



# A Computational Logic Approach to Human Spatial Reasoning

E.-A. Dietz

S. Hölldobler

R. Höps

## KRR Report 15-02

Mail to  
Technische Universität Dresden  
01062 Dresden

Bulk mail to  
Technische Universität Dresden  
Helmholtzstr. 10  
01069 Dresden

Office  
Technische Universität Dresden Room 2006  
Nöthnitzer Straße 46  
01187 Dresden

Internet  
<http://www.wv.inf.tu-dresden.de>

# A Computational Logic Approach to Human Spatial Reasoning

Emmanuelle-Anna Dietz, Steffen Hölldobler, Raphael Höps\*

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

**Abstract.** In this paper we present a new approach with respect to spatial reasoning problems by using logic programs. Because the weak completion of a logic program admits a least model under the three-valued Łukasiewicz semantics and this semantics has been successfully applied to other human reasoning tasks, conditionals are evaluated under these least  $\mathbb{L}$ -models. In this paper we show that the weak completion semantics can also handle spatial relations in a way humans do. In particular, we develop a computational logic approach to spatial reasoning and show that the weak completion semantics computes preferred mental models.

## 1 Introduction

In the last century the classical (propositional) logic has played an important role as a normative concept for psychologists investigating human reasoning episodes. Psychological research, however, showed that humans systematically deviate from the classical logically correct answers.

Until now there are no widely accepted theories that express formal representations of human reasoning. Before we can formalize human behavior we need to understand why humans draw certain conclusions given some specific information. We address this issue by exploring current results from the literature. Conventional formal approaches such as classical logic are not appropriate for this purpose because they cannot deal with the elementary aspects that humans permanently need to reason with, for instance with incomplete information or non-monotonicity. Various alternatives where these properties hold have been proposed. However, most of them are purely theoretical and have never been applied to real case studies. Instead, for most of these theories, artificial examples have been constructed, which only show that this theory works within this very specific context. But what is the value of a theory for human reasoning that has never been tested on how humans actually reason?

Strube [32] claims that just modeling is not satisfying. He argues that knowledge engineering should also aim at being *cognitively adequate*. Accordingly, when evaluating computational approaches which try to explain human reasoning we insist on assessing their cognitive adequacy. Strube distinguishes between *weak* and *strong* cognitive adequacy: Weak cognitive adequacy requires the system to be ergonomic and user-friendly,

---

\* The authors are mentioned in alphabetical order.

whereas strong cognitive adequacy involves an exact model of human knowledge and reasoning mechanisms that follows the relevant human cognitive processes.

The concept of adequacy [31] originally has been defined in a linguistic context to compare and explain language theories and their properties, for which there are two different measures: *conceptual adequacy* and *inferential adequacy*. Conceptual adequacy reflects in how far the language represents the content correctly. Inferential adequacy is about the procedural part when the language is applied on the content.

Knauff, Rauh Schneider and Knauff, Rauh and Renz [19, 20] define cognitive adequacy in the setting of qualitative spatial reasoning, where they make a similar distinction: The degrees of conceptual adequacy reflects to which extent a system corresponds to human conceptual knowledge. Inferential adequacy focuses on the procedural part and indicates whether the reasoning process of a system is structured similarly to the way humans reason. This is analogous to the proposition made by Stenning and van Lambalgen [29, 30] to model human reasoning by a two step process: Firstly, human reasoning should be modeled by setting up an *appropriate representation* (conceptual adequacy) and, secondly, the *reasoning process* should be modeled with respect to this representation (inferential adequacy).

In Cognitive Science, it is common to evaluate theories about human reasoning processes by performing reasoning experiments on subjects directly. For instance, Knauff and Renz, Rauh and Knauff [18, 28] investigate which kind of information humans use when representing and remembering spatial arrangements in Allen's interval calculus [1].

However, a lot of theories in Cognitive Science are formulated in natural language and are not formalized. This might be a drawback because it leaves a lot of space for ambiguity and makes them incomparable to other ones. Most of them are only investigated within a particular setting and not general enough to be applied on other experimental results. From a Computer Science perspective, this seems to be a deficiency and especially unnecessary: If a problem is extensively studied and well-understood, then we should be able to formalize it. The formal approaches in Computer Science and the flexible techniques of Artificial Intelligence seem to offer the required ingredients for this issue.

In the following we will focus on human spatial reasoning, where we investigate and formalize how new knowledge is understood given some information about a certain arrangement of objects. For instance, the information, that a ferrari is left of a porsche and right of the porsche there is a beetle: We can without difficulty conclude that the ferrari is left of the beetle. But how exactly do we come to this conclusion? What happens if we have a set of premises with which more than one arrangement is possible? Consider the following example taken from Ragni and Knauff [26] which consists of four premises above the line and a query below the line.

1. The ferrari is left of the porsche.
  2. The beetle is right of the porsche.
  3. The porsche is left of the hummer.
  4. The hummer is left of the dodge.
- 
- Q.* The porsche is (necessarily) left of the dodge.

Would a human immediately notice that the first two premises are not relevant and that the query follows from the transitivity given the left relation of the third and the fourth premise?

The question which we shall be discussing in this paper is how to automatically construct what humans may have in mind while reading the premises. Next, accordingly, the query should be evaluated such as humans do. For instance, the mental model theory [16] assumes that humans construct one model, verifies the query and possibly constructs the next one. On the other hand, the preferred model theory [26] claims that humans only have exactly one simplified model in mind. This requires less effort than constructing all possible models and in most situations of our everyday life this one model is absolutely enough to reason with. According to Ragni and Knauff [26], people will only think of alternatives, if they are asked to search for other models. But even then, they do not randomly construct them from scratch. Instead of that, they first start generating models which are most similar to the preferred one by changing them as little as possible. This theory seems to be very promising and is empirically supported. Furthermore, it can give us an explanation about how people deal with ambiguity when modeling spatial reasoning problems and why, while evaluating a query, they might come to conclusions which are classical logically wrong.

We will formalize the preferred model theory in a computational logic setting based on the *weak completion semantics*. This approach has its origin in the work by Stenning and van Lambalgen [30]. Their work was based on the Kripke's and Kleene's three-valued logic [17] and contained a technical bug which was corrected in [12] by considering the three-valued Łukasiewicz logic [21] instead. [12, 13] also showed that each logic program as well as its weak completion admits a least model under Łukasiewicz logic, which can be computed by iterating Stenning and van Lambalgen's semantic operator [30]. Consequently, reasoning should be performed with respect to this least model.

Somewhat surprisingly, under the weak completion semantics we were able to adequately model the suppression [5, 14] as well as the selection task [6], the belief-bias effect [23], contextual abductive reasoning with side effects [24] and indicative conditionals [4]. Moreover, the approach can be implemented within an connectionist setting based on the core method [2, 11].

The goal of this paper is to show that human spatial reasoning can also be adequately modeled under the weak completion semantics. In order to do so, we first discuss spatial relations in human reasoning and, in particular, the preferred model theory in Section 2. Thereafter, we introduce the computational logic approach based on the weak completion semantics in Section 3. Finally, in Section 4 we show how the preferred model theory can be implemented under the weak completion semantics.

