Monadic Reasoning using Weak Completion Semantics

Ana Oliveira da Costa[1] ana.oli.costa@gmail.com

Emmanuelle-Anna Dietz Saldanha[1] dietz@iccl.tu-dresden.de

Steffen Hölldobler[1,2*] sh@iccl.tu-dresden.de

[1]International Center for Computational Logic, TU Dresden, Germany
[2]North-Caucasus Federal University, Stavropol, Russian Federation

Abstract

A recent meta-analysis carried out by Khemlani and Johnson-Laird showed that the conclusions drawn by humans in psychological experiments about syllogistic reasoning deviate from the conclusions drawn by classical logic. Moreover, none of the current cognitive theories predictions fit the empirical data. In this paper a Computational Logic analysis clarifies seven principles necessary to draw the inferences. We propose a modular approach towards these principles and show how human syllogistic reasoning can be modeled under a new cognitive theory, the Weak Completion Semantics.

1 Introduction

In this paper we present an approach to monadic reasoning using a new cognitive theory based on the Weak Completion Semantics (WCS) [Höl15]. In monadic reasoning, premises are assertions that assign properties to entities. In a meta-analysis by Khemlani and Johnson-Laird [KJ12] the authors argue that the development of a comprehensive theory of monadic reasoning is of major importance for the progress of Cognitive Science.

The Weak Completion Semantics (WCS) has its roots in the ideas first expressed by Stenning and van Lambalgen [SvL08]. Those had some technical errors corrected in [HK09a] by using the three-valued Łukasiewicz logic. WCS has been successfully applied, among others, to the suppression task [DHR12], the selection task [DHR13], the belief-bias effect [PDH14a, PDH14b, Die17], to reasoning about conditionals [DH15, DHP15], to spatial reasoning [DHH15] and to syllogistic reasoning [CDHR16, CDHR17]. Hence, we believe that the development of a general monadic reasoning theory based on WCS is the natural next step.

Our approach is based on reasoning principles, which are motivated by findings made in Cognitive Science and Computational Logic. Moreover, we identify two key reasoning tasks related to monadic reasoning and show the encoding of the principles for each of those distinct reasoning activities. In resume, we present a first steps in the direction of a modular encoding of monadic quantified assertions.

The study of monadic assertions by psychologists was mainly concerned with Aristotelian syllogisms. We follow the same strategy. In this work we extend the work in syllogistic reasoning presented in [CDHR16, CDHR17]. A syllogism consists of two premises and a conclusion. Each of them is a quantified statement using one of the four quantifiers *all* (A), *no* (E), *some* (I), and *some are not* (O)[1] about sets of entities which we denote in the following by a, b, and c. An example of two quantified statements is:

[*] The authors are mentioned in alphabetical order.

In: S. Hölldobler, A. Malikov, C. Wernhard (eds.): *YSIP2 – Proceedings of the Second Young Scientist's International Workshop on Trends in Information Processing, Dombai, Russian Federation, May 16–20, 2017*, published at http://ceur-ws.org.

Mood	Natural language	First-order logic	Short
affirmative universal	*all a are b*	$\forall X(a(X) \rightarrow b(X))$	Aab
affirmative existential	*some a are b*	$\exists X(a(X) \wedge b(X))$	Iab
negative universal	*no a are b*	$\forall X(a(X) \rightarrow \neg b(X))$	Eab
negative existential	*some a are not b*	$\exists X(a(X) \wedge \neg b(X))$	Oab

Table 1: The four moods and their logical formalization.

	1st Premise	2nd Premise
Fig. 1	a-b	b-c
Fig. 2	b-a	c-b
Fig. 3	a-b	c-b
Fig. 4	b-a	b-c

Table 2: The four figures.

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 3: The truth tables for the connectives under three-valued Łukasiewicz logic. Note that $(\mathsf{U} \leftarrow \mathsf{U}) = \top$.

Some b are a

No b are c

In experiments, participants are normally expected to complete the syllogism by drawing a logical consequence from the first two premises, e.g. in this example '*some a are not c*'. The participants' given response – the conclusion – is evaluated as true if it can be derived in classical first-order logic (FOL), otherwise as false. The four quantifiers and their formalization in FOL are given in Table 1. The entities can appear in four different orders called *figures* as shown in Table 2. Hence, a problem can be completely specified by the quantifiers of the first and second premise and the figure. The example discussed above is denoted by IE4. Altogether, there are 64 syllogisms and, if formalized in FOL, we can compute their logical consequence in classical logic. However, the meta-analysis by Khemlani and Johnson-Laird [KJ12] based on six experiments has shown that humans do not only systematically deviate from the predictions of FOL but from any other of 12 cognitive theories. In the case of IE4, besides the above mentioned logical consequence, a significant number of humans answered '*no valid conclusion*', which does not follow from IE4 in FOL, as *'some a are not c'* follows from IE4.

After we discuss our solution for syllogistic reasoning, we compare the predictions under WCS with the results of FOL, the syntactic rule based theory PSYCOP [Rip94], the Verbal Model Theory [PN95] and the Mental Model Theory [JL83]. The two model-based theories performed the best in the meta-analysis [KJ12].

2 Weak Completion Semantics

2.1 Logic Programs

A *(logic) program* $\mathcal{P}$ is a finite set of 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, respectively. Clauses are assumed to be universally closed. 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 $A \leftarrow \bot$[2] are called *facts* and *assumptions*,[3] respectively. $\neg A$ is *assumed* in $\mathcal{P}$ iff $\mathcal{P}$ contains an assumption with head A and no other clause with head A occurs in $\mathcal{P}$. We restrict terms to be constants or variables only, i.e. we consider so-called data logic programs. For each $\mathcal{P}$ the underlying alphabet consists precisely of the symbols occurring in $\mathcal{P}$ and that non-propositional programs contain at least one constant.

