A Computational Theory for the Model Construction, Inspection and Variation Phase in Human Spatial Reasoning

Julia Mertesdorf[1] and **Emmanuelle-Anna Dietz Saldanha**[2] and **Steffen Hölldobler**[2,3] and **Marco Ragni**[1]
[1]Cognitive Computation Lab, Technische Fakultät, Universität Freiburg, 79110 Freiburg, Germany
[2] International Center for Computational Logic, TU Dresden, 01187 Dresden, Germany
[3] North-Caucasus Federal University, Stavropol, Russian Federation

Abstract

Our long-term research goal is the development of a cognitive theory for adequately modeling human reasoning tasks. The theory should be *computational* and on the other hand *comprehensive*. The Weak Completion Semantics (WCS) seems to be a good candidate, as it has previously shown to adequately model a wide range of human reasoning tasks. By means of human spatial reasoning, we show here that the WCS can fully cover all three stages of reasoning that have been suggested by the preferred mental model theory. The contribution comprises aspects within the area of Computer Science and Psychology. Through the formal process of modeling, in particular through the computation of alternative models within the variation phase, we have gained new insights and put forward assumptions that need to be verified.

Keywords: Computational Theory, Spatial Reasoning, Preferred Mental Model Theory, Weak Completion Semantics

Introduction

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

Currently the *Weak Completion Semantics (WCS)* is a very good, if not the best candidate for such an comprehensive and computational cognitive theory. The WCS is based on ideas initially proposed by Stenning and van Lambalgen (2005, 2008), but is mathematically sound: As Hölldobler and Kencana Ramli (2009) have shown, the three-valued logic used in Stenning and van Lambalgen (2008) is inadequate for the suppression task. Surprisingly, the suppression task can be adequately modeled if the three-valued Łukasiewicz (1920) logic is used. Since then, the WCS has been applied to various human reasoning tasks (cf. Wason, 1968; Byrne, 1989) summarized in Hölldobler (2015), has outperformed twelve cognitive theories considered by Khemlani and Johnson-Laird (2012) in syllogistic reasoning (Oliviera da Costa, Dietz Saldanha, Hölldobler, & Ragni, 2017), and can be implemented as a neural network (Dietz Saldanha, Hölldobler, Kencana Ramli, & Palacios Medinacelli, 2018).

Given a human reasoning task, the first step within the WCS is to construct a logic program representing the task. The construction of these programs is based on several principles, some of which are well-established like using *licenses for inferences*, *existential import* (Johnson-Laird, 1983; Rips, 1994; Stenning & van Lambalgen, 2008), or *Gricean implicature* (Grice, 1975), whereas others are novel like *unknown generalization* (Oliviera da Costa et al., 2017). If interpreted under the three-valued logic of Łukasiewicz (1920), the programs have a unique supported model, which can be computed by iterating the semantic operator introduced by Stenning and van Lambalgen (2008). Reasoning is performed and answers are computed with respect to these models. Skeptical abduction is applied if some observations in the given human reasoning task can not be explained otherwise.

Human Spatial Reasoning

In this paper we apply the WCS to spatial reasoning. Suppose you were given (in this sequence) the following information:

The Audi is left of the Beetle.
The Audi is left of the Cadillac.
The Cadillac is left of the Dodge.

Given these premises, what, if anything, follows for the Beetle and the Dodge? A psychological finding by Ragni and Knauff (2013) is that many human reasoners do construct the preferred (mental) model $a \quad b \quad c \quad d$.[1]

Based on the spatial representation of this preferred model, a reasoner could infer that *the Beetle is to the left of the Dodge*. In fact, most human reasoners seem to do this. Yet, the preferred model is not the only model for the given premises. If a reasoner would construct these alternative models he/she may find a counter-example and may answer that *nothing follows.* Under First-order Logic, there might be more than one model for the given premises of the task, letting unspecified which one to choose as the preferred one. Ragni and Knauff (2013) presented an algorithmic approach – the preferred mental model theory – to construct and manipulate mental models. Based on this theory, Dietz, Hölldobler, and Höps (2015) modeled the preferred models for human spatial reasoning in the WCS. They represented the relations among objects, transitivity properties, and the first-free-fit-principle suggested by Ragni and Knauff (2013) as logic programs and showed that the supported model in the WCS corresponds to the preferred mental model.

