Bachelor Thesis

Individualizing a Spatial Reasoner for the Mental Model Theory

Christian Breu

Examiner: apl. Prof. Dr. Dr. Marco Ragni

Advisers: apl. Prof. Dr. Dr. Marco Ragni

Albert-Ludwigs-University Freiburg

Faculty of Engineering

Department of Computer Science

Cognitive Computation Lab

January 19th, 2019

Writing Period

19. 10. 2018 – 19. 1. 2019

Examiner

apl. Prof. Dr. Dr. Marco Ragni

Advisers

apl. Prof. Dr. Dr. Marco Ragni

Declaration

I hereby declare, that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work. I hereby also declare, that my Thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

Place, Date	Signature

Abstract

The research topic of this thesis is to find out whether it is possible to model individual reasoning behavior in the domain of spatial reasoning of cognitive science. Cognitive models are usually optimized to fit a general result for a group of people. There is no research about predictive modeling of individual answers to reasoning problems from single participant data. The goal of this work is to find possible ways to individualize a spatial model to be able to predict the results of a single person better.
An individualization is a modification of the problem-solving process of a spatial model. With the individualizations, a model can be adapted to predict the results of a specific participant. The individualized spatial model is used to statistically evaluate the different individualization factors implemented in the model. In general, it is possible to achieve better predictions for individual participant data.

Zusammenfassung

Diese Arbeit beschäftigt sich damit herauszufinden, ob es möglich ist individuelles Schlussfolgern in der Domäne des räumlichen Schlussfolgerns zu modellieren. Kognitive Modelle sind normalerweise darauf optimiert generelle Ergebnisse einer Gruppe von Personen möglichst genau vorherzusagen. Zurzeit gibt es noch keine Forschung zur Vorhersage der Antworten einzelner Personen aus experimentellen Daten. Das Ziel dieser Arbeit ist es, Möglichkeiten zu finden ein kognitives(räumliches) Modell zu individualisieren, so dass die Ergebnisse einer einzelnen Person besser modelliert werden können. Eine Individualisierung ist eine Veränderung im Problemlöseprozess eines Modells. Mit solchen Individualisierungen kann ein räumliches Modell angepasst werden auf eine bestimmte Person. Ein solches individualisiertes Modell wird statistisch ausgewertet, um die Individualisierungen zu evaluieren. Es ist prinzipiell möglich, mit solchen Individualisierungen bessere Ergebnisse zu erziehlen.

Contents

1 Introduction

Commonly, experiments in psychological research collect data from a number of individual participants and then analyze them to be able to draw a conclusion for a group of people. Sometimes, the data of the participants differ in a wide range. This is why usually the average of the data is inspected. There are various statistical analysis methods to find out whether there is a general tendency in the data or not.
In cognition science, based on such experiments, researchers create cognitive models to fit the data and to predict answers for unseen participants.
If a model can predict the results of an experiment, it can possibly give a new insight into how cognitive processes actually work. Usually, cognitive models are created based on group results obtained from psychological experiments. Because of this, the predictions can only be interpreted as a group result.
However, since cognition science is about how actual individuals solve problems, it is more interesting to be able to predict the experimental results for a single person instead of an average value for a whole group of different people.
With an individual model, predictions for a single person can be made based on prior results. This way, the modeling of the cognitive process of a human should be more precise. There might be a bigger scientific value on such a model, compared to a model that is predicting an average value for a group.

This work is focused on the mental model theory for spatial reasoning by Philip N. Johnson-Laird [1][2][3]. In order to obtain a cognitive model for individual people, a

spatial model that was implemented to fit data for a group of people, can be modified. In this work, a reimplemented version of the spatial model "space-5" from Johnson-Laird [1] is used.

The first step is a brief comparison between this model and data from an actual experiment about spatial reasoning. The results of individual participants of the experiment are compared with the predictions that are made from the spatial model. Since the results of a single person differ from the results of the whole group, significant differences to the results of the model are expected. To find those differences is the goal of the first step.

From there on, the model is modified, according to the differences that have been found, to be able to make predictions about individual results. The result of this process is a model that takes data from a single person as input. With such input, the model can predict the results of this individual person. This input changes the way the model builds or inspects a mental model of a given spatial problem. Since the participants do not give the same answers every time, it is likely that there are some factors that have an impact on the individual cognition.
Only a certain proportion of all the problems in the experiment are used for tuning the model. The given answers lead to the parameters for the model. The rest of the answers are used to test the performance of the model to predict how the individual participants will answer for specific problems.
If a certain set of modifications is able to model different individuals with good precision, this can hopefully lead to a better understanding of the cognitive processes in general and specifically for individual people and the differences between them.

2 Related work

In this chapter the theoretical background about the mental model theory will be explained. The CCOBRA (Cognitive Computation for Behavioral Reasoning Analysis) framework from the cognitive computation lab of the University of Freiburg will be introduced as well.

2.1 Mental Model Theory

The mental model theory by Philip N. Johnson-Laird and Ruth M.J. Byrne is a theory about human reasoning [1]. The theory presents a possible explanation of how people solve a given problem, which requires drawing inferences. Problems like this can be found in relational reasoning, where there are different objects with the relation between them defined in premises.

The main idea is that people will create an own mental representation of objects, according to the pieces of information that are given in the problem. Following Johnson-Laird and Byrne [3], a mental model is an integrated representation of these kinds of information. Johnson-Laird and Byrne [3] defined three phases of the reasoning process. In the "comprehension phase", the mental model is created from the problem information. In the next phase, the "description phase", this mental model will then be used to do a deduction in order to give an answer to the problem. The problem often contains a premise that needs to be checked, whether it holds or not. With this premise and the mental model, the reasoner will make an initial conclusion about the problem. The reasoner will try to

falsify this initial conclusion by building all possible mental representations of the given problem, as described by Johnson-Laird et al. [1]. The process of trying to falsify the initial conclusion is the last phase which is called the "verification phase".
According to Goodwin and Johnson-Laird [4], there are indeterminate relational problems that can be interpreted in various ways, because at least one relation of the objects inside the model is not determined. With such a problem, an indeterminate model will be created. Depending on the reasoner, the initial model and conclusion can differ based on the interpretation of the indeterminate problem. By checking all possible arrangements of the objects in the problem, the correct conclusion can be made.
In case the premise to be checked does not hold in the mental model, the reasoner will try to verify this premise by searching for a model where it holds. The mental model theory assumes that the order of the premises is not relevant for the resulting answer since the reasoner will try to build all possible models when searching for counterexamples for the previously made conclusion. As Johnson-Laird et al. [5] explained, it is difficult for humans to create more than one model, so indeterminate problems are significantly harder to solve than determinate ones. Various experiments give evidence that the use of mental representations by humans is very likely, as proposed by Johnson-Laird [2].

2.2 Preferred Mental Model Theory

The preferred mental model theory is based on the mental model theory and was presented by Marco Ragni and Markus Knauff [6]. Both theories support the assumption that humans create a mental model from a given problem. In the preferred mental model theory, the three phases are supported, but there are different assumptions. In the "comprehension phase" which is called "model construction phase" in the preferred mental model theory, the mental model is created. According to Ragni and Knauff [6], for determined problems, there is exactly one preferred model. In the next phase, the "model inspection phase" in the preferred mental model theory, new relations between the objects

will be inferred from this preferred model. In the last phase in the mental model theory, new models will be created to validate a previous conclusion. The preferred mental model theory proposes that the reasoner only modifies the preferred mental model to verify the inference that was made. Furthermore, the reasoner will not try to falsify the conclusion, if not explicitly told to do so in the experiment, as explained by Ragni and Knauff [6]. In Knauff [7], an experiment has shown evidence for the existence of preferred models in spatial reasoning. The preferred mental model is the model that will be created from the premises as proposed by Ragni and Knauff [6]. This process is deterministic and therefore results in the same preferred mental model for most of the people. The preferred model is easier to construct for the people and thus is preferred over other possible models. Since only one possible interpretation will be examined by the reasoner, errors are more likely to occur compared to the mental model theory approach. On the other hand, creating and remembering different mental models consumes cognitive resources. Only creating the preferred mental model will, therefore, be easier for the reasoner.

2.3 CCOBRA framework

The CCOBRA(Cognitive Computation for Behavioral Reasoning Analysis) framework was implemented by Nicolas Riester and Daniel Brand [8]. It can be downloaded from the cognitive computation lab [9]. There is also a website where the framework can be used without having to install it first [10]. CCOBRA is a framework, which enables the comparison of cognitive models in various domains. The spatial-relational domain is one of those, so it is suited to evaluate the spatial model and the variations made with individualizations. One of the main goals of the framework is to evaluate models with respect to their ability to predict actual results from experiments. The goal is not to be able to predict the results from aggregated data, but single predictions made by individual participants. The input for this are problems from actual experiments and the corresponding data from the participants. In the framework, there are two different types

of learning-functions that can be used by the model to adapt itself to the data.
The first training function is called *pretrain*. In this function, a set of experimental data for different individuals is given. The data consists of problems and the corresponding answers, which are used to adjust the model before the evaluation. The other training function is named *adapt* and will be called after each prediction in the evaluation. It can be used to adapt the model to the actual participant since in this function the answer of the participant is given. The model can then compare this result with its own prediction and change some settings inside the program to be possibly more precise in the following predictions.
In order to add a cognitive model to CCOBRA, a few functions from the model interface of the framework have to be implemented. To be able to run a model, only the predict function has to be implemented in a CCOBRA model class. The predict function simply returns the answer of the model to a given problem. There are no other requirements on the models themselves to be able to be evaluated, as explained in the CCOBRA documentation [8]. Since the restrictions are not so high for the models, there is more freedom to change the behavior of the model in order to gain more knowledge about cognitive processes e.g. the way how a mental model is created inside the program. The *pretrain* and the *adapt* functions are optional and do not have to be used to be able to evaluate a model. Also, there is a function *start participant*, which can be used to reset the model to the state after the pre-training.
Any benchmark can be used for the evaluation and new benchmarks can be set up as well. In this case, all benchmarks for the given experimental data are already in the framework. With the data from the experiment and the answers generated from the model, CCOBRA will evaluate the performance of the model. Figure 1 shows the basic evaluation flow of CCOBRA. The result of the evaluation will be presented in a plot. Through this method, it is possible to conveniently compare a lot of different models with the same problems, if they are all added to the framework and evaluation routine. The version of the framework from December 6th, 2018 was used.

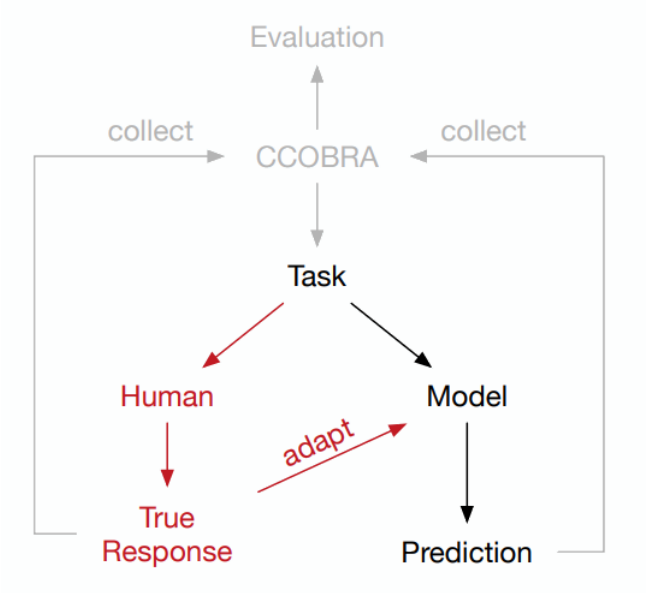

Figure 1: The basic workflow of CCOBRA, as explained in the slides of a presentation of the framework [11]. The model will be evaluated with real responses from participants. With this data, the model can be adapted to fit the data better in the following problems.

2.3.1 CCOBRA baseline models

In the framework, there are two baseline models with which the spatial model is compared. The first of these models is the "random model". This model always chooses the answer randomly from the given set of answer choices in the problem. There is no usage of the *pretrain* or *adapt* function, since the prediction is always a random choice.

The second model is the "transitive closure" model. This model checks all given premises for transitive dependencies between the objects. For each transitive dependency that was found, the model creates a new premise to represent this transitive relation. All relations of all objects in the problem are explicitly computed like this. The conclusion premise of the problem that needs to be verified, is then searched in the previously computed set of premises. If the conclusion premise can be found in these premises, "true" is returned. If the conclusion premise cannot be found in the computed premises, "false" is returned. This model also does not use the *pretrain* and *adapt* function.

3 Background

3.1 The Spatial Model

The spatial model used in this work is a program called "space-5" by Philip N. Johnson-Laird and was originally coded in Lisp. It can be found on a research website of the Princeton University [12]. The version that is used here is a semantically equal program written in Python 3.

The spatial model from Johnson-Laird is an implementation of the mental model theory for spatial relations. The model can solve problems consisting of relational premises. With these premises, an initial mental model will be constructed and then the conclusion given for the problem (as a premise as well) will be checked with this initial model. The result of this check is the initial conclusion, which states, whether the problem premise holds in the initial model or not. This initial model will be modified later in order to verify or falsify this conclusion and to come to a final conclusion about the given problem. The model will always try to falsify the initial conclusion from the initial model. The output is the result of this verification process and the model that led to this result. If the initial conclusion could not be falsified, the initially created model will be returned. The program only modifies the primarily created model, so it also fits the preferred mental model theory from Ragni and Knauff [6]. This section is about the general functionality and all important steps with regard to the mental model theory of the "space-5" program.

The premises are written out sentences like "the square is on the right of the circle" (as used in the source code of the program [12]), which encode the relation between two different objects. These premises will first be parsed so that only the relevant information remains. Every premise consists of the relative relation between two objects. The relation of the premise will be parsed into coordinate form. Relations are translated into a Cartesian coordinate system. The encoding for "right" is (1, 0, 0) and "left" is encoded as (-1, 0, 0). Suitably, the "above" and "below" relations are encoded as (0, 1, 0) and (0, -1, 0).
For example, the sentence from before will be parsed to a list containing "square", "(1, 0, 0)" and "circle". After parsing each premise, the program will process it in order to build the correct model resulting from the premises.

The program can create a new model with two objects and their relation. This happens, if there is no model that contains at least one of the objects in the premise. This part of the model creation will be called "model start" from now on.
To create a new model, the program will initialize a dictionary with the coordinates as key and the objects of the model as items. The parsing process already extracts the relation in coordinate form. For example for the premise "the x is on the left of the y", the relation will be parsed to (-1, 0, 0). The relation describes the relative position of the first item, in this case, "x" to the second item "y". The program will start by inserting the "y" at the default coordinates (0, 0, 0). After this, the "x" is inserted at the position of "y" with the added relation. This means "x" will be added at (-1, 0, 0).

If there is already a model that contains exactly one object in the parsed premise, the program will add the other object according to the relation in the premise. The program will search the closest free spot in the model, which satisfies the relation given

in the premise. If there is a model consisting of "x" at (0, 0, 0) and "y" at (1, 0, 0) with a premise "the z is in front of the y", the program will search the first free position for the "z". This leads to "z" being inserted at (1, 1, 0).
In some cases, there are multiple ways to add a new object into an existing model. However, the program always searches the first free spot in the model that satisfies the relation of the premise. This part of the model creation will be called "model insert" from now on.