$\mathsf{g}\mathcal{P}$ denotes the set of all ground instances of clauses occurring in $\mathcal{P}$, where a ground instance of clause C is obtained from C by replacing each variable occurring in C by a constant. A ground atom A is *defined* in $\mathsf{g}\mathcal{P}$ iff $\mathsf{g}\mathcal{P}$ contains a clause whose head is A; otherwise A is said to be *undefined*. $def(A, \mathcal{P}) = \{ A \leftarrow Body \mid A \leftarrow Body \in \mathsf{g}\,\mathcal{P}\}$ is called *definition* of A in $\mathcal{P}$. The interested reader is referred to e.g. [Höl09, Llo84] for more details about classical logic and logic programs

2.2 Three-Valued Łukasiewicz Logic

We consider the three-valued Łukasiewicz logic [Łuk20] for which the corresponding truth values are *true* ($\top$), *false* ($\bot$) and *unknown* (U). A *three-valued interpretation* I is a mapping from the set of formulas to the set $\{\top, \bot, \mathsf{U}\}$. The truth value of a given formula under I is determined according to the truth tables in Table 3. We represent an interpretation as a pair $I = \langle I^\top, I^\bot \rangle$ of disjoint sets of ground 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 $\mathcal{P}$ if it is an interpretation, which maps each clause occurring in $\mathsf{g}\mathcal{P}$ to $\top$. I is the *least model* of $\mathcal{P}$ iff for any other model J of $\mathcal{P}$ it holds that $I \subseteq J$.

2.3 Integrity Constraints

An *integrity constraint* is an expression of the form $\mathsf{U} \leftarrow \mathit{Body}$, where *Body* is a conjunction of literals and U denotes the unknown. An interpretation I maps an integrity constraint $\mathsf{U} \leftarrow \mathit{Body}$ to $\top$ iff $I(\mathit{Body}) \subseteq \{\bot, \mathsf{U}\}$. Given an interpretation I and a finite set IC of integrity constraints, I *satisfies* IC iff all clauses occurring in IC are true under I.

2.4 Forms of Reasoning

The philosopher Peirce identified three forms of reasoning [PHW74]: deduction, induction and abduction. We focus here on deduction and abduction. For deduction we use the semantic operator $\Phi_\mathcal{P}$, defined below. For abduction we search explanations for some observation inspired by ideas introduced in Logic Programming.

2.4.1 Reasoning with Respect to Least Models

For a given program $\mathcal{P}$, consider the following transformation: (1) For each ground atom A which is defined in $\mathsf{g}\mathcal{P}$, replace all clauses of the form $A \leftarrow \mathit{Body}_1, \ldots, A \leftarrow \mathit{Body}_m$ occurring in $\mathsf{g}\mathcal{P}$ by $A \leftarrow \mathit{Body}_1 \vee \ldots \vee \mathit{Body}_m$. (2) Replace all occurrences of $\leftarrow$ by $\leftrightarrow$. The obtained set of formulas is called *weak completion* of $\mathcal{P}$ or $\mathit{wc}\mathcal{P}$.[4]

It has been shown by [HK09b] that programs as well as their weak completions admit a least model under three-valued Łukasiewicz logic. Moreover, the least model of $\mathit{wc}\mathcal{P}$ can be obtained as the least fixed point of the following semantic operator, which is due to Stenning and van Lambalgen [SvL08]: Let $I = \langle I^\top, I^\bot \rangle$ be an interpretation. $\Phi_\mathcal{P}(I) = \langle J^\top, J^\bot \rangle$, where

$$\begin{array}{rcl} J^\top & = & \{A \mid A \leftarrow \mathit{Body} \in \mathit{def}(A, \mathcal{P}) \text{ and } I(\mathit{Body}) = \top\}, \\ J^\bot & = & \{A \mid \mathit{def}(A, \mathcal{P}) \neq \emptyset \text{ and } I(\mathit{Body}) = \bot \text{ for all } A \leftarrow \mathit{Body} \in \mathit{def}(A, \mathcal{P})\}. \end{array}$$

The Weak Completion Semantics (WCS) is the approach to consider weakly completed programs, to compute their least models, and to reason with respect to these models. We write $\mathcal{P} \models_{wcs} F$ iff formula F holds in the least fixed point of $\Phi_\mathcal{P}$ (which is the least model of $\mathit{wc}\mathcal{P}$).

2.4.2 Backward Reasoning with Abduction

Abduction is a reasoning process that searches for explanations given a program and some observations, which do not follow from the program [KKT93]. Explanations are usually restricted to certain formulas called *abducibles*. The *set of abducibles w.r.t.* $\mathcal{P}$ is

$$\mathcal{A}_\mathcal{P} = \{A \leftarrow \top \mid A \text{ is undefined in } \mathsf{g}\mathcal{P}\} \cup \{A \leftarrow \bot \mid A \text{ is undefined in } \mathsf{g}\mathcal{P}\} \cup \{A \leftarrow \top \mid \neg A \text{ is assumed in } \mathsf{g}\mathcal{P}\}.$$

An *abductive framework* consists of a program $\mathcal{P}$, a finite set $\mathcal{A}_\mathcal{P}$ of abducibles, a finite set IC of integrity constraints, and an entailment relation. Let $\langle \mathcal{P}, \mathcal{A}_\mathcal{P}, \mathsf{IC}, \models_{wcs} \rangle$ be an abductive framework, $\mathcal{E} \subseteq \mathcal{A}_\mathcal{P}$, and $\mathcal{O}$ a non-empty set of literals called *observation*. An observation $\mathcal{O} = \{o_1, \ldots, o_n\}$ is *explained by* $\mathcal{E}$ *given* $\mathcal{P}$ *and* IC iff $\mathcal{P} \cup \mathcal{E} \models_{wcs} o_1 \wedge \ldots \wedge o_n$ and $\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathsf{IC}$. $\mathcal{O}$ is *explained given* $\mathcal{P}$ *and* IC iff there exists an $\mathcal{E}$ such that $\mathcal{O}$ is explained by $\mathcal{E}$ given $\mathcal{P}$ and IC. We prefer subset-minimal explanations. An explanation $\mathcal{E}$ is subset-minimal iff there is no explanation $\mathcal{E}'$ such that $\mathcal{E}' \subset \mathcal{E}$.

