options would be selected by a human, where the correctness of the choice has been validated by teams of human volunteers. Three examples of Triangle-COPA questions are as follows:

44: The triangle opened the door, stepped outside and started to shake. Why did the triangle start to shake?
`(and (exit' E1 LT) (shake' E2 LT) (seq E1 E2))`
a. The triangle is upset.
`(unhappy' e3 LT)`
b. The triangle is cold.
`(cold' e4 LT)`

58: A circle and a triangle are in the house and are arguing. The circle punches the triangle. The triangle runs out of the house. Why does the triangle leave the house?
`(and (argueWith' E1 C LT) (inside' E2 C) (inside' E3 LT) (hit' E4 C LT) (exit' E5 LT) (seq E1 E4 E5))`

a. The triangle leaves the house because it wants the circle to come fight it outside.
`(and (attack' e6 C LT) (goal' e7 e6 LT))`

b. The triangle leaves the house because it is afraid of being further assaulted by the circle.
`(and (attack' e8 C LT) (fearThat' e9 LT e8))`

83: A small triangle and big triangle are next to each other. A circle runs by and pushes the small triangle. The big triangle chases the circle. Why does the big triangle chase the circle?
`(and (approach' E1 C LT) (push' E2 C LT) (chase' E3 BT C) (seq E1 E2 E3))`

a. The big triangle is angry that the circle pushed the small triangle, so it tries to catch the circle.
`(angryAt' e4 BT C)`

b. The big triangle and circle are friends. The big triangle wants to say hello to the circle.
`(and (friend' e5 BT C) (goal' e6 e7 BT) (greet' e7 BT C))`

As a benchmark set of challenge problems for automated reasoning systems, Triangle-COPA has a number of attractive characteristics. By providing first-order logic representations as inputs and outputs, Triangle-COPA focuses the efforts of competitors specifically on the central interpretation problem. At the same time, it places no constraints on the particular reasoning methods that are actually used to select the correct answer, affording comparisons between systems that use radically different knowledge resources and reasoning algorithms. The relational vocabulary of Triangle-COPA literals are fixed [13], but the semantics of these predicates are not tied to any one ontology or theory. The correct answers of Triangle-COPA are randomly sorted, so the quality of any given system can be gauged between that of random guessing (50%) and human performance (near 100%).

Thus far, only Maslan et al. [13] has demonstrated an approach to solving Triangle-COPA problems. Using five axioms and an implementation of weighted abduction [10], the authors demonstrated that the least-cost proof of the observables in Question 83 (above) entailed answer "a", that the big triangle (BT) was angry at the circle (C).

In the following two sections, we show two alternative approaches to solving the scenario described in Question 83. Our aim is to demonstrate that this benchmark set of questions can serve as a grounds for comparison of different logical formalisms, algorithms, and knowledge bases, and help the larger automated

reasoning community make progress on the difficult challenges of automated commonsense reasoning.

3 Probabilistic Abductive Reasoning

Triangle-COPA problems can be viewed as a choice between two alternative interpretations of a sequence of observable actions. Hobbs et al. [8] describes how interpretation of natural language can be cast as a problem of logical abduction, and solved using automated abductive reasoning technologies. Abduction, as distinct from logical deduction or induction, is a form of logical reasoning that identifies a hypothesis that, if it were true, would logically entail the given input. In classical logic, abduction is not a sound inference mechanism; asserting the truth of an antecedent given an observable consequent is a logical fallacy, "affirming the consequent." Still, automated abductive reasoning is a natural fit for many commonsense reasoning problems in artificial intelligence, including the interpretation problems in Triangle-COPA.

Automated abductive reasoning requires two mechanisms: a means of generating sets of hypotheses that entail the input, and a scoring function for preferential ordering these hypotheses. Hobbs et al. [8] described "Weighted Abduction," where hypotheses are generated by backchaining from the given input using the implicature form of knowledge base axioms, unifying literals across different antecedents wherever possible. The process generates an and-or proof graph similar to that created when searching for first-order proofs by backchaining, but where every solution in the and-or graph identifies a set of assumptions that, if true, would logically entail the given observables. Weighted Abduction orders these hypotheses by computing the combined cost of all assumed literals (those without justification), through a mechanism of propagating initial costs to antecedents during backchaining. Maslan et al. [13] demonstrated how Weighted Abduction can be used to solve Triangle-COPA problems by searching for the least-cost set of assumptions that entailed the literals in one of the two alternatives.

Several researchers have pursued probabilistic reformulations of Weighted Abduction, eschewing the use of ad-hoc weights for probabilities that might be learned from empirical data. Ovchinnikova et al. [17] and Blythe et al. [2] describe two recent probabilistic reformulations, each casting the and-or proof graph as a Bayesian network whose posterior probabilities can be calculated using belief propagation algorithms for graphical models. These efforts help to position abductive reasoning among current approaches to uncertain inference, and to take advantage of recent advances and tools for reasoning with Markov Logic Networks [20]. However, a simpler formulation of probabilistic abduction may be more appropriate when the task is only to rank possible hypotheses.

As in other probabilistic reasoning tasks, the calculation of the joint probability of a set of events is trivially easy if we assume that they are all conditionally independent: the joint probability of the conjunction is the product of their prior probabilities. If we know the prior probabilities of all assumed literals in an abductive proof (those without justification), then the naive estimate of their joint

probability is simply their product [19]. This calculation can be applied to any solution in an and-or graph created by backchaining from the given input, giving us a convenient means of ranking hypotheses.

This approach allows us to use standard first-order logic and familiar technologies of lifted backchaining instead of belief propagation in graphical models. However, by using logical inference (rather than uncertain inference) we require that the consequent of an implication is always true when the antecedent holds, i.e. the probability of the consequent given the antecedent is always one. Hobbs et al. [8], building on McCarthy's [15] formulation of circumscription, describes how defeasible first-order axioms can be authored by the inclusion of a special etcetera literal (etc) as a conjunct in the antecedent. These literals are constructed with a unique predicate name that appears nowhere else in the knowledge base, and therefore can only be assumed (via abduction), never proved. The arguments of this predicate are all of the other variables that appear in the axiom, restricting its unification with other etcetera literals of the same predication that may be assumed in the proof.

The probabilities of etcetera literals can be quantified if we interpret them as being an unspecified conjunction of all of the unknown factors of the world that must also be true for the antecedent to imply the consequent. Etcetera literals are true in exactly the cases where the remaining antecedent literals and the consequent are all true. As such, their prior probabilities are equal to the conditional probability of the consequent given the remaining conjuncts in the antecedent.

Using etcetera literals, we can author defeasible versions of the five axioms used by Maslan et al. [13] to correctly solve Triangle-COPA Question 83, above. Here the prior probabilities of the etcetera literals are encoded directly in the literal as its first argument, appearing a numerical constant.

