Modelling the Accuracy Rates of Spatial Relational Reasoning Problems: An Analysis Facilitating ACT-R and PRISM Theory

Nicolas Valentin Turcas

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfilment of the requirements for the degree of Master of Cognitive Science

Department of Cognitive Science

Carleton University
Ottawa, Ontario

Abstract

Relational reasoning involves evaluating relations between representations. Spatial relational reasoning problems have long been used in psychology to study deductive inference abilities. Research incorporating such tasks has resulted in findings of a variety of effects and reasons as to why some problems are more difficult than others.

Computational accounts of the relational reasoning of spatial information offer valuable insights, such as how individuals may construct a mental model to infer conclusions and why some cognitive strategies might be preferred over others. However, many of these accounts either fail to incorporate or remain general to the impact of other factors affecting the difficulty of these reasoning problems, such as the effects of working memory errors.

This thesis aims to investigate how the different types of memory errors of omission and commission may be computationally modelled to provide a theoretically unifying account of qualitatively different cognitive reasoning processes and quantitatively different accuracy rates on experimental measurements of spatial relational reasoning problems. Our models demonstrate that modelling memory errors of omission and commission in Python ACT-R based on PRISM theory produces a similar negative relationship of a decrease in accuracy rates with each increase of premise and dimensionality complexity per question as found in the relational reasoning experimental literature.

Our results highlight the need for future modelling to consider individual differences in participant micro-strategy preferences, how reasoning processes may be affected by different memory errors, and how future measures may be constructed to better address raised concerns.

Acknowledgements

Thank you, Dr Jim Davies, for believing in me enough to take me on as a student, for all the valuable time put aside for our meetings, and for all your guidance along the way! Thank you, Dr Robert West, for all your insight on ACT-R and cognitive modelling. Thank you to all my other academic mentors: Dr Alan Scoboria, Dr Brian MacPherson, Dr Jill Singleton-Jackson, and Dr Douglas MacDonald.

Shout out: Kitu, Cristina, Olivia, Cristina, John, Valentin, Aurora, Jeff, Tim, Jordan, Ryan, Lauren, Joe, and Cole.

Table of Contents

Ch 1) Introduction

Relational reasoning involves evaluating relations between representations. One effective method of psychological research that has investigated how humans reason about relations of spatial information has been by examining how inferences are made from syllogistic deductions (Hunter, 1957; de Soto et al., 1965; Johnson-Laird, & Bryne, 1991; Cortes et al., 2021). Relational reasoning problems are lab-based tasks that require participants to evaluate a set of given premises and then generate or verify a conclusion that logically follows. For example, in an *n-term series task*, the first premise could be "A is above B", with the second premise being "C is below B", and the conclusions to be inferred would be "from the perspective of A, what direction is C?" (below).

This thesis aims to investigate how the different types of memory errors of omission and commission may be computationally modelled to provide a theoretically unifying account of qualitatively different cognitive reasoning processes and quantitatively different accuracy rates on experimental measurements of spatial relational reasoning problems. Four models were constructed based on four questions of varying complexity taken from the *Multidimensional Relational Reasoning Task* (MRRT) (Cortes et al., 2021). To establish optimal parameter settings four variations of each model were also examined, resulting in four models based on four questions of the MRRT, with four variations per model.

Our models demonstrate that modelling memory errors of omission and commission in Python ACT-R based on PRISM theory produces a similar negative relationship of a decrease in accuracy rates with each increase of premise and dimensionality complexity per question as found in the relational reasoning experimental literature.

A prominent set of related theories that have explored how individuals represent and reason about such relational problems has been theories of how the mind constructs and facilitates the use of mental models (Held et al., 2006). *Mental model theory* (MMT) (Johnson-Laird 1983; 1991; 2001) argues that humans leverage their visuospatial faculties to construct mental models of the essential information provided by the premises, which can then be used to infer a conclusion. While many mental models are constructed based on amodal spatial relations for spatial inferences, spatial properties can also be used to reason about non-spatial relations by representing non-spatial relations such as "cleaner than" in a spatial format.

Several factors have been identified that influence individuals' difficulty reasoning over these spatial relationships. For example, a commonly cited factor that has been found to influence the difficulty of mental model construction is whether a set of premises leads to a determinate problem in which only one conclusion can be drawn or an indeterminate problem in which the premises can lead to multiple correct inferences (Bryne & Johnson-Laird, 1989).