However, the approach of Dietz et al. (2015) is restricted to computing and reasoning with respect to the preferred mental model and does not cover the inspection and variation phase

[1] a denotes Audi, b Beetle, c Cadillac, and d Dodge.

reported by Ragni and Knauff (2013). More generally, it does not cover the flesh-out process after the initial mental model has been constructed. The goal of this paper is to show that the whole process – construction of an initial mental model, inspection, and variation – can be modeled by the WCS in the context of spatial reasoning.

Programs

Here, we consider programs similar to the ones introduced by Dietz Saldanha, Hölldobler, and Pereira (2017). A *(contextual logic) program* is a finite set of (positive) *facts* of the form $A \leftarrow \top$, (negative) *assumptions* of the form $A \leftarrow \bot$ and *rules* of the form $A \leftarrow L_1 \wedge \cdots \wedge L_m \wedge (\neg)\mathsf{ctxt}\, L_{m+1} \wedge \cdots \wedge (\neg)\mathsf{ctxt}\, L_{m+p}$, where A is an atom, L_i, $1 \leq i \leq m+p$, are literals (i.e. L_i is an atom or a negated atom), $\top$ denotes truth, $\bot$ denotes falsehood, and ctxt is a unary *context* operator. The interpretation of the connectives is given in Table 1.

The ctxt operator is similar to *negation as failure* (Clark, 1978) or *default negation* locally, and helps to provide a natural formalization of defeasible rules. To explain its behavior let us return to the spatial reasoning problem in the introduction. After reading the first premise, most participants seem to assume that the space right of the Beetle is not occupied. However, (classical) logically, it can neither be proven that the space is occupied, nor that it is not. Here, the application of ctxt allows us to conclude that the space is not occupied.

The example from the introduction can be represented by the facts $\mathit{left}(a,b) \leftarrow \top$, $\mathit{left}(a,c) \leftarrow \top$, $\mathit{left}(c,d) \leftarrow \top$. In addition, the rule $\mathit{right}(X,Y) \leftarrow \mathit{left}(Y,X)$ denotes the symmetry of left and right. Such a rule is considered to be a schema. Ground instances of this rule are obtained by replacing the variables occurring in it by the constants occurring in the program. In this example, these are a, b, c, and d. Let $\mathcal{P}$ be a program. $\mathsf{g}\,\mathcal{P}$ denotes the set of ground instances of clauses occurring in $\mathcal{P}$.

Computation of Supported Models

The connectives in Table 1 can be read as *not* ($\neg$), *and* ($\wedge$), *or* ($\vee$), *if* ($\leftarrow$), *only if* ($\leftrightarrow$) and *not, if not true* (ctxt). It remains to specify the meaning of ground atoms. A ground atom A may be true ($\top$), false ($\bot$), or unknown ($\cup$). An interpretation I can be represented by a pair $\langle I^\top, I^\bot \rangle$, where $I^\top = \{A \mid I(A) = \top\}$ and $I^\bot = \{A \mid I(A) = \bot\}$. As interpretations are mappings, $I^\top$ and $I^\bot$ must be disjoint. Ground atoms which do not occur in $I^\top \cup I^\bot$ are mapped to $\cup$. I is a *model* for a program $\mathcal{P}$ if and only if I maps all ground instances of clauses occurring in $\mathcal{P}$ to true.

Under the WCS a program $\mathcal{P}$ may admit a unique supported model which can be computed by iterating the semantic operator $\Phi_\mathcal{P}$ on the space of interpretations provided by Stenning and van Lambalgen (2008). Let I be an interpretation, then $\Phi_\mathcal{P}(I) = \langle J^\top, J^\bot \rangle$, where

$$J^\top = \{A \mid \text{there is } A \leftarrow \mathit{body} \in \mathsf{g}\,\mathcal{P} \text{ such that } I(\mathit{body}) = \top\},$$
$$J^\bot = \{A \mid \text{there is } A \leftarrow \mathit{body} \in \mathsf{g}\,\mathcal{P} \text{ and for all } A \leftarrow \mathit{body} \in \mathsf{g}\,\mathcal{P}, \text{ we find } I(\mathit{body}) = \bot\}.$$

Under certain conditions $\Phi_\mathcal{P}$ has a unique fixed point which can be computed by iterating the operator starting with an arbitrary interpretation.[2] In this case, this fixed point is the supported model of the weak completion of the given program $\mathcal{P}$. For example, considering the program presented in the previous section and starting with the empty interpretation $\langle \emptyset, \emptyset \rangle$ the fixed point $\langle I^\top, \emptyset \rangle$ is reached after two iterations, where

$$\begin{aligned} I^\top &= \{\mathit{left}(a,b),\ \mathit{left}(a,c),\ \mathit{left}(c,d)\} \\ &\cup \{\mathit{right}(b,a),\ \mathit{right}(c,a),\ \mathit{right}(d,c)\}. \end{aligned}$$

All instances of *left* are added in the first iteration, whereas all instances of *right* are added in the second iteration.