If both objects are already in a model, but they are in different ones, the program will combine those two models according to the relation between the objects. This will fuse two smaller models into one big model. The program will compute the coordinate offset for all objects for one of the two models. With this offset added to the respective coordinates, all objects of one model can be copied into the other one to combine the two models. For instance, with two given models, the first containing object "x" at (0, 0, 0) and "y" at (1, 0, 0), and the second model containing object "z" at (0, 0, 0) and "q" at (1, 0, 0). The premise triggering the combination of these models is "the z is on the right of the y". The program will find a new origin for the second model in order to combine the models according to the relation. As a result, the two models are combined into one bigger model with object "x" at coordinates (0, 0, 0), "y" at (1, 0, 0) , "z" at (2, 0, 0) and "q" at(3, 0, 0).
This process will be called "model combine" from now on.

When both objects of the processed premise can be found in the same model, the program will try to verify the model by checking, if the relation in the parsed premise does hold in the actual model. The result will be "true" if the relation between the two objects in the premise holds in the model and "false" if the premise does not hold in the model. This part of the program will be called "model verify" from now on.
If "model verify" comes to the result that the premise holds in the model, the program will try to falsify this result. If the premise does not hold in the initial model, the program

will try to verify this result. In order to verify or falsify a model, the program will try to find a new model, where all premises hold, except the initial conclusion premise given in the problem. The program will invert the conclusion premise. With this changed premise, the program will modify the current model in order to make all premises hold. If an alternative model exists, where the inverted question premise does hold, the problem cannot be answered with the initial conclusion. A new conclusion will be returned by the program. If such a model does not exist, the program will draw the final conclusion based on the initial model. The general procedure for the parsed premise in the model can be seen in figure 2. After the process is done, the output of the program will be the answer to the given task, according to the model. If the initial conclusion could be falsified, the answer will be given accordingly. The program does not exhaustively check all possible ways to modify a model to make a premise hold. There may be other modifications which will result in a suitable model in some cases.

3.2 The experimental data

This chapter presents the given experimental data for this work.
Three data sets of different experiments were given for this research. The experiments were conducted to check for pieces of evidence for different theories within the mental model theory for spatial reasoning. All three of them were online experiments and thus had no restrictions on the participants.

3.2.1 First experiment

The first data set is from an experiment which took place in 2012. In the first part of this experiment, the participants had to solve 48 different problems that were presented to them one after another. In the next part, 16 similar but different problems were presented

as well. So in total, the participants had to solve 64 problems. All problems had an unique identification number assigned to them. 33 participants took part in this experiment. The data of the first part of the experiment will be referred to as "premise order" data. The second part of the experiment will be called "figural" data in later chapters.
The problems for the first part of the experiment contained 4 premises, each with the spatial relation between two different objects. The only possible relations for the premises were "left" and "right". The objects could be any letter from "A", "B", "C" and "D". After processing all of the premises, the resulting model contains all four letters in a specified order. These premises were presented one after another. The participant could decide when to go to the next premise. The time between the premises was measured as well as the total time used for each problem.
According to the mental model theory, the participant should build up their own mental representation of the relations and objects that were given. The fourth and last premise contained objects that were already mentioned before. This premise is the question that can now be answered with true or false by pressing a specified button. Supposedly, there are undetermined models in the problems, so there is always a correct and incorrect answer. This experiment is about the verification of a given model. The question premise only contains objects that were given before, so the participant only has to check, whether this premise holds in the model or not.

However, out of the 48 problems, three problems are undetermined. For these problems, the correct answer depends on the interpretation of the premises. Therefore they were not used for this research since the correctness of the answer needs to be decidable to find possible reasons for errors made by the participants. This could be a result of an error in the data set or in the creation of the problems.
There were also a few participants that only answered to a fraction of the problems that are required to finish the experiment. Since it is a web experiment such cases are bound to happen.
Because a certain amount of answered problems is needed to find out as much as possible

about the participant, only the data of 30 participants could be used. The goal of this part of the experiment was to find potential influences of the order of the premises on the results. The order of the presented premises can make a difference how the reasoner builds a model. In this experiment, the performance of the participants was tested for different premise arrangements. In the data, all problems were categorized into three types regarding the premise order effect. The problems of type 1 start to build up the model from A and B or C and D. The following premises only add one more object to the model each, so this problem type is continuous with respect to the model building phase. The new objects are added to the model on the same side because the problem is continuous. Type 2 problems always start with B and C (in any order) and then build the model by adding A and the D (in any order). These problems are semi-continuous because there is a change of direction in the model building process. This change of direction is represented through the change of the side, where the last new object will be inserted. The A is added on the left side, while the D is added on the right side of the existing model.
Problems of type 3 will lead to two sub-models of A and B and D and C with a combination of both of them resulting in the final model to be built. This happens in such a way that the problem is built non-continuous. These problem types will be referred later on with "problem type 1", "problem type 2" and "problem type 3".

The second part of this experiment had 16 problems, each containing three premises.
The third and last premise was the question that was to be answered with true or false. The presentation and form of the premises are the same as in the other set of problems. Again, the times used for each premise and the question were measured. These problems were created to test for the figural effect. The figural effect is the influence that the order of the objects in the premises has on the processing of them, as explained by Knauff et al.[13]. Following this, the figure of a problem is the order of the objects in each premise.

3.2.2 Newer experiments

The other two data sets are from experiments from 2018. Anyone could participate in these studies through the website of the cognitive computation lab of the University of Freiburg [14]. The experiments take about 20-40 minutes, depending on the participant.

The second experiment is called "spatial1", but it will be referred to as the "single choice" data in later chapters. 49 people participated in the experiment. There is also demographic data about the participants including age, the highest education, profession, gender, handedness and a few answers about the motivation of the participant.
The participants had to solve 64 problems each. The problems were split into two parts. In the first part, the participant will see two premises which state a relation between two different objects each. After reading the first premise, the participants can go to the next premise by pressing space. The relation between two objects can be "north", "south", "east", "west", "north-east", "south-east", "north-west" and "south-west". The objects were buildings which can be found inside a city, e.g. "mall", "station" and "bar". The second part of the problem is a question and the answer that the participant has to give. The question is a sentence containing two objects and free space for the relation between these two. This is a "single choice" experiment, so the participant has to choose one single answer to fill into the sentence. Eight different choices were presented to the participants. Each of those choices is a relation that will complete the sentence of the question to a premise. In order to select a certain relation, the participants had to type the short version of the relation, e.g. "nw" for "north-west". The time used by the participant for part one and part two of the problem was measured separately. This experiment is not about the verification of a given question premise, but the participant has to choose the correct answer from eight different choices.

The third experiment is called "spatial2", but it will be referred to as "verification" data in later chapters. 51 people participated in this experiment. They had to solve 48

problems each. Again, in each problem, a set of premises was presented to the participants. In the first part of the problem, four different premises were presented at once. The premises contained again two objects and their relation. The relation could be "west", "left", "right" or "east", but only one relation is used for each premise of the problem. The objects in the premises were fruits or fruit trees. For example "lime", "plum" or "mango tree". The four premises in the first part of the problem contain a total of five objects.

In the second part of the problem, a set of four premises, containing the same five objects were presented at once. The participant has to give an answer, whether the model created in the first part of the problem can be created with these premises as well. The answer can be "True" (yes) or "False" (no), with respect to the possible equality of the presented models. If all the spatial relations between the objects in the new set of premises are equal to the relations of the objects in the model of the first part of the problem, the participant should answer "True". If a relation is not equal in the two models, this does not necessarily mean that it is not possible to create the same model with the respective premises. If there is a relation that is not equal and cannot be interpreted in a different way to match the model from the first part of the problem, the answer should be "False". In this experiment, a whole model is to be verified in contrast to just one relation between two objects in the other experiments. The times for the first and second part of the problems were recorded like in the previous experiment. This experiment is, like the "premise order" and "figural" experiment, just about the verification of a given model. There are more premises to be checked by the participant, but the principle is the same.

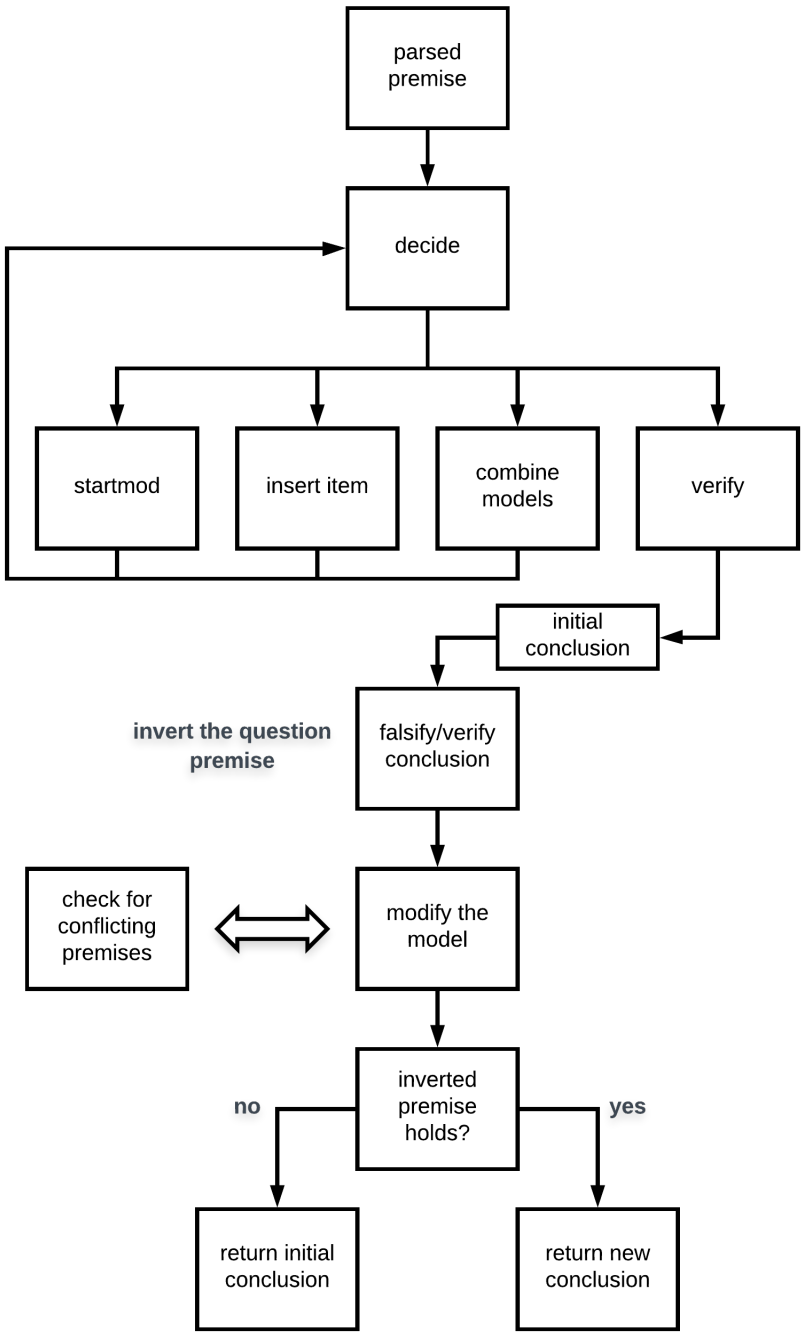

Figure 2: This figure shows the general procedure of the spatial model from Philip N. Johnson-Laird. In the *decide* function, the program chooses the next operation to process the problem, based on the current premise.

4 Parameter assignments approach

This chapter explains the problem and the general approach in detail. The first step is an analysis of the experimental data, which is presented after the problem definition. Then the implementation of some of the ideas resulting from the analysis, which are called individualizations, are explained. Finally, these individualizations are tested in this approach.

4.1 Problem definition

The goal is to find individualizations for single persons in spatial reasoning and to apply them in the spatial model from Johnson-Laird [12]. These individualizations are expected to have an effect on the processing of problems in the model. The model with individualizations can be adapted specifically to experimental data of a single person. This adapted model potentially makes better predictions for this specific person. Therefore, the main task is to find suitable individualizations for the given experimental data. This includes the process of finding possible reasons for why the participants in the experiment make errors in some problems and in others not. An individualization is relevant if there is at least data of a few participants that shows evidence of an effect on the result of a problem. In other words, if a mistake can be explained by a change in the problem-solving process for more than one person, it can be the base for a relevant individualization. A very important quality for such an individualization is that the individualization can still result in correct answers for other, similar problems. If an individualization can explain all

of the errors made by a person but also changes the results that were previously correct, the individualization is not beneficial.

4.2 General approach

In order to find suitable individualizations for the model, the "premise order" and "figural" data sets from the first experiment were analyzed. From this analysis, ideas for possible reasons of errors made by the participants were developed.
Some of these ideas for possible errors were picked, to implement them in the spatial model. The individualizations can be turned on and off to be able to test all possible combinations of them in the spatial model. The program has a parameter list of boolean values, to specify which individualization should be activated for this run of the program. In the program, there will be a statistical evaluation of all possible parameter assignments for the spatial model. With this method, the best parameter assignments, with regard to the precision of the participant prediction, can be found. The best parameter assignments can then lead to new conclusions about the usefulness of the tested individualizations.

4.3 First analysis of the data

In this step, a manual analysis of the data of the first experiment was carried out. The goal of this analysis was to find any indication for individual factors that can be used in the spatial model. The focus was on finding possible reasons for errors that the participants might have made during the experiment. The problems that were answered incorrectly, were observed with the aim to find out if there is a systematical error in all wrong answered problems of the same person.
The spatial model implements the mental model theory, so it contains all the essential steps for processing a problem an hence is suitable to describe the way the participants make errors. Therefore, an error can be described by specifying the step that results

in (erroneously) changing the procedure to process a problem in the spatial model. For example, an error in the combination process in the spatial model will lead to a different model.

Data of all participants were checked, if they had an error type in common or if they tend to make errors for the same problems. A lot of the incorrect answers of the participants cannot be explained because there are very often other problems with the same kind of premises (so they should be similar in the way that the participant is solving them) that were answered correctly by the same person. However, the brief analysis resulted in the following options for possible errors made by the participants:

- Misinterpretation of a premise:

  When reading the premises, some relations can be harder to understand than others. The participant could misread one or multiple relations in the premises and therefore build up a wrong model because the premise was not understood correctly. The premises only contain the relation between two objects, so by misreading the relation of the premise, the participant inverts the relation of the premise. There is no other way to misinterpret a one-dimensional relation between two objects.

  The other way of misreading a premise is to omit the relation and just create the mental representation of the premise by the relation of the two objects when they are presented. For example, the premise D R C (D is on the right of C) should lead to the model CD, but when the premise is misread like mentioned before, the resulting model might be DC. For this way of misreading a premise, the inversion of the relation can be used to realize the individualization.

- Model building direction and premise direction:

  When a model is built up step by step by successively adding the objects, the direction in which the next object has to be inserted can make a difference. If there is a model AB and the next premise states that C is on the right of B, the constellation of the objects can be understood quite easily. But if the next premise would state that B is on the left of C, the participant might think that something

has to be added to the left of some object, while the correct action to take would be to add the C to the right of B.