[2] We consider the weak completion of programs and, hence, a clause of the form $A \leftarrow \bot$ is turned into $A \leftrightarrow \bot$ provided that this is the only clause in the program where A is the head of.

[3] $A \leftarrow \bot$ is called an assumption because it can be overwritten under the Weak Completion Semantics, as we will discuss later.

[4] If $\mathcal{P} = \{A \leftarrow \bot, A \leftarrow \top\}$ then $\mathit{wc}\mathcal{P} = \{A \leftrightarrow \bot \vee \top\}$. This is semantically equivalent to $\mathit{wc}\mathcal{P} = \{A \leftrightarrow \top\}$. $A \leftarrow \bot$ is overwritten.

3 Reasoning with Monadic Quantified Assertions

In their meta-study [KJ12], Khemlani and Johnson-Laird enumerated some tasks related to monadic reasoning that have been investigated experimentally. Those include: given two premises and a conclusion, the task is to evaluate if the conclusion follows necessarily or possibly from the premises; given two premises, the task is to check if they are consistent or to formulate a conclusion that follows from them; and, given two premises and an invalid conclusion, the task is to formulate counterexamples, which refute this syllogism. They argue that not only most individuals can understand those tasks, but are also able to develop procedures to carry them out. Our approach to model monadic reasoning provides tools to derive conclusions similar to conclusions drawn by humans.

We identify two types of reasoning: *reasoning towards a representation* and *reasoning from a representation*. The first has the representation of assertions as output and the second has a representation of assertions as input. In this work a representation of assertions is a three-value interpretation. In the principles discussed below we define one logic program for the first type of reasoning and one logic program for the second type. Principles can be combined independently.

3.1 Reasoning Principles

Our reasoning principles are based on findings in Cognitive Science and Logical Programming. Note that assertions can be either positive or negative. We explain only the encoding of positive assertions. The encoding of negative assertions follows analogously. Table 4 shows which clauses need to be added to the logic program, which encodes either the task to reason towards a representation or to reason from a representation, when one of our principles is considered.

3.1.1 Assertion as Rule (rule)

Under WCS we can only encode assertions as rules with two or more distinct predicates. Therefore, the encoding towards a representation of an assertion that establishes, for example, a relation from the predicate y to the predicate z includes the rule $z(X) \leftarrow y(X)$. In the case that we reason from a representation we have nothing to check related to this principle and encode this principle as the fact $rule_{yz} \leftarrow \top$.

3.1.2 Licenses for Inferences (licenses)

Stenning and van Lambalgen [SvL08] propose to formalize conditionals by *licenses for inferences*. For example, the conditional *for all* X*, if* $p(X)$ *then* $q(X)$ is represented by the program $\{q(X) \leftarrow p(X) \wedge \neg ab(X),\ ab(X) \leftarrow \bot\}$. Its first clause states that for all X, $q(X)$ holds if $p(X)$ holds and nothing abnormal for X is known. Clauses are assumed to be universally closed and, hence, the universal quantifier can be omitted. Licenses are encoded by abnormality predicates, which are usually of the form $ab(X)$. This principle is encoded in the same way for both types of reasoning.

3.1.3 Existential Import and Gricean Implicature (import)

Humans seem to understand quantifiers differently due to a pragmatic understanding of language. For instance, in natural language we normally do not quantify over things that do not exist. Consequently, *for all* implies *there exists*. This appears to be in line with human reasoning and has been called the *Gricean Implicature* [Gri75]. Several theories like the theory of mental models [JL83] or mental logic [Rip94] assume that the sets we quantify about are not empty. Likewise, Stenning and van Lambalgen [SvL08] have shown that humans require existential import for a conditional to be true.

Consider the conditional *for all* X*, if* $y(X)$ *then* $z(X)$ encoded by the clause $z(X) \leftarrow y(X)$. The principle import implies that there is an object that belongs to both predicates y and z. In reasoning towards a representation, we encode import by adding the fact $y(o) \leftarrow \top$. If in the current reasoning task we consider licenses as well, then we need to add the assumption $ab_{yz}(o) \leftarrow \bot$ to assert that nothing abnormal between these two predicates is known for this object. In reasoning from a representation we just need to check that in the current representation there exists this object that is in both predicates. We encode it with the rule $exist_{yz} \leftarrow y(X) \wedge z(X)$.