A formula F *follows from* $\mathcal{P}$ *under the WCS* ($\mathcal{P} \models_{wcs} F$) if and only if the supported model of $\mathcal{P}$ maps F to true.

Construction/ Inspection for Preferred Models

Relations between objects can be easily represented in programs. However, there is no straightforward way in which we can express the order in which the premises are given. But exactly this information is crucial if we want to formalize the preferred mental model theory. For this purpose, we explicitly express phases, where each premise is read at one particular phase.

Let $\mathcal{S}$ be a spatial reasoning problem consisting of a finite sequence of premises and a conclusion. 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. Within $\mathcal{P}_\mathcal{S}$ we will use the following relations, whose informal meanings are as follows:

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

where $I \in [1,n]$, n is the number of premises, and X and Y are objects. The construction of the program $\mathcal{P}_\mathcal{S}$ is initialized by specifying all premises of $\mathcal{S}$ as facts of the form

$$l(u,v,i) \leftarrow \top, \tag{1}$$

given that the i-th premise of $\mathcal{S}$ was *object u is left of object v*. Thereafter, the following rules are added:[3]

$$nl(X,Y,I) \leftarrow \mathsf{ctxt}\, l(X,Y,I) \wedge \overline{\mathsf{ctxt}}\, ol(Y,I) \wedge \overline{\mathsf{ctxt}}\, or(X,I). \tag{2}$$
$$nl(X,Y,J+1) \leftarrow nl(X,Y,J). \tag{3}$$
$$ol(Y,J+1) \leftarrow nl(X,Y,J). \tag{4}$$
$$or(X,J+1) \leftarrow nl(X,Y,J).$$
$$l(X,Z,J+1) \leftarrow l(X,Y,J+1) \wedge nl(Z,Y,J). \tag{5}$$
$$l(Z,Y,J+1) \leftarrow l(X,Y,J+1) \wedge nl(X,Z,J).$$
$$\mathit{left}(X,Y) \leftarrow nl(X,Y,n). \tag{6}$$
$$\mathit{left}(X,Z) \leftarrow \mathit{left}(X,Y) \wedge \mathit{left}(Y,Z). \tag{7}$$
$$\mathit{right}(X,Y) \leftarrow \mathit{left}(Y,X). \tag{8}$$

[2] See, Dietz Saldanha et al. (2017) for details. For each program $\mathcal{P}$ presented in this paper $\Phi_\mathcal{P}$ has a unique fixed point.

[3] Here and in the sequel, $\overline{\mathsf{ctxt}}$ is used as abbreviation for $\neg\mathsf{ctxt}$.

Table 1: Three-valued Łukasiewicz logic with ctxt. F is a formula, L a literal, and $\top$, $\bot$, and $\cup$ denote *true*, *false*, and *unknown*, respectively.

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

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

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

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

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

L	ctxt$L$
$\top$	$\top$
$\bot$	$\bot$
$\cup$	$\bot$

These rules are schemas and need to be instantiated such that $I \in [1,n]$, $J \in [1,n-1]$, and X,Y,Z are different constants denoting the objects occurring in the premises of $\mathcal{S}$. We assume that the addition $J+1$ is computed while instantiating a rule. The rule in (2) states that if in phase I object X should be placed to the left of Y and the space to the left of X as well as the space to the right of X are empty, then X is placed as the left neighbor of Y. The rule in (3) keeps neighbors for succeeding phases. The rules in (4) ensure that neighbors take space, i.e., if X has become the left neighbor of Y in phase J, then the space to the left of Y as well as the space to the right of X are occupied in phase $J+1$. The rules in (5) implement the first free fit technique from (Ragni & Knauff, 2013), thus if X should be placed to the left of Y but there is already a left neighbor Z of Y, then X is placed to the left of Z. Likewise, if X should be placed to the left of Y but X is already the left neighbor of some other object Z, then Z should be placed to the left of Y. The final neighbors are derived by the rule in (6): If X is left neighbor of Y after processing all premises, then X is (finally) to the left of Y. The rules in (7) and (8) express that *left* is transitive and *right* is the inverse of *left*.