## 2 Spatial Relations in Human Reasoning

We will first address the spatial reasoning problem following the introduction of [26]. We assume binary spatial relations between two objects and restrict ourselves in this paper to the *is left of* and *is right of* relations. We use the notation  $leftOf(X, Y)$  and

$rightOf(X, Y)$  to express that  $X$  is left of  $Y$  and  $X$  is right of  $Y$ , respectively. Reconsidering the example from the introduction we obtain:

*Example 1.*

1.	$leftOf(ferrari, porsche)$
2.	$rightOf(beetle, porsche)$
3.	$leftOf(porsche, hummer)$
4.	$leftOf(hummer, dodge)$
$Q. leftOf(porsche, dodge)$	

Formally, a *spatial reasoning problem* consists of a finite list of *premises* and a *query*, and the question whether the premises entail the query. We assume that each premise is a ground atom of the form  $p(a, b)$  specifying some spatial relation  $p$  between the *objects*  $a$  and  $b$ . The query is also a ground atom of the form  $p(a, b)$ , where  $p$  is some spatial relation and the objects  $a$  and  $b$  must occur in the premises.

## 2.1 Inference Rule Approach

The *inference rule approach*, as presented by Byrne and Johnson-Laird [3], is based on following assumptions:

1. Humans know a set of inference rules, which they can apply on the premises of the spatial reasoning problem in order to derive new knowledge.
2. In case they encounter the query, it is proven. Only when all possibilities of applying the rules are exploited without proving the query, the query is refuted.
3. The order of the premises is not important.

In [3], a system with nine inference rules for spatial reasoning problems is specified, which does not only consider the *leftOf* and *rightOf* relations, but also the *frontOf* relation. As in this paper we will restrict ourselves to the *leftOf* and the *rightOf* relation, we will only consider a simplified version of the rule system. The inference rules are the following:

- (1)  $\forall X, Y, Z (leftOf(X, Z) \leftarrow leftOf(Z, Y) \wedge leftOf(Y, Z))$
- (2)  $\forall X, Y (leftOf(X, Y) \leftrightarrow rightOf(Y, X))$

Rule (1) represents transitivity of the relations within the objects and rule (2) the symmetry between the *leftOf* and *rightOf* relation. In this system, Example 1 can be solved by considering premises 3 and 4 together with rule (1) specifying the transitivity of *leftOf*, which allows to conclude the query. These rules seems appropriate and are necessary to identify the relations between the objects. However, the major drawback of this approach is that it does not consider the order in which premises are read.

## 2.2 Mental Model Theory

The *mental model theory* [16] does not assume that humans apply inference rules but that they construct so-called *mental models*. In spatial reasoning, a mental model is

understood as the presentation of the spatial arrangements between objects that correspond to the premises. Consider the following example, again taken from [26], which is similar to Example 1 except from the third premise:

*Example 2.*

1.	<i>leftOf(ferrari, porsche)</i>
2.	<i>rightOf(beetle, porsche)</i>
3.	<i>leftOf(beetle, hummer)</i>
4.	<i>leftOf(hummer, dodge)</i>
<i>Q. leftOf(porsche, dodge)</i>	

A mental model corresponding to the premises is constructed by writing the objects next to each other exactly how one would imagine the arrangement in the mind. In this case, there exists only one possible mental model:

*ferrari porsche beetle hummer dodge*

A spatial reasoning problem which has exactly one mental model is called a *deterministic* problem. On the other hand, Example 1 is a *non-deterministic* problem, for which we have three mental models:

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

The mental model theory assumes that when solving a spatial reasoning problem, humans behave as follows:

1. Any of the mental models is constructed.
2. If the query does not hold in this model, then it is refuted or rejected.
3. If the query holds in this model, then, one after another of the mental models will be constructed and verified. If the query holds in all these models, then it is proven.

In this approach, deterministic problems are easier to solve than non-deterministic ones as in this case Step 3. can be omitted.

In the original version of the mental model theory as well as in the inference rule approach it is assumed that the order of the premises is irrelevant. However, [25] already refers to the studies made by [22] which show that the order actually influences the construction of mental models and, in particular, the construction of the first mental model in Step 1.

### 2.3 Preferred Model Theory

The preferred model theory [26] is based on the mental model theory. The major difference is the assumption that humans do not consider all but only certain mental models. These certain ones in turn depend on the order in which the premises are presented. In

this section we will discuss this theory together with the computational system PRISM, developed by Ragni and Knauff [26] as well.

One should note that  $leftOf(a, b)$  or  $rightOf(a, b)$  denote that object  $a$  is left or right of object  $b$ , respectively. However, this does not mean that  $a$  is a neighbor of  $b$ . There might be other objects between  $a$  and  $b$ . In case  $a$  is a neighbor of  $b$ , i.e., there is no other object between them, then we will refer to this relation as *left neighbor of* or *right neighbor of* and denote this as  $ln(a, b)$  and  $ln(b, a)$ , respectively.

**Model Phases** The preferred model theory assumes that the phases in which humans solve spatial reasoning problems, can be divided as follows:

**Model construction** Based on the presented order of the premises, a particular preferred mental model is constructed.

**Model inspection** It is verified, whether the query holds in this model; in most cases, the (possibly wrong) answer for the whole problem is derived; only when the problem assignment explicitly suggests to search for other possible arrangements, the next phase is initiated.

**Model variation** Other mental models are considered; for this purpose, the mental models are stepwise modified; at first, alternative models which are as similar as possible to the preferred one, are considered. The similarity measure is computed by a neighborhood graph, also presented in [26].

**PRISM - Preferred Inferences in Reasoning with Spatial Mental Models** PRISM is an implementation of the preferred model theory. It only allow the following four types of premises for a spatial reasoning problem:

**Type 1** Exactly the first premise.

**Type 2** Premises containing exactly one new object and one which occurs already in the model constructed so far.

**Type 3** Premises, which contain two new objects, but which are not from type 1.

**Type 4** Premises, which relate two objects occurring in two different *submodels* to each other, where submodels are already constructed arrangements of objects which are not in relation to each other yet.

Based on a valid list of premises, PRISM constructs the preferred mental model by stepwise adding new objects to an initially empty arrangement. For this purpose, one premise after each other is read and, depending on its type, is processed as follows:

1. If the premise is of type 1 then the two objects will be placed directly next to each other.
2. If the premise is of type 2 then the new object will be inserted directly next to the already existing one provided that the space next to the existing one is free. If this space is already occupied then the new object is placed in the next available space; This procedure is called *first free fit* or *3f-strategy*.
3. If the premise is of type 3 then a new arrangement – i.e., a new submodel – is constructed in which both objects are arranged directly next to each other.
4. If the premise is of type 4, the first arrangement will be placed directly next to the second arrangement; the object within these arrangements do not change their places

*An Example* We will illustrate PRISM by reconsidering Example 1. Reading the premises one by one, the preferred mental model is constructed as follows: After reading the first premise,  $leftOf(ferrari, porsche)$ , which is of type 1, the ferrari and the porsche are placed next to each other as follows:

*ferrari porsche*

After reading the second premise,  $rightOf(beetle, porsche)$ , which is of type 2, the beetle is added next to the porsche as its right neighbor:

*ferrari porsche beetle*

After reading the third premise,  $leftOf(porsche, hummer)$ , which is again of type 2, we notice that the hummer cannot be placed directly right of the porsche because this space is already occupied. Therefore, the hummer will be placed on the next free space right of the porsche:

*ferrari porsche beetle hummer*

Finally, after reading the forth premise,  $leftOf(hummer, dodge)$ , the dodge is placed as right neighbor of the hummer and we obtain the preferred mental model:

*ferrari porsche beetle hummer dodge*

In the second phase, we check whether the query holds. As *porsche* is left of *dodge* in the above model, humans normally respond ‘yes’, if they are not explicitly pointed to consider other models. If this is the case, the third phase starts and we try to change the model with the least possible operations. In Example 1, the models

*ferrari porsche hummer beetle dodge*

and

*ferrari porsche hummer dodge beetle*

are generated subsequently. The query holds for all three models and, thus, indeed the classical logically correct answer is ‘yes’. However, most humans appear to infer their answer immediately after generating the preferred mental model.