Principle with $\mathsf{licenses}$	Towards a Representation	From a Representation
rule	$z(X) \leftarrow y(X) \wedge \neg ab_{yz}(X)$	$rule_{yz} \leftarrow \neg ab_{yz}(X)$
rule *negative*	$z'(X) \leftarrow y(X) \wedge \neg ab_{yz}(X)$	$rule_{yz'} \leftarrow \neg ab_{yz}(X)$
import	$ab_{yz}(o) \leftarrow \bot$ $y(o) \leftarrow \top$	$exist_y \leftarrow y(X) \wedge z(X) \wedge \neg ab_{yz}(X)$
import *negative*	$ab_{yz}(o) \leftarrow \bot$ $y(o) \leftarrow \top$	$existneg_{yz} \leftarrow y(X) \wedge \neg z(X) \wedge \neg ab_{yz}(X)$
rule *with* import	$z(X) \leftarrow y(X) \wedge \neg ab_{yz}(X)$ $ab_{yz}(o) \leftarrow \bot$ $y(o) \leftarrow \top$	$rule_{yz} \leftarrow exist_{yz}$ $exist_{yz} \leftarrow y(X) \wedge z(X) \wedge \neg ab_{yz}(X)$
rule *with* import *negative*	$z'(X) \leftarrow y(X) \wedge \neg ab_{yz}(X)$ $ab_{yz}(o) \leftarrow \bot$ $y(o) \leftarrow \top$	$rule_{yz} \leftarrow existneg_{yz}$ $existneg_{yz} \leftarrow y(X) \wedge \neg z(X) \wedge \neg ab_{yz}(X)$
norefutation	$ab_{yz}(X) \leftarrow \bot$	$norefute_{yz} \leftarrow \neg refute_{yz}$ $refute_{yz} \leftarrow y(X) \wedge \neg z(X) \wedge \neg ab_{yz}(X)$
norefutation *negative*	$ab_{yz}(X) \leftarrow \bot$	$norefuteneg_{yz} \leftarrow \neg refuteneg_{yz}$ $refuteneg_{yz} \leftarrow y(X) \wedge z(X) \wedge \neg ab_{yz}(X)$
unknownGen	$y(o) \leftarrow \top$	$gen_{yz} \leftarrow y(X) \wedge uz(X) \wedge \neg ab_{yz}(X)$ $gen_{yz} \leftarrow y(X) \wedge \neg z(X) \wedge \neg ab_{yz}(X)$
unknownGen *negative*	$y(o) \leftarrow \top$	$genneg_{yz} \leftarrow y(X) \wedge uz(X) \wedge \neg ab_{yz}(X)$ $genneg_{yz} \leftarrow y(X) \wedge z(X) \wedge \neg ab_{yz}(X)$
doubleNeg	$ab_{nzz}(o) \leftarrow \bot$	–
transformation	$z(X) \leftarrow \neg z'(X) \wedge \neg ab_{nzz}(X)$	–

Table 4: Encoding of reasoning principles. We show the encoding for the two types of reasoning considered in our approach: *reasoning towards a representation* and *reasoning from a representation*. Note that, o is a new object not occurring in the program. Logic programs in this table incorporate the use of licenses. To encode these principles without licenses it suffices to remove all references to abnormality predicates..

3.1.4 Unknown Generalization (unknownGen)

Humans seem to distinguish between '*some y are z*' and '*all y are z*'. Accordingly, if we observe that an object o belongs to y and z then we do not want to conclude both, '*some y are z*' and '*all y are z*'. In order to prevent such unwanted conclusions we introduce the following principle: if we know that '*some y are z*' then there must not only be an object o_1 which belongs to y and z (by Gricean implicature) but there must be another object o_2 which belongs to y and for which it is unknown whether it belongs to z.

In reasoning to a representation, the encoding of this principle is only possible when the principle licenses is used in the current reasoning task, too. Consider the same clause used as example in previous principles extended with abnormalities, $q(X) \leftarrow p(X) \wedge \neg ab(X)$. We encode it by adding the fact $p(o) \leftarrow \top$ and not adding a clause with $ab(o)$ in the head. $ab(o)$ will be then evaluated to unknown, and as consequence $q(o)$ is unknown, too. In reasoning from a representation we check if there exist an object that it is in p and that either it is not in q or it is unknown if it belongs to q.

3.1.5 Refutation by Counterexample (norefutation)

Empirical findings support the hypothesis that people spontaneously use counterexamples in monadic reasoning [KJ12]. When we reason towards a representation we use this principle by explicitly stating which objects are not expected to be part of a refutation. This is done using abnormality predicates. For example, if we don't expect the object o to be used in a counterexample for a rule $z(X) \leftarrow y(X) \wedge ab(X)$ we add the assumption $ab(o) \leftarrow \bot$ to the program. We generalize this to universally quantified assertions by assuming that no counterexamples are expected for any object, i.e. we add to our program $ab(X) \leftarrow \bot$. Note again that the principle norefutation can only be used in reasoning towards a representation if licenses are considered, too.

Finally, when we reason from an interpretation we add rules with *head refute* that describe our counterexamples. This principle is then defined by the rule $norefute_{yz} \leftarrow \neg refute_{yz}$.

3.1.6 Converse Interpretation (converse)

Although there appears to be some evidence that humans seem to distinguish between '*some y are z*' and '*some z are y*' (see the results reported in [KJ12]) we propose that Iab implies Iba and vice versa. If there is an object which belongs to y and z, then there is also an object which belongs to z and y. Consider the conditional *for all X, if $p(X)$ then $q(X)$*. Then, by converse we encode *for all X, if $q(X)$ then $p(X)$*, too. This applies to both types of reasoning.

3.1.7 No Derivation by Double Negation (doubleNeg)

Consider the following two negative sentences (i.e. including negation) and the positive one: ' If not a, then b. If not b then c. a is true.' The program representing these sentences is $\mathcal{P} = \{b \leftarrow \neg a,\ c \leftarrow \neg b,\ a \leftarrow \top\}$. The weak completion of $\mathcal{P}$ is $\mathsf{wc}\,\mathcal{P} = \{b \leftrightarrow \neg a,\ c \leftrightarrow \neg b,\ a \leftrightarrow \top\}$. Its least model is $\langle\{a, c\}, \{b\}\rangle$, and thus a and c are true: a is true because it is a fact and c is true by the negation of b. b is derived to be false because the negation of a is false. This example shows that under WCS, a positive conclusion (c being true) can be derived from two clauses, which include negation. Considering the results of the participants' responses in [KJ12], they seem not to draw conclusions through double negatives. Accordingly, we block them through abnormalities.

3.1.8 Negation by Transformation (transformation)

A negative literal cannot be the head of a clause in a program. In order to represent a negative conclusion $\neg p(X)$ an auxiliary atom $p'(X)$ is introduced together with a clause $p(X) \leftarrow \neg p'(X)$ and the integrity constraint $\mathsf{U} \leftarrow p(X) \wedge p'(X)$. This is a widely used technique in logic programming. Together with the principle *licences for inferences*, the additional clause becomes $p(X) \leftarrow \neg p'(X) \wedge \neg ab(X)$.