In each phase, one premise is processed and understood as a request to place the mentioned objects in the required order. Objects are placed in the first available space like in PRISM (see, Ragni & Knauff, 2013). Once the fixed point of $\Phi_{\mathcal{P}_\mathcal{S}}$ is computed the preferred model can be identified: Given a problem $\mathcal{S}$, X is the left neighbor of Y if and only if it holds that $\mathcal{P}_\mathcal{S} \models_{wcs} nl(X,Y,n)$. Queries involving the *left* and *right* relation can be answered with respect to the preferred model of $\mathcal{S}$.

Variation/ Inspection for Alternative Models

We now present the main result of this paper, viz. an approach to the model variation phase. Figure 1 shows the modeling process of the variation phase, which consists of several steps: First, all initial left placement requests (*il*), all positive neighborhood left relations (*nl*), and all positive *ambiguous* relations from the preferred model are extracted (*Extract relevant information from preferred mental model*). Thereafter, the program is constructed (*Create program*). Based on all extracted ambiguities, all permutations of all length are computed (*Compute all permutations*). The order of the items in each permutation is kept by the variation program through the phase-indices in the relations: The first item in a permutation is assigned the phase-index 1 and the last item the phase-index v. All different ways of swapping ambiguous objects are simulated. Considering all permutations, all alternative models of the spatial reasoning problem $\mathcal{S}$ can be found. Until all permutation have been processed, the following is done (*Iteration through list of permutations*): One trial of swapping objects is done per iteration, by swapping objects through *ambiguous relations* (*Program construction*). This is realized by adding the *ambiguous* relations of the current permutation as positive facts to the program (see (14) below). Moreover, the program needs to know which objects are affected by these swap-requests. Therefore, the implementation adds two positive facts for each added *ambiguous* fact, encoding that the objects in question need to be adjusted (see (13) below). Thereafter, it is checked whether the relation encoding the conclusion of the spatial reasoning problem is mapped to true or false in the fixed point of the semantic operator (*Compute fixed point*). The answer is saved, provided that the model is not marked as abnormal (see (21) and (22)). The program is reset, which includes deleting all facts regarding the *ambiguous* and *adjust* relations and resetting all atoms occurring in the program to unknown (*Reset program*). The iteration continues until all permutations have been processed. The final answer is given by checking whether the list of collected answers contains the answer "No" (*Final answer*). If that is the case, the final answer to the query is "No" because an alternative model has been found that does not support the conclusion. Otherwise, the final answer is "Yes".

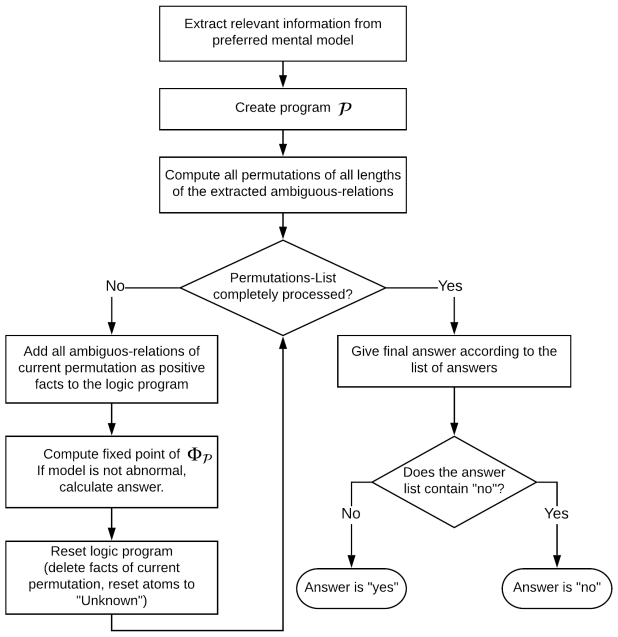

Figure 1: Flowchart of the variation phase.

Ambiguity Identification

We first record the set of initial placements of the spatial reasoning problem by replacing each fact of the form (1) by

$$il(X,Y) \quad \leftarrow \quad \top. \tag{9}$$

We extend $\mathcal{P}_S$ to mark ambiguities in the model construction:

$$amb(Z,X,J+1) \leftarrow l(Z,Y,J+1) \wedge nl(X,Y,J). \tag{10}$$
$$amb(Z,Y,J+1) \leftarrow l(X,Z,J+1) \wedge nl(X,Y,J).$$
$$amb(Z,X,I) \leftarrow l(Z,Y,I) \wedge amb(X,Y,J+1). \tag{11}$$
$$amb(Z,Y,I) \leftarrow l(X,Z,I) \wedge amb(X,Y,J+1).$$

X, Y, Z, I, and J must be instantiated as before and we assume that $I > J$. Let $\mathcal{A}_S$ be the program consisting of all ground instances of clauses mentioned in this paragraph.