- Push: Maybe you are attacking

```
(implies (and (attack_ e1 x y)
              (goal_ e2 e1 x)
              (etc1_push 0.9 e1 e2 x y))
         (push_ e3 x y))
```

- Approach: Maybe you want to attack

```
(implies (and (goal_ e1 e2 x)
              (attack_ e2 x y)
              (etc1_approach 0.9 e1 e2 x y))
         (approach_ e4 x y))
```

- AngryAt: Maybe they attacked someone you like

```
(implies (and (attack_ e1 y z)
              (like_ e2 x z)
              (etc1_angryAt 0.9 e1 e2 x y z))
         (angryAt_ e x y))
```

– Attack: Maybe you are angry at them

```
(implies (and (angryAt_ e1 x y)
              (etc1_attack 0.9 e1 x y))
         (attack_ e x y))
```

– Chase: Maybe you want to attack

```
(implies (and (attack_ e1 x y)
              (goal_ e2 e1 x)
              (etc1_chase 0.9 e1 e2 x y))
         (chase_ e3 x y))
```

Etcetera literals also afford a means of encoding the prior probabilities of other literals directly in the knowledge base. Below we provide eight additional axioms, one for each predicate used in either the Triangle-COPA question or in the axioms above, where an etcetera literal is the antecedent of each predicate form. By adding these axioms to the knowledge base, we can conduct our search for unique sets of assumptions by backtracking on all axioms to construct an and-or graph that terminates with etcetera literals. The probability of any solution in this and-or graph (assuming conditional independence) is simply the product of the priors of each etcetera literal.

```
(implies (etc0_push 0.01 e x y) (push_ e x y))
(implies (etc0_approach 0.01 e x y) (approach_ e x y))
(implies (etc0_angryAt 0.01 e x y) (angryAt_ e x y))
(implies (etc0_attack 0.01 e x y) (attack_ e x y))
(implies (etc0_chase 0.01 e x y) (chase_ e x y))
(implies (etc0_goal 0.9 e x y) (goal_ e x y))
(implies (etc0_like 0.9 e x y) (like_ e x y))
(implies (etc0_seq3 1.0 x y z) (seq x y z))
```

Figure 2 shows a visual representation of the most probable proof ($Pr = 0.0043$) of the given observables of Triangle-COPA Question 83, found amongst a set of 6038 possible proofs found by backchaining on the axioms listed above. The approach happens because the circle (C) had the goal to attack the little triangle (LT). The push happens for this same reason, and these explanations are unified. The chase happens because the big triangle (BT) had the goal to attack the circle, because it was angry at the circle, because the circle's attack on someone that the big triangle likes. The attacks are unified, and we infer that the big triangle likes the little triangle. Left unexplained are why the circle had the goal of attacking the little triangle, why the big triangle likes the circle, why attacking was the goal chosen by the big triangle, and why these eventualities happened in this sequence. The correct alternative appears in the most-probable proof, namely that the big triangle is angry at the circle.

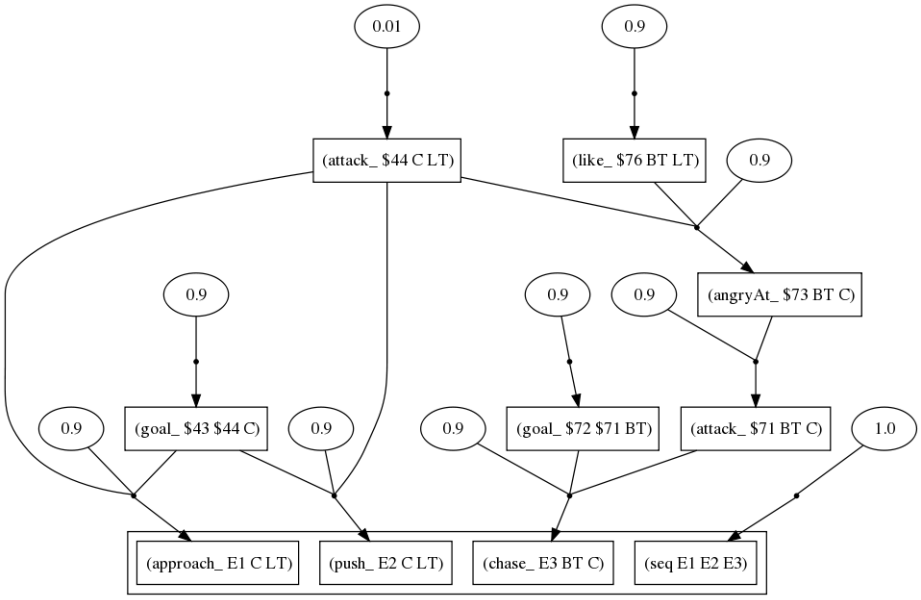

Fig. 2: The most-probable proof of Triangle-COPA Question 83.

4 Standard Deontic Logic

Now we consider a different approach, namely deontic logic, to tackle the Triangle-COPA benchmarks. This approach appears to be promising, since in [4] it was demonstrated that deontic logic is very well-suited for modelling different kinds of human reasoning. There are interesting examples from cognitive psychology, e.g. the Wason-selection task, or the suppression task, which can be formalized in a way that they are accessible for automated reasoning systems. We are using the tableau prover Hyper [1], a first order refutational theorem prover, which is able to decide standard deontic logic as well.

Standard deontic logic (SDL) is obtained from the well-known modal logic K by adding the seriality axiom D:

$$\mathsf{D}: \quad \Box P \rightarrow \Diamond P$$

In this logic, the $\Box$-operator is interpreted as "it is obligatory that" and the $\Diamond$ as "it is permitted that". The $\Diamond$-operator can be defined by the following equivalence:

$$\Diamond P \equiv \neg\Box\neg P$$

The additional axiom D: $\Box P \rightarrow \Diamond P$ in SDL states that if a formula has to hold in all reachable worlds, then there exists such a world. With the deontic reading of $\Box$ and $\Diamond$ this means: Whenever the formula P ought to be, then there exists a world where it holds. In consequence, there is always a world, which is

ideal in the sense that all the norms formulated by the 'ought to be'-operator hold.

SDL can be used in a natural way to describe knowledge about norms or licenses. The use of conditionals for expressing rules which should be considered as norms seems likely, but holds some subtle difficulties. If we want to express that *if P then Q* is a norm, an obvious solution would be to use

$$\Box(P \to Q)$$

which reads *it is obligatory that Q holds if P holds.* An alternative would be

$$P \to \Box Q$$

meaning *if P holds, it is obligatory that Q holds.* In [24] there is a careful discussion which of these two possibilities should be used for conditional norms. The first one has severe disadvantages. The most obvious disadvantage is that P together with $\Box(P \to Q)$ does not imply $\Box Q$. This is why we prefer the latter method, where the $\Box$-operator is in the conclusion of the conditional. For a more detailed discussion of such aspects we refer to [5].

For the examples in Triangle-COPA we argue that one can understand norms as expectation—many emotions in everyday life can be explained with unmet expectations. The husband not bringing flowers on the wedding anniversary and the friend arriving delayed to a date are only two examples, where unmet expectations cause negative feelings. On the other hand, expectations met can cause positive feelings. The husband helping with the dishes causes the wife to be content. We consider the scenario described in Question 83 from Triangle-COPA corresponding to the following set of facts:

$$approach(e1, c, lt). \tag{1}$$
$$push(e2, c, lt). \tag{2}$$
$$chase(e3, bt, c). \tag{3}$$
$$seq(e1, e2, e3). \tag{4}$$

The last fact states that the eventualities $e1$, $e2$ and $e3$ constitute a sequence of events.