- Implicit model combination:

  Another way how a wrong model could emerge is, when the first two premises lead to the creation of two different models, like for problems of type 3 in the "premise order" data. Those two models do not have any connection at this point, but the participant might intuitively arrange the two models and perform an implicit model combination that way. When the third premise is read, it may be misread or simply ignored due to the fact that the model is already completed.

- Guessing:

  The participant could be guessing all the time and randomly click on the given answer choices. In the "premise order" data there is also another option for guessing. Since the model ABCD results quite often from the premises, the participant might think that the premises always lead to this model and starts to skip the premises and answers the question based on this one standard model.

- Verbal Memory:

  This option is derived from the idea that the participants possibly do not even build a model in such a way the mental model theory states. If a participant is answering with only the use of the verbal memory, the answer only depends on the premises and the question premise. No model will be built. If a similar premise to the question premise was already given within the other premises, the participant can answer the question right away without building a model.

- Modify the initial model:

  According to the preferred mental model theory, the initially created model will be used to solve the problem most of the time [6]. However, it is not clear if there are some participants, who actually try to modify the initial model to verify or

falsify their conclusion. In the spatial model from Johnson-Laird, this functionality is already implemented.

- Alternative Combination:

  In the problems of type 3 from the "premise order" data, the model creation phase can be changed with an alternative way to combine two sub-models. In this option, the model will be assembled correctly, but while putting the two sub-models in correct order, they overlap. This leads to an effective switch of two objects. For example the two sub-models AB and CD would result in a model ABCD with the premise "B L C". With this change, the resulting model will be ACBD. The way how the models are combined is correct, with respect to the order of the sub-models. But the models do overlap now, which results in a model where one premise does not hold. In combination with the inversion of one of the three premises, this can lead to different models and hence possibly lead to better modeling of the participants' answers. Since the combine operation is only needed for problems from the "premise order" experiment, this option can only be used for these problems.

- Alternative Insert:

  There are two different ways to insert a new element into a given model with a given premise. The new object has a relation to one of the objects in the model. This relation can be interpreted in two ways. A strict interpretation would lead to the new model being inserted into the model directly next to the other object mentioned in the premise. If the relation in the premise is "right", the strict interpretation could be described with "directly to the right of". The meaning of the relation can also be interpreted less strict, into "somewhere to the right of". So the new object can be placed at the next free spot, which satisfies the relation.

4.4 Derivation of individual factors for the participants of the experiment

In the first raw analysis of the data, possible errors of the participants were found. However, these are just ideas about how the problems are answered incorrectly. To actually find out if these possible factors have an impact on the participants, a statistical evaluation is needed.

To do this, the individual factors that were proposed in the results of the first raw analysis, were implemented into the spatial model. All factors are in the program and can be activated separately. This way the program can be run with any combination of individualization factors.

Since the goal is to emulate the results of a given participant, the parameter assignments for the individualizations are chosen based on the answer that the participant gave to a specific problem. The answer to that problem should be wrong, so the resulting parameter assignments can lead to actual errors in the answers of the model and thus probably make a better prediction for the answers of the participants .

The program will find all parameter assignments that lead to the same answer as the given one. All of these assignments are tested on a certain amount of problem data from the corresponding participant. This way each assignment has a precision for the prediction of the answers given by the participant. The program will then return the most precise assignment for active individualizations that can be used for future predictions of answers of participants.

4.4.1 The individualization factors in the program (spatial model)

In this section, all individualizations that were used in the spatial model are going to be explained in detail. Also, the way how they were implemented is explained. Before explaining the additions to the program, the changes necessary for the spatial model to be used with the given experimental data will be briefly mentioned.

The spatial model has a parser, but the parser always returns the correctly parsed premise, so it is easier to change the premise directly to simulate a misunderstanding of a premise or any other modification in the parsing process that leads to a different result. For this reason, the parser is not used in this version of the spatial model. Also, the experimental data cannot be processed by the parser because the problems are stored in a different format.

Since there is no parsing, the functions for the verification or falsification of the initial conclusion need to be changed accordingly. There is no change in the actual semantics of the function.

The program can process each problem type of the three experiments. The "single choice" experiment has a different type of problems than the other experiments. Usually, the task is to verify a specified model and then answer whether a certain premise (or multiple premises) holds in the model or not. In the "single choice" experiment, the participant has to choose one answer from eight different answer premises given in the problem. Therefore the model will have two top-level functions to process the two types of tasks differently. For the "single choice" experiment, the model will return a premise, which resembles one of the eight possible answers. For all other problems, the top-level function will return a boolean value.

Besides these changes, some minor adaptions had to be done in order to use the spatial model for all experimental data. The output of the actual model is still the same, but the information, whether the answer is "True" or "False", is integrated into the returned model, in order to not change the whole structure of the program.

The individualization factors used for this approach are the guessing , misinterpretation of a premise, verbal memory usage and premise incorporate incorrectly option. The misinterpretation of a premise and the premise incorporate incorrectly individualizations are implemented for each of the three task premises of a problem.

The guessing option triggers a way to answer the problem, which is separate from the normal model building process. For the problems of the "premise order" and "figural"

data, the answer will be given based on the standard model ABCD and the question premise. Since this standard model occurs almost always in the problems of this experiment, there is a chance that some participants will be bored of reading all the premises and then just skip the premises and answer with this standard model in mind. For the other experiments, the guessing is implemented as a randomized answer between the presented choices of the task. This represents the case when the participant is just randomly guessing. There is no model that is mostly created like the one in the first experiment.

The "verbal memory" individualization will decide if the program answers with the model that was built or with the verbal memory instead. The verbal memory is represented through the problem since it contains all the premises and the question. In the program, the assumption is made that the participant can correctly remember all of the premises that were presented. The program will then search for a premise which answers the question premise clearly. This means that the program will search for a premise that contains the same two objects as the question premise. With such a premise, the program can decide whether the question premise holds or not without building a mental model. If there is no such premise in the problem or, one of the premises falsifies the question premise, the program will return "False" as the answer. If there is one, the program will return "True".

The other parameters affect the way the model is built up in the program and the answer will be given. This represents the normal case that the participant is actually trying to build a mental representation of the premises that are presented. The six parameters for the model building individualizations are used one after another when building the model. There is no parameter that affects the way how the model will be verified in this approach. For each premise that the program takes from a problem, two of the parameters will be used.
The first of the two taken parameters will determine if the current premise is understood

correctly or not. This parameter can activate the misinterpretation of a premise individualization. If a certain premise is misunderstood, the relation of the premise will be inverted. The premises only contain two objects and their relation, so there is only one way to misinterpret the premise. For example, the premise A L B, which means A is on the left of B, will be interpreted as A R B (A is on the right of B). The second parameter will be used in the model building process and can activate the premise incorporate incorrectly individualization. This individualization can take effect in "model start", "model insert" or "model combine" in the spatial model. For "model start" and "model combine", the error implemented by the individualization will connect the two objects or sub-models with the wrong relation. If a premise states that A has to be to the left of B, the program would place A to the right of B, if the corresponding parameter is true (active). In "model insert", an active parameter will also result in the object being inserted into the model with the opposite relation. So instead of adding the new object to the right of a certain object in the model, the program is adding the new object to the left. Effectively, the program is inverting the premise before performing the actual action to build the model. This means that if a premise is not understood correctly and after this, the same premise is not correctly incorporated into the model, there will be no effect on the model that has been built.
In a later approach, three other individualizations were implemented into the spatial model. The modification of the initial model is already implemented in the original spatial model. After the initial model is created and the question premise is verified, the model will try to falsify the conclusion with that initial model. If more than one premise is verified with the initial model, the model will try to falsify the corresponding initial conclusion for each of these premises. This individualization is named "modify initial model".

The alternative insertion individualization was implemented additionally to the insert functionality from the spatial model. With the alternative insertion individualization, the model inserts the new object directly next to the object that is already in the model.

If this position is already occupied, the object at this position will be moved by the relation of the premise. Generally, all objects with certain coordinates will be moved in the direction of the relation, to maintain all relations between the other objects in the model. With the model AC and the premise B R A("the B is on the right of A"), the "alternative insert" will put the B between the A and the C, whilst moving the C in the "right" direction. This will result in the model ABC.

The alternative combination can be activated after the combination is done. If the individualization is active, the model will be normalized with respect to the coordinates of the objects in the model. Then the two objects next to the former sub-models are swapped. For the model ABCD(with former sub-models AB and CD) after the combination, the alternative combination will change the model to ACBD. B and C are swapped because the two models overlap with this combination style. This individualization is only applicable to "premise order" problems. This individualization is called "alternative combine".

Figure 3 shows the spatial model with all individualizations. The individualizations are visualized with a colorization at the point in the program where they are used. Some of the individualizations change the model while others can directly lead to an answer.

4.4.2 Parameter assignments and evaluation

The goal is to be able to predict the answers of a specified participant. The program parameters for the individualizations will be computed with one specified problem and the corresponding answer given by a participant.
The program will iterate through all possible assignments of the parameters and then check whether the result of the program with these parameters is equal to the answer that the participant gave. For each of those parameter assignments where the given answer

matched, the program will compute the answers for a set of problems and check if the computed answer is equal to the corresponding answer given by the participant. Every parameter assignment will then have a precision value (from 0.0 to 1.0), which represents how good the parameter is modeling the results of the participant. The parameter assignment which has the best precision will be chosen as the assignment to be used for future problems. Usually, there are a lot of different assignments with the same precision value. Therefore the program is choosing one of them randomly since the precision value is the same for all of them.

The parameters are evaluated in the program as shown by the Binary Decision Diagram (BDD) in figure 4. This figure is not a real BDD, but this representation is more compact since the different steps in the model building phase are not dependent on the previous step. All individualizations are shown in the order in which they are used in the model. A parameter assignment represents a path through the BDD. The program will find the path with the best precision for each participant. Depending on the path chosen by the program, different conclusions can be made regarding the way the participants process the problems.

This approach is inspired by a similar approach from Klauer et. al [15]. In this paper, a number of probabilities are statistically optimized. With a given set of answers to a given task, Klauer et. al computed the most likely path through a BDD, containing individualization factors.

4.4.3 Implementation of the spatial model into the CCOBRA framework

In this implementation of the spatial model, the predict function of the CCOBRA framework just converts the problems into a format that can be processed by the spatial model. According to the problem, a suitable top-level function in the spatial model will be called. The output of the spatial model is also changed to fit the benchmarks for the experiments that are used by the framework.

The *pretrain* function is not used in this approach. This is because the data given by the framework in *pretrain* can only be used to change general parameters in the model. Since this model and this general approach are only based on data from one single person, there is no use for this function.

In order to find good parameter assignments for the individualizations, the program needs a certain amount of data from one person. In the *adapt* function, the given problem and the answer made by the participant are stored. If a specific number of problems has been processed and stored, the model searches for good parameter assignments. First, all possible parameter assignments are generated from one problem and the corresponding answer. These assignments are evaluated with all previous problems and corresponding answers. The resulting precision for each parameter assignment, which represents how good the results match the participants' answers, is used to pick the best assignment to be used for future problems. The precision value of the best parameter assignment has to be bigger than a certain threshold in order to be used. If there is more than one assignment with the same best precision value, one of them will be randomly chosen. Figure 5 shows this general process. If a certain amount of processed data is reached, the best parameter assignment is found as described in the flowchart.

In a later version of the model, the parameter assignment search was done in a specified interval. This should make the model more precise since the individualizations can be deactivated if the current data shows that there is no expected precision gain with the active individualization.

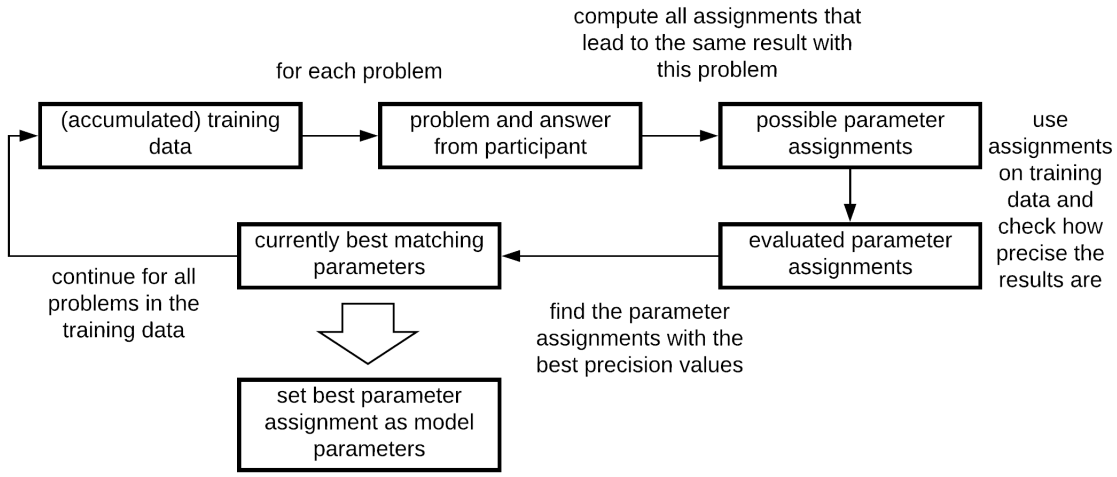

Figure 5: This figure shows the general procedure to find the best matching parameters for given participant data. This is processed in the *adapt* function of the CCOBRA model for the parameter assignment approach.

4.4.4 Results and analysis of the parameter approach

After implementing the spatial model into the CCOBRA framework, the model without any parameter set to active was evaluated in order to measure the baseline precision in all of the experiments. The spatial model was evaluated with the "figural", "premise order", "verification" and "single choice" benchmarks. These benchmarks contain all data from the three given experiments.

For the results of the baseline of the spatial model with all eight parameters set to "False", see figure 6. The baseline model represents the original spatial model from Johnson-Laird. The overall prediction precision is very high for the "figural" and "premise order" benchmarks, considering the model only computes the correct response to the given problem. This leads to the conclusion that the participants in this experiment gave the correct answer for most of the problems. Therefore it is assumed that the tasks are not very difficult for human reasoners. The precision values of the baseline model for the "verification" and the "single choice" problems are less precise, so these kinds of problems seem to be harder for the participants

The parameters model was tested for all four experiment benchmarks. The spatial model with parameters only uses a different assignment than the baseline if the precision of this assignment is better than a certain threshold. The results presented in figure 6 were achieved with a relatively low threshold of 0.7 precision over the training problems. This threshold value is still reasonable because the baseline in each experiment is less precise. For the "figural" benchmark, the threshold was set to 0.8 for the parameter model to be potentially able to beat the baseline. If the threshold is higher than the best precision value, none of the parameter assignments is activated and the results can therefore not be better than the baseline. The first version of the model that was tested, processed the parameter search only once after a specified amount of problems that was already answered by the model. The number of problems that were used for the training in the evaluation is 15. The parameter model cannot beat the baseline in terms of precision, with this implementation of the *adapt* function.

In a later version of the model, the *adapt* function will trigger the parameter search in specified intervals. After the first 10 problems, the model will try to find better parameter assignments always after 10 more problems are processed. The model was tested with various numbers for the interval and the first parameter search. The interval length seems to not have a big impact on the result. The found parameters are not precise enough to be used for the model, so the model is still only as good as the baseline at best. If the threshold for the use of the found assignments is lower, the results will get worse. These results show that this approach of finding a good assignment with all the individualizations together is not trivial. One of the main problems of this approach are the unknown side effects of the individualizations in the model for each other. For example, the "verbal memory" individualization can only be used exclusively in the program, so other individualizations cannot be used at the same time. There are potentially a lot of dependencies, so a different approach is needed.