*Submodels* Let us consider yet another example.

*Example 3.*

1.	$leftOf(ferrari, porsche)$
2.	$rightOf(beetle, hummer)$
3.	$leftOf(ferrari, beetle)$
$Q. leftOf(porsche, beetle)$	

After reading the first premise, the ferrari and the porsche are placed next to each other:

*ferrari porsche*



## VIII

The second premise is of type 3, that means, both objects are placed next to each other in a new empty order:

*beetle hummer*

Only after reading the third premise which is of type 4, both submodels are put in relation to each other and we obtain following preferred mental model:

*ferrari porsche beetle hummer*

Accordingly, the query will be answered with ‘yes’. Only in exceptional cases, humans try do some model variation, and figure out that there is also another model which agrees with the premises:

*ferrari beetle porsche hummer*

In this model, the query does not hold. However, the preferred model theory assumes that the majority of the people would answer with ‘yes’.

## 3 Logic Programs under the Weak Completion Semantics

We assume the reader to be familiar with logic and logic programming, but we repeat basic notions and notations in this section. A *(logic) program* is a finite set of (program) clauses of the form  $A \leftarrow B_1 \wedge \dots \wedge B_n$  where  $A$  is an atom and  $B_i$ ,  $1 \leq i \leq n$ , are literals or of the form  $\top$  and  $\perp$ , denoting truth- and falsehood, respectively.  $A$  is called *head* and  $B_1 \wedge \dots \wedge B_n$  is called *body* of the clause. We restrict terms to be constants and variables only, i.e., we consider so-called *data logic programs*. Clauses of the form  $A \leftarrow \top$  and  $A \leftarrow \perp$  are called *positive* and *negative facts*, respectively.

Let us clarify the notation defined until now by Example 1 from the previous section. The program, which contains all four premises is denoted as a set of positive facts about the *leftOf* or *rightOf* relation between the objects:

$$\mathcal{P}_{ex} = \{ \begin{array}{ll} \text{leftOf}(\text{ferrari}, \text{porsche}) & \leftarrow \top, \\ \text{leftOf}(\text{porsche}, \text{beetle}) & \leftarrow \top,^1 \\ \text{leftOf}(\text{porsche}, \text{hummer}) & \leftarrow \top, \\ \text{leftOf}(\text{hummer}, \text{dodge}) & \leftarrow \top, \\ \text{rightOf}(X, Y) & \leftarrow \text{leftOf}(Y, X) \end{array} \}$$

where the last clause is not a fact but simply a clause which represents the symmetry between *leftOf* and *rightOf* relations. Assume that the ferrari would actually not be left of the porsche. We can represent this as a negative fact:

$$\text{leftOf}(\text{ferrari}, \text{porsche}) \leftarrow \perp.$$

In this paper we assume for each program that the alphabet consists precisely of the symbols mentioned in the program. When writing sets of literals we will omit curly brackets if the set has only one element.

Let  $\mathcal{P}$  be a program.  $g\mathcal{P}$  denotes the set of all ground instances of clauses occurring in  $\mathcal{P}$ . As the set of constants, predicate symbols and variables is finite,  $\mathcal{P}$  is finite, and thus  $g\mathcal{P}$  is finite as well. We assume a fixed set of constants, denoted by  $\mathcal{C}$ , which is nonempty and finite. If  $\mathcal{P}$  is a program, then  $con(\mathcal{P})$  denotes the set of all constants occurring in  $\mathcal{P}$ . If not stated otherwise, we assume that  $\mathcal{C} = con(\mathcal{P})$ . Consider again our example:

$$con(\mathcal{P}_{ex}) = \{ferrari, porsche, beetle, hummer, dodge\}.$$

A ground atom  $A$  is *defined* in  $g\mathcal{P}$  if and only if (in the following abbreviated with iff)  $g\mathcal{P}$  contains a clause whose head is  $A$ ; otherwise  $A$  is said to be *undefined*. Let  $\mathcal{S}$  be a set of ground literals.

$$def(\mathcal{S}, \mathcal{P}) = \{A \leftarrow body \in g\mathcal{P} \mid A \in \mathcal{S} \vee \neg A \in \mathcal{S}\}$$

is called *definition* of  $\mathcal{S}$ .  $def(\mathcal{S}, \mathcal{P})$  for one of our facts in  $\mathcal{P}_{ex}$  is:

$$def(leftOf(ferrari, porsche), \mathcal{P}_{ex}) = \{leftOf(ferrari, porsche) \leftarrow \top\}$$

However,  $def(leftOf(porsche, porsche), \mathcal{P}_{ex})$  is empty.

Let  $\mathcal{P}$  be a program and consider the following transformation:

1. For each defined atom  $A$ , replace all clauses of the form  $A \leftarrow body_1, \dots, A \leftarrow body_m$  occurring in  $g\mathcal{P}$  by  $A \leftarrow body_1 \vee \dots \vee body_m$ .
2. If a ground atom  $A$  is undefined in  $g\mathcal{P}$ , then add  $A \leftarrow \perp$  to the program.
3. Replace all occurrences of  $\leftarrow$  by  $\leftrightarrow$ .

The ground program obtained by this transformation is called *completion* of  $\mathcal{P}$ , whereas the ground program obtained by applying only the steps 1. and 3. is called *weak completion* of  $\mathcal{P}$  or  $wc\mathcal{P}$ . The weak completion of the previously introduced program is:

$$\begin{aligned} wc\mathcal{P}_{ex} = \{ & leftOf(ferrari, porsche) \leftrightarrow \top, \\ & rightOf(beetle, porsche) \leftrightarrow \top, \\ & leftOf(porsche, hummer) \leftrightarrow \top, \\ & leftOf(hummer, dodge) \leftrightarrow \top \} \cup \\ & \{ rightOf(o_1, o_2) \leftrightarrow leftOf(o_2, o_1) \mid o_1, o_2 \in con(\mathcal{P}_{ex}) \}. \end{aligned}$$

We consider the three-valued Łukasiewicz (or Ł-) logic [21] (see Table 1) and represent each *interpretation*  $I$  by a pair  $\langle I^\top, I^\perp \rangle$ , where  $I^\top$  contains all atoms which are mapped to *true* by  $I$ ,  $I^\perp$  contains all atoms which are mapped to *false* by  $I$ , and  $I^\top \cap I^\perp = \emptyset$ .