While many of these difficulty factors have been identified over decades of research, surprisingly, it has only been recently that these properties have been investigated within the same set of stimuli. Cortes et al. (2021) conducted a study that employed the Multidimensional Relational Reasoning Task (MRRT), a task that measures error rates and response times on relational reasoning stimuli that systematically vary in six different ways: number of premises (two or three), number of dimensions per premise (one or two), relational type (spatial or non-spatial), solution (true, false, or indeterminate), premise order (continuous or discontinuous), and conclusion phrasing ("A first" or "A second"). Using a mixed-effects framework, the primary finding of Cortes et al. (2021) was that those reasoning problems containing a greater number of premises and dimensions led to greater task difficulty as measured through response times and

accuracy rates. The Multidimensional Relational Reasoning Task (MRRT) and related normative data on the task difficulty of each problem can be found at (https://osf.io/qfvp2/). Models built for this thesis will be based on specific questions of the MRRT.

While classic mental model theory (MMT) provides a framework for understanding human spatial reasoning, it makes general assumptions about how mental models are constructed and what makes some reasoning problems more difficult than others. Typically, reasons for the difficulty of a problem are explained by the number of viable alternative mental models and demands placed on working memory required to solve the problem. Related theories such as the *preferred model theory* (Knauff, Rauh, & Scheller, 1995; Jahn et al., 2008; Ragni, & Knauff, 2013; Ragni, Brand & Riesterer, 2021) attempt to explain how human cognition is biased toward constructing specific mental models over others and the implications that come from these preferences. In *preferred inferences in reasoning with spatial mental models* (PRISM), Ragni and Knauff (2013) present an implementation of preferred model theory through a computational model simulating how such mental models are cognitively constructed, inspected, and varied to make inferences for relational reasoning problems. The difficulty of a relational reasoning problem, according to PRISM, can be measured by moves of their theoretical *spatial focus*.

While PRISM provides a more detailed computational account for how individuals construct and use mental models over classical mental model theory, like mental model theory, it remains general and underspecified with its assumptions towards other pertinent components of human cognition such as working memory. One way this shortcoming may be alleviated is by using cognitive architectures such as ACT-R, which are intended to be unifying theories of cognition that computationally implement theories of human memory to replicate data found primarily in experimental cognitive psychology studies. The PRISM model is a good candidate

for the continuation of modelling relational reasoning as its predecessor *spatial reasoning with models* (SRM) (Knauf, et al., 2005; Ragni, & Knauff 2008) has previously been successfully implemented in ACT-R without the need for an architectural extension (Boeddinghaus et al., 2006).

Implementing computational theories such as PRISM/SRM in a cognitive architecture such as ACT-R allows for more comprehensive, unified, and detailed accounts of how a theory of mental models may be used by the human mind. Previous work implementing SRM into ACT-R (Boeddinghaus et al., 2006) primarily explored the construction and inspection of mental models through response times garnered by production times of the ACT-R models. In this thesis, the effects of ACT-R theory are presented instead on accuracy rates by constructing four models of relational reasoning problems of varying complexity. These four ACT-R models vary in the number of premises and dimensions per problem and seek to replicate the negative relationship found in Cortes et al. (2021) of an increase in the mentioned properties correlating with a decrease in accuracy rates.

Ch 2) Literature Review

2.1) Theories of Mental Models

One of the most prominent theories for understanding how individuals reach inferences about spatial relations through deductive reasoning is mental model theory (MMT) (Johnson-Laird 1980; 1983; 1989; 1991; 1998; 2001; 2006; 2010; 2013; Johnson-Laird & Bryne, 1991; Tversky, 1993). The origin of mental model theory, however, goes as far back to work done in the logic of diagrammatic accounts of reasoning by Charles Sanders Pierce (Peirce, 1931-1958, vol. 4), as well as to Kenneth Craik's writings of the human nervous system as a calculating

machine capable of modelling its external environment (1943). See Johnson-Laird (2004) for a history of mental models with examples of other scientists who anticipated such a theory.

A mental model is "an internal model of the state of affairs that the premises describe" (Johnson-Laird, & Byrne, 1991, p.35). According to mental model theory, the information provided by premises is not held onto in the working memory of the reasoner's mind as separate pieces of information, but rather, humans primarily leverage their visuospatial faculties to construct an integrated representation, i.e., a mental model in working memory of information presented in a reasoning problem which can be used to infer conclusions.