The question we are asking is "How does the little triangle feel?". The two alternatives provided are as follows:

a. The little triangle feels relieved:
   $relief(e4, lt, e3)$
b. The little triangle is angry at the big triangle:
   $angryAt(e5, lt, bt)$

The notion of fulfilled expectations can be helpful to answer this question. The big triangle observes the circle attacking the little triangle. The little triangle expects the big triangle to defend it. The big triangle chases the circle away from the little triangle which corresponds to defending it. The little triangle is relieved that the big triangle hurried to its defense.

We need some background knowledge in this example:

– Pushing someone means attacking someone:

$$push(E, X, Y) \rightarrow attack(E, X, Y). \tag{5}$$

– Chasing an attacker means defending the person under attack:

$$attack(E, X, Y) \wedge chase(E', Z, X) \wedge after(E, E') \rightarrow defend(E', Z, Y). \tag{6}$$

Where *after* is a transitive predicate, stating that one eventuality occurs after another. $after(e1, e2)$ means that event $e2$ occurs after $e1$.

It is possible to model expectations with the help of deontic logic. Normative statements are used to model expected behavior. In our example, we use deontic logic to model the fact that one should defend someone who is attacked by someone else. This set of deontic formulae is the set of ground instances of the following formula:

$$attack(E, Z, X) \rightarrow \Box defend(E, Y, X) \vee Z = Y. \tag{7}$$

Formula (7) is not a SDL formula. However, we use it as an abbreviation for its set of ground instances. The ground instance interesting for our example is:

$$attack(e2, c, lt) \rightarrow \Box defend(e2, bt, lt) \vee c = bt. \tag{8}$$

With the help of formula (8), it is possible to derive that the big triangle ought to defend the little triangle in event $e2$.

Formula 8 states that in the ideal world following eventuality $e2$, the big triangle defends the little triangle. Another possibility to express this, would be to use the eventuality, which is part of every atom. We could state $defend(e9, bt, lt)$ for some new eventuality $e9$ and add some information stating that eventuality $e9$ is the ideal successor of $e2$. For this it would be necessary to introduce a new relation, connecting eventualities with its ideal successor. Since this is rather cumbersome, we use standard deontic logic instead.

Ground instances of the following formula can be used to deduce that someone is relieved if someone ought to be defended by someone and is actually defended:

$$(\Box defend(E, X, Y) \wedge defend(E', X, Y) \wedge after(E, E'))) \rightarrow (\bigwedge_{\substack{\forall E'' \\ after(E', E'')}} relief(E'', Y, E'))$$

A ground instance interesting for our example is:

$$\begin{aligned} &(\Box defend(e2, bt, lt) \wedge defend(e3, bt, lt) \wedge after(e2, e3) \rightarrow \\ &(relief(e4, lt, e3) \wedge relief(e5, lt, e3)) \end{aligned} \tag{9}$$

We want to use a theorem prover in order solve example 83 together with the above introduced question. To accomplish this, the following formulae are combined to one set of formulae S:

– Formulae (1) - (4) describing the scenario,
– the background knowledge given (5) and (6),
– some additional formulae formalizing the *after* predicate,
– the deontic logic formulae (8) and (9) stating the information about expectations and
– some formulae stating that bt, lt and c are pairwise different.

We use Hyper to solve example 83 with the question introduced above. It is possible to deduce that the little triangle is relieved in $e4$ by transforming this reasoning task into a satisfiability test. Hyper constructs a closed hyper tableau for $S \cup \{\neg \mathit{relief}(e4, lt, e3)\}$ which implies that $\mathit{relief}(e4, lt, e3)$ is entailed by S.

Referring to the question "How does the little triangle feel?" formulated before, we can use the derived $\mathit{relief}(e4, lt, e3)$ to show that the second alternative given is the correct one.

Of course, it is not desirable to formalize all rules manually. Rules like (9) can be generated automatically by formalizing a metarule stating that: whenever x and y are friends and y is obliged to do something for x and actually does it, x is relieved. This metarule can then be instantiated by the respective obligation.

5 Integration of Background Knowledge

In the previous section we used standard deontic logic to tackle one of the examples from Triangle-COPA. In addition to the formulae for normative statements, we used formulae (5) and (6) stating some essential background knowledge. In order to solve all Triangle-COPA benchmarks, an extensive background knowledge on psychology is essential. It is labor intensive and error-prone to state the whole background knowledge manually. Therefore it is desirable to use existing knowledge bases. There are several detailed ontologies like Yago [22], Cyc [11], and Sumo [16], stating knowledge about common sense. The very size of these ontologies however forbids to use these ontologies entirely. For example Yago contains more than 10 million entities (like persons, organizations, cities, etc.) and contains more than 120 million facts about these entities. ResearchCyc contains more than 500,000 concepts, forming an ontology in the domain of human consensus reality. Nearly 5,000,000 assertions (facts and rules) using more than 26,000 relations interrelate, constrain, and, in effect, (partially) define the concepts. And even the smallest version of Cyc, OpenCyc, still contains more than 3 million formulae.

Therefore it is necessary to extract relevant parts from these ontologies. However brute-force extraction by selecting for example all assertions from OpenCyc containing the word "attack" results in a set of 13,184 assertions. The vast majority of these assertions contains irrelevant information. For example assertions about the movie "Mars attacks" are selected. These irrelevant assertions potentially thwart the reasoning process, making it worthwhile to invest some effort into carefully selecting assertions suitable as background knowledge. Partitioning techniques used to handle large theories with theorem provers like the SInE (Sumo Inference Engine) [9] metaprover could be helpful to address this problem.

6 Discussion

Benchmark problems have helped to spur new ideas and compare technologies across many areas of computer science and beyond. For researchers interested in logical approaches to automated reasoning, as in other fields, the most useful benchmarks will be those that focus specifically on the core research challenge, but are not prejudice for or against any one technical approach. In this paper we have argued that the Triangle-COPA set of challenge problems is a useful tool for exploring the relationship between human and logical reasoning. We described two different logical approaches for solving Triangle-COPA questions, a probabilistic form of logical abduction and deontic logic. In so doing, we demonstrate that the questions are agnostic to the particular logic framework that is used to solve them. By tackling the same questions with different approaches, we gain new insights into both the similarities and differences afforded by different techniques. We encourage other research groups in our community to apply their unique approaches to the same questions, to consider the similarities and differences among approaches that go beyond the shallow characteristics of various logical notation, and to focus their efforts on overcoming the enormous challenges of humanlike commonsense reasoning.