<sup>1</sup> Strictly speaking, we should write  $rightOf(beetle, porsche) \leftarrow \top$  instead. However, for a simplified representation of the programs in the following, we will only allow *leftOf* relations as facts in the programs. Note that, because of the last clause, we can still conclude the corresponding *rightOf* relation.

$F \mid \neg F$	$\wedge \mid \top \text{ U } \perp$	$\vee \mid \top \text{ U } \perp$	$\leftarrow \mid \top \text{ U } \perp$	$\leftrightarrow \mid \top \text{ U } \perp$
$\top \mid \perp$	$\top \mid \top \text{ U } \perp$	$\top \mid \top \top \top$	$\top \mid \top \top \top$	$\top \mid \top \text{ U } \perp$
$\perp \mid \top$	$\text{U} \mid \text{U} \text{ U } \perp$	$\text{U} \mid \top \text{ U } \text{ U}$	$\text{U} \mid \text{U} \top \top$	$\text{U} \mid \text{U} \top \text{ U}$
$\text{U} \mid \text{U}$	$\perp \mid \perp \perp \perp$	$\perp \mid \top \text{ U } \perp$	$\perp \mid \perp \text{ U } \top$	$\perp \mid \perp \text{ U } \top$

**Table 1.** Truth tables for the  $\mathbb{L}$ -semantics, where we have used  $\top$ ,  $\perp$  and  $\text{U}$  instead of *true*, *false* and *unknown*, respectively, in order to shorten the presentation.

Atoms occurring neither in  $I^\top$  nor in  $I^\perp$  are mapped to *unknown*. Let  $I = \langle I^\top, I^\perp \rangle$  and  $J = \langle J^\top, J^\perp \rangle$  be two interpretations. We define

$$I \subseteq J \text{ iff } I^\top \subseteq J^\top \text{ and } I^\perp \subseteq J^\perp.$$

An interpretation  $I$  which maps a formula  $F$  to *true* under  $\mathbb{L}$ -logic, is written as  $I(F) = \text{true}$ . A *model* of  $\mathcal{P}$  is an interpretation which maps each clause occurring in  $\mathcal{P}$  to *true*.  $I$  is the *least model* of  $\mathcal{P}$  iff for any other model  $J$  of  $\mathcal{P}$  it holds that  $I \subseteq J$ .

Under  $\mathbb{L}$ -logic we find  $F \wedge \top \equiv F \vee \perp \equiv F$  for each formula  $F$ , where  $\equiv$  denotes semantic equivalence. Hence, occurrences of the symbols  $\top$  and  $\perp$  in the bodies of clauses can be restricted to those occurring in facts.

One should observe that in contrast to two-valued logic,  $A \leftarrow B$  and  $A \vee \neg B$  are not semantically equivalent under  $\mathbb{L}$ -logic. Consider, for instance, an interpretation  $I$  such that  $I(A) = I(B) = \text{unknown}$ . Then,  $I(A \vee \neg B) = \text{unknown}$  whereas  $I(A \leftarrow B) = \text{true}$ .

It has been shown in [12] that logic programs as well as their weak completions admit a least model under  $\mathbb{L}$ -logic. Moreover, the least  $\mathbb{L}$ -model of  $wc\mathcal{P}$  can be obtained as least fixed point of the following semantic operator, which is due to Stenning and van Lambalgen [30]:  $\Phi_{\mathcal{P}}(\langle I^\top, I^\perp \rangle) = \langle J^\top, J^\perp \rangle$ , where

$$\begin{aligned} J^\top &= \{A \mid A \leftarrow \text{body} \in g\mathcal{P} \text{ and } I(\text{body}) = \text{true}\}, \\ J^\perp &= \{A \mid \text{def}(A, \mathcal{P}) \neq \emptyset \text{ and} \\ &\quad I(\text{body}) = \text{false for all } A \leftarrow \text{body} \in \text{def}(A, \mathcal{P})\}. \end{aligned}$$

The *weak completion semantics* (WCS) is the approach to consider weakly completed logic programs and to reason with respect to the least  $\mathbb{L}$ -models of these programs. We write  $\mathcal{P} \models_{wcs} F$  iff formula  $F$  holds in the least  $\mathbb{L}$ -model of  $wc\mathcal{P}$ . As least  $\mathbb{L}$ -model of  $wc\mathcal{P}_{ex}$  we obtain  $\langle I^\top, \emptyset \rangle$ , where

$$\begin{aligned} I^\top &= \{ \text{leftOf}(\text{ferrari}, \text{porsche}), \text{leftOf}(\text{beetle}, \text{porsche}), \\ &\quad \text{leftOf}(\text{porsche}, \text{hummer}), \text{leftOf}(\text{hummer}, \text{dodge}), \\ &\quad \text{rightOf}(\text{porsche}, \text{ferrari}), \text{rightOf}(\text{porsche}, \text{beetle}), \\ &\quad \text{rightOf}(\text{hummer}, \text{porsche}), \text{rightOf}(\text{dodge}, \text{hummer}) \}. \end{aligned}$$

The  $\Phi$  operator differs from the semantic operator defined by Fitting in [8] in the additional condition  $\text{def}(A, \mathcal{P}) \neq \emptyset$  required in the definition of  $J^\perp$ . This condition

states that  $A$  must be defined in order to be mapped to *false*, whereas in the Kripke-Kleene-semantics considered by Fitting an atom may be mapped to *false* even if it is undefined in the underlying program. This reflects precisely the difference between the weak completion and the completion semantics, namely Step 2. of the program transformation. The Kripke-Kleene-semantics was also applied in [30]. However, as shown in [12] 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 episodes. Both, the technical bug as well as the non-adequate modeling, can be avoided by using WCS.

As shown in [7], WCS is related to the well-founded semantics (WFS) as follows: Let  $\mathcal{P}$  be a program which does not contain a positive loop and let

$$\mathcal{P}^+ = \mathcal{P} \setminus \{A \leftarrow \perp \mid A \leftarrow \perp \in \mathcal{P}\}.$$

Let  $u$  be a new nullary relation symbol not occurring in  $\mathcal{P}$  and  $B$  be a ground atom in

$$\mathcal{P}^* = \mathcal{P}^+ \cup \{B \leftarrow u \mid \text{def}(B, \mathcal{P}) = \emptyset\} \cup \{u \leftarrow \neg u\}.$$

Then, the least  $\mathbb{L}$ -model of  $wc\mathcal{P}$  and the well-founded model for  $\mathcal{P}^*$  coincide. The programs specified in [5] and in [6] to model the suppression and the selection task, respectively are acyclic and, thus, tight. Therefore, our results hold for both, WCS and WFS. The programs presented in the sequel of this paper are not tight. However, the positive cycles in our programs do not have any effect on the results as has been shown by Höps [15] and therefore they also hold for WFS.

## 4 Human Spatial Reasoning under WCS

Following the preferred model theory, we show how the preferred mental model of a spatial reasoning problem can be computed by logic programs under WCS. This approach covers the model construction and the model inspection phase.