According to mental model theory, human reasoning consists of three distinct stages: comprehension, description, and validation. At the model comprehension phase, an integrated unified mental model is generated based on the situation described by the premises, with reasoners drawing on relevant general knowledge of language, semantics, perception, tacit world knowledge, etc. The second phase of model description is named as such because it is conceived as a stage in which the parsimonious model is examined to check a putative conclusion or relations that were not explicitly stated (Johnson-Laird, & Byrne, 1991, p. 35). Finally, a conclusion is generated or validated at the model validation phase based on the previously constructed model. In this final validation phase, however, according to the classical interpretation of mental model theory, reasoners attempt to find alternative models of the presented premises which may contradict the conclusion as false. Reasoners will iterate through each phase if such a contradiction is found until all possible models are generated and examined. If such a contradictory alternative model cannot be found, the conclusion is considered true (Johnson-Laird & Bryne, 1991). Therefore, the difficulty of a problem, according to classic

mental model theory, is dependent on the amount of all possible alternative models and the demands placed on the working memory of the individual.

In contrast to the mental model theory, related theories such as the preferred model theory (Knauff et al., 1995; Jahn et al., 2008; Ragni & Knauff, 2013; Ragni, Brand & Riesterer, 2021) argue that people only construct a single mental model in most situations and remain almost blind to other interpretations unless explicitly told to acknowledge alternatives. In preferred model theory, the three stages of comprehension, description, and validation are renamed to better characterise what occurs in each respective stage as model generation, inspection, and variation phases (Ragni & Knauff, 2013).

Another radical departure from the mental model theory that the preferred model theory puts forth is that reasoners are biased towards specific solutions for reasoning problems. Variations of models are only conducted as local transformations, in contrast to the generation of a whole new model and are typically undertaken during indeterminate problems in which approaches such as the *fff-strategy* (first free fit) or *ff-strategy* (first fit) are often employed - both of which will be elaborated on in the next section.

Although the mental model theory has developed since its inception when Johnson-Laird attempted to present a unified approach to comprehension and reasoning over mental models (1980; 1983), what has remained consistent through the years are three fundamental axiom-like assumptions that mental model theory rests on (Johnson-Laird, 2006; 2010). First, an essential characteristic of each mental model is that they represent a single possibility that is common to a distinct set of possibilities. Second, models are iconic insofar as they can be so that the structure of a representation corresponds to the structure of what it represents. As (Johnson-Laird 1998, p447) puts it, “Like a diagram (Maxwell, 1911) or an architect's model, the parts of the model

correspond to the relevant parts of what it represents, and the structural relations between the parts of the model are analogous to the structural relations in the world". Finally, mental models are constructed only to represent information about what is considered valid or possible at the time; this is referred to as the *principle of truth* (Johnson-Laird, 2013). This is not to say that mental models cannot represent what might be false and temporarily assumed to be true, such as in the case of counterfactuals. Instead, the reason why the negation of a premise is not represented in mental models is argued as a consequence of how the mind operates to represent as little information as possible to reduce the cognitive load on working memory, i.e., *the principle of economy* (Johnson-Laird, 2006, p.112). So mental models, according to the principle of truth, represent a proposition in the premise only when the proposition is true in possibility unless something exceptional occurs to overrule it.

While many psychologists agree that mental model theory is one of the most preferred frameworks to explain spatial relational reasoning, it is not without its critics, as it was initially developed to stand in contrast to a competing theory of *mental logic*. The traditional theory of mental logic argues that the human ability to reason depends on an implicit mental logic that uses formal rules of inference much like those in logical calculus and related areas. The hypotheses of the mental logic theory argue that deductive reasoning from premises entails using formal rules to infer a valid conclusion much akin to a proof analogous to explicit proofs in logic. Mental logic theory preserves the logical form of premises for the reasoning task and supports the use of formal rules to provide a valid conclusion. Johnson-Laird (2010), in contrast, argues multiple distinct weaknesses of mental logic theory, such as the problem of being able to recover a logical form of assertions, that valid conclusions arrived at in human reasoning are not monotonical with an increasing number of premises, and that manipulations of content can often affect an

individual's choices of which cases refute general hypotheses. While the theory of mental logic predates mental model theory (Beth & Piaget, 1966; Braine, 1978), it has continued to have supporters (Rips 1994; Braine & O'Brien, 1998; O'Brien 2010). See Ragni, and Knauff (2008) and Van der Henst (2002) for a comparison of the two theories.