References

1. Peter Baumgartner, Ulrich Furbach, and Björn Pelzer. Hyper tableaux with equality. In Frank Pfennig, editor, *CADE 21*, volume 4603 of *LNCS*, 2007.
2. James Blythe, Jerry R. Hobbs, Pedro Domingos, Rohit J. Kate, and Raymond J. Mooney. Implementing weighted abduction in markov logic. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pages 55–64, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
3. David Ferrucci, Anthony Levas, Sugato Bagchi, David Gondek, and Erik T. Mueller. Watson: Beyond jeopardy! *Artificial Intelligence*, 199–200(0):93 – 105, 2013.
4. Ulrich Furbach and Claudia Schon. Deontic logic for human reasoning. In Thomas Eiter, Hannes Strass, Miroslaw Truszczynski, and Stefan Woltran, editors, *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation - Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, volume 9060 of *Lecture Notes in Computer Science*, pages 63–80. Springer, 2014.
5. D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. van der Torre, editors. *Handbook of Deontic Logic and Normative Systems*. College Publications, 2013.
6. Fritz Heider. *The Psychology of Interpersonal Relations*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1958.
7. Fritz Heider and Marianne Simmel. An experimental study of apparent behavior. *American Journal of Psychology*, 57(2):243–259, 1944.
8. Jerry R. Hobbs, Mark E. Stickel, Douglas E. Appelt, and Paul Martin. Interpretation as abduction. *Artif. Intell.*, 63(1-2):69–142, October 1993.
9. Krystof Hoder and Andrei Voronkov. Sine qua non for large theory reasoning. In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *Automated Deduction - CADE-23 - 23rd International Conference on Automated Deduction, Wroclaw,*

Poland, July 31 - August 5, 2011. Proceedings, volume 6803 of *Lecture Notes in Computer Science*, pages 299–314. Springer, 2011.
10. Naoya Inoue and Kentaro Inui. Ilp-based inference for cost-based abduction on first-order predicate logic. *Journal of Natural Language Processing*, 20(5):629–656, 2013.
11. Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.
12. Hector J. Levesque. The Winograd Schema Challenge. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI, 2011.
13. Nicole Maslan, Melissa Roemmele, and Andrew S. Gordon. One hundred challenge problems for logical formalizations of commonsense psychology. In *Twelfth International Symposium on Logical Formalizations of Commonsense Reasoning, Stanford, CA*, 2015.
14. John McCarthy. Programs with common sense. In *Semantic Information Processing*, pages 403–418. MIT Press, 1968.
15. John McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.
16. Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9. ACM, 2001.
17. Ekaterina Ovchinnikova, Andrew S. Gordon, and Jerry. R. Hobbs. Abduction for discourse interpretation: A probabilistic framework. In *Joint Symposium on Semantic Processing*, pages 42–50, 2013.
18. David Pautler, Bryan L. Koenig, Boon-Kiat Quek, and Andrew Ortony. Using modified incremental chart parsing to ascribe intentions to animated geometric figures. *Behavior Research Methods*, 43(3):643–665, 2011.
19. David Poole. Representing bayesian networks within probabilistic horn abduction. In Bruce D'Ambrosio and Philippe Smets, editors, *UAI '91: Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence, University of California at Los Angeles, Los Angeles, CA, USA, July 13-15, 1991*, pages 271–278. Morgan Kaufmann, 1991.
20. Matthew Richardson and Pedro M. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.
21. Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.
22. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A large ontology from wikipedia and wordnet. *Web Semant.*, 6(3):203–217, September 2008.
23. Robert H. Thibadeau. Artificial perception of actions. *Cognitive Science*, 10(2):117–149, 1986.
24. Frank von Kutschera. *Einführung in die Logik der Normen, Werte und Entscheidungen.* Alber, 1973.

Interactive Theorem Proving – Modelling the User in the Proof Process *

Bernhard Beckert and Sarah Grebing
{beckert, sarah.grebing}@kit.edu

Karlsruhe Institute of Technology (KIT)

Abstract. Proving complex problems requires user interaction during proof construction. A major prerequisite for user interaction is that the user is able to understand the proof state in order to guide the prover in finding a proof.
Previous evaluations using focus groups for two interactive theorem provers have shown that there exists a gap between the user's model of the proof and the actual proof performed by the provers' strategies.
In this paper, we sketch a process model of the interactive proof process that helps to analyze this gap. Additionally, we give insight into the results of a usability test of the interactive verification System KeY, which provides evidence that this model is consistent with the actual proof process.

1 Introduction

Motivation. The degree of automation of interactive theorem provers (ITPs) has increased to a point where complex theorems over large formalisations for real-world problems can be proven effectively. But even with a high degree of automation, user interaction is still required on different levels. On a global level, users have to find the right formalisation and have to decompose the proof task by finding useful lemmas. On a local level, when automatic proof search for a lemma fails, they have to either direct proof search or understand why no proof can be constructed and fix the lemma or the underlying formalisation. As the degree of automation increases, the number of interactions decreases. But the remaining interactions get more and more complex as ITPs are applied to more and more complex problems.

We report on work in progress using the method of usability testing for several goals: (a) to gain insight into the interactive proof process using an ITP, (b) insight into problems in the interactive proof process and (c) insights for possible improvements. We carried out an experiment performing usability testing of the interactive verification system KeY [7]. In this paper we will briefly introduce a model of the interactive proof process, introduce our experiment and briefly give insights into the first results of the experiment, which relate to the

* The work presented here is part of the project Usability of Software Verification Systems within the BMBF-funded programme Software Campus.

proof process. In earlier work [6] we identified a gap between the user's model of the proof process and the actual proof process in the system. For illustrating the gap we developed a first informal model of the interactive proof process which we will extend in this paper.

In Section 2 we present related work of usability evaluations of interactive theorem provers and attempts to find a suitable model of the proof process. A first abstract model of the proof process follows in Section 3 and the gap between the user's and the prover's state is described in Section 4. The experiment and insights into the results are given in Section 5; and Section 6 concludes our work and shows future work.

2 Related Work

The usability of interactive theorem provers has been evaluated using various evaluation methods. Related work is concerned with usability evaluations of interactive theorem provers based on models defined prior to the evaluations. In addition, related work is also concerned with the derivation of models of the interactive proof process from evaluation results.

Merriam and Harrison [13] have evaluated interfaces of three theorem provers: CADiZ, IMPS and PVS. In this work they have identified four key activities in the interactive proof process where the user needs support from the proof system: planning, reuse, reflection and articulation. The three theorem provers have been examined with respect to these activities. Based on these results, gaps in user support of the theorem provers have been identified as well as points in the systems' interfaces where the user can make errors that cost him or her a lot of time to recover from.

Merriam [14] developed two approaches for the description of user activities in the proof process. He formalized a generic formal model of the proof using Z as formal language. This model is used to enable to gain insight into which kind of information is necessary for the user to conduct a proof effectively. Merriam assumes in this model that the user forms an opinion during the proof process about the provability of a proof goal using heuristics. He remarks that to model this assumption, a suitable cognitive model of the user is necessary. Interactions the user performs in the system are outside this model and are modelled in a second model of Merriam on the basis of Newman's Action cycle. Both models together were used to evaluate the PVS proof system.

Norbert Voelker [15] published a discussion paper on requirements and design issues of user interfaces for provers. He presented difficulties in the design of user interfaces of theorem provers developed in academia. In addition, a requirement analysis based on the scenarios using the scenario method has been carried out and resulted in a high-level description of the interaction with the proof system.