The rules in (10) record the ambiguities from neighbors: if object X is the direct left neighbor of object Y in phase I and there is a request to place a new object Z to the left of Y, then there is an ambiguity between Z and X, because both could possibly be the direct left neighbor of Y. Similarly, if X should be placed to the left of the new object Z, but X and Y are already in a direct left neighborhood, then Z and Y are ambiguous and could be swapped in order to obtain an alternative model. The rules in (11) record the inherited ambiguities: If a new object Z is requested to be set to the left of Y, but Y is already marked as ambiguous with respect to another object X, then Z is ambiguous with respect to X, too. Likewise, in case X is requested to be placed to the left of a new object Z with X and Y already being marked as ambiguous objects, then Z will also be ambiguous with respect to Y. It is important to note that these clauses need to be created for all phases I and J with $I > J$. This means that the *amb*-relation with phase index J does not necessarily need to be in the phase directly before I, but it can also be that $I = J + 2$.

Program Construction for Alternative Models

Likewise to the construction of the program for the preferred model, the construction of the programs during variation uses phases as well. The neighbor left relations that have been generated by the preferred model will be used as starting point:

$$\{nl(X,Y,n) \leftarrow \top \mid \mathcal{P}_S \models_{wcs} nl(X,Y,n)\}. \tag{12}$$

First, the number of programs for the computation of alternative models (i.e. one program for one alternative model) is specified by the number of *amb* relations in the fixed point of $\Phi_{\mathcal{A}_S \cup \mathcal{P}_S}$:

$$\#\mathsf{perm} = \sum_{i=1}^{|amb|} \prod_{k=1}^{i} (|amb| - k + 1),$$

where

$$|amb| = |\{amb(X,Y,I) \mid \mathcal{A}_S \cup \mathcal{P}_S \models_{wcs} amb(X,Y,I), I \in [2,n]\}|.$$

The amount of phases v for each program depends on the number of *ambiguous*-relations with respect to the current permutation *pm*, i.e.

$$v_{pm} = |\{amb(X,Y,I) \mid amb(X,Y,I) \in pm, I \in [1,|pm|]\}.$$

Second, the *amb* relations of the current permutation *pm* tells us which objects can be adjusted within the variation phase:

$$\{adj(X,I) \leftarrow \top \mid amb(X,Y,I) \in pm, I \in [1,|pm|]\} \cup \tag{13}$$
$$\{adj(Y,I) \leftarrow \top \mid amb(X,Y,I) \in pm, I \in [1,|pm|]\}.$$

In each phase I of the variation, two objects X, Y are swapped according to a swap-request of the form $amb(X,Y,I) \leftarrow \top$. Accordingly, the maximum phase index v refers to the number of phases in the variation process and the phase index n refers to the number of phases in the construction process of the preferred model. Since the model variation starts with the preferred model, that is, the last phase n of the preferred model, the last overall phase in the variation program is $n+v$. Together with the fact in (9), which will serve as constraint to prevent violating the premises of the given spatial reasoning problem, the set of neighbor relations in (12) of the preferred model, and the objects in (13) that have to be adjusted, each program $\mathsf{var}\mathcal{P}_S$ with its according considered permutation *pm*, where $I \in [1,|pm|]$, consists of the following clauses:

$$amb(X,Y,I) \leftarrow \top. \tag{14}$$
$$amb(Y,X,I) \leftarrow amb(X,Y,I). \tag{15}$$
$$left(X,Y) \leftarrow nl(X,Y,n+v). \tag{16}$$
$$left(X,Z) \leftarrow \mathsf{ctxt}\, left(X,Y) \wedge \mathsf{ctxt}\, left(Y,Z). \tag{17}$$
$$right(X,Y) \leftarrow left(Y,X). \tag{18}$$
$$nl(X,Y,Q) \leftarrow nl(X,Y,P) \wedge \overline{\mathsf{ctxt}}\, adj(X,I) \wedge \overline{\mathsf{ctxt}}\, adj(Y,I). \tag{19}$$
$$nl(X,Y,Q) \leftarrow \mathsf{ctxt}\, amb(Y,X,I) \tag{20}$$
$$\wedge\ nl(Y,X,P) \wedge \overline{\mathsf{ctxt}}\, il(Y,X).$$
$$nl(X,Y,Q) \leftarrow \mathsf{ctxt}\, amb(Z,X,I) \wedge nl(Z,Y,P) \wedge \overline{\mathsf{ctxt}}\, il(Y,X).$$
$$nl(X,Y,Q) \leftarrow \mathsf{ctxt}\, amb(Y,Z,I) \wedge nl(X,Z,P) \wedge \overline{\mathsf{ctxt}}\, il(Y,X).$$