3.2 Encoding of a Representation

When reasoning from a representation we need to encode the representation, i.e. the three-value interpretation, as a logic program. We start by defining predicates and objects that we are interested to reason about, as not all elements in our interpretation are relevant to derive conclusions. Ground atoms defined by those predicates and objects are called *relevant atoms*. Then, given an interpretation, we build a logic program by adding clauses that encode the evaluation of the relevant atoms. Thus, if the relevant atoms are evaluated to *true* or *false* we add facts or assumptions to our program, respectively. If some relevant atom is *unknown* we consider a new predicate that is defined by the original atom predicate name prefixed by u (from unknown). We then add a fact to the program with that new atom in the head. For example, if the relevant atom $y(o)$ is evaluated to *unknown* we add the fact $uy(o) \leftarrow \top$ to our program.

3.3 Entailment of Quantified Assertions

The goal of reasoning from a representation is to entail conclusions. A conclusion is a quantified assertion. For each of the four possible moods we define the principles to be considered in their encoding. A conclusion is entailed if all principles defined for the mood of the conclusion are satisfied in the current representation.

There is a rule for each conclusion. The head of that rule is the predicate related to that conclusion; and its body is a conjunction of atoms related to the principles used in the encoding of that conclusion. The predicate related to conclusion Azy, Izy, Ezy or Ozy is the predicate a_{zy}, i_{zy}, e_{zy} or o_{zy}, respectively. Therefore, a conclusion is entailed iff its related atom is entailed. We add the rule $ab_{yz}(X) \leftarrow uy(X)$ to avoid atoms that are evaluated to *unknown* interfering with the entailment of conclusions. For example, to entail the conclusion Aac we don't consider ground atoms with predicate a that are evaluated to *unknown*. In the next chapter we present our encoding of syllogisms and we show how the entailment rules are defined for each mood.

3.4 Search Alternative Conclusions to NVC : Abduction

Our hypothesis is that when participants are faced with a NVC conclusion (*'no valid conclusion'*), they might not want to accept this conclusion and proceed to check whether there exists unknown information that is relevant. This information may be explanations to facts in our program, and we model such repair mechanism

Mood	Principles Used	From a Representation
Ayz	rule *with* import and norefutation licenses	$a_{yz} \leftarrow rule_{yz} \wedge norefute_{yz} \wedge \neg ab_{ayz}$ $ab_{ayz} \leftarrow \bot$
Eyz	rule with import *negative* and norefutation *negative* licenses	$e_{yz} \leftarrow ruleneg_{yz} \wedge norefuteneg_{yz} \wedge \neg ab_{eyz}$ $ab_{eyz} \leftarrow \bot$
Iyz	rule with import, unknownGen and converse licenses	$i_{yz} \leftarrow rule_{yz} \wedge gen_{yz} \wedge rule_{zy} \wedge gen_{zy} \wedge \neg ab_{iyz}$ $ab_{iyz} \leftarrow \bot$
Oyz	rule with import *negative* and unknownGen *negative* licenses	$o_{yz} \leftarrow ruleneg_{yz} \wedge genneg_{yz} \wedge \neg ab_{oyz}$ $ab_{oyz} \leftarrow \bot$

Table 5: Reasoning principles used for syllogistic reasoning and clauses scheme to encode conclusions' entailment.

using skeptical abductive reasoning. Facts in our programs come either from an existential import or from unknown generalization. We use only the first as source for observations, since they are used directly to infer new information.

Each head of an existential import generates a single observation. We apply abduction sequentially to each of them. To prevent empty explanations we remove from the current program the fact that generated the observation. For each observation and each of its minimal explanations we compute the least model of the weak completion of the program extended with the explanation and collect all entailed syllogistic conclusions. Observations that cannot be explained are filtered out. Let Answers consist of all entailed conclusions obtained in that way. The final conclusion is obtained by following a skeptical reasoning, i.e. the final answer to the current syllogism is given by $\mathsf{FinalAnswer} = \cap_{A \in \mathsf{Answers}} A$. In the case that FinalAnswer is empty, we entail the NVC conclusion.

4 Syllogisms: Use Case

The principles used in our syllogistic encoding are enumerated in Table 5. Note that we are only interested in conclusions between predicates a and c. When such a conclusion is not possible we entail NVC. As explained before, a NVC conclusion triggers the use of abduction.

4.1 Accuracy of Predictions

We follow the evaluation proposed by [KJ12]: There are nine different answers for each of the 64 syllogisms that can be ordered in a list: Aac, Eac, Iac, Oac, Aca, Eca, Ica, Oca, and NVC. For each answer (e.g., Aac) we assign a 1 to it, if this is predicted under WCS and else a 0. Analogously, for the percentages of participants' responses we can use a threshold-function like [KJ12] that any value above 16% [5]assigned a 1 and else a 0. Both lists can then be compared for their congruency as follows, where i is the ith element of both lists:

$$\textsc{comp}(i) = \begin{cases} 1 & \text{if both have the same value for } i\text{th element,} \\ 0 & \text{otherwise.} \end{cases}$$

The matching percentage of this syllogism is then computed by $\sum_{i=1}^{9} \textsc{comp}(i)/9$. Note that the percentage of the match does not only take in account when WCS correctly predicts a conclusion, but also whenever it correctly rejected a conclusion. The average percentage of accuracy is then simply the average of the matching percentage of all 64 syllogisms.

[5]Given that there are nine different conclusion possibilities the chance that a conclusion has been chosen randomly is $1/9 = 11.1\%$; moreover, a binomial test shows that if a conclusion is drawn in more than 16% of the cases by the participants it is unlikely that has been chosen by just random guesses. The statistical analysis is elaborately explained by [KJL12].