Aitken and Melham evaluated the interactive proof systems Isabelle and HOL using recordings of user interactions with the systems in collaboration with HCI experts. During the proof process the users were asked to think aloud and afterwards the users were interviewed. The authors goal was to study the activities

performed by users of interactive provers during the proof process to obtain an interaction model of the users. They propose to use typical user errors as usability metric and they compared provers w.r.t. these errors [3,4,2]. Also, suggestions for improvements of the systems have been made by the authors based on the evaluation results, including improved search mechanisms and improved access to certain proof relevant components.

The systems Isabelle and HOL have been evaluated by Aitken [1] using records of interactions. A semi-formal interaction model was extracted from the results, by identifying the actions that were performed during proof construction. Of the fifteen actions that have been identified, some relate to mental work of the users and some were direct actions in the system. All actions were modelled as activity diagram and it was distinguished between actions on the logical level and actions on the interaction level. In this work the relation between the problem class, the proof plan and the implementation is depicted.

In the work of Goguen [9] three user roles that can be represented by one single user have been identified: the prover, the reader and the specifier. Each of these roles has different requirements for the interactive proof system and some of the requirements can be conflicting. The authors claim that users of theorem provers need precise feedback on the failure of a proof attempt at the (sub)goal level. Further they argue that an unstructured proof tree is not easy to use as the users need to orient themselves in the proof tree. They present a proof approach where users should form the high-level proof plan and leave the "low-level computations" to the automatic prover. They implement their user interface for the proof assistance tool Kumo.

Similar to our findings in previous work, Archer and Heitmeyer [5] also realized the gap between the prover's and the user's model of the proof and have developed the TAME interface on top of the prover PVS to reduce the distance between manual proofs and proofs by automation. TAME is able to prove properties of timed automata using so called *human-style reasoning*. Proof steps in TAME are intended to be close to the large proof steps performed in manual proofs. The authors have developed strategies on top of the PVS strategies that correspond more to proof steps performed by humans. The goal is to provide evidence and comprehension of proofs for domain but not proof experts.

3 A Model of the System Consisting of User and Prover

In order to be able to describe the interactive proof process and to describe what influences the gap between the user and the prover states, a precise model of the proof process has to be developed. Our idea is to have an interactive proof system that consists of two main components that exchange information during the proof process: the user U and the prover P. We model both components as simple transition systems with three different transition functions: one that decides the next action for the user (f_{UDec}) resp. the next proof step for the prover (f_{PDec}), one that computes the next state of the user (f_{UCh}) resp. prover (f_{PCh}) according to the action/proof step and one function that computes the

next state of the user according to the prover's current state (f_{insp}) resp. the next state of the prover according to the action of the user ($f_{trigger}$).

Definition 1 (The Prover). *We model the prover as a transition system*

$$Prover = (P, PS, f_{trigger}, f_{PDec}, f_{PCh}, p_0, P_T) \ ,$$

where

- P *is a set of prover states*
- PS *is a set of actions which we call proof steps*
- $f_{trigger} : P \times A \to P$ *is a transition function*
- $f_{PCh} : P \times PS \to P$ *is a transition function*
- $f_{PDec} : P \to PS$ *is a choice function*
- $p_0 \in P$ *is the initial state*
- $P_T \subseteq P$ *is the set of terminating states*
- $P_{Proof} \subseteq P_T$ *is the set of terminating states in which a proof has been found*

Definition 2 (The User). *We model the user as a transition system*

$$User = (U, A, f_{insp}, f_{UDec}, f_{UCh}, u_0, U_T) \ ,$$

where

- U *is a set of user states*
- $A = (A_{proc} \cup A_{man})$ *is a set of actions, being the union of the proof manipulating and the process-oriented actions*
- $stopProcess \in A_{proc}$ *is the action to stop the proof process*
- $f_{insp} : U \times P \to U$ *is a transition function*
- $f_{UCh} : U \times A \to U$ *is a transition function*
- $f_{UDec} : U \to A$ *is a choice function*
- $u_0 \in U$ *is the initial state*
- $U_T \subseteq U$ *is the set of terminating states*

Definitions 1 and 2 depend on each other, as Definition 2 uses a component of Definition 1 (namely P) and vice versa (namely A). Both definitions could be combined to one system definition, but for simplicity we have two definitions, one for each component.

A prover's state P includes a partial proof (tree). We assume that the user states U at least consist of a mental model of the provers' state. We do not characterize this model in full detail, as we believe it is different for every user and depends on the experience with the system and mathematical background knowledge. Determining this model in full detail goes beyond the scope of our work.

In our model we focus on the interaction between the user and the prover. We further assume that the user's model of the prover's state is more abstract than the actual prover's state. We believe the user has a proof plan, that is formed

when developing the proof obligation. We consider that plan to be encoded in the user's state. Furthermore, we assume that the user has an idea about the effect of performing actions on the prover's state. This knowledge is included in the function f_{UCh}, as the user calculates a successor state from the current state and the action that the user performs in the system. As the successor state of the user also includes an abstraction of the prover's state, the user updates this model according to the expectations of the effect of performing the action.

Actions of the user can be of two kinds: proof manipulating (A_{man}) (e.g., applying a single proof rule or invoking a specific automatic strategy in the prover) and process-oriented actions (A_{proc}) (e.g., inspecting the prover's state further or stop the proof process (*stopProcess*)).

A proof step in the prover is an application of a calculus rule onto the current proof state.

Terminating states in the prover's model can be of two kinds: either a state in which a proof is found, i.e., $t_{Proof} \in P_T$ or states in which the automatic strategies stops, i.e., $p_t \in P_T$. These terminating states mark the beginning of the user interacting with the prover.

Interaction between both, the user and the prover, involves an information exchange. This exchange happens through the functions $f_{trigger}$ (user to prover) and f_{insp} (prover to user). The function f_{insp} involves the user inspecting the proof state of the prover (which can be either $p_t \in P_T$ in case the automatic strategies stop or the initial state p_0 in case the proof process is at the beginning) and updating the user's model by changing the state. This update may involve changing the proof plan by concretizing proof states in the plan or changing the mental model of the proof state by refining states.

The function $f_{trigger}$ represents the state change in the prover when a user action involves input to the prover and corresponds to the user invoking an action. This action is then accepted by the prover and translated by the prover into the strategy that should be used. The strategy of the prover is responsible for choosing the next proof step that should be applied to the prover's state. We model this by a state change performed using the prover's decision function (f_{PDec}).

The user also has such a decision function, which we call f_{UDec}. This function decides which action the user will perform next, depending on the user's state.

In our model of the user the functions f_{insp} and f_{UDec} follow each other. After the user has made his or her decision, the corresponding action is performed and function $f_{trigger}$ is applied in case the user decides to invoke the prover's strategies. Function f_{PDec} follows $f_{trigger}$ and then a sequence of function applications of f_{PCh} apply until a terminating state is reached.

In the following, we will define the interaction between the user and the prover in the interactive proof system.

Definition 3 (The Interactive Proof Process in an Interactive Proof System).