A final important point to be mentioned is that mental model theory is not equivalent to theories of propositional representations (Pylyshyn 1981; 2001; 2002; 2003) or mental imagery (Kosslyn 1980; 1994; Kosslyn et al., 2006; Pearson, & Kosslyn, 2015). Johnson-Laird (1998, p. 463) argues that thinking is dependent on propositional representations that capture the meaning of premises and that they are used to construct mental models that generate an integrated representation capturing the meaning of the premises. Mental model representations are abstract, amodal, and purely conceptual, whereas visual mental images are primarily understood as how something looks from a particular point of view. Instead, Knauff & Johnson-Laird (2002) present the *visual-imagery-impedance hypothesis*, which argues that relations that elicit visual images containing details irrelevant to an inference, such as mental images, may actually impede the process of reasoning. One of the primary ways mental model theories have been empirically tested has been through the use of *relational reasoning problems*, the subject of the following section.

2.2) Relational Reasoning Problems

Relational reasoning problems are lab-based assessments that have long been employed to test the deductive inferences of participants in psychology experiments (Hunter, 1957; de Soto et al., 1965; Huttenlocher, 1968; Johnson-Laird, 1972; Bryne & Johnson-Laird, 1989). In relational reasoning problems, participants are presented with a series of premises, typically two to four, and are requested to either generate or verify a conclusion that logically follows to

complete the task. Research on relational reasoning problems has identified several distinct ways these problems may vary to produce their own unique effects on the difficulty of the problem as measured through accuracy rates or response times. A summary follows, along with examples of common variations found throughout the literature.

The first significant distinction between relational reasoning problems is spatial or non-spatial relation types between premises. This thesis will focus on spatial reasoning problems; however, this distinction is important since empirical data on how visual and spatial mental representations interact and relate to one another leaves much unresolved, with the subject remaining a topic of debate (Schultheis et al., 2007). Due to the nature of mental models, it is argued that abstract notions such as "nicer than" or "better than" may be represented in a spatial-analogical manner. Empirical evidence has supported the association of mental model reasoning as correlated with activation in the parietal lobe, an area associated with several spatially related cognitive processes (Goel & Dolan, 2001). In contrast, more visual relations such as "cleaner than" or "dryer than" in reasoning tasks have been found to lead to increased activation in early visual cortex areas (Knauff et al., 2003), and have been argued, as mentioned in the previous section, to even at times hinder the reasoning process itself (Knauff & Johnson-Laird, 2002). Sima et al. (2013) used head-mounted eye-tracking instruments to investigate the differences between spatial and visual mental representations and found significant eye movements along with spatial relations only when participants were asked to employ a visual mental representation but not when a spatial mental representation was expected of being used.

Relation Type: Spatial

A is above B

C is below B

Is A below B? (True/False)

Relation Type: Non-Spatial

A is cleaner than B

C is less clean than B

Is A cleaner than C? (True/False)

The inclusion of additional premises and dimensions per problem has been found to be one of the major increases in a problem's difficulty. This decrease in accuracy rates or increase in response time is often credited to an increase in working memory demand due to additional premises and the number of relational dimensions per premise necessitating the construction of more complex mental models to be reasoned over (Goodwin, & Johnson-Laird, 2005; Johnson-Laird, 1989). A recent study conducted by Cortes et al. (2021), whose Multidimensional Relational Reasoning Task will be used to construct the models used for this thesis, demonstrated an interaction effect between problems with a greater number of premises and dimensions leading to greater task difficulty. The ACT-R models built in this thesis will vary on the number of premises and dimensions to compare accuracy rates, matching the combination of premises and dimensions as illustrated below.

Relation Type: 2 Premises & 1 Dimension

A is cleaner than B

A is less clean than C

Conclusion: B is less clean than C? (True/False)

Relation Type: 2 Premises & 2 Dimension

A is more organised and less helpful than B

A is less organised and less helpful than C

Conclusion: A is less organised and less helpful than C? (True/False)

Relation Type: 3 Premises & 1 Dimension

A is above B

C is below B

D is above A

Conclusion: D is above C? (True/False)

Relation Type: 3 Premises & 2 Dimensions

A is above and to the left of B

B is below and to the left of C

C is above and to the left of D

Conclusion: A is above and to the left of D? (True/False)

Another common way a relational problem may be altered to affect construction and inspection difficulty is the order in which premises are presented in a continuous, semi-continuous, or discontinuous manner. In a continuous and semi-continuous order, it is possible to integrate the information from the first two premises into one single model. However, when presented with a discontinuous premise order, one must wait for later premises such as the third

or fourth to integrate all information into a single model. Ehrlich and Johnson-Laird (1982) found that in four-term series problems, discontinuous premise orders produced a significant amount more error rates (60% errors) compared to continuous (37% errors) and semi-continuous (39% errors) premise orders, which were very similar to one another.