### 4.1 Preferred Mental Models in Logic Programs

The running example in Section 3 shows us that relations between objects can be easily represented in logic programs. However, there is no straightforward way in which we can express the order in which the premises are read. But exactly this information is crucial if we want to formalize the preferred model theory. For this purpose, in the approach we will propose now, we explicitly express phases, where each premise is read at one particular phase. This allows us to define the order in which the premises are processed. In contrast to PRISM, we do not strictly separate between the model construction and model inspection phase but process them at the same time.

Let  $\mathcal{S}$  be a spatial reasoning problem. The program  $\mathcal{P}_{\mathcal{S}}$  represents the premises of  $\mathcal{S}$  and the necessary background knowledge in order to construct the preferred mental model.  $\mathcal{P}_{\mathcal{S}}$  is called a PPM-program with respect to the problem  $\mathcal{S}$ . Within  $\mathcal{P}_{\mathcal{S}}$  we will use the following notation with informal meaning as follows:

## XII

$l(X, Y, i)$	in phase $i$ , $X$ is placed to the left of $Y$ ,
$ln(X, Y, i)$	in phase $i$ , $X$ is the left neighbor of $Y$ ,
$ol(X, i)$	in phase $i$ , the space directly left of $X$ is occupied,
$or(X, i)$	in phase $i$ , the space directly right of $X$ is occupied,

where  $i$  starts with 1 and indicates the number of premises processed so far. Given a spatial reasoning problem  $\mathcal{S}$ , the corresponding program  $\mathcal{P}_{\mathcal{S}}$  is constructed as follows.

1. We start by reading the premises. For each premise do: If the  $i$ th premise is of the form  $leftOf(o_1, o_2)$  or  $rightOf(o_1, o_2)$ , then add

$$l(o_1, o_2, i) \leftarrow \top \quad \text{or} \quad l(o_2, o_1, i) \leftarrow \top, \quad (1)$$

respectively, to the (initially empty) program  $\mathcal{P}_{\mathcal{S}}$ , where  $o_1$  and  $o_2$  are assumed to be different objects. Let  $n$  be the number of premises.

2. We make a closed world assumption for the  $l$  relation in phase 1:

$$\{l(o_1, o_2, 1) \leftarrow \perp \mid o_1, o_2 \in con(\mathcal{P}_{\mathcal{S}}), diff(o_1, o_2)\}, \quad (2)$$

where  $diff$  specifies that its arguments are different objects. One should observe that programs are weakly completed. For example, if the first premise of a spatial reasoning problem is of the form  $leftOf(porsche, hummer)$  then the ground facts

$$l(porsche, hummer, 1) \leftarrow \top \quad \text{and} \quad l(porsche, hummer, 1) \leftarrow \perp$$

are generated in the first two steps, respectively. Their weak completion is

$$l(porsche, hummer, 1) \leftrightarrow \top \vee \perp \quad \equiv \quad l(porsche, hummer, 1) \leftrightarrow \top.$$

Under WCS, positive information overwrites negative information. In other words, there is no obligation to place  $o_1$  to the left of  $o_2$  in phase  $i$  unless explicitly stated in the  $i$ th premise.

3. As at the beginning no objects have been placed, the space to the left and to the right of each object is initially empty. This can be expressed by

$$\{ol(o, 1) \leftarrow \perp \mid o \in con(\mathcal{P}_{\mathcal{S}})\} \cup \{or(o, 1) \leftarrow \perp \mid o \in con(\mathcal{P}_{\mathcal{S}})\} \quad (3)$$

In our running example, we find that the space to the left and to the right of both cars, the porsche and the hummer, are empty in phase 1:

$$\{ ol(porsche, 1) \leftarrow \perp, \quad or(porsche, 1) \leftarrow \perp, \\ ol(hummer, 1) \leftarrow \perp, \quad or(hummer, 1) \leftarrow \perp \}.$$

4. We start to place objects. If in phase  $i$  object  $o_1$  should be placed to the left of object  $o_2$  and the space to the left of  $o_1$  as well as the space to the right of  $o_1$  are empty, then  $o_1$  is placed as the left neighbor of  $o_2$ :

$$\begin{aligned} \{ ln(o_1, o_2, i) \leftarrow l(o_1, o_2, i) \wedge \neg ol(o_2, i) \wedge \neg or(o_1, i) \\ \mid o_1, o_2 \in con(\mathcal{P}_{\mathcal{S}}), diff(o_1, o_2), i \in [1, n] \} \end{aligned} \quad (4)$$

In the running example we obtain for phase 1 the clause

$$\begin{aligned} ln(porsche, hummer, 1) \leftarrow & l(porsche, hummer, 1) \wedge \\ & \neg ol(hummer, 1) \wedge \neg or(porsche, 1) \end{aligned}$$

among others. Given the facts in (1), (2) and (3), the body of this clause will be true under WCS and, consequently, the porsche will be placed as the left neighbor of the hummer in phase 1.

5. Once an object  $o_1$  has become the left neighbor of another object  $o_2$  in phase  $i$ , this relation holds until the preferred mental model is constructed, i.e., for all phases  $j \in [i, n]$ .

$$\begin{aligned} \{ln(o_1, o_2, i + 1) \leftarrow & ln(o_1, o_2, i) \\ & | o_1, o_2 \in con(\mathcal{P}_S), diff(o_1, o_2), i \in [1, n - 1]\} \end{aligned} \quad (5)$$

6. If  $o_1$  has become the left neighbor of  $o_2$  in phase  $i$ , then the space to the left of  $o_2$  as well as the space to the right of  $o_1$  are occupied in phase  $i + 1$ .

$$\begin{aligned} \{ol(o_2, i + 1) \leftarrow & ln(o_1, o_2, i) \mid o_1, o_2 \in con(\mathcal{P}_S), diff(o_1, o_2), i \in [1, n - 1]\} \\ \cup \\ \{or(o_1, i + 1) \leftarrow & ln(o_1, o_2, i) \mid o_1, o_2 \in con(\mathcal{P}_S), diff(o_1, o_2), i \in [1, n - 1]\} \end{aligned} \quad (6)$$

In combination with (5) the space to the left of  $o_2$  and the space to the right of  $o_1$  are occupied in all future phases. For example, after the porsche has been placed as left neighbor of the hummer in phase 1, the clauses

$$ol(hummer, 2) \leftarrow ln(porsche, hummer, 1)$$

and

$$or(porsche, 2) \leftarrow ln(porsche, hummer, 1)$$

determine that there is no space anymore immediately to the left of the hummer and immediately to the right of the porsche at phase 2.

7. If  $o_1$  should be placed to the left of  $o_2$  but there is already a left neighbor  $o_3$  of  $o_2$ , then  $o_1$  should be placed to the left of  $o_3$ :

$$\begin{aligned} \{l(o_1, o_3, i + 1) \leftarrow & l(o_1, o_2, i + 1) \wedge ln(o_3, o_2, i) \\ & | o_1, o_2, o_3 \in con(\mathcal{P}_S), diff(o_1, o_2, o_3), i \in [1, n - 1]\} \end{aligned} \quad (7)$$

One should observe that this can only happen from phase 2 onwards, as in the first phase none of the objects has a left neighbor. This is the reason for writing  $i + 1$  in the atom  $l(o_1, o_2, i + 1)$  occurring in the bodies of the clauses in (7).