We model the interactive proof system consisting of a user U and prover P as a transition system. The state space S of the interactive proof system is the

set of triples

$$S = U \times P \times \{automode, interactive, inspected, decided, fail, success\} \ .$$

The initial state is $s_0 = (u_0, p_0, interactive)$*, where* p_0 *and* u_0 *are the initial states of the user* U *resp. the prover* P*.*

For all states $s \in S$*, the successor state* s' *of* s *is defined as follows, where* $a = f_{UDec}(u)$*:*

(a)	*if* $s = (u, p, interactive)$	*then* $s' = (f_{insp}(u, p), p, inspected)$
(b)	*if* $s = (u, p, inspected)$	*then* $s' = (u, p, decided)$
(c)	*if* $s = (u, p, decided)$ *and* $a = stopProcess$	*then* $s' = (f_{UCh}(u, a), p, userStop)$
(d)	*if* $s = (u, p, decided)$ *and* $a \in A_{man}$	*then* $s' = (f_{UCh}(u, a), f_{trigger}(p, a), auto)$
(e)	*if* $s = (u, p, decided)$ *and* $a \in A_{proc} \setminus \{stopProcess\}$	*then* $s' = (f_{insp}(u, p), p, inspected)$
(f)	*if* $s = (u, p, auto)$ *and* $p \notin P_T$	*then* $s' = (u, f_{PCh}(p, f_{PDec}(p)), auto)$
(g)	*if* $s = (u, p, auto)$ *and* $p \in P_T$	*then* $s' = (u, p, interactive)$
(h)	*if* $s = (u, p, userStop)$ *and* $p \notin P_{Proof}$	*then* $s' = (u, p, fail)$
(i)	*if* $s = (u, p, userStop)$ *and* $p \in P_{Proof}$	*then* $s' = (u, p, success)$

For states of the form $s = (u, p, success)$ *and* $s = (u, p, fail)$*, the successor state* s' *is undefined. They do not have a successor state.*

In the following, we will give a brief description of the Definition 3. In addition we have depicted the interactive proof process in Figure 1.

(a) If the system is in the state *interactive*, the user inspects the proof state and updates the own state using the information gained from the inspection $f_{insp}(u, p)$ (e.g., at the beginning of the proof process after the user has formulated the proof obligation and the system has translated it into the prover's representation, i.e., $s = (u_0, p_0, interactive)$ or after the prover has reached a terminating state, i.e., $s = (u_0, p_0, interactive)$)

(b) If the system is in the state *inspected*, the user makes a decision about the next action in the process according to the updated own model of the proof state. The action is decided by the internal choice function $f_{UDec}(u)$, (e.g., when the user has inspected the formula in the proof obligation and now determines what to do next in the proof process)

(c) If the system is in the state *decided* and the user has chosen the process-oriented action *stopProcess*, the user has decided to stop the proof process (e.g., when the user discovers a mistake in the specification or the program)

(d) If the system is in the state *decided*, and the user has chosen a proof manipulating action the user interacts with the prover (e.g., the user has inspected the formula in the proof obligation and encounters that a quantifier instantiation has to be performed or the induction rule has to be applied). The function $f_{trigger}$ translates the user's actions into the prover's strategies.

(e) If the system is in the state *decided*, and the user has chosen a process-oriented action (except *stopProcess*) the user inspects the proof state further (e.g., the user wants to inspect the proof tree in more detail).

(f) If the system is in the state *auto*, the prover applies a proof step according to the provers internal choice function resp. strategies until the automatic strategies can not apply more rules and therefore a terminating state is reached (e.g., the user has invoked the automatic strategies of the prover and the prover applies consecutive proof rules. Each rule application corresponds to one state transition.)

(g) If the system is in the state *auto* and the prover has reached a terminating state after proof step application, the user now can interact with the prover (e.g., the prover's strategies cannot apply proof rules anymore and presents the remaining proof obligation to the user)

(h) + (i) If the system is in the state *userStop*, the user decides whether the proof process was successful respec. failed (e.g., the user has either found a proof or discovered a mistake in the formalization and decides to terminate the proof process)

We model that while the prover's strategies apply proof steps (so state transitions in the prover's model according to f_{PCh} are made), the user can not interact with the prover until it reaches a terminating state p_t.

In our model the user's state also consists of a proof plan that the user formed when formulating the proof obligation. The *(partial) proof plan* of the user is a sequence of abstract proof states, denoted by $abs(P)$, related to each other by actions. These actions can be identical to the actions defined in the user's model and abstractions of the proof steps of the prover. We assume that this proof plan of the user consists of abstract proof states – they may either be identical to some prover states, possibly with intermediate prover states in between the abstract states of the user's proof plan. Some abstract states in the plan may also correspond to a sequence of prover states and, which are summarized as one abstract state.

The user might not always have a clear proof plan, e.g., at the beginning of the proof process. In this case, the user may consider several actions that he or

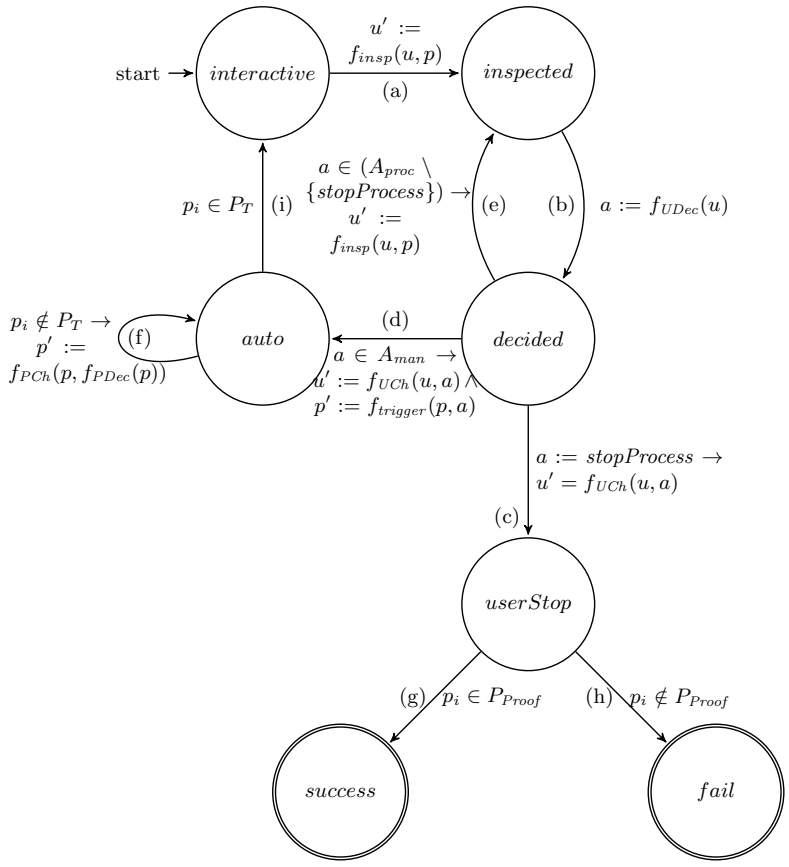

Fig. 1. Model of the interactive proof process ((a)-(i) are references to Definition 3)

she deems worthwhile to pursue, and for each of these actions he or she likely only has a rough idea of the resulting proof state. Of course, in certain situations, this set of possible actions to continue a proof is empty, as the user is unable to come up with a proof plan.