4.2 IA2: Perfect Match (100%)

The two syllogistic premises of IA2 are as follows:

Some b are a. (Iba) *All c are b. (Acb)*

First we develop a program to reason *towards a representation* by considering the principles listed in Table 5. After that, we need to encode their respective rules that are presented in the second column in Table 4. Program $\mathcal{P}_{\mathsf{IA2}}$ consists of the following clauses:

$$
\begin{array}{ll}
a(X) \leftarrow b(X) \wedge \neg ab_{ba}(X). & (\mathsf{rule\&licenses}) \\
ab_{ba}(o_1) \leftarrow \bot. & (\mathsf{import\&licenses}) \\
b(o_1) \leftarrow \top. & (\mathsf{import}) \\
b(o_2) \leftarrow \top. & (\mathsf{unknownGen}) \\
b(X) \leftarrow a(X) \wedge \neg ab_{ab}(X). & (\mathsf{converse\&rule\&licenses}) \\
ab_{ab}(o_3) \leftarrow \bot. & (\mathsf{converse\&import\&licenses}) \\
a(o_3) \leftarrow \top. & (\mathsf{converse\&import}) \\
a(o_4) \leftarrow \top. & (\mathsf{converse\&unknownGen}) \\
b(X) \leftarrow c(X) \wedge \neg ab_{cb}(X). & (\mathsf{rule\&licenses}) \\
ab_{cb}(X) \leftarrow \bot. & (\mathsf{norefutation\&licenses}) \\
c(o_5) \leftarrow \top. & (\mathsf{import})
\end{array}
$$

The least model of $\mathsf{wc}\,\mathcal{P}_{\mathsf{IA2}} = \langle I^\top, I^\bot \rangle$ is

$$\langle \{a(o_1), a(o_3), a(o_4), b(o_1), b(o_2), b(o_3), b(o_5), c(o_5)\}, \{ab_{ba}(o_1), ab_{ab}(o_3)\} \cup \{ab_{cb}(o_i) \mid i \in \{1,2,3,4,5\}\}\rangle. \quad (1)$$

Next, in order to derive from this model some conclusion, we construct a new program to *reason from a representation*, i.e. to reason from the least model. We consider again the principles in Table 5, but now we use the third column in Table 4. Additionally, we need to encode the representation of our relevant atoms by adding facts and assumptions according to the least model of $\mathsf{wc}\,\mathcal{P}_{\mathsf{IA2}}$. The relevant atoms are $\mathcal{X} = \{a(o), c(o) \mid o \in \{o_1, o_3, o_4, o_5\}\}$. The new program $\mathcal{P}^*_{\mathsf{IA2}}$ consists of the following clauses:

$$
\begin{array}{ll}
\{\, A \leftarrow \top \mid \mathcal{P}_{\mathsf{IA2}} \models_{wcs} A \textit{ and } A \in \mathcal{X}\} \,\cup & \\
\{\, A \leftarrow \bot \mid \mathcal{P}_{\mathsf{IA2}} \models_{wcs} \neg A \textit{ and } A \in \mathcal{X}\} \,\cup & \\
\{\, uA \leftarrow \top \mid \mathcal{P}_{\mathsf{IA2}} \not\models_{wcs} (A \vee \neg A) \textit{ and } A \in \mathcal{X}\} \,\cup & \\
\{\, ab_{yz}(X) \leftarrow uy(X) \mid y \in \{a,c\} \text{ and } z \in \{a,c\}\} \,\cup & \\
\{\, a_{yz} \leftarrow \mathit{rule}_{yz} \wedge \mathit{norefute}_{yz} \wedge \neg ab_{ayz}, ab_{ayz} \leftarrow \bot, & (\mathsf{Ayz}) \\
\;\; e_{yz} \leftarrow \mathit{ruleneg}_{yz} \wedge \mathit{norefuteneg}_{yz} \wedge \neg ab_{eyz}, ab_{eyz} \leftarrow \bot, & (\mathsf{Eyz}) \\
\;\; i_{yz} \leftarrow \mathit{rule}_{yz} \wedge \mathit{gen}_{yz} \wedge \mathit{rule}_{zy} \wedge \mathit{gen}_{zy} \wedge \neg ab_{iyz}, ab_{iyz} \leftarrow \bot, & (\mathsf{Iyz}) \\
\;\; o_{yz} \leftarrow \mathit{ruleneg}_{yz} \wedge \mathit{genneg}_{yz} \wedge \neg ab_{oyz}, ab_{oyz} \leftarrow \bot & (\mathsf{Oyz}) \\
\qquad\qquad \mid y \in \{a,c\} \text{ and } z \in \{a,c\} \,\} \,\cup & \\
\{\, \mathit{exist}_{yz} \leftarrow y(X) \wedge z(X) \wedge \neg ab_{yz}(X), \mathit{existneg}_{yz} \leftarrow y(X) \wedge \neg z(X) \wedge \neg ab_{yz}(X), & (\mathsf{import}\&\mathit{negative}) \\
\;\; \mathit{rule}_{yz} \leftarrow \mathit{exist}_{yz}, \mathit{exist}_{yz} \leftarrow y(X) \wedge z(X) \wedge \neg ab_{yz}(X), & (\mathsf{rule}\ \mathit{with}\ \mathsf{import}) \\
\;\; \mathit{rule}_{yz} \leftarrow \mathit{existneg}_{yz}, \mathit{existneg}_{yz} \leftarrow y(X) \wedge \neg z(X) \wedge \neg ab_{yz}(X), & (\mathsf{rule}\ \mathit{with}\ \mathsf{import}\ \mathit{negative}) \\
\;\; \mathit{norefute}_{yz} \leftarrow \neg \mathit{refute}_{yz}, \mathit{refute}_{yz} \leftarrow y(X) \wedge \neg z(X) \wedge \neg ab_{yz}(X), & (\mathsf{norefutation}) \\
\;\; \mathit{norefuteneg}_{yz} \leftarrow \neg \mathit{refute}_{yz}, \mathit{refuteneg}_{yz} \leftarrow y(X) \wedge \neg z(X) \wedge \neg ab_{yz}(X), & (\mathsf{norefutation}\ \mathit{negative}) \\
\;\; \mathit{genneg}_{yz} \leftarrow y(X) \wedge uz(X) \wedge \neg ab_{yz}(X), \mathit{genneg}_{yz} \leftarrow y(X) \wedge \neg z(X) \wedge \neg ab_{yz}(X) & (\mathsf{unknownGen}\ \mathit{negative}) \\
\;\; \mathit{gen}_{yz} \leftarrow y(X) \wedge uz(X) \wedge \neg ab_{yz}(X), \mathit{gen}_{yz} \leftarrow y(X) \wedge \neg z(X) \wedge \neg ab_{yz}(X), & (\mathsf{unknownGen}) \\
\qquad\qquad \mid y \in \{a,c\} \text{ and } z \in \{a,c\} \,\}. &
\end{array}
$$