8. Likewise, if  $o_1$  should be placed to the left of  $o_2$  but  $o_1$  is already the left neighbor of some other object  $o_3$ , then  $o_3$  should be placed to the left of  $o_2$ :

$$\begin{aligned} \{l(o_3, o_2, i + 1) \leftarrow & l(o_1, o_2, i + 1) \wedge ln(o_1, o_3, i) \\ & | o_1, o_2, o_3 \in con(\mathcal{P}_S), diff(o_1, o_2, o_3), i \in [1, n - 1]\} \end{aligned} \quad (8)$$

9. Finally, in order to determine whether the query is true, we add the following clauses to  $\mathcal{P}_S$ . If  $o_1$  is the left neighbor of  $o_2$  after processing all premises, then  $o_1$  is to the left of  $o_2$  in the preferred mental model:

$$\{leftOf(o_1, o_2) \leftarrow ln(o_1, o_2, n) \mid o_1, o_2 \in con(\mathcal{P}_S), diff(o_1, o_2)\} \quad (9)$$

The *leftOf* relation is transitive:

$$\begin{aligned} leftOf(o_1, o_3) \leftarrow leftOf(o_1, o_2) \wedge leftOf(o_2, o_3) \\ \mid o_1, o_2, o_3 \in con(\mathcal{P}_S), diff(o_1, o_2, o_3) \end{aligned} \quad (10)$$

The *rightOf* relation is the inverse of the *leftOf* relation

$$\{rightOf(o_1, o_2) \leftarrow leftOf(o_2, o_1) \mid o_1, o_2 \in con(\mathcal{P}_S), diff(o_1, o_2)\} \quad (11)$$

In each phase, one premise is read and understood as a request to place the mentioned objects in the required order. Objects are placed in the first available space like in the PRISM approach, where again in each phase exactly one request to place objects is processed and the objects in the request are placed. Once the least fixed point of the weak completion of  $\mathcal{P}_S$  has been reached we can identify the preferred mental model: Given a problem  $S$ ,  $o_1$  is the left neighbor of  $o_2$  iff  $ln(o_1, o_2, n)$  holds in the least fixed point. Additionally, a queries involving the *leftOf* and *rightOf* relation can now be answered with respect to the preferred mental model of  $S$ . This will be illustrated by two examples in the next subsection.

## 4.2 Examples

*A Simple Example* We consider the following spatial reasoning problem:

$$\begin{array}{ll} \text{Example 4.} & \begin{array}{l} 1. \text{ leftOf(porsche, hummer)} \\ 2. \text{ leftOf(dodge, hummer)} \\ \hline Q. \text{ leftOf(dodge, porsche)} \end{array} \end{array}$$

Let  $\mathcal{P}_4$  be the logic program corresponding to Example 4 and  $\Phi_{\mathcal{P}_4}$  the corresponding semantic operator. To save space we abbreviate the constants representing cars by their first letter, i.e.,  $d$ ,  $h$  and  $p$  are abbreviations for *dodge*, *hummer* and *porsche*, respectively. In Table 2 we illustrate the computation of the least fixed point of  $\Phi_{\mathcal{P}_4}$  step by step, where  $\Phi \uparrow 0 = \langle \emptyset, \emptyset \rangle$  and  $\Phi \uparrow (i + 1) = \Phi(\Phi \uparrow i)$  for all  $i > 0$ . Focusing on atoms which are mapped to true, i.e., on  $I^\top$ , we find:

- In the first iteration of the  $\Phi_{\mathcal{P}_4}$  operator the requests to place the porsche to the left of the hummer in phase 1 and the dodge to the left of the hummer in phase 2 are recorded.
- In the second iteration of  $\Phi_{\mathcal{P}_4}$  the porsche becomes the left neighbor of the hummer in phase 1.

iteration	$I^\top$	$I^\perp$	reason
$\Phi_{\mathcal{P}_4} \uparrow 0$	$\emptyset$	$\emptyset$	
$\Phi_{\mathcal{P}_4} \uparrow 1$	$l(p, h, 1)$ $l(d, h, 2)$	$l(d, h, 1), l(d, p, 1), l(h, d, 1)$ $l(h, p, 1), l(p, d, 1)$ $ol(d, 1), ol(h, 1), ol(p, 1)$ $or(d, 1), or(h, 1), or(p, 1)$	(1) (1) (2) (2) (3) (3)
$\Phi_{\mathcal{P}_4} \uparrow 2$	$ln(p, h, 1)$	$ln(d, h, 1), ln(d, p, 1), ln(h, d, 1)$ $ln(h, p, 1), ln(p, d, 1)$	(4) (4)
$\Phi_{\mathcal{P}_4} \uparrow 3$	$ln(p, h, 2)$ $ol(h, 2)$ $or(p, 2)$ $l(d, p, 2)$	$ol(d, 2), ol(p, 2)$ $or(d, 2), or(h, 2)$ $l(h, p, 2), l(p, d, 2), l(p, h, 2), l(h, p, 2)$	(5) (6) (6) (7,8)
$\Phi_{\mathcal{P}_4} \uparrow 4$	$ln(d, p, 2)$ $leftOf(p, h)$	$ln(d, h, 2), ln(h, p, 2), ln(p, d, 2)$ $l(h, d, 2)$	(4) (7,8) (9)
$\Phi_{\mathcal{P}_4} \uparrow 5$	$leftOf(d, p)$ $rightOf(h, p)$	$ln(h, d, 2)$	(4) (9) (11)
$\Phi_{\mathcal{P}_4} \uparrow 6$	$leftOf(d, h)$ $rightOf(p, d)$		(10) (11)
$\Phi_{\mathcal{P}_4} \uparrow 7$	$rightOf(h, d)$		(11)

**Table 2.** The computation of the least fixed point of  $\Phi_{\mathcal{P}_4}$ , where in each iteration only atoms are listed which appear in  $I^\top$  and  $I^\perp$  for the first time.

- In the third iteration of  $\Phi_{\mathcal{P}_4}$  we learn that the space to the left of the hummer as well as the space to the right of the porsche are occupied in phase 2. As the porsche is the left neighbor of the hummer in phase 1, this relationship is preserved in phase 2 and the dodge must be placed to the left of the porsche in phase 2.
- In the fourth iteration of  $\Phi_{\mathcal{P}_4}$  the dodge becomes the left neighbor of the porsche in phase 2 and we find that the porsche and the hummer are in the *leftOf* relation.
- In the fifth iteration of  $\Phi_{\mathcal{P}_4}$  we find that the dodge and the porsche are in the *leftOf* relation, whereas the hummer and the porsche are in the *rightOf* relation.
- In the sixth iteration of  $\Phi_{\mathcal{P}_4}$  we find by transitivity that the dodge and the hummer are in the *leftOf* relation, whereas the porsche and the dodge are in the *rightOf* relation.
- Finally, in the seventh iteration of  $\Phi_{\mathcal{P}_4}$  we find that the hummer and the dodge are in the *rightOf* relation.