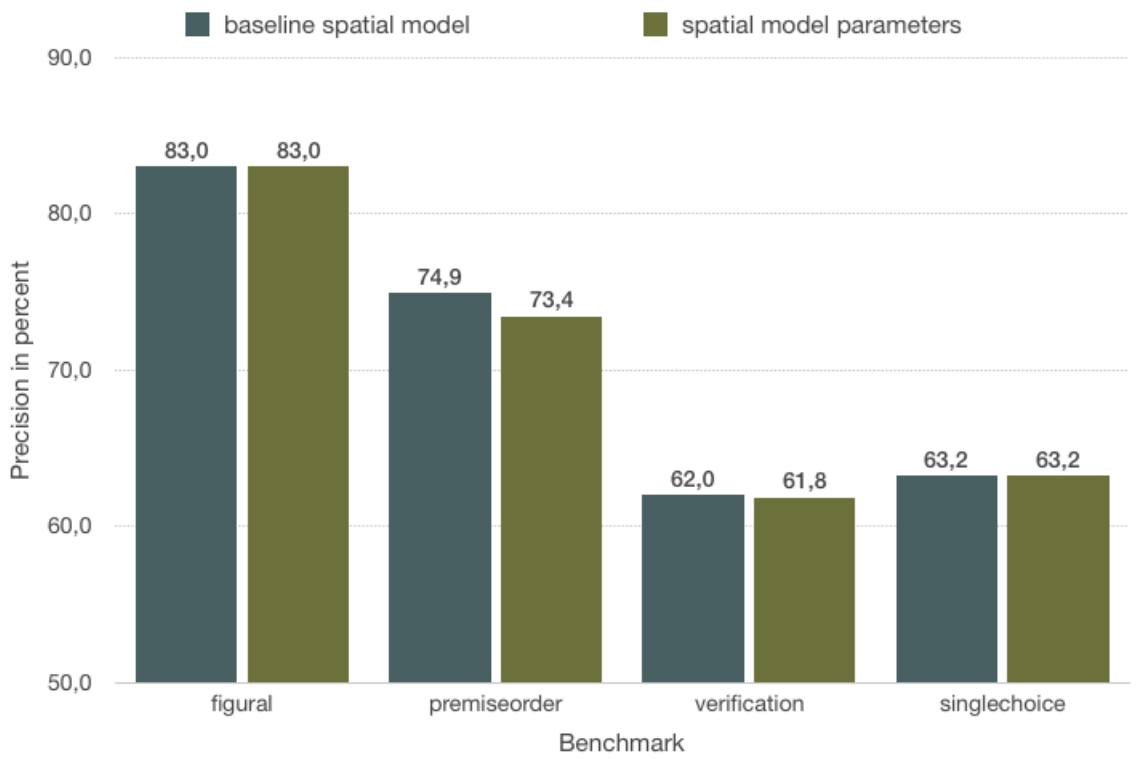

Figure 6: The results of the original spatial model without parsing and the parameter model. The values are the overall precision of the tested model in the CCOBRA framework. The results are computed with a specified amount of problems and participant data from the respective benchmark.

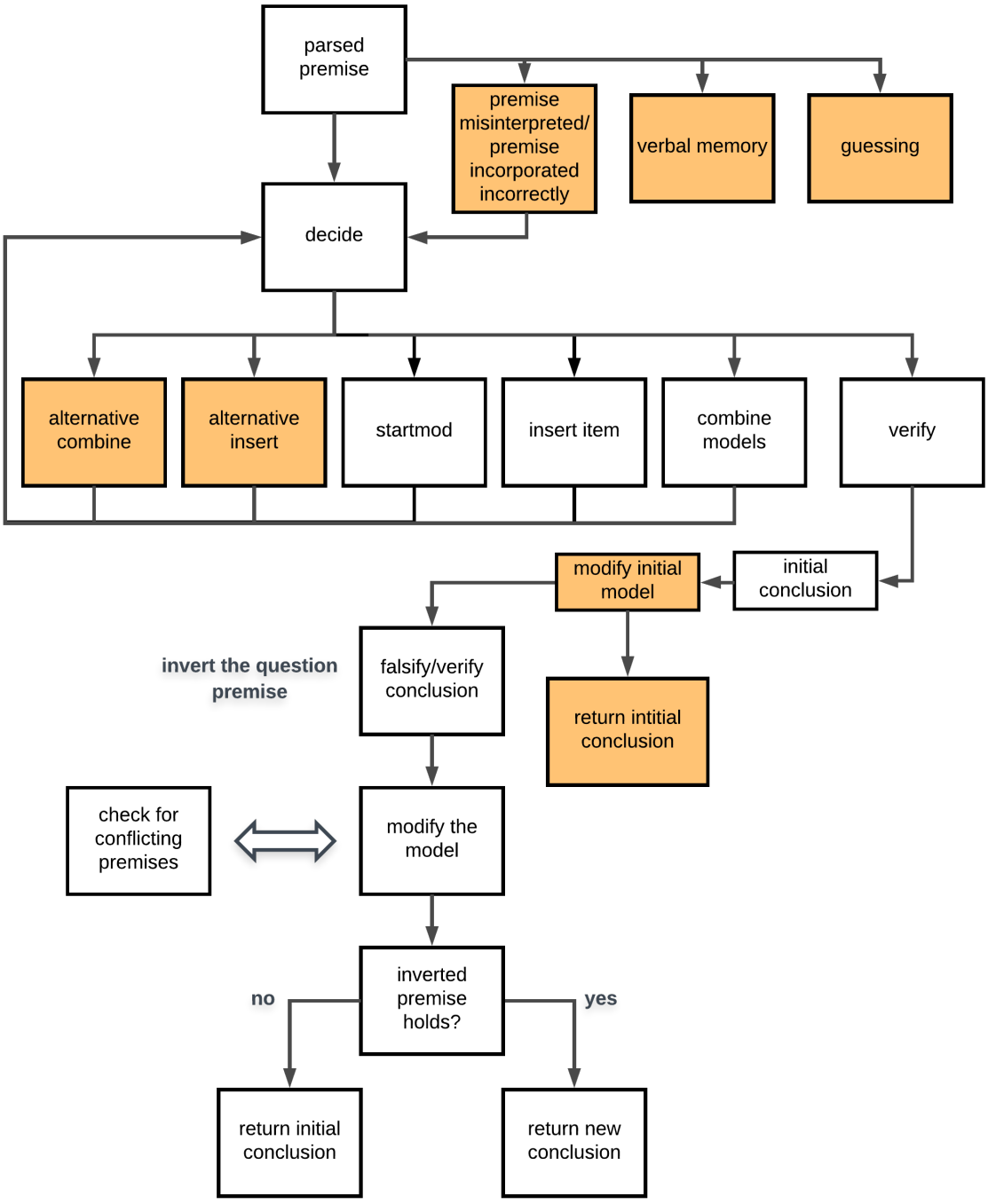

Figure 3: This figure shows the general processing of the parsed premises in the individualized spatial model. The individualizations are colorized. There are Individualizations that will change the general flow of the program completely. Other individualizations will just change the created model.

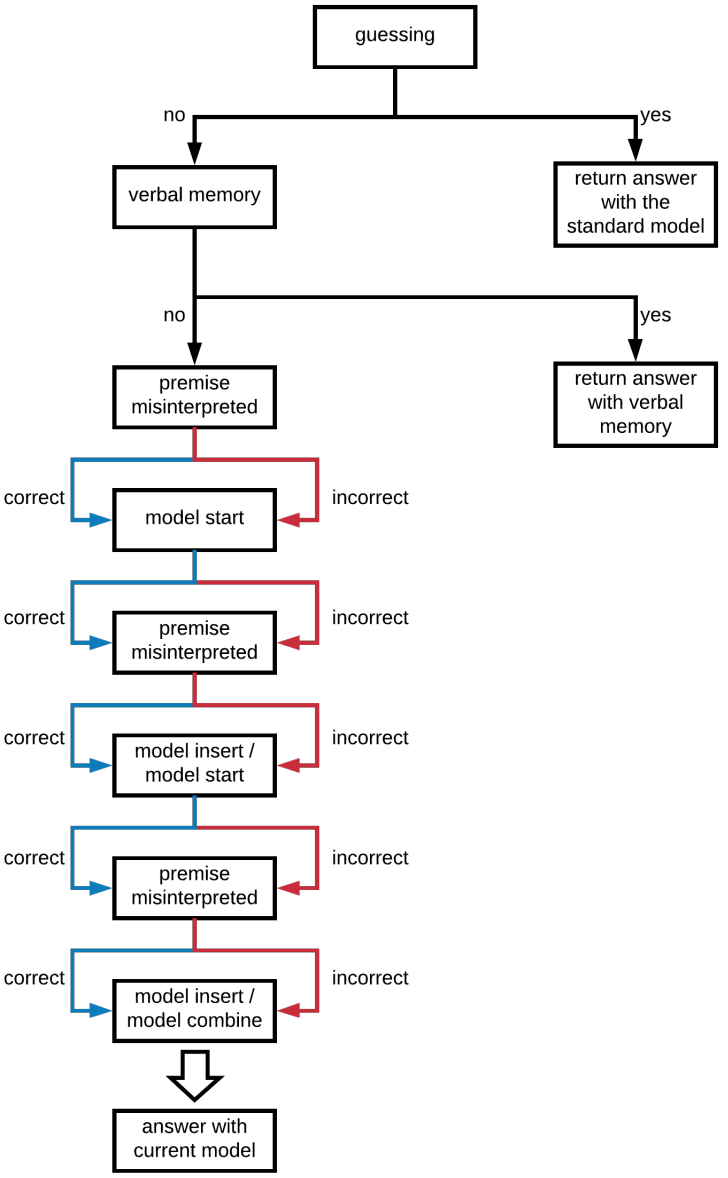

Figure 4: The parameters for the individualizations in the program in the order as they are evaluated in the program. The parameters for the model building process lead to a very big BDD with 32 possible paths. However, the decisions are always the same, since they are not dependant on the outcome of the prior decision. For the decision order to be clearer, the graphic shows all possible paths through the diagram, but it is not the actual BDD on which the program is based on.

5 Categories approach

This chapter is about the second approach to find good individualizations for the spatial model. The new idea is to categorize the problems to learn more about the differences between them. The goal of this approach is to be able to find good parameter assignments or even new possible individualizations.

5.1 Categorization approach

Since there is not much data given in the evaluation, the problems that were given should be analyzed as precise as possible. For instance, a participant made a mistake in a certain problem but gave the correct answer to another, similar problem. The challenge is, to find out what the difference between those two problems is. Obviously, there are various other factors that have an influence on the given answer by the participant, but the problem itself can make a big difference as well.

In this approach, the problems of the "premise order" experiment were used for a new analysis to find possible categories. In order to categorize the problems, for systematic differences between them was searched. For example, if there is a group of problems which leads to a certain way to build up a model, this could be a possible category. Other categories would result in a model that will be built up differently. Since the problems are very alike, every difference can lead to a possible category.
Categorizations can be used to find possible error patterns of the participants. If all

problems of a certain category are answered incorrectly, but another category only contains problems with correct answers, a pattern was found for this participant. The difference between the two categories probably leads to the error, so an individualization that handles this part of the problem-solving process can be activated. The goal is to find a categorization that can indicate answer patterns for as many participants as possible.

5.1.1 The categories

The problems of the "premise order" experiment with four premises are already categorized. The first category for these problems is the previously explained problem type. The problem type already categorizes the problems into three categories. These three categories are the continuous (or problem type one), the semi-continuous (or problem type two) and the non-continuous (problem type three) problems. As explained in chapter 3, the types of problems change the way how the model is built. Within these problem types, the problems are categorized into another two subcategories. More than that seems not applicable to the number of problems that were given. With more than three categories for the in total 45 problems, there would be not enough problems per category to get meaningful results.
Further analysis of the problems resulted in possible categories that seemed to be worth to be investigated. In the following paragraphs, the possible categories are explained and why they may help with understanding the reason why the participants make errors. Each category is only applicable to one problem type.

The first category group is about the direction in which the model is built. There are various possibilities to split the problems into such categories. These categories are about the order of the objects in the premises.
For problem type one, this means that the problems can be separated by the elements in the first premise. Since problems of type one are continuous, the model will be built up from A and B or from D and C. For problems of type three, the direction in which

the model will be built will be represented by the first premise as well. There is either A and B or C and D, which decides the elements that are contained in the following premises.

For problems of type two, this building direction can be determined with the second premise. The second premise contains A and B or C and D, this is also the way how the category will split the problems. The differentiation of the two subcategories will be made through the elements in the second premise. This also decides, in which direction the model will be completed first.

The next categories are about the model building direction with respect to which side a new object is added to the model.
For problem type two, the model building direction is different. Since these problems are semi-continuous, there will be a change of the direction at some point. This category will differentiate if the relation direction of the second premise is the same as the model building direction. The model building direction describes, at which side of the model a new object is added. If a new object is added to the left of the model, the model building direction is left. If the relation in the second premise is the same as the model building direction for the new object, the directions will be different for the third premise and vice versa.
For problem type one a similar category is possible. Since these problems are continuous, the premise relation direction can be the same as the model building direction for all premises or none of them. This is the way the problems are separated in this category.

All three premises that are used for the model have the same relation. The question premise can have a different relation than the other premises. This category can be applied to every problem type since all problems have this characteristic. The problems are split up by whether the relations of all four premises are the same or not.

The next possible category is just about the question and has, therefore, no connec-

tion to the problem type. This category can be used for all problems. The problems are separated by the objects contained in the question premise. There is a category for all problems with all question premises containing A and D. The other category contains problems with A and B in the question premise. The latter can be answered without even building a mental representation, for example by the verbal memory.
Based on these options, a few categorizations to test with the data were made.

The following is one of the categorizations that resulted from the analysis of the problems of the "premise order" experiment. This is only an example, so the categorization is not complete but simple.
The first category is always the type of problem. As earlier explained, the problems are already roughly categorized by these types of problems. The second category that is used is the model building process.

For every type of problem, there is a different categorization. In the problem type one, the problems with the same relation direction and model building direction will be categorized. The relation direction that a problem has will be determined by the three premises of the task, not by the relation in the question. In the problem type two, the problems are separated by in which premise the model building direction is different from the relation direction, as explained previously. In problem type 3, the problem distinction will be made by the objects in the first premise as explained above.

The categorization with these two categories is shown in figure 7. For each category, two example problems are given. The premises are represented like this example: "object1 relation object2". The question premise is the fourth premise and separated by a vertical line.
To illustrate the categorization, the problem containing the premises C L D, B L C and A L B with the question premise A L D will be categorized. Since the problem starts with C and D and then two insertions follow in the model building process, the problem is of type one. The relation in the task premises is left, so the relation direction for the

problem is left as well. The model building direction is left because B and A will be added to the model from the left side. The task relation and the building direction are the same and therefore this problem is in the "task relation == building direction" category.
Another example is the problem with the premises C R B, D R C, and B R A. This problem has C and B in the first premise, so it belongs to problems of type two. In the second premise, the D will be inserted into the model. The relation of the premises is right. The D was added to the right side of the model, so the model building direction is right. The model building direction and the relation of the premises are the same, which means the change of the building direction has to be in the last premise. In the last premise, the A is added to the left side of the model, but the relation in the premise is right. The relation and the model building direction are different in the third premise.