4 The Gap in the Proof Process

In former evaluations we have already identified a gap between the user's abstract states of the proof process and the concrete state of the prover. Based on this gap we have identified three major challenges an interactive theorem prover has

to meet in order to be more usable: (a) keeping the gap small, (b) bridging the gap and (c) allowing for effective interactions [6].

Here, we will now give a more precise description of the gap between the users abstract states and the provers states using a proof system with an explicit proof object, a proof tree as proof state and a sequent calculus as underlying proof calculus as an example. The following two problems can occur in such a system:

Provers Strategies applied too many Proof Rules To determine which next action to apply (modelled by the function f_{UDec}) the user has to inspect the proof state and update the own state according to the information gained by inspecting the prover's state (f_{insp}).

In this situation a gap between the users model of the proof state and the prover's state can occur when the automatic strategies applied too many proof steps or the proof steps were too "complicated". The user has to inspect the provers state and update the own model according to the information gained in the inspection. However, for this update the user has to find a correspondence for the current prover's state p_t in the own model too. As described above, p_t is a terminating state after the application of several proof steps decided by the strategy. If the user often iterates between the states *decided* and *inspected*, this may be a sign of a gap caused by the prover's strategies. Here the user needs a lot of time to find a correspondence between the own model of the prover's state and the current prover's state.

User Expectations Not Met. Another possibility for a gap is that the user performed the proof according to the proof plan he or she has made before the proof process and at a terminating prover's state p_t the prover's state does not correspond to the expectations of the user (it does not correspond to the state in the user's plan). The user has to inspect the prover's state further in order to determine whether he or she has made a mistake in the proof plan or the proof steps in the proof plan have been to abstract and have to be concretized by the inspection process.

If the user only has a partial proof plan, a gap can occur during the proof process when the difference between the prover's current state and the last state in the user's proof plan is large and the user is not able to relate the states to each other anymore. In this case the user has to inspect the proof state in order to retrace the proof steps the prover's strategies have applied and update the own state according to the gathered information.

The user has expectations about the effect of his or her actions on the proof state. If such an expectation is not met by the prover's strategies, a gap may occur as well. The user now has to try to understand what the effect of the performed action was by closely inspecting the proof state.

To summarize, we assume that the gap occurs at the point in the proof process where the prover reaches a terminating state p_t and the user applies function f_{insp} in order to apply function f_{UDec}. A hint that a gap has occurred can be, when the user needs a lot of time for the inspection process. In this case the loop between the states *inspected* and *decided* is traversed several times.

5 Insights into the Usability Test of the KeY system

To gain insights into the interactive proof process and to find evidence that our model is consistent with the proof process we conducted a usability test. Based on earlier results of two focus group discussions we conducted a formative, explorative usability test for the KeY system as the target of evaluation. Usability tests are structured interviews guided by a moderator following a script, which consists of all tasks and questions in the order they should be posed. While the participants perform the tasks, they should use the "thinking-aloud" technique. In addition their actions on the screen are recorded. The recorded data is then transcribed and anonymized. Later on, a qualitative content analysis [12,11,10] is performed to evaluate the test results.

In the following, we will first briefly describe the target of evaluation, give details about the usability test sessions and give insights into first observations and first analysis results.

The Target of Evaluation: KeY. The KeY system is an interactive verification system for programs written in Java and specified using the Java Modeling Language (JML). As such it is mostly used for the verification of Java programs w.r.t. a formal specification (usually a functional specification but also, e.g., information-flow properties). KeY has an explicit proof object, i.e., all intermediate proof states can be inspected by the user. The underlying calculus is a sequent calculus for Java Dynamic Logic [8]. Its user interface represents proofs as a tree. The nodes of the tree are intermediate proof goals (i.e., sequents). Each node is annotated with the rule that was applied to some formula in its direct parent node that lead to the current node.

The Participants. Nine KeY users took part in our usability tests, either intermediate or expert users. We excluded novice users, as our hypothesis was that advanced users perform more complex or larger proofs than novice users and therefore suffer more from efficiency problems in the proof process.

The Usability Test. Our goal of performing the usability test was (a) to gain insight into the proof process using the KeY system and (b) to determine whether a new mechanism, prototypically introduced into KeY, helps the user in bridging the gap between the concrete proof state and the model of the proof. We also wanted to gain information about further room for improvement of the target of evaluation. We planned a session time of approximately 70 minutes.

We structured the usability test into different phases[1]: introduction, warm-up, task and cool-down phase. In the *introduction-phase* the users were interviewed by the moderator about their experiences using the KeY system. The *warm-up phase* started with an interview about the proof process of the participants using the KeY system. Then the participants were asked to specify and

[1] The testing script can be found at `http://formal.iti.kit.edu/~grebing/SWC/` in German.

verify a Java method within the time frame of 10-15 minutes. We did not restrict the usage of system features in the warm-up phase. Our intention for this phase was to get insight into how the user uses the system to find a proof.

Based on earlier focus group discussions we prototypically implemented a mechanism to support the display of the history of a formula in the KeY system: It allowed the user to select a formula in the open goal and retrieve the path from the open goal to the original proof obligation in which the formula was affected by rule applications, in the following also called *history of a formula*. This mechanism should help to bridge the gap between the user's model of the proof and the current proof, as the user is able to trace back the history of a selected formula and see the changes during the proof process.

For the *task phase* we developed tasks that should help to evaluate the mechanism. We divided this phase into two parts with two different tasks each, one with and one without the new mechanism. One of the two different task types involved showing the user a partial proof for a proof obligation in first-order logic, obfuscating the predicate and function symbol names. The second task type involved a partial proof for the correctness of a method contract of a Java method.

For both types of tasks and both parts of the task phase the questions were identical: the user should describe the proof situation, they should name the history of two formulas of the open goal and name the next step to continue the proof process. At the end of the task phase the users were asked about their expectations about parent formulas of a given formula and proof.

In the *cool-down* phase participants were interviewed again about the new mechanism and generally about room for improvement in the system.

Insights into the results of the Usability Test. As the analysis is still work in progress we only give insights into the results of the warm-up phase and not a full analysis[2].

Almost all testing sessions have taken longer than we planned beforehand. Solving a task or answering the interview questions took longer than anticipated, as we didn't want to interrupt the participants.

In the warm-up phase we wanted to see how the participants use the KeY system to solve the task in order to gain insights into the proof process. Before the task, we wanted to know detailed information about the expectations of the users in a certain proof situation and about the proof process in general.

The interview questions have been:

1. Please imagine you are sitting in front of the KeY system and the automatic strategies stop with a lot of open goals/proof branches and quite a large sequent formula. What could have happened? What could have been reasons that KeY opened a lot of proof branches and was not able to close them? (In addition a screenshot of a proof with open goals in the KeY system has been used as stimulus)

[2] The sessions have been conducted in German, as it was the native language of the users. We translate the tasks and answers to the best of our knowledge.