Query  $Q$  is answered positively; the dodge is to the left of the porsche.

*Submodels* We return to Example 3 from Section 2.3 which contains premises from type 3 and type 4, i.e., premises that generate submodels. Let  $\mathcal{P}_3$  be the logic program



iteration	$I^\top$	$I^\perp$	reason
$\Phi_{\mathcal{P}_3} \uparrow 0$	$\emptyset$	$\emptyset$	
$\Phi_{\mathcal{P}_3} \uparrow 1$	$l(f, p, 1)$ $l(h, b, 2)$ $l(f, b, 3)$	$l(b, f, 1), l(b, h, 1), l(b, p, 1), l(f, b, 1), l(f, h, 1), l(h, b, 1)$ $l(h, f, 1), l(h, p, 1), l(p, b, 1), l(p, f, 1), l(p, h, 1)$ $ol(b, 1), ol(f, 1), ol(h, 1), ol(p, 1)$ $or(b, 1), or(f, 1), or(h, 1), or(p, 1)$	(1) (1) (1) (2) (2) (3) (3)
$\Phi_{\mathcal{P}_3} \uparrow 2$	$ln(f, p, 1)$	$ln(b, f, 1), ln(b, h, 1), ln(b, p, 1), ln(f, b, 1)$ $ln(f, h, 1), ln(h, b, 1), ln(h, f, 1), ln(h, p, 1)$ $ln(p, b, 1), ln(p, f, 1), ln(p, h, 1)$	(4) (4) (4)
$\Phi_{\mathcal{P}_3} \uparrow 3$	$ln(f, p, 2)$ $ol(p, 2)$ $or(f, 2)$	$ol(b, 2), ol(f, 2), ol(h, 2)$ $or(b, 2), or(h, 2), or(p, 2)$ $l(b, h, 2), l(b, p, 2), l(f, b, 2), l(f, h, 2)$ $l(f, p, 2), l(h, p, 2), l(p, b, 2)$	(5) (6) (6) (7,8) (7,8)
$\Phi_{\mathcal{P}_3} \uparrow 4$	$ln(h, b, 2)$ $ln(f, p, 3)$ $ol(p, 3)$ $or(f, 3)$	$ln(b, h, 2), ln(b, p, 2), ln(f, b, 2), ln(f, h, 2)$ $ln(h, p, 2), ln(p, b, 2), ln(p, f, 2)$  $l(b, f, 2), l(h, f, 2), l(p, h, 2)$	(4) (4) (5) (6) (6) (7,8)
$\Phi_{\mathcal{P}_3} \uparrow 5$	$ln(h, b, 3)$  $ol(b, 3)$ $or(h, 3)$ $l(f, h, 3)$ $leftOf(f, p)$	$ln(b, p, 3), ln(f, b, 3), ln(f, h, 3)$ $ln(h, d, 2), ln(b, f, 2), ln(h, f, 2), ln(h, p, 3), ln(p, h, 2)$  $l(h, p, 3), l(p, f, 3)$	(5) (5) (6) (6) (7,8) (9)
$\Phi_{\mathcal{P}_3} \uparrow 6$	$l(p, h, 3)$  $leftOf(h, b)$ $rightOf(p, f)$	$ln(p, b, 3), ln(p, f, 3)$ $ol(f, 3), ol(h, 3), or(b, 3), or(p, 3)$ $l(b, h, 3), l(b, p, 3), l(f, p, 3),$ $l(h, b, 3), l(h, f, 3), l(p, b, 3), ln(h, f, 3)$	(5) (6) (7,8) (7,8) (9) (11)
$\Phi_{\mathcal{P}_3} \uparrow 7$	$ln(p, h, 3)$  $rightOf(b, h)$	$ln(b, h, 3)$ $l(b, f, 3)$	(4) (7,8) (11)
$\Phi_{\mathcal{P}_3} \uparrow 8$	$leftOf(p, h)$	$ln(b, f, 3)$	(5) (9)
$\Phi_{\mathcal{P}_3} \uparrow 9$	$leftOf(f, h)$ $leftOf(p, b)$ $rightOf(h, p)$		(10) (10) (11)
$\Phi_{\mathcal{P}_3} \uparrow 10$	$leftOf(f, b)$ $rightOf(b, p)$ $rightOf(h, f)$		(10) (11) (11)
$\Phi_{\mathcal{P}_3} \uparrow 11$	$rightOf(b, f)$		(11)

**Table 3.** The computation of the least fixed point of  $\Phi_{\mathcal{P}_3}$ , where in each iteration only atoms are listed which appear in  $I^\top$  and  $I^\perp$  for the first time.

corresponding to Example 3 and  $\Phi_{\mathcal{P}_3}$  be the corresponding semantic operator. Again to save space, we abbreviate the constants representing *beetle*, *hummer*, *ferrari* and *porsche*, by their first letter, i.e.,  $b$ ,  $h$ ,  $f$  and  $p$ , respectively. In Table 2 we depict the computation of the least fixed point of  $\Phi_{\mathcal{P}_4}$ . Focusing on  $I^\top$  we find:

- In the first iteration the three requests to place objects are recorded.
- In the second and the forth iteration, the ferrari becomes the left neighbor of the porsche and the hummer becomes the left neighbor of the beetle, respectively, thus generating two submodels which are not connected at this step.
- In the fifth and the sixth iteration the request to place the ferrari to the left of the beetle ( $l(f, b, 3)$ ) is processed. It is changed to  $l(f, h, 3)$  and, thereafter, to  $l(p, h, 3)$ .
- Consequently, the porsche becomes the left neighbor of the hummer in the seventh iteration leading to the preferred mental model.

Query  $Q$  is answered positively; the porsche is to the left of the beetle.

## 5 Conclusion

We have shown that our computational logic approach based on the weak completion semantics can compute preferred mental models for spatial reasoning problems. We have restricted our presentation to the *leftOf* and *rightOf* relation, but the formalization can be extended to include additional ones like the *frontOf* or the *backOf* relations. Likewise, we should be able to handle the four cardinal directions. Different than other approaches such as described in [10], the preferred model theory explains how a model is constructed and seems to be able to predict conclusions humans make given a spatial reasoning problem. This allows us to understand why classical logically wrong inferences can happen and how they influence the model construction.

We believe that our computational logic approach can also be extended to compute preferred mental models in more complex calculi for spatial reasoning like the RCC-8 calculus [27], but this needs to be thoroughly investigated in the future. From a practical point of view, this might be interesting when considering a great amount of objects and relations. Computing all possible models for a given problem is quite costly and possibly unnecessary, as one typical model might already give us enough information. As the preferred model theory shows us, the majority of models is not even taken into account by humans.

Höps has shown in [15] that although the logic programs in this paper are not tight anymore, the relationship between the weak completion and the well-founded semantics mentioned in Section 3 can be preserved and, hence, preferred mental models can also be computed within state-of-the-art reasoning systems based on answer set programming like CLINGO [9]. Thus, large scale applications seems to be feasible.

As shown in [11] the least fixed point of the  $\Phi$  operator can be computed in a connectionist setting based on the core method [2]. In other words, the core method allows to compute and to reason with respect to preferred mental models within a purely connectionist network.