To find out how good a categorization is, it has to be applied and the resulting categories need to be evaluated with problems from the experimental data. The categorization will assign each problem into a specific category. The problems in each category can be evaluated with regard to the correctness of the participants' answers. For each participant, each category has a precision value which represents the number of correct answers given by the participant for the specified problems in the category. If there are a lot of participants, where the categorization results in categories with mostly correct or incorrect answers, the categorization might be suited for this experiment. A good category can separate the correct and incorrect answered problems.
A small helper program is used in order to check how many problems of each category are correct and incorrect. The program takes an assignment of all problems into categories as input. It then computes with a specified participant for each of these categories, how many of the answers are correct. The output of the program will be the ratios of correct answers in each category.
These results can be analyzed to find out if there are any patterns that can be observed over multiple participants. The problem with such precision values is that it is hard to decide how big the difference between the two categories has to be, in order to actually

find a difference in the problem that could have caused this precision gap. Sometimes, a participant gives a wrong answer to a problem because of lack of concentration or other reasons. If every category only has a few problems, one wrong answer has a big impact on the evaluation. This is the reason why the judgment was very strict. The precision gap between the two categories has to be at least 0.75 to be classified as a difference based on the categories.

From the analysis, the following categorization resulted. For each problem type, there are two categorizations. In total there are four subcategories for each problem type category. For problem type one, the problems are categorized by the model building direction in comparison to the relation direction. One of the subcategories contains all the problems, where the two directions are the same and the other one holds all the problems where the directions are different. The next categorization is made through the first premise. One subcategory includes all the problems that contain A/B in the first premise. The other subcategory holds D/C in the first category.

The most promising categorization for the problem types two and three appeared to be almost the same. The first categorization is the model building direction. This will be decided with the elements in the first premise for problems of type three. Type two problems always start with C and B in the first premise, so the objects in the second premise will decide in which direction the model is built first. The next categorization is about the relation of the task premises and the question premises. The problems of both types will be filtered by the equality of those two relations.

The complete categorization is shown in figure 8. In most categories, at least 10 out of 32 participants allow for individualizations. This means that there are clear differences between at least two categories, so a conclusion on the problem-solving ability of the respective participant could be made. In the program, an individualization could be triggered for the respective problem category.

5.1.2 Implementation of the categorization

This approach is also implemented in the CCOBRA framework. The spatial model with categories is called from the *predict* function, the *adapt* function will be used in a similar manner as in the parameter assignments approach. The individualizations can be activated in the same way as the parameter assignments via a class variable in the models. Since the *pretrain* function cannot be used to train the model for the actual participants in the evaluation, the *adapt* function is used for the training.

Before the *adapt* function is called, the predict function is executed. In the predict function, activated individualizations can be processed to take effect in the processing of the model. Since there are different individualizations for the three problem types, the predict function also checks to which problem type the given problem belongs. If the first premise contains B and C, the problem has to be of type two. If the last premise contains B and C, it is a problem of type three. Problems of type one are all problems that do not satisfy the conditions for problem type two and three.
The problem type and the answer that the baseline model has given are stored for later use in the *adapt* function. Every problem and the corresponding answers from the participant and the baseline are stored, in the *adapt* function, into separate lists for every problem type.
If the already processed problems reach a certain number, the actual categorization is executed, like the parameter search in the approach before. In a later version, the categorization is processed at specified intervals. This can make the predictions more precise if the rest of the *adapt* function is adapted to the intervals.
The distribution of the problem types in this specified amount of problems can differ, so the *adapt* function checks how many problems of each type are stored in the corresponding lists. If enough problems of a certain type are stored, the categorization for this problem type will be processed. If there are just a few problems of a problem type, the categorization is not accurate in predicting the answers given in future problems. However, since

the *adapt* function is used for training, the number of problems collected before doing the categorization should be kept to a minimum to achieve the best possible results. Because of the relatively low amount of training data, only two categories are used. The problem type and one other category. It is still possible to use various categorizations, but not at once as it was intended previously.

The *adapt* function will then call the *categorization* function for the respective problem type. Inside these functions, the problems are split up into two lists. For example, if the categorization is about the task relation and the question relation, every problem will be checked, whether the two relations are the same or not and then put into a corresponding list.

After the two category lists are computed, the precision of each category is computed with the answers of the baseline model. The precision for all problems of the problem type is computed as well.

With these precision values for the categories, the decision will be made whether an individualization parameter should be activated or not. For each problem type, only one parameter can be activated at a time. The reason for this is the inference of the individualizations. One individualization can revert the changes from another one.

If the difference of the precision between the two categories is bigger than a specified threshold, the corresponding individualization parameter is activated. If the overall precision of a problem type is below a certain threshold, a special individualization is activated, which will invert the answers given from the model. This will lead to a wrong answer with respect to the baseline model. This is a way to simulate that the participant generally gives the wrong answers for some reason.

In the predict function, the individualization activation parameters for each problem type are checked and then the spatial model with categorizations is called with the specified parameter assignment for each problem. The predict function will check to which category the currently processed problem belongs, to only use the individualizations for the corresponding category. Since the individualizations are only for specific problems, they have to be activated for only these problems in order to achieve a better result.

Figure 9 shows the general categorization procedure inside of the *adapt* function. The accumulated data in the figure are previously prepared in the *adapt* function as well. The described process is only triggered when enough problems are accumulated from previous predictions.

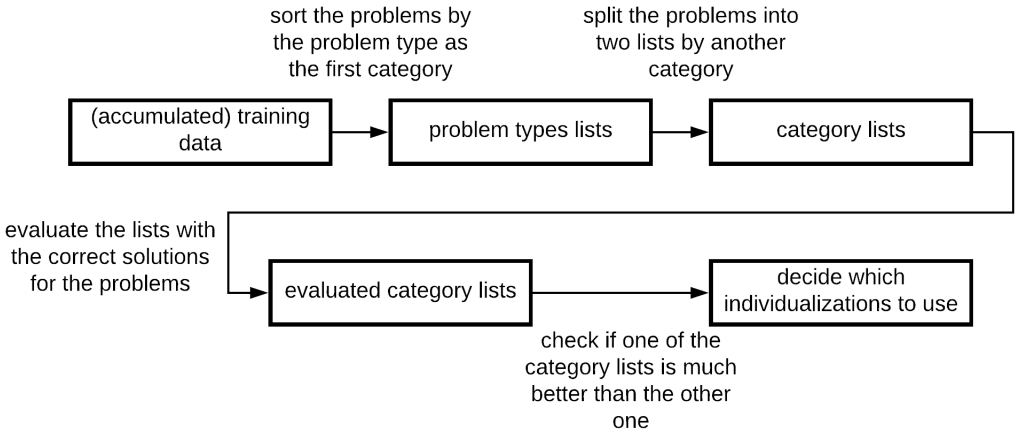

Figure 9: This figure shows the general categorization procedure for the CCOBRA model of the categorization approach. With a specified amount of accumulated training data separated into problem types, the program will categorize the problems in the problem type lists. After this each category will be evaluated to decide which individualization should be activated or not.

5.1.3 Results of the categorization

The evaluation is made with the CCOBRA framework. The baseline and the categorization model are compared with respect to the general precision result from the evaluation by the framework. Since only the data of the "premise order" experiment is used, only the results for this experiment can be compared.

Despite the fact that no real training data is used and the majority of the evaluation data is actually used as training data, the categorization model could achieve a better overall precision than the baseline of the spatial model. As figure 10 shows, the categories model is better than the baseline with a precision of 0.755 compared to 0.749 from the baseline.

In the evaluation, 27 problems were accumulated to do the categorization. The categorization threshold for the separate problem types was nine. So there have to be at least nine problems of a problem type to do the categorization for this specific problem type. The threshold, for how much better one category has to be than the other one is set to 0.6. This means that one category has to have an at least 0.6 higher precision value to trigger an individualization. If the overall precision (with respect to the correct solution) of the problems of a certain type is below 0.4, the model will give the wrong answer for all problems of this type.

The way how the model learns about the participant is not optimal. For this "premise order" benchmark, 27 out of 45 problems are used to find good categories. The fact that it is still possible to achieve a better result than the baseline with this setup, shows that there is potential for this categorization approach, to reach even more precision. However, it is also worth to note that mostly the parameters set with the overall precision of a problem type lead to better results. The real categories are activated as well, but the effect on the result is very small compared to the general precision parameters. The categories model is not using the *pretrain* function to learn something about the usual answer patterns. Also, there is no adaption for the activated individualizations. This is why so many problems are needed to activate meaningful individualizations.

In a later version, the model uses the categorization more often. The thresholds are not changed, except the threshold for the overall precision of a problem type is lowered to 0.3. The number of problems accumulated until the first categorization is lower with 25 problems. The biggest change to the *adapt* function is that the categorization is processed after each problem, starting from 25 problems. With this change to the *adapt* function, a slightly better result could be achieved.

To summarize the categories approach, it is worth to further investigate how the learning phase of the model can be implemented in a better way. The results are better than the baseline, even when the actual categorization individualizations are only activated for 18 problems out of 45 problems in total.

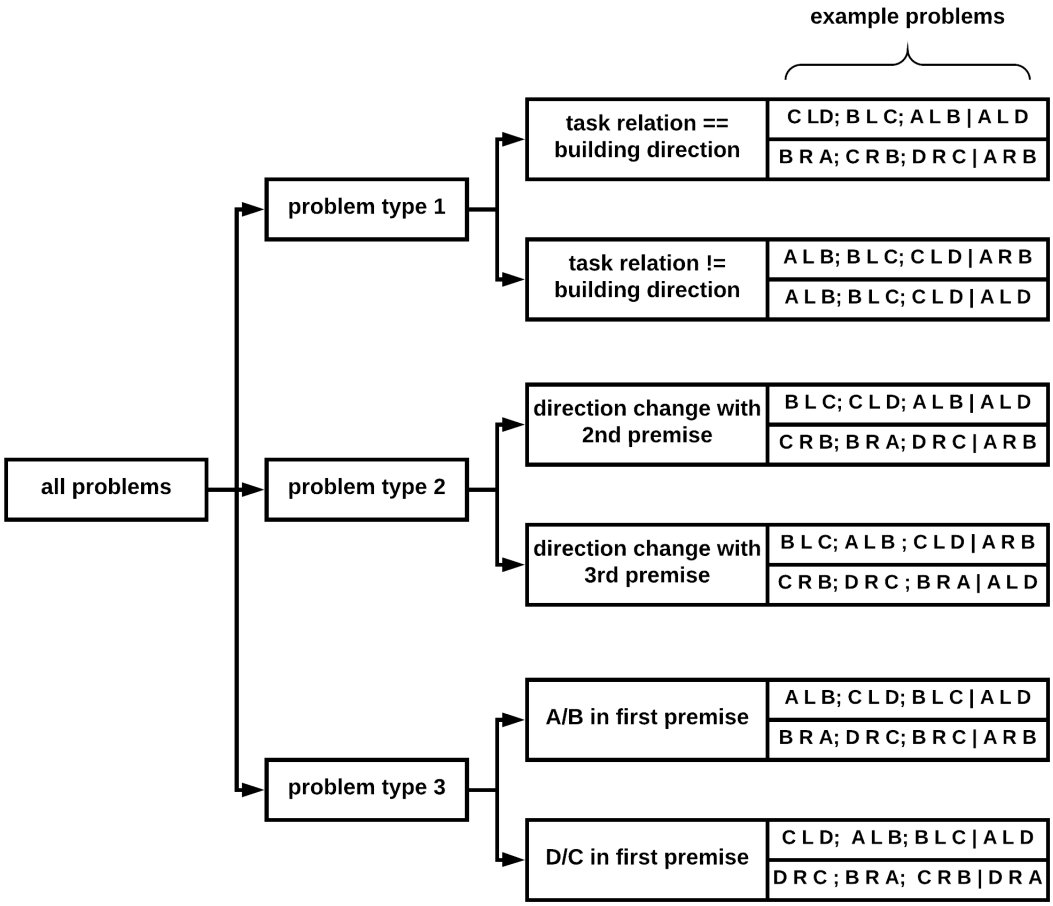

Figure 7: The graph shows how the six categories result from the different subcategories of the problem type categories. Next to each subcategory description are two example problems given. The three task premises are separated by a semicolon, while the question premise at the end is separated by a vertical bar. The relations "left" and "right" are notated as "L" and "R".

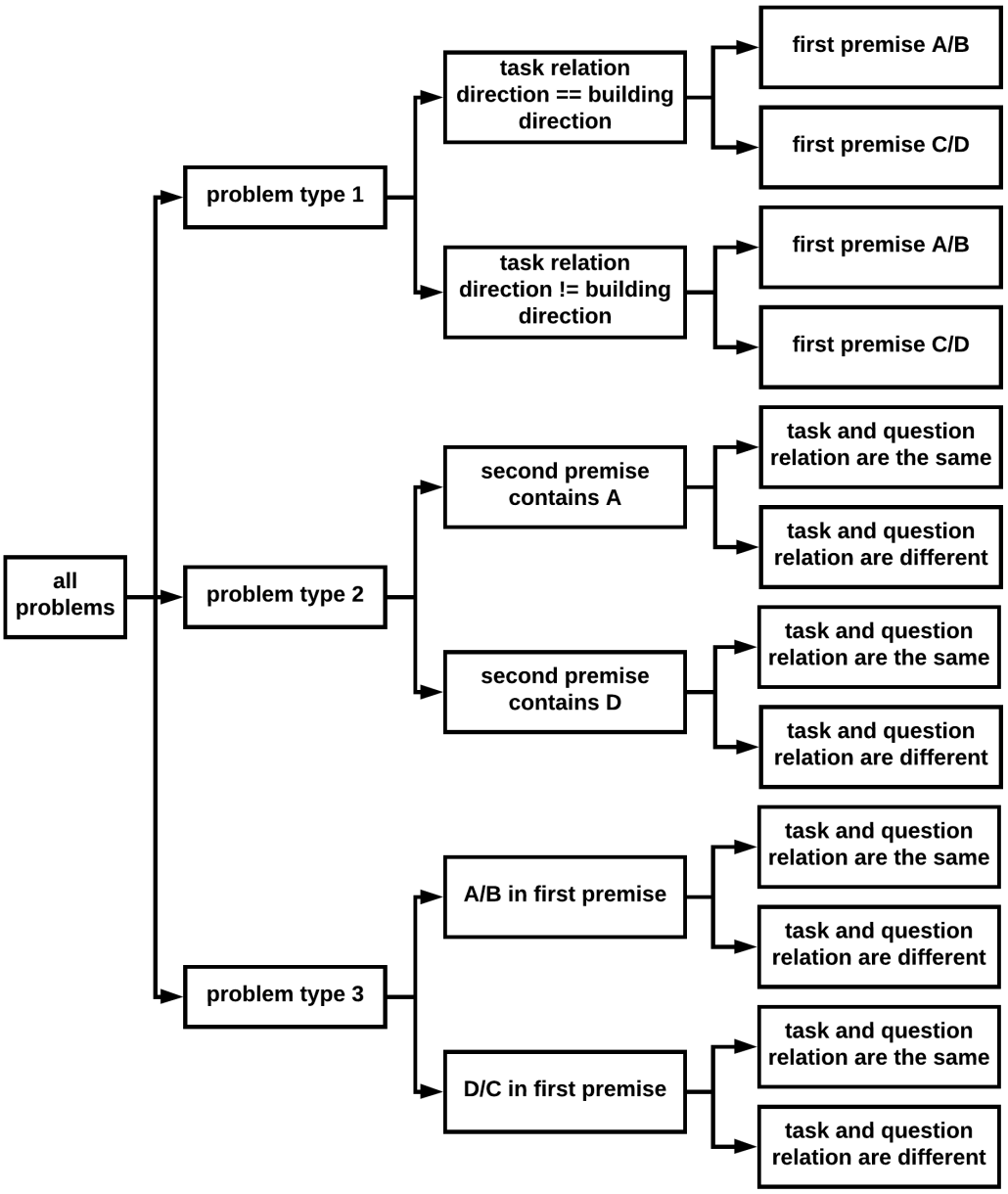

Figure 8: The most promising categorization. The problems are separated by two categorizations in each problem type category. This results in 12 subcategories in total. Every subcategory contains about four problems, this means the problems are overall well balanced and the subcategories can be evaluated in the same manner.