X, Y, Z, I, and J must be instantiated as before, n is the number of premises, v is the last phase in the variation model construction, $P = n - 1 + I$, and $Q = n + I$.

The fact in (14) ensures that all *amb*-relations from the current permutation are added as facts. The permutation order within a permutation *pm* is specified by the phase index I, starting from 1 to $|pm|$. These facts are requests to swap two objects in phase I. The rule in (15) expresses that the order of objects in an *ambiguous*-relation is irrelevant, as all orders lead to the same result. The rule in (16) specifies final neighbors. The rules in (17-18) are analogous to (7-8). The rule in (19) is similar to (3) except to the additional constraint that none of the concerned objects has to be adjusted in the current phase I. The rules in (20) encode the actual swapping of two objects. Altogether, there are three different cases how two objects can be swapped: Either the objects to be swapped are in the same *nl*-relation, or the left object in the *nl*-relation has to be swapped, or the right object in the *nl*-relation is requested to be swapped.

Incomplete Model or Constraint Violation

As the models constructed in the variation may be incomplete due to violated constraints, we include *abnormality* clauses in order to consider only *normal* models for computing answers with respect to the given problem query:

$$chain \leftarrow left(X_1,X_2) \wedge left(X_3,X_4) \wedge \cdots \wedge left(X_{n-1},X_n). \quad (21)$$

$$ab \leftarrow \neg chain.$$

$$ab \leftarrow left(X,Y) \wedge \mathsf{ctxt}(il(Y,X)). \quad (22)$$

$X,Y,X_1,\ldots,X_n \in con(\mathsf{init}\Pi\mathsf{rem})$ and $X,Y,X_1,\ldots,X_n$ are different to each other. The rules in (21) denote the case when the alternative model is not complete: If no chain can be constructed from the *left*-relations, then this model is marked as abnormal. The rule in (22) denotes the case when the alternative model violates some constraint. The case of constraint violation only concerns big, non-deterministic problems with five or more objects, for which the constraints contained in the bodies of the rules in (20) cannot prevent some of the violating swaps anymore.

In each phase of the model variation, two objects are swapped according to the swap-requests (by $amb(X,Y,i)$), until all requests in the current permutation have been processed. Likewise to the preferred model construction, the variation program will then proceed with mapping the *nl*-relations to *left*- and *right*-relations by the rules in (16), (17) and (18). After all *left*- and *right*-relations are determined, we can check in the alternative model whether there are any abnormalities. As soon as the fixed point of Φ with respect to the given program is computed, the alternative model can be identified, provided that the model is not abnormal, i.e. the atom *ab* is false.

Is the beetle (necessarily) left of the dodge?

Consider again the example from the introduction, where the preferred model is $a \quad b \quad c \quad d$.

This example has additionally two valid alternative models. Due to the limited space, we do not show the complete computation of the preferred mental model with marking ambiguities. The result of the computation are two marked ambiguities, one between the objects *c* and *b* and one between *d* and *b*. For a detailed explanation on what happens in each iteration when computing preferred models under the WCS, see the examples in Dietz et al. (2015).

The implementation determines four different permutations of the two *ambiguous*-relations, which are (1) $amb(c,b,1)$, (2) $amb(d,b,1)$, (3) $amb(c,b,1)$ and $amb(d,b,2)$, and (4) $amb(d,b,1)$ and $amb(c,b,2)$. We show the variation program exemplary for permutation (3) in Table 2, starting with the empty interpretation, leading to the alternative and valid model $a \quad c \quad d \quad b$.

The atoms *ambiguous* and *adjust* are abbreviated to *amb* and *adj* to fit the table. Furthermore, Table 2 only shows the atoms that appear in $I^\top$ and $I^\perp$ for the first time to maintain readability, as was done in Dietz et al. (2015). The column on the right side of the table signifies the clause which leads to the atoms shown in the respective row.