It appears to be that the computational logic approach based on the weak completion semantics can cover a wide range of human reasoning episodes. As already mentioned in the introduction, it can handle the suppression as well as the selection task, the belief-bias effect, contextual abductive reasoning with side effects and indicative conditionals. With the results presented in this paper, these tasks can now be combined with reasoning over spatial relationships. We are unaware of any other logic based approach which covers such a wide range of human reasoning episodes.

An aspect that we haven't considered yet is the validation phase in the preferred model theory. This phase applies when someone is explicitly asked to search for alternative models. As already mentioned in the introduction, Ragni and Knauff [26] do not assume that these alternative models are chosen randomly. Instead, the models are constructed based on the least amount of effort that needs to be done to create a new model. This means, models which are the most similar to the preferred one, will be considered first. This similarity is based on a measure defined through a neighborhood graph which allows for minimal revision. Until now, we do not have formalized this procedure in WCS. This would imply an extension of the current approach which allows some kind of revision with respect to a given program.

## References

1. James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26:832–843, 1983.
2. Sebastian Bader and Steffen Hölldobler. The core method: Connectionist model generation. In Stefanos Kollias, Andreas Stafylopatis, Włodzisław Duch, and Erkki Oja, editors, *Artificial Neural Networks - ICANN 2006*, volume 4132 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2006.
3. R. Byrne and Johnson-Laird. Spatial reasoning. In *Journal of Memory and Language*, volume 28, pages 564 – 575, 1989.
4. 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/>.
5. Emmanuelle-Anna Dietz, Steffen Hölldobler, and Marco 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, CogSci 2013*, pages 1500–1505. Austin, TX: Cognitive Science Society, 2012.
6. Emmanuelle-Anna Dietz, Steffen Hölldobler, and Marco Ragni. A computational logic approach to the abstract and the social case of the selection task. In *Proceedings of the 11th International Symposium on Logical Formalizations of Commonsense Reasoning, COMMON-SENSE 2013*, Aeyia Nappa, Cyprus, 2013.
7. Emmanuelle-Anna Dietz, Steffen Hölldobler, and Christoph Wernhard. Modeling the suppression task under weak completion and well-founded semantics. *Journal of Applied Non-Classical Logics*, 2014.
8. Melvin Fitting. A Kripke-Kleene semantics for logic programs. *Journal of Logic Programming*, 2(4):295–312, 1985.
9. M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub. *Clingo = ASP + control*: Preliminary report. In M. Leuschel and T. Schrijvers, editors, *Technical Communications of the Thirtieth*

- International Conference on Logic Programming (ICLP'14)*, volume arXiv:1405.3694v1, 2014. Theory and Practice of Logic Programming, Online Supplement.
10. Geoffrey P Goodwin and PN Johnson-Laird. Reasoning about relations. *Psychological review*, 112(2):468, 2005.
  11. Steffen Hölldobler and Carroline Dewi Kencana Ramli. Logic programs under three-valued Łukasiewicz semantics. In Patricia M. Hill and David Scott Warren, editors, *Logic Programming, 25th International Conference, ICLP 2009*, volume 5649 of *Lecture Notes in Computer Science*, pages 464–478, Heidelberg, 2009. Springer.
  12. Steffen Hölldobler and Carroline Dewi Kencana Ramli. Logics and networks for human reasoning. In Cesare Alippi, Marios M. Polycarpou, Christos G. Panayiotou, and Georgios 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.
  13. Steffen Hölldobler and Carroline Dewi Puspa Kencana Ramli. Contraction properties of a semantic operator for human reasoning. In Lei Li and K. K. Yen, editors, *Proceedings of the Fifth International Conference on Information*, pages 228–231, 2009.
  14. Steffen Hölldobler, Tobias Philipp, and Christoph 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.
  15. Raphael Höps. Menschliches räumliches Schließen und Ansätze aus der Computational Logic. Bachelor's thesis, TU Dresden, Germany, 2014.
  16. P. N. Johnson-Laird. *Mental models: towards a cognitive science of language, inference, and consciousness*. Harvard University Press, Cambridge, MA, 1983.
  17. Stephen Cole Kleene. *Introduction to Metamathematics*. North-Holland, Amsterdam, 1952.
  18. Markus Knauff. The cognitive adequacy of allen's interval calculus for qualitative spatial representation and reasoning. *Spatial Cognition and Computation*, 1:261–290, October 1999.
  19. Markus Knauff, Reinhold Rauh, and Jochen Renz. A cognitive assessment of topological spatial relations: Results from an empirical investigation. In *Proceedings of the 3rd International Conference on Spatial Information Theory, COSIT'97*, volume 1329 of *Lecture Notes in Computer Science*, pages 193–206, Heidelberg, 1997. Springer.
  20. Markus Knauff, Reinhold Rauh, and Christoph Schlieder. Preferred mental models in qualitative spatial reasoning: A cognitive assessment of Allen's calculus. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society, CogSci 1995*, pages 200–205, Hillsdale, NJ, 1995. Lawrence Erlbaum.
  21. Jan 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.
  22. K. Manktelow. *Reasoning and Thinking*. Psychology press, 2000.
  23. Luís Moniz Pereira, Emmanuelle-Anna Dietz, and Steffen Hölldobler. A computational logic approach to the belief bias effect. In *14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, 2014. short paper.
  24. Luís Moniz Pereira, Emmanuelle-Anna Dietz, and Steffen Hölldobler. Contextual abductive reasoning with side-effects. *Journal of Theory and Practice of Logic Programming (TPLP)*, 14(4-5):633–648, 2014.
  25. M. Ragni, M. Knauff, and B. Nebel. A computational model for spatial reasoning with mental models. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, pages 1797–1802, 2005.
  26. Marco Ragni and Markus Knauff. A theory and a computational model of spatial reasoning with preferred mental models. In *Psychological Review 2013*, volume 120, pages 561 – 588, 2013.

27. David A. Randell, Zhan Cui, and Anthony Cohn. A spatial logic based on regions and connection. In Bernhard Nebel, Charles Rich, and William Swartout, editors, *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, pages 165–176. Morgan Kaufmann, San Mateo, California, 1992.
28. Jochen Renz, Reinhold Rauh, and Markus Knauff. Towards cognitive adequacy of topological spatial relations. In *Spatial Cognition II, Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications*, pages 184–197, London, UK, 2000. Springer-Verlag.
29. Keith Stenning and Michiel van Lambalgen. Semantic interpretation as computation in non-monotonic logic: The real meaning of the suppression task. *Cognitive Science*, 6(29):916–960, 2005.
30. Keith Stenning and Michiel van Lambalgen. *Human Reasoning and Cognitive Science*. A Bradford Book. MIT Press, Cambridge, MA, 2008.
31. G. Strube. *Wörterbuch der Kognitionswissenschaft*. Klett-Cotta, Stuttgart, 1996.
32. Gerhard Strube. The role of cognitive science in knowledge engineering. In Franz Schmalhofer, Gerhard Strube, and Thomas Wetter, editors, *Contemporary Knowledge Engineering and Cognition, First Joint Workshop*, volume 622 of *Lecture Notes in Computer Science*, pages 159–174. Springer, Heidelberg, 1992.