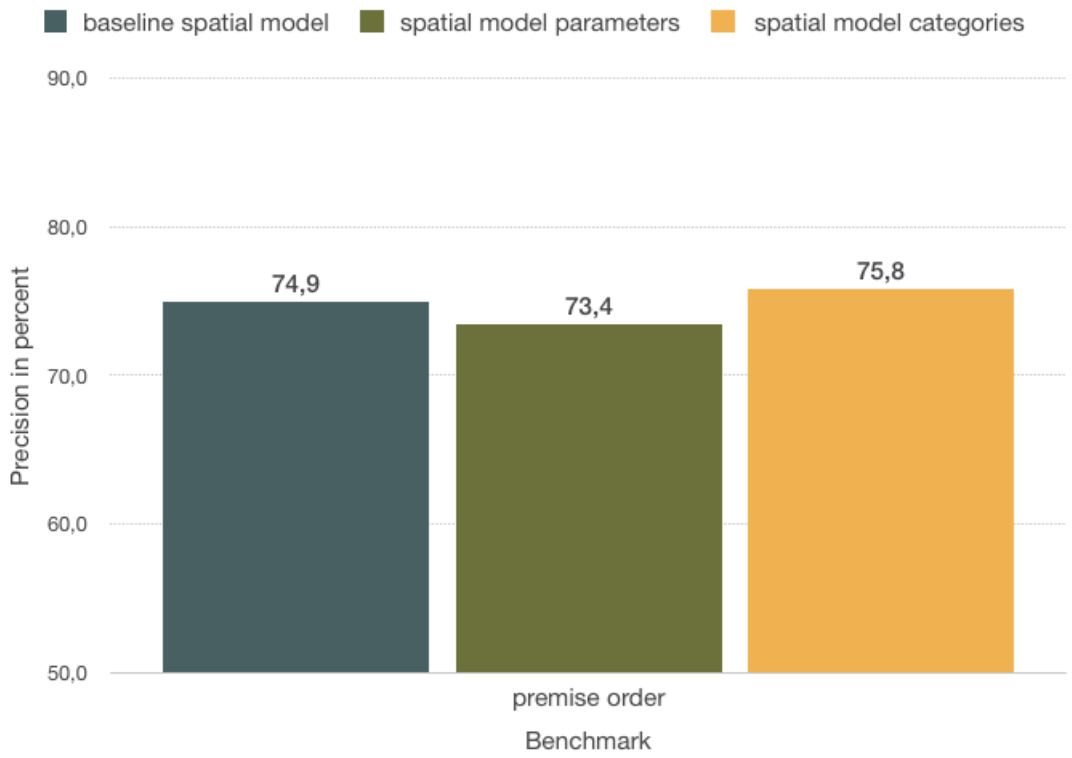

Figure 10: This figure shows the results of the categorization model compared to the baseline results and the parameter models. The categorization is only tested for the "premise order" benchmark, so the comparison with other benchmarks is not possible.

6 Third approach

This chapter is about the third and last approach in this thesis. In the previous approaches, the CCOBRA framework is not utilized in an optimal way. A good model should make use of the *pretrain* function, since the *pretrain* data contains a lot of information about the general behavior of the participants in an experiment. Another aspect to be improved from the previous approaches is the usage of the *adapt* function in the models. The model should be able to dynamically adapt the individualizations used in the model for future problems.
Finally, the individualizations are probably not independent from each other, so this approach will go one step back and check how the separated individualizations behave.

6.1 General procedure of the approach

The goal of this approach is to find out which individualization can achieve a better result and if it is possible to combine them to accomplish even better results. In order to do this, separate CCOBRA models are created for each individualization used in the parameter approach. The individualizations for misinterpreting a premise and for incorporating a premise incorrectly into the model are semantically the same, so the individualizations for the incorporation of the premises are omitted for the model building process.

6.1.1 Separate individualizations

With the five models for misinterpreting premise 1-3, verbal memory usage and guessing usage, it was tested how they behave in the CCOBRA framework with the "premise order" benchmark. The CCOBRA models always use the parameter assignment of the corresponding individualization for each prediction. Since the models are only different in the parameters they use for the prediction, they are comparable. The individualizations are also completely independent.

The *adapt* and *pretrain* functions are not used in this first test. For this test, only the "premise order" data is used. The individualizations are created based on the results of this experiment, so they should work at least in this data set. The other experiments are used later to check how generalized the individualizations are. If the individualizations can also achieve good results in other, different experiments, the individualizations are generally usable and thus can show that such factors have an impact on human reasoners.

The "premise order" data is used for all five models to get the precision results from the CCOBRA evaluation. In all models, every prediction is made with the corresponding individualization. The *adapt* and *pretrain* functions are not used. As a result, all of the models have a worse precision than the baseline, as shown in figure 11. However, it is unexpected to see that the "guessing" model has the best results within the individualization models. In contrast to the baseline spatial model, this "guessing" model could predict every single answer from one of the participants. No individualization could surpass the baseline precision like this.

6.1.2 Adapt function for the individualizations

The next step is the implementation of the *adapt* function for all models. The *adapt* function is the same for all of the models, in order to make them comparable. The only thing that is different is the configuration of the thresholds and the used individualization. The models start with the baseline parameters assignment. After each prediction in the

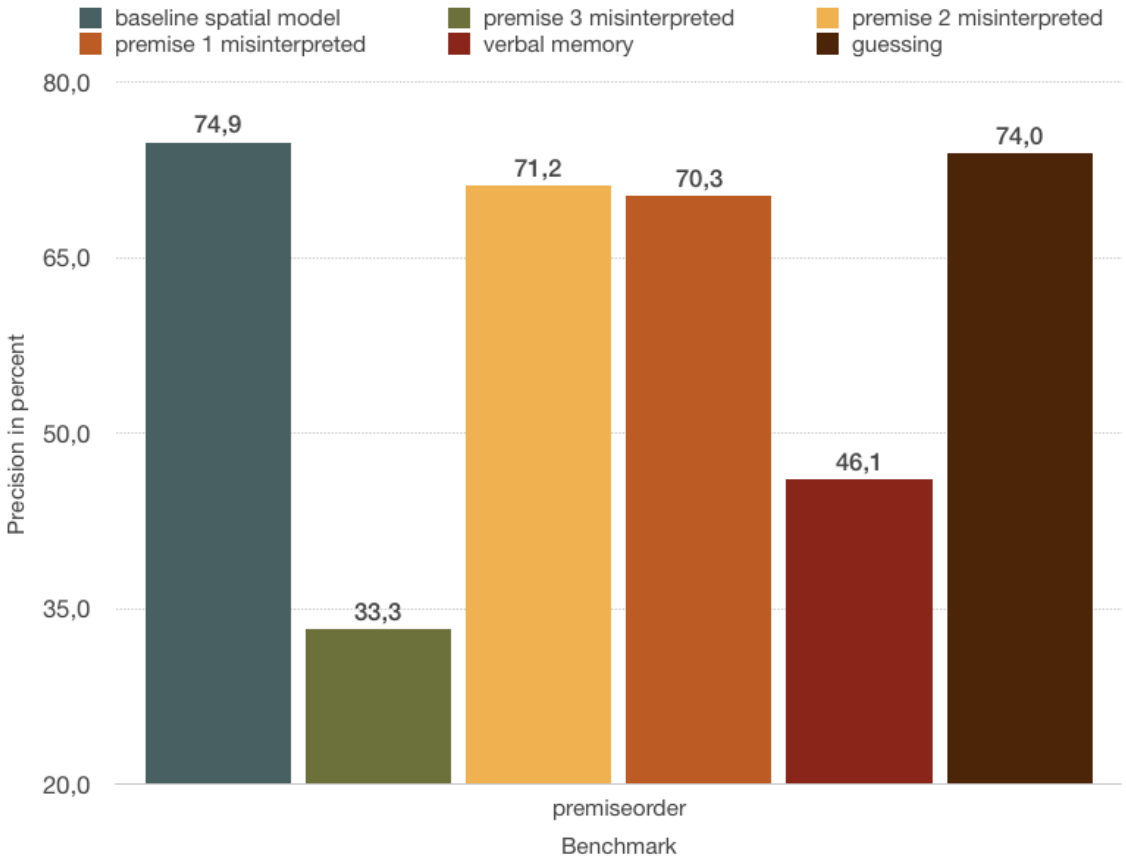

Figure 11: This diagram shows the results of the separate individualization models. The corresponding individualizations are always activated. The *adapt* and *pretrain* function are not used. None of the models is as precise as the baseline model. The "guessing" model could predict one person completely correct.

evaluation, the *adapt* function is called. The predict function stores the previous answer and problem like in the other approaches before.

If the predicted answer from the model differs from the participants' answer, the *adapt* function will check, if the model can make a better prediction with the activated individualization. In this case, the *adapt* function then makes a prediction with the corresponding activated individualization. If the answer is the same as the answer from the actual participant, the individualization has a good influence on the prediction. If an individualization is good enough at predicting the actual results, it is activated. In order to decide whether

to activate a certain individualization or not, each individualization has a rating variable. The rating value will have an initial value of 0.0 at the start of the evaluation. If the previously wrong answer can be made correct with respect to the participants' answer, the rating will be increased by a specified gain value. If not, the rating will be decreased by a negative gain value. In the tested models the gain value is between 0.1 and 0.2 usually. If the rating is above a certain threshold, the individualization is activated. If the rating falls below this threshold, it is deactivated, because it is not precise enough anymore.

If the participants' answer can be predicted correctly by the model, the *adapt* function has no effect, since the current prediction is good for this problem. The configuration of a model is composed of the activation threshold for the individualization and the gain values for the rating variable. The gain values can be different for increasing and decreasing the rating.

For each model, the configuration can be changed in order to get the best result possible. The models with *adapt* functions all have better results. The model for misinterpreting the third premise and for the verbal memory usage have better precision than the baseline model, as shown in figure 12. The "verbal memory" model configuration has a bigger gain value and a lower threshold than the "premise misinterpreted model" configurations. This means this individualization corrects less problem predictions than the "premise misinterpreted model". For each individualization, a good configuration has to be found. This configuration is optimized manually to find out if there can be a better result with these individualizations. The optimization for the configuration is not very complex since the gain values and the thresholds are usually between 0.7 and 1.5. So for each of the values of a configuration, various numbers were tested. By observing the frequency of increases and decreases of the rating value, a good configuration can be easily found. If the rating of individualization is not increased very often, the threshold and gain value can be adapted to activate the individualization. With this method, individualizations can be pushed to be activated more or less often.

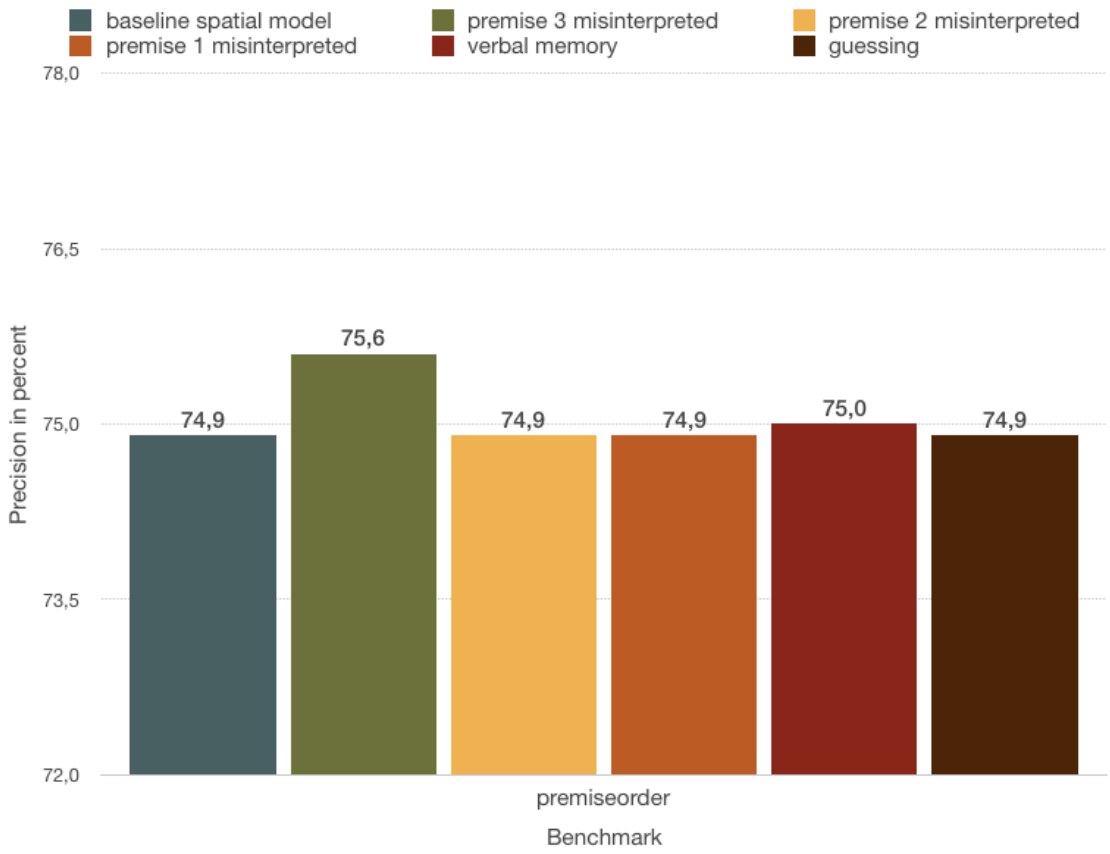

Figure 12: This figure shows the results of the models with the *adapt* function. With the use of adapt, all models are more precise. The "premise three misinterpreted" and "verbal memory" models are even better than the baseline precision.