In iteration 2 and 3 in Table 2, the model obtained after processing the first swap-request is computed, which is $a \quad c \quad b \quad d$ (phase 4).

Thereafter the final alte rnative model is computed $a \quad c \quad d \quad b$, determining all *nl*-relations that hold in the model, as can be seen in iteration 3 and 4 (phase 5).

The answer to the query of the problem, $left(b,d)$ is determined in iteration 4. Since $left(b,d)$ is *False* in the fixed point of Φ_Π, this relation does not hold in the alternative model. It does however hold in the preferred model $a \quad b \quad c \quad d$. Conclusively, the final answer of our implementation is "No", because there was at least one model in which the relation described in the query did not hold.

Table 2: Alternative model computation with two swaps.

Φ_Π	$I^\top$	$I^\perp$	clause nr./ program
↑ 1	$il(a,b), il(a,c),$		(9)
	$il(c,d),$		(9)
	$nl(a,b,3), nl(b,c,3),$		(12)
	$nl(c,d,3)$		(12)
	$amb(c,b,1),$		(14)
	$amb(d,b,2),$		(14)
	$adj(b,3), adj(c,3),$		(13)
	$adj(b,4), adj(d,4),$		(13)
↑ 2	$amb(b,c,1),$		(15)
	$amb(b,d,2),$		(15)
		$nl(a,b,4), nl(b,a,4), nl(c,a,4),$	(19)
		$nl(c,b,4), nl(c,d,4), nl(d,b,4),$	(19)
		$nl(d,c,4), nl(a,b,5), nl(b,a,5),$	(19)
		$nl(c,b,5), nl(d,a,5), nl(d,b,5),$	(19)
		$nl(d,c,5),$	(19)
	$nl(b,d,4),$		(20)
	$nl(a,c,4)$		(20)
↑ 3		$left(a,b), left(b,a), left(c,b),$	(16)
		$left(d,a), left(d,b), left(d,c)$	(16)
	nl(a, c, 5),	$nl(a,d,5), nl(b,c,5), \ln(b,d,5),$	(19)
		$nl(c,a,5),, nl(c,d,5)$	(19)
	$nl(c,b,4),$		(20)
	nl(d, b, 5)		(20)
↑ 4	$left(a,c),$	$left(a,d), left(b,c),$ ***left(b, d)***,	(16)
	$left(d,b),$	$left(c,a), left(c,d),$	(16)
		$right(a,b), right(a,d), right(b,a),$	(18)
		$right(b,c), left(b,d), right(c,d)$	(18)
	nl(c, d, 5)		(20)
↑ 5	$left(c,d),$		(16)
	$right(c,a),$	$right(a,c), right(c,b), right(d,a),$	(18)
	$right(b,d),$	$right(d,b), right(d,c)$	(18)
		$chain$	(21)
↑ 6	$left(a,d),$		(17)
	$left(c,b),$		(17)
	$right(d,c),$		(18)
	$chain$		(21)
	ab		(21)
↑ 7	$left(a,b),$		(17)
	$right(d,a),$		(18)
	$right(b,c)$		(18)
		ab	(21)
↑ 8	$right(b,a)$		(18)

Discussion and Conclusions

The contribution of this paper comprises various aspects within both the area of Computer Science and Psychology. Through the formal process of modeling the spatial reasoning task, we have had to put forward new hypotheses on the model variation phase which need to be verified in the future:

Cognitive complexity of alternative models The variation phase starts with the information provided on the preferred model. How is this related to the *cognitive complexity* for the construction of the individual alternative models?

List of permutations Is the list of permutations cognitively adequate? Do humans keep track of such a list, or does one permutation trigger the next one? If humans keep such a list, how likely do they make mistakes? Are these mistakes related to the *distance* of the preferred model?

Ambiguity identification We suggested to rigorously identify ambiguities within the task. Yet, humans might be sloppy in the sense that they recognize certain ambiguities more easily. If so, which are the selection criteria?

Default and explicit knowledge Two notions of negation, weak and strong negation, were necessary for modeling this task. How does this distinction relate to other tasks?