2. How do you solve the problem of determining what has happened and what the next steps are?
3. Which possibilities do you have for that? Please arrange them in the order relatively to each other how often you use the possibilities.
4. Are there other alternatives in this situation, or are you missing a mechanism which is better suitable than the ones implemented in the system?
5. If you could wish for a functionality that could support you in proving using the KeY system, which one would it be?

The practical task for the users in the warm-up consisted of proving a method that removes the k-th element of a given array and returns the rest of the array, given the following task description:

> Please verify that the method fulfills its contract. Please conduct the proof like you are used to do it. Please complete or add something to the specification or the program if necessary. Please think-aloud what you are searching for and please explain before you click why you would like to click on that element on the screen.

The first specification of the method which we provided did not formalize the requirements we described to the user but was already a partial contract.

Our Intentions for the Tasks and Questions. Our intention behind these questions and tasks in the warm-up phase was to gain insight into how users use the KeY system to conduct a proof. In the practical task we intentionally did not show the method and its specification from the beginning on. We wanted to see which user directly proceeds to use the system to find out whether the method meets its specification and which user requests for the specification from the beginning on. In the first case, if the user found a proof, and if task time was not too far advanced, we asked the user whether the specification is an adequate rendition of the requirements.

First Results of the Usability Test. In the following, we will briefly mention those observations from the warm-up phase that contribute to our model. Our observation was that the users try to abstract from the concrete proof tree to gain an overview over the proof by using a feature of the KeY system that hides all intermediate proof states after using the automatic strategy. Almost all participants either used this feature in the practical tasks or mentioned the usage of this feature for proof inspection in the answer to the second question. This relates to the user's having or trying to build an abstract model of the prover's state.

When determining whether a proof is closeable, some users first tried the provers strategies again, as they assume the amount of user defined proof steps is not sufficiently high, before inspecting the proof tree in detail. In this case, we assume the users to have an expectation about the prover's strategies.

There were also users who noted that they would prune the proof tree when the strategies "went too far". They would prune the tree at a proof node from which they know its meaning, e.g., after finishing the symbolic execution and would then "apply rules in a controlled way." Here we have a hint for the gap, when the strategies of the prover apply too many rules and open too many goals without closing them again, the user goes back to a state from which he has a model.

At least one participant noted that he or she always switches from local to global proving, i.e. the participant first has an idea on the global, more abstract level how a proof should be performed, and during the proof process, when the automatic strategies are not able to close all goals he or she switches to inspecting the proof in detail on the local level and therefore inspecting the sequent in the proof node more closely. When a proof branch is closed, the user switches back to the global level and tries to close the next open goals. This indicates that the user has different abstraction layers of the proof.

6 Discussion and Future Work

We have presented a model of the components involved in an interactive proof process and briefly described a usability test of the KeY system to gain insights into the interactive proof process and to find evidence that the proposed model is consistent with the actual proof process. We described the point in the proof process where a gap between the user's model and the prover's actual proof state can occur using our model. We are aware that it is not possible to find evidence for all parts of the model, as some parts, such as how the precise user state can be characterized cannot be assessed by the "thinking-aloud" technique. Participants do not always verbalize everything they are thinking.

Our model does not yet include how users form a proof plan. This is a research field of its own and it remains for future work to include results of this research field into our model. We did not consider different user types and their special requirements on the prover yet, but we are confident that it is possible to include this in our model as well. The model of the proof process has to be enhanced, as it does not capture yet that the user and the prover are parallelized. It is not yet captured that the user can make decisions while the prover searches for a proof, as well as the user is able to interrupt the proof process. The role of the user interface is not yet captured by the model. We assume it can be modelled as a filter function for the prover's state p, which only shows parts of the prover's state to the user and the user only inspects this filtered state in the function f_{insp}.

As the evaluation of the usability tests is work in progress a full analysis and evaluation of the results is ongoing work.

Acknowledgements. We thank the participants in our focus group discussions on the usability of KeY and of Isabelle and our participants of the usability tests of the KeY system. In particular, we also thank the three moderators for their

great work. In addition, we thank our project partners from DATEV eG for sharing their expertise in how to prepare and analyse focus group discussions.

References

1. J. S. Aitken. Problem solving in interactive proof: A knowledge-modelling approach. In *Proceedings of the European Conference on Artificial Intelligence 1996 (ECAI96): 335-339, Edited by W. Wahlster*, pages 335–339, 1996.
2. J. S. Aitken, P. Gray, T. Melham, and M. Thomas. Interactive theorem proving: An empirical study of user activity. *J. of Symbolic Comp.*, 25(2):263–284, 1998.
3. J. S. Aitken and T. F. Melham. An analysis of errors in interactive proof attempts. *Interacting with Computers*, 12(6):565–586, 2000.
4. S. Aitken, P. Gray, T. Melham, and M. Thomas. A study of user activity in interactive theorem proving. In *Task Centred Approaches To Interface Design*, pages 195–218. Dept. of Computing Science, 1995. GIST Technical Report G95.2.
5. M. Archer and C. Heitmeyer. Human-style theorem proving using PVS. In *Theorem Proving in Higher Order Logics*, LNCS 1275. Springer, 1997.
6. B. Beckert, S. Grebing, and F. Böhl. A usability evaluation of interactive theorem provers using focus groups. In *Software Engineering and Formal Methods – SEFM 2014 Collocated Workshops*, Lecture Notes in Computer Science. Springer, 2014.
7. B. Beckert, R. Hähnle, and P. H. Schmitt, editors. *Verification of Object-Oriented Software: The KeY Approach*. LNCS 4334. Springer-Verlag, 2007.
8. B. Beckert, V. Klebanov, and S. Schlager. Dynamic logic. In Beckert et al. [7], chapter 3, pages 69–175.
9. J. Goguen. Social and semiotic analyses for theorem prover user interface design. *Formal Aspects of Computing*, 11:11–272, 1999.
10. U. Kuckartz. *Qualitative Inhaltsanalyse. Methoden, Praxis, Computerunterstützung*. Weinheim und Basel: Beltz Juventa, 2014.
11. P. Mayring. *Einführung in die qualitative Sozialforschung – Eine Anleitung zu qualitativem Denken (Introduction to qualitative social research)*. Weinheim: Psychologie Verlags Union, 1996.
12. P. Mayring. Qualitative content analysis. *Forum : Qualitative Social Research*, 1(2), June 2000. Online Journal, 1(2). Available at: `http://qualitative-research.net/fqs/fqs-e/2-00inhalt-e.htm` [Date of access: 04, 2014].
13. N. Merriam and M. Harrison. Evaluating the interfaces of three theorem proving assistants. In F. Bodart and J. Vanderdonckt, editors, *Design, Specification and Verification of Interactive Systems '96*, Eurographics, pages 330–346. Springer Vienna, 1996.
14. N. A. Merriam. Two modelling approaches applied to user interfaces to theorem proving assistants. In *Proceedings of the 2nd International Workshop on User Interface Design for Theorem Proving Systems.*, pages 75–82. Department of Computer Science, University of York, 1996.
15. N. Völker. Thoughts on requirements and design issues of user interfaces for proof assistants. *Electron. Notes Theor. Comput. Sci.*, 103:139–159, Nov. 2004.