For $y \in \{a,c\}$ and $z \in \{a,c\}$, $\mathcal{P}^*_{\mathsf{IA2}} \not\models_{wcs} a_{yz} \vee e_{yz} \vee i_{yz} \vee o_{yz}$, because $\mathcal{P}^*_{\mathsf{IA2}} \not\models_{wcs} \mathit{exist}_{yz}$ and $\mathcal{P}^*_{\mathsf{IA2}} \not\models_{wcs} \mathit{existneg}_{yz}$. Thus, we entail *'no valid conclusion'* (NVC). However, a significant percentage of participants answered Iac and Ica, despite IA2 being an invalid syllogism in classical FOL. According to our sixth principle, abduction, our hypothesis is that these participants might have searched for alternatives to NVC.

The observations are $\mathcal{O}_1 = \{b(o_1)\}$, $\mathcal{O}_2 = \{a(o_3)\}$ and $\mathcal{O}_3 = \{c(o_5)\}$. If we examine $\mathcal{O}_i = \{o\}$ with $i \in \{1,2,3\}$, then we will try to find an explanation for $\mathcal{O}_i$ with respect to $\mathcal{P}_{\mathsf{IA2}} \setminus \{o \leftarrow \top\}$.[6] The set of abducibles is:

[6] We remove the fact from the program that generated the observation, because otherwise the explanation would be empty.

	Participants	FOL	PSYCOP	Verbal Models	Mental Models	WCS
OA4	Oca, Ica, Iac	Oca	Oca,	Oca, NVC	Oca, Oac, NVC	Oca
IE4	Oac, NVC	Oac	Oac, Iac, Ica	Oac, NVC	Oac, NVC Eac, Eca, Oca	Oac
IA2	Iac, Ica	NVC	NVC	Ica, NVC	Iac, Ica, NVC	Iac, Ica
	100%	-	77%	84%	83%	89%

Table 6: Conclusions drawn by a significant percentage of participants are highlighted in gray and compared to the predictions of the theories FOL, PSYCOP, Verbal, and Mental Models as well as WCS for the syllogisms OA4, IE4, and IA2.

$$\begin{array}{rll} \mathcal{A}_{\mathcal{P}_{\mathsf{IA2}}} & = \{ab_{ba}(o_i) \leftarrow \top,\ ab_{ba}(o_i) \leftarrow \bot & \mid i \in \{2,3,4,5\}\} \\ & \cup\ \{ab_{ab}(o_i) \leftarrow \top,\ ab_{ab}(o_i) \leftarrow \bot & \mid i \in \{1,2,4,5\}\} \\ & \cup\ \{c(o_i) \leftarrow \top, \quad c(o_i) \leftarrow \bot & \mid i \in \{1,2,3,4\}\} \\ & \cup\ \{ab_{cb}(o_5) \leftarrow \top & \mid i \in \{1,2,3,4,5\}\} \\ & \cup\ \{ab_{ba}(o_1) \leftarrow \top,\ ab_{ab}(o_3) \leftarrow \top\ \}. & \end{array}$$

$\mathcal{E}_1 = \{c(o_1) \leftarrow \top\}$ and $\mathcal{E}_2 = \{c(o_3) \leftarrow \top,\ ab_{ba}(o_3) \leftarrow \bot\}$ are the minimal explanations for $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. Note that for $\mathcal{O}_3$ there is no explanation.

Consider $\mathcal{O}_1 = \{b(o_1)\}$, where the program to be taken into account is $\mathcal{P}^1_{\mathsf{IA2}} = (\mathcal{P}_{\mathsf{IA2}} \setminus \{b(o_1) \leftarrow \top\}) \cup \mathcal{E}_1$. Given the least model of $\mathsf{wc}\,\mathcal{P}_{\mathsf{IA2}} = \langle I^\top, I^\bot \rangle$ as defined in (1), the least model of $\mathsf{wc}\,(\mathcal{P}^1_{\mathsf{IA2}})$ is $I = \langle I^\top \cup \{c(o_1)\}, I^\bot \rangle$, i.e. $c(o_1)$ is newly entailed to be true after applying abduction. This model entails what participants concluded, namely Iac and Ica.

For the observation $\mathcal{O}_2 = \{a(o_3)\}$ we consider the program $\mathcal{P}^2_{\mathsf{IA2}} = (\mathcal{P}_{\mathsf{IA2}} \setminus \{a(o_3) \leftarrow \top\ \}) \cup \mathcal{E}_2$. The least model of $\mathcal{P}^2_{\mathsf{IA2}}$ also entails the conclusions Iac and Ica.

$\mathsf{Answers}(\mathcal{P}_{\mathsf{IA2}}) = \{\{\mathsf{I}ac, \mathsf{I}ca\},\ \{\mathsf{I}ac, \mathsf{I}ca\}\}$ is the set of all conclusions. $\mathsf{FinalAnswer}(\mathcal{P}_{\mathsf{IA2}}) = \{\mathsf{I}ac, \mathsf{I}ca\}$ consists of the skeptically entailed conclusions, i.e. it is the intersection of all conclusions, which in this case are '*some a are c*' (Iac) and '*some c are a*' (Ica).