6.1.3 Pretrain function for the individualizations

In the next step, the *pretrain* function is additionally included. In the *pretrain* function, training data of a few participants are given. For each of those participant data sets, the *pretrain* function will simulate a normal evaluation with a custom *adapt* function. So for each problem in the problem set of a participant, predict will be executed before an alternative version of *adapt* is used. This custom *adapt* function counts the number of answers that are corrected with respect to the participants' answer, instead of changing a rating value. Like described, the function computes the ratio of corrected answers over the whole data set, which is provided by the framework. This value, the prior, represents

in percent, how many problems can be corrected. This number is normalized for each model to be comparable. In the *adapt* function, the prior will be added to the normal rating value to get the total rating for this individualization. The activation condition for each individualization is represented with: $((multiplier * prior) + rating) \geq threshold$
In each model, the prior can be multiplied by any number to use it for the activation decision. With this, the configuration for an individualization model is composed of the activation threshold, the prior multiplier, and the positive and negative gain value.
The model for misinterpreting the third premise gained a lot on precision and now reaches 76% as shown in figure 13. This is a significant difference to the baseline performance. Only the model for misinterpreting the first premise is not better than the baseline. This shows that the individualizations generally make the model more precise if the model is tuned for this parameter activation configuration.

Later, three new individualization models were added. Alternative ways to perform the "model insert" and "model combine" operations in the spatial model are implemented. The third individualization named "modify initial model", enables the modification of the initial model built by the program. With this third individualization, the model tries to falsify the conclusion made with the initial model. The way how the individualizations are realized in the spatial model is described in chapter 4. These individualization models were directly implemented with the *pretrain* and *adapt* function. These new individualizations, especially the alternative insertion and combination, are expected to be much better in combination with one or more of the "premise misinterpreted" models. However, the first step is to test how good each individualization is separately. As shown in figure 14, the results of these two models are better than the baseline, so there is potential for these individualizations.

6.1.4 Combination of the individualizations

In the next step, the single individualizations are combined to get an even better prediction precision. In a combination model, both individualizations have their own prior and rating

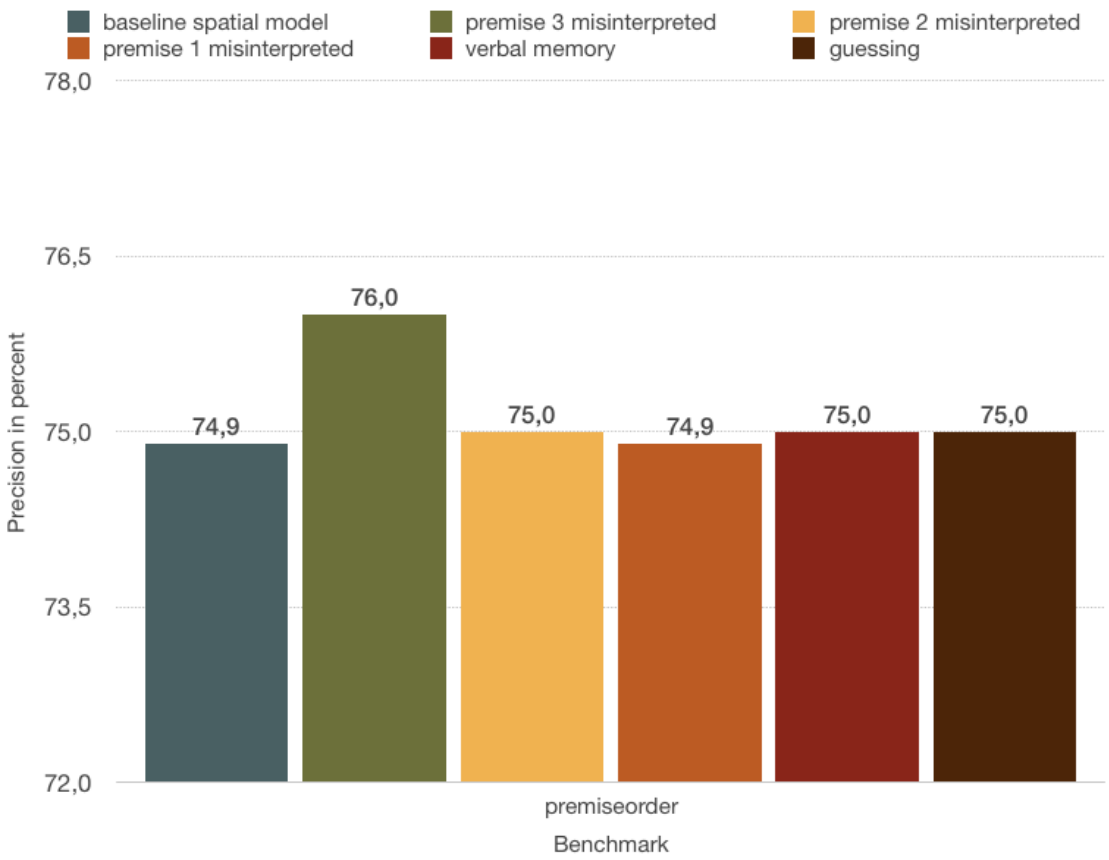

Figure 13: This diagram shows the results of the complete CCOBRA models for each of the individualizations. Most of the models are better than the baseline. "premise three misinterpreted" is again the most precise model.

variables. Also, the decision to activate the individualizations is independent as a default. There is a chance that the two individualizations have negative side effects on each other, so it is possible to only use one of the individualizations to be independently activated. The two individualizations have a rating and prior for the combination, where they are activated at the same time. If the prior and the rating is above a certain threshold, both individualizations are activated. If not, one of the individualizations is deactivated. This is important for the case when both individualizations are separately activated. If this happens, the result might be worse than with only one active individualization. For a combination model, the configuration for the combination variables has to be optimized.

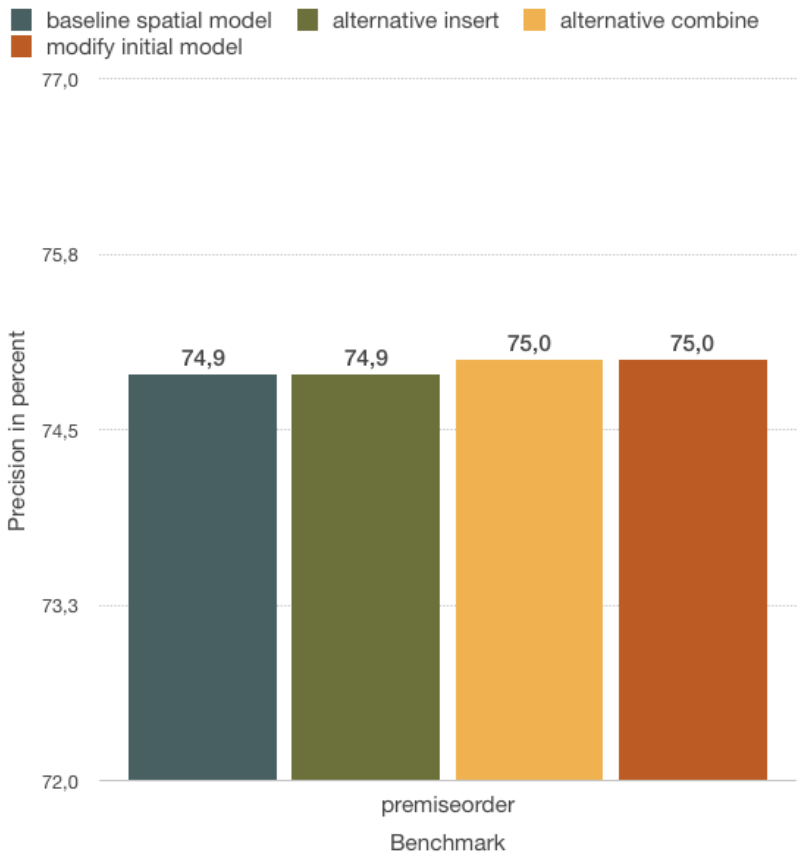

Figure 14: This diagram shows the results for the complete CCOBRA models for the three new individualizations. Two of the new models have better results than the baseline model. These individualizations are expected to be much more precise when combined with other individualizations.

Another approach for the combination is to only activate the individualizations separately, but to give the ratings a certain boost if the combination can correct an answer. This approach was tested, however, there are too many side effects from the individualizations, so the results were worse than the baseline. Therefore the approach with the combination and separated rating and prior variables is tested.

Since the "premise three misinterpreted" model has the best result in the separate evaluation, it is preferred for combinations.

The first combination that is implemented uses the "premise three misinterpreted" in-

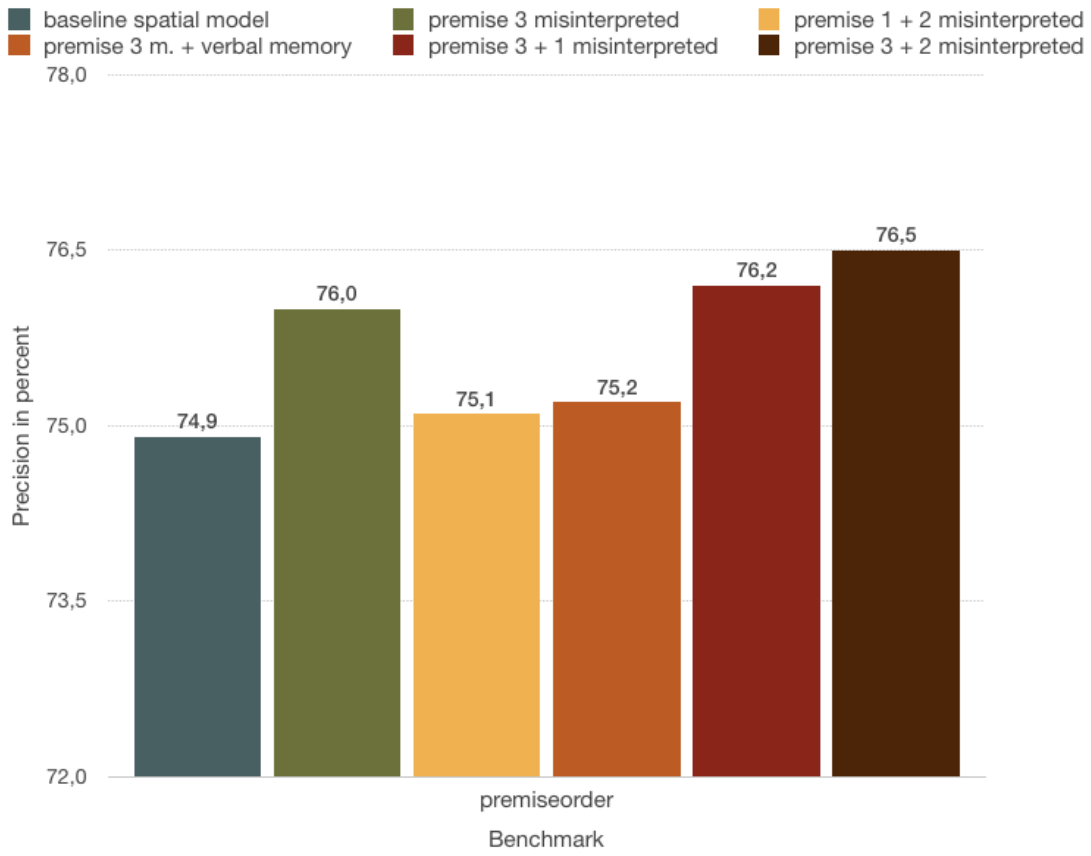

Figure 15: This figure shows the results of the combinations of the individualizations. As it can be seen, all of the combinations surpassed the baseline, but only two of them were better than the "premise three misinterpreted" model. These results show that the combination is not trivial, but can lead to better results. "misinterpreted" is shortened to "m." for readability.

dividualization and the "verbal memory" individualization. The result is better than the baseline, but not as good as the "premise three misinterpreted" model, as shown in figure 15. The threshold is really low with 0.4 and gain values of 0.1. This means that the combination needs to correct a wrong answer from the model four times in a row, without a prior. Usually, the threshold is much higher. The results are not as good as expected. The reason for this could be the "verbal memory" individualization. This individualization leads to an answer without even creating a model. This is why such individualizations cannot be used together at the same time.

The next combination is the "premise three misinterpreted" model together with the "premise two misinterpreted" model. These individualizations can both be used at the same time. The combination can correct a lot of the wrong answers. The prior value is even better than the one from the "premise three misinterpreted" model. The independent activation is not used for "premise two misinterpreted" since the precision gain through this individualization is not much. With only the "third premise misinterpreted" individualization and the combination individualization in the model, the result is better than the separate individualization results. This shows that a combination can indeed make use of both of the individualizations, but it is not trivial to keep the individualizations from interfering with each other. The combination has a threshold of 1.2 because the rating is increased very often. This means that these individualizations can be used in the data frequently.
"premise one misinterpreted" did not lead to a better result, but the combination of the "premises three misinterpreted" and "premise one misinterpreted" can also potentially explain more wrong answers from the data. Similar to the previous combination, the result is better than the "premise three misinterpreted" model. Again, "premise three misinterpreted" is preferred since the individual results are better for this model, so the "premise one misinterpreted" individualization will not be activated independently in the combination model.

The combination of the misinterpretation of premise one and two also leads to a better result than the baseline. This time, there is no preferred individualization, since the separate results are not very different. Because of this, the two individualizations are both activated independently. The prior values are very low for the separate individualizations. The combination prior is bigger, this means that the combination again is able to correct answers that were previously not possible to be corrected.

These results show that it is possible to combine two different individualizations to

be able to predict more of the participants' answers correctly. The individualizations to combine need to be chosen with care, since there can be interferences between the individualizations.

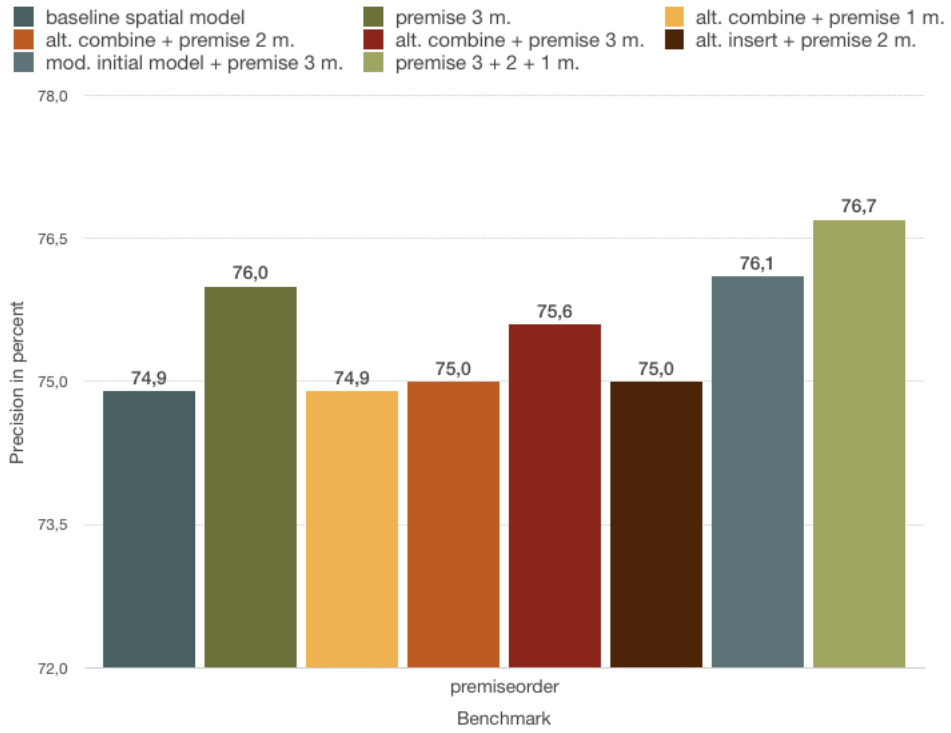

Figure 16: This figure shows the results of the combinations with the new individualizations. "misinterpreted" is shortened to "m." for readability. The results are worse than expected, as only one model of the combinations of two individualizations could be merely better than the "premise three misinterpreted" model. The combination model with all three "premise misinterpreted" individualizations merged together has the best result. The combination of the three individualizations is beneficial for the prediction precision of the model.