The Weak Completion Semantics has shown again to be a good candidate for a comprehensive and computational cognitive theory, as it seems to adequately model yet other aspects of human reasoning task not considered so far. The WCS can fully cover all three stages of reasoning that have been suggested by the preferred mental model theory. This is novel as the WCS has previously never been considered to model the variation phase or alternative models in such a rigorous way. In particular, it seems that only few approaches (e.g., *mReasoner* Khemlani & Johnson-Laird, 2013) deal with the processes of alternative model construction. From a cognitive point of view, this is a central step if we intend to understand actual human reasoning, as one main part of it is concerned with the construction of counter examples. Future work includes the application of the current approach to other human reasoning tasks, such as syllogistic reasoning and reasoning with (counterfactual) conditionals. Furthermore, a metric among the alternative models and with respect to the model transformation should be specified. Possibly this could depend on the cardinality of the list of permutations or, more interestingly, on the amount of steps within the fixed point computation of the Φ operator. An interesting starting point of investigation would be whether a certain experimental setup could make it possible to mimic the operator iteration, by providing participants the information sequentially.

References

Byrne, R. M. J. (1989). Suppressing valid inferences with conditionals. , *31*, 61–83.

Clark, K. L. (1978). Negation as failure. In H. Gallaire & J. Minker (Eds.), *Logic and data bases* (Vol. 1, pp. 293–322). New York, NY: Plenum Press.

Dietz, E.-A., Hölldobler, S., & Höps, R. (2015). A Computational Logic Approach to Human Spatial Reasoning. In *IEEE symposium series on computational intelligence, (SSCI 2015)* (pp. 1627–1634). IEEE.

Dietz Saldanha, E.-A., Hölldobler, S., Kencana Ramli, C., & Palacios Medinacelli, L. (2018). A Core Method for the Weak Completion Semantics with Skeptical Abduction. *J. of A. I. Res. Special Track on Deep Learning, Knowledge Representation, and Reasoning*, *63*, 51 – 86.

Dietz Saldanha, E.-A., Hölldobler, S., & Pereira, L. M. (2017). Contextual Reasoning: Usually Birds Can Abductively Fly. In M. Balduccini & T. Janhunen (Eds.), *Proceedings of 14th international conference on logic programming and nonmonotonic reasoning (lpnmr)* (Vol. 10377, pp. 64–77). Springer International Publishing.

Grice, H. P. (1975). Logic and Conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and semantics: Vol. 3: Speech acts* (p. 41-58). New York: Academic Press.

Hölldobler, S. (2015). Weak Completion Semantics and its Applications in Human Reasoning. In U. Furbach & C. Schon (Eds.), *Proc. of workshop on bridging the gap between human and automated reasoning* (pp. 2–16). CEUR-WS.org.

Hölldobler, S., & Kencana Ramli, C. D. (2009). Logic Programs under Three-Valued Łukasiewicz Semantics. In P. M. Hill & D. S. Warren (Eds.), *Iclp* (Vol. 5649). Springer.

Johnson-Laird, P. N. (1983). *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge, MA: HARVARDUP.

Khemlani, S., & Johnson-Laird, P. N. (2012). Theories of the Syllogism: A Meta-Analysis. *Psy. Bulletin*, 427-457.

Khemlani, S., & Johnson-Laird, P. N. (2013). The processes of inference. *Argument & Computation*, *4*(1), 4-20.

Łukasiewicz, J. (1920). O logice trójwartościowej. *Ruch Filozoficzny*, *5*, 169-171.

Oliviera da Costa, A., Dietz Saldanha, E.-A., Hölldobler, S., & Ragni, M. (2017). A Computational Logic Approach to Human Syllogistic Reasoning. In G. Gunzelmann, A. Howes, T. Tenbrink, & E. J. Davelaar (Eds.), *Proceedings of the 39th annual conference of the cognitive science society* (p. 883-888). Austin, TX: Cognitive Science Society.

Ragni, M., & Knauff, M. (2013). A Theory and a Computational Model of Spatial Reasoning With Preferred Mental Models. In *Psy. rev. 2013* (Vol. 120, pp. 561 – 588).

Rips, L. J. (1994). *The Psychology of Proof: Deductive Reasoning in Human Thinking*. Cambridge, MA: MIT Press.

Stenning, K., & van Lambalgen, M. (2005). Semantic Interpretation as Computation in Nonmonotonic Logic: The Real Meaning of the Suppression Task. *Cognitive Science*, *6*(29), 916–960.

Stenning, K., & van Lambalgen, M. (2008). *Human Reasoning and Cognitive Science*. Cambridge, MA.

Wason, P. (1968). Reasoning about a Rule. , *20*(3), 273–281.