4.3 Overall Accuracy of 89%

The results of the three examples formalized under WCS are summarized and compared to FOL, PSYCOP, the Verbal, and the Mental Model Theory in Table 6. For some syllogisms the conclusions drawn by the participants and WCS are identical and for some syllogisms the conclusions drawn by the participants and WCS overlap. WCS differs from the other cognitive theories. Combining the syllogistic premises representation and entailment rules for all 64 syllogistic premises and applying abduction when NVC was entailed (which happened in 43 cases), we accomplished an average of 89% accuracy in our predictions. In 18 cases we have a perfect match, in 30 cases the match is 89%, in 13 cases the match is 78%, and in the remaining three cases the match is 67%. Compared to the other cognitive theories, we achieve the best performance, as their best results were accomplished by the Verbal Models Theory (84%) and the Mental Model Theory (83%).

5 Final Remarks

We presented a theory that is modular, i.e. each of our encoding principles can be considered independently. This feature allow us to consider any combination of principles in the encoding of quantified assertion, and this is particularly relevant if we want to encode the reasoning processes as preform by an individual our by a group of individuals. Moreover, in our approach the task to represent information and the task to derive new conclusions are decoupled. This means that we can consider different principles for each of those tasks and further investigate differences and similarities between them.

Following this approach we encoded syllogistic reasoning and compared to other cognitive theories. We perform the best with an overall accuracy of 89% in our predictions.

References

[CDHR16] A. Costa, E.-A. Dietz, S. Hölldobler, and M. Ragni. Syllogistic reasoning under the weak completion semantics. In U. Furbach and C. Schon, editors, *Bridging 2016 – Bridging the Gap between Human and Automated Reasoning*, volume 1651 of *CEUR Workshop Proceedings*, pages 5–19. CEUR-WS.org, 2016.

[CDHR17] A. Costa, E.-A. Dietz, S. Hölldobler, and M. Ragni. A computational logic approach to human syllogistic reasoning. In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*, 2017. accepted.

[DH15] 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*, volume 9345 of *Lecture Notes in Artificial Intelligence*, pages 265–278. Springer, 2015.

[DHH15] E.-A. Dietz, S. Hölldobler, and R. Höps. A computational logic approach to human spatial reasoning. In *IEEE Symposium Series on Computational Intelligence*, pages 1627–1634. IEEE, 2015.

[DHP15] E.-A. Dietz, S. Hölldobler, and L. M. Pereira. On conditionals. In G. Gottlob, G. Sutcliffe, and A. Voronkov, editors, *Global Conference on Artificial Intelligence*, Epic Series in Computing, 2015.

[DHR12] 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.

[DHR13] 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.

[Die17] E.-A. Dietz. A computational logic approach to the belief bias in human syllogistic reasoning. In *10th International and Interdisciplinary Conference on Modeling and Using Context*, volume 10257 of *Lecture Notes in Computer Science*. Springer, 2017.

[Gri75] H. P. Grice. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and semantics*, volume 3. New York: Academic Press, 1975.

[HK09a] S. Hölldobler and C. D. Kencana Ramli. Logic programs under three-valued Łukasiewicz semantics. In P. M. Hill and D. S. Warren, editors, *25th International Conference on Logic Programming*, volume 5649 of *Lecture Notes in Computer Science*, pages 464–478, Heidelberg, 2009. Springer.

[HK09b] S. Hölldobler and C. D. Kencana Ramli. Logics and networks for human reasoning. In *International Conference on Artificial Neural Networks*, pages 85–94, 2009.

[Höl09] S. Hölldober. *Logik und Logikprogrammierung*, volume 1: Grundlagen. Synchron Publishers, Heidelberg, 2009.

[Höl15] S. Hölldobler. Weak completion semantics and its applications in human reasoning. In U. Furbach and Claudia Schon, editors, *Proceedings of the Workshop on Bridging the Gap between Human and Automated Reasoning on the 25th International Conference on Automated Deduction*, pages 2–16. CEUR-WS.org, 2015.

[JL83] P. N Johnson-Laird. *Mental models: Towards a cognitive science of language, inference, and consciousness.* Harvard University Press, 1983.

[KJ12] S. Khemlani and P. N. Johnson-Laird. Theories of the syllogism: A meta-analysis. *Psychological Bulletin*, pages 427–457, 2012.

[KJL12] Sangeet Khemlani and P. N. Johnson-Laird. Theories of the Syllogism: A Meta-Analysis. *Psychological Bulletin*, 138:427–457, 2012.

[KKT93] A. C. Kakas, R. A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1993.

[Llo84] J. W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag New York, Inc., New York, NY, USA, 1984.

[Łuk20] Jan Łukasiewicz. O logice trójwartościowej. *Ruch Filozoficzny*, 5:169–171, 1920.

[PDH14a] L. M. Pereira, E.-A. Dietz, and S. Hölldobler. An abductive reasoning approach to the belief-bias effect. In C. Baral, G. De Giacomo, and T. Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 14th International Conference*, pages 653–656, 2014.

[PDH14b] 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*, volume 14, pages 633–648. Cambridge University Press, 2014.

[PHW74] C.S. Peirce, C. Hartshorne, and P. Weiss. *Collected Papers of Charles Sanders Peirce.* Collected papers of Charles Sanders Peirce. Belknap Press of Harvard University Press, 1974.

[PN95] Thad A Polk and Allen Newell. Deduction as verbal reasoning. *Psychological Review*, 102(3):533, 1995.

[Rip94] L. J. Rips. *The psychology of proof: Deductive reasoning in human thinking.* MIT Press, Cambridge, MA, 1994.

[SvL08] K. Stenning and M. van Lambalgen. *Human Reasoning and Cognitive Science.* A Bradford Book. MIT Press, Cambridge, MA, 2008.