The next combination that is implemented uses the "alternative combine" individualization. This individualization leads to several new models to be created when combined with a premise misinterpretation individualization. This is why the combination is tested with all three "premise misinterpreted" individualizations. The combinations of the individualizations do surprisingly not lead to more answers being corrected. The results are only

merely better than the baseline, as shown in figure 16. In all three combination models, the threshold is lowered to even make it possible for the combined individualizations to be activated. The quality of the result decreases if the combined individualizations are activated frequently. In all three combination models, the result could not be improved compared to the separate models. This means that the "alternative combine" is not a good individualization for this experiment. In other experiments, the participants probably build models like this.

The "alternative insert" individualization is combined with "premise two misinterpreted" individualization. The alternative insertion has no effect if this individualization is activated alone on the "premise order" data. There is no premise that can be interpreted in two ways regarding the insert operation. This is why the alternative insertion is combined with the "premise two misinterpreted" individualization. If the second premise leads to a different model, the third premise can lead to an alternative insertion. The result is not as good as expected. The combination seems not to increase the number of corrected answers. Therefore the combination of the individualizations provides no explanation for participant errors in the experimental data. Maybe the participants in the "premise order" experiment never inserted new objects like this individualization. Another possibility is that the participants may never misinterpret a premise and then insert the object in an alternative way. It will be interesting to see, whether this combination will have better results with other data sets.

Another combination merges the "modify initial model" and "premise three misinterpreted" models. This combination leads to a slightly better result than the "premise three misinterpreted" model, as shown in figure 16. However, the combination needs a very low threshold of 0.3 to be even activated. The prior value was also very low. Since the result was still better than the separated models, the combination is probably used in some way by human reasoners.
The results for the combinations with the new individualizations were not as good as

expected. The individualizations are probably too complex for the given problems.

Since the combination of two individualizations leads to better results, the combination of three individualizations might result in even better ones. The combination of "premise three misinterpreted" with "premise two misinterpreted" and with "premise one misinterpreted" has the best result for the "premise order" experiment, as shown in figure 15. Therefore, a combination of all three individualizations is implemented. The combination for the three individualizations has an own rating and prior, in accordance with the previous combinations. The combination with the three individualizations has the highest priority, so it is activated if the rating is good enough. If the rating is not higher than the threshold, the two other combinations are checked, whether they should be activated or not. This new model could achieve a better result than the two other combination models, so this model proves that the precision can improve with such a combination as well. However, the gain in precision is not as much as expected. With more individualizations to be possibly activated, more interactions between those can occur. This means that it will become more and more difficult to prevent interactions if more individualizations or combinations of individualizations are used. In this model, there are already three different ways for a single individualization to be activated in the evaluation. Since the individualizations need to be deactivated as well, the expenditure of more combinations might not outweigh the gain in precision.

6.1.5 Results

The results of the models for the "premise order" benchmark show that the individualizations proposed in this work lead to better results than the original spatial model. All results are shown in table 1. Combinations of individualizations lead to a better result than single individualizations. The gain in prediction precision is only 1.8% for the best

combination model, which is not a big gain in precision compared to the baseline model. It is unexpected that the baseline has a precision of 74.9% already. This indicates that the participants in the experiment did fairly well in solving the problems since the baseline solves all problems correctly. With an already precise baseline, the potential precision gains of the individualizations are limited.

A possible reason for this low gain of only about two percent, might be the way the individualizations are activated. Before an individualization is activated, the corresponding rating has to be increased. This happens if the individualization corrected a previously incorrect prediction by the model. With the limited number of problems in the data set, depending on the activation threshold, a lot of problems are already processed at the time when the individualization is activated. This decreases the possible gains by the individualizations.

The other benchmarks are tested with all individualization models as well. The results are generally not better than the baseline. The configuration of the models needs to be optimized again since the problem and the results of the participants are different. The prior values for most of the individualization and combination models are very low. This means that the corresponding individualization could not correct the results in the pretraining very often. This is already an indicator of the effectiveness of the individualization. In some cases, the prior value was 0.0, which translates to a never increased rating in the evaluation. For these models, the corresponding individualization has no benefit for the tested experimental results. For other models, the activated individualizations worsen the results. In table 1, it can be observed that there are some models, which are as good as the baseline and some models which are not as good as the baseline. In some of the models, the individualizations are not activated in the evaluation. This way the individualized model is the same as the baseline model. Various configurations were tested and it was not possible to achieve results better than the baseline, with a few exceptions for the "single choice" experiment. This shows that it is possible to beat the baseline with individualized models. However, the general trend in comparison to the "premise order" experiment shows that the individualizations implemented in this approach are not

beneficial for other experiments. This leads to the conclusion that the individualizations do not work in general. The individualizations are derived from one specific experiment and thus are limited to a certain problem type.

In general, this approach showed that the individualizations can be used to achieve better results than the original model. However, when combining models, the individualizations that are coupled need to be picked carefully in order to not trigger unwanted side effects. The more individualizations or combinations are used, the harder it gets to prevent interactions between them. The benchmark tests for the other experiments show that the individualizations need to be adapted for other experiments in order to get the best possible results.

Table 1: Complete results for all implemented models on all benchmarks

	Benchmark name			
Model name	premise order	figural	verification	single choice
random model	50.0	52.0	52.0	12.0
transitive closure	74.1	83.0	58.0	22.0
baseline spatial model [12]	74.9	83.0	62.0	63.2
spatial model parameters	73.4	83.0	61.8	63.2
spatial model categories	75.8	no result	no result	no result
premise 3 misinterpreted	76.0	83.0	62.0	63.2
premise 2 misinterpreted	75.0	82.7	62.0	63.2
premise 1 misinterpreted	74.9	83.0	62.0	63.2
verbal memory	75.0	83.0	62.0	63.2
guessing	75.0	83.0	62.2(varies)	63.2
alternative insert	74.9	83.0	62.0	63.4
modify initial model	75.0	83.0	62.0	63.2
alternative combine	75.0	83.0	62.0	63.2
The following models are combination models which use two or more individualizations. "misinterpreted" is shortened to "m." for readability.				
premise 1 and 2 m.	75.1	82,7	62.0	63.2
premise 3 m. and verb. mem.	75.2	83.0	62.0	63.2
premise 3 and 1 m.	76.2	83.0	62.0	63.2
premise 3 and 2 m.	76.5	83.0	62.0	63.2
premise 1 m. and alt. combine	74.9	83.0	62.0	63.2
premise 2 m. and alt. combine	75.0	82.7	62.0	63.2
premise 3 m. and alt. combine	75.6	83.0	62.0	63.2
premise 2 m. and alt. insert	75.0	82.7	62.0	63.5
premise 3 m. and modify i. mod.	76.1	83.0	62.0	63.2
premise 3, 2 and 1 m.	76.7	83.0	62.0	63.2

This table shows the results of all tested models including the baseline models of the CCOBRA framework. Each row represents one of the models and the results for each benchmark in the framework. The results are represented in precision values in percent. The results are colored in olive if the result is better than the baseline spatial model. Results that are worse than the baseline are colored in orange. The guessing model results vary, so the result is not always better than the baseline. The categorization model can only process problems from the "premise order" experiment. Therefore, there is no result for this model on the other benchmarks.

7 Conclusion

In conclusion, there are a few things to be learned from the three approaches. The first approach with all the individualizations being activated in all possible combinations showed that there are potential side effects, so the optimization with all eight individualizations is very difficult. This approach did not have better results than the original spatial model. The reason for this might be that the model does not change the parameter assignment dynamically. Also, through the parameter assignments, the individualizations were activated together. Another reason for the negative result could be the way in which the CCOBRA framework was used. By omitting the *pretrain* function and misusing the *adapt* function, a lot of potential was not used.

The categorization approach focused more on the distinction of when and which individualization should be activated, depending on the specific problem itself. The categories are made specifically for one experiment, so they cannot be used for other, different experimental data. Despite that, this approach is promising since the results were better than the baseline, even with the same unfavorable use of the framework for the evaluation. In this case, there were not many problems for which the activated categorization was used.

The third and last approach was addressing the problems of the first two approaches by splitting the individualizations into separate models. The procedure for the *pretrain* and *adapt* functions was made equal for all of these models. This ensures that a change in the result can only come from a different individualization and the activation configuration

of it. Several individualizations showed that in principle, it is possible to achieve better results with only one individualization. Various combinations of individualizations also demonstrate that the combination can achieve more precision for the prediction. It is worth to mention that the combination is not trivial and that there are often side effects, even when using only two individualizations at the same time. The activation of the combination of two individualizations will displace the usage of the separated individualizations. This way, a previously corrected problem might be predicted wrong by the model. There are more possible corrections, but there is also a risk of overlapping with the corrections made by the separate individualizations. The combination of more individualizations, as far as it was implemented, resulted in a precision gain. The usage of all of the proposed individualizations and their combinations will need sophisticated management of the individualizations in the model.

The test of the individualizations with the data of the other experiments leads to the conclusion that the individualizations in the way they are implemented, cannot be used in general for other experiments. The reason for this is that the individualizations are derived from the "premiseorder" experiment, and hence only have an effect in this experiment. The individualizations had a negative effect on the results when activated in the model. There are exceptions to this that show that in some experiments individualizations also had a positive effect. However, the general trend shows worse results than for the "premiseorder" experiment.

7.1 Future work

Further research on this topic can be conducted by combining more than three individualizations. This is not a trivial task, as the most promising individualizations with possibly no negative side effects have to be chosen for this. The way how the individualizations are actually combined in the model may also be worth to be investigated more.

The way how the configuration for each model is found can also be optimized. Values for the variables, found via manual testing, are currently used. The optimization is not difficult, but for more individualizations in a model, it is more complicated to do this. If more individualizations are used in a model, it would be good to automatize this within the model or in the framework.
In chapter 6, a possible reason for the low precision gains of the individualizations is described. In the framework, only a general pre-training is processed by the model. With more data from a specific person, the models could be trained for each participant. This will lead to a prior that fits the participant data much better than the prior processed with general data. With a more precise configuration of the models, the individualizations can be activated more often, depending on the participant. The way the thresholds of the models are set only allows for individualizations to be activated more often or less often for every participant. With a different prior for each participant, individualizations can be activated with less previous corrections of the prediction. This can allow for a more frequent usage of the individualizations and thus increases the potential of the models.
A promising next step could be to integrate a categorization into the combination models. This way, for each problem type, an own individualization or combination can be used. The categorization can also introduce the activation of new individualizations without interfering directly with the activation process of the other individualizations. Depending on the category, a specific set of parameters can be used for each prediction of the spatial model. This way the use of the individualizations can be adjusted more precisely to only use the individualizations when they are actually beneficial. Since the categorization has potential, judging from preliminary tests, it is worth to try out.

The individualizations could only achieve better results for one of the experimental data sets. It would be interesting to know, how the individualizations need to be changed to get better results for more than one specific experiment. In order to do this, a first step can be to derive individualizations from the other experimental data as well. The new individualizations can give new insights into why the individualizations from this

approach did not work for the other experiments. If similarities between the specific individualizations for the different experiments can be found, generally applicable individualizations can possibly be derived.

This work demonstrated that specific modifications for specific data can lead to better results. In general, more experiments need to be investigated to learn more about the utilization of individualizations into cognitive models.

8 Acknowledgments

First and foremost, I would like to thank professor Marco Ragni, my research supervisor for this thesis. I would like to offer my special thanks to professor Ragni as my adviser for the many in depth explanations and patient guidance for this work.

My special thanks are extended to Nicolas Riesterer and Daniel Brand for many valuable and constructive suggestions regarding the implementation. The quick and reliable feedback has been very much appreciated.

I would also like to thank to Paulina Friemann for the the experimental data.

I would like to express my appreciation to Thomas Leyh, Till Steinmann, Wolfgang Breu and Julia Mertesdorf for proofreading the thesis.

Finally, I would like to thank my parents and friends for their support and encouragement throughout my study.

Bibliography

[1] R. M. Byrne and P. N. Johnson-Laird, "Spatial Reasoning," *Journal of Memory and Language*, vol. 28, no. 5, pp. 564–575, 1989.

[2] P. N. Johnson-Laird, "Mental Models in Cognitive Science," *Cognitive Science*, vol. 4, no. 1, pp. 71–115, 1980.

[3] P. N. Johnson-Laird and R. M. J. Byrne, *Deduction*. Essays in cognitive psychology, Lawrence Erlbaum, 1991.

[4] G. P. Goodwin and P. N. Johnson-Laird, "Reasoning About Relations," *Psychological Review*, vol. 112, no. 2, pp. 468–493, 2005.

[5] P. N. Johnson-Laird, "Mental models and human reasoning," *Proceedings of the National Academy of Sciences*, vol. 107, no. 43, pp. 18243–18250, 2010.

[6] M. Ragni and M. Knauff, "A Theory and a Computational Model of Spatial Reasoning With Preferred Mental Models," *Psychological Review*, vol. 120, no. 3, pp. 561–588, 2013.

[7] M. Knauff, R. Rauh, and C. 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*, pp. 200–205, 1995.

[8] N. Riesterer and D. Brand, "CCOBRA Documentation." http://orca.informatik.uni-freiburg.de/orca_sylwebsite/orca/documentation/usage/tutorial.html#introduction-to-ccobra. Retrieved December 9, 2018.

[9] N. Riesterer and D. Brand, "Repository of the CCOBRA source code." https://github.com/CognitiveComputationLab/ccobra. Retrieved December 16, 2018.

[10] N. Riesterer and D. Brand, "CCOBRA Evaluation Website." http://orca.informatik.uni-freiburg.de/orca_sylwebsite/orca/index.html. Retrieved December 16, 2018.

[11] N. Riesterer, D. Brand, and P. Friemann, "Setting up the CCOBRA Framework." "http://www.cc.uni-freiburg.de/teaching/teaching_files/2018-ws-setup. Retrieved December 9, 2018.

[12] S. Khemlani, "Mental models webpage." "http://mentalmodels.princeton.edu/models/". Retrieved December 15, 2018.

[13] M. Knauff, R. Rauh, C. Schlieder, and G. Strube, "Mental models in Spatial Reasoning," in *Spatial Cognition*, pp. 267–292, 1998.

[14] Cognitive Computation Lab University of Freiburg, "Experiments." http://www.cc.uni-freiburg.de/experiments. Retrieved December 26, 2018.

[15] K. C. Klauer, C. Stahl, and E. Erdfelder, "The Abstract Selection Task: New Data and an Almost Comprehensive Model," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 33, no. 4, pp. 680–703, 2007.

List of Figures

List of Tables