The difficulty of integrating objects within a mental model of differing premise orders to increase mean error rates from continuous, semi-continuous, to discontinuous (Knauff et al., 1998; Najasmic et al., 2011) is referred to as the *Continuity Effect* or *Order of Premises Effect*. Discontinuous premise orders are believed to be more difficult due to the construction of multiple models, which need to merge into a unified mental model or a unified sub-set of all constructed models. All ACT-R models in this thesis will adopt a continuous premise order to ensure equivalent conditions between groups.

Premise Order: Continuous

A is Left of B

B is left of C

C is Left of D

Premise Order: Semi-continuous

B is left of C

C is left D

A is left of B

Premise Order: Discontinuous

C is left of D

A is left of B

B is left of C

The investigation of the effects of premise order gives rise to a related inquiry into the effect of the order of objects within each premise or conclusion. This area has been extensively studied in research on syllogistic reasoning and has come to be referred to as the *Figural Bias Effect,* also referred to as the *Order of Terms Effect*. Consider the spatial relational of "A is to the left of B" and "B is left of C", there exist four possible ways, or "figures", that allow reformulating this sequence using the same binary relations (left/right) leading to the same spatial arrangement. "A is to the left of B" is equivalent and would lead to the same figure produced as "B is to the right of A". Therefore, another manipulation which may be performed on a spatial relational reasoning task is the order by which objects of a premise or conclusion are presented. This differs from the order of premises manipulation mentioned above, as in the order of terms effect manipulation, the premise order remains the same between conditions, but the order of objects presented in the premises or conclusion to be verified are manipulated and often reversed.

Conclusion Phrasing: A first

Adrian (A) is below Brian (B)

Brian (B) is below Caleb (C)

Conclusion: Adrian (A) is below Caleb (C)

Conclusion Phrasing: A last

Adrian (A) is below Brian (B)

Brian (B) is below Caleb (C)

Conclusion: Caleb (C) is above Adrian (A)

With the possibility of multiple figures being equally valid in responses and producing the same mental models, the question arises as to what order objects of a figure are preferred when generating a conclusion? An experiment conducted by Johnson-Laird and Bara (1984) found that given the two different types of non-spatial relation problems, as shown below, when participants are asked to generate conclusions regarding the relation between A and C, type 1 syllogisms elicit more conclusions of form A-relation-C, while conclusions of type 2 syllogisms elicit conclusions of type C-relation-A.

Type 1

A is related to B

B is related to C

What is the relation between A and C?

Type 2

B is related to A

C is related to B

What is the relation between A and C?

Johnson-Laird and Bara (1984) argue that type two problems are more difficult because to integrate all information into a unified model, object B needs to be brought into the middle of the model. This is not the case when compared to relations of problem type one, which saves cognitive resources as the information provided repeats with a term already encountered as the first term of the second premise.

At least regarding syllogisms of spatial reasoning, Knauff et al. (1998) found no general bias for A-C relations, but rather a figural bias conclusion of the reverse order C - A was preferred making up 62.8% of all conclusions. The reason for such contradictory results, Knauff

et al. (1998) explain, is the cognitive processes that inspect a constructed mental model - primarily due to the facilitation of a spatial focus that begins on the last object inserted during the construction phase. Therefore, cognitive effort is saved by starting on object C as the reference object, and then locating object A, resulting in a conclusion phrasing of C - A.

A final important distinction that needs to be made for relational reasoning is between determinate and indeterminate problems. So far, all the above examples have been determinate problems in which there is only one constructed model that is logically valid. Indeterminate problems rather have led researchers to argue for the existence of a *preference effect of indeterminate problems* in which differing mental models may be constructed based on the same premises. For example, given the premises "A is to the left of B" and "C is to the right of A" leads to two possible figures: "A-B-C" or the equally valid model of "A-C-B". The difference between these two constructions demonstrates two separate micro-strategies, the first figure being constructed via the first free fit (fff) strategy in which the spatial focus inserts a token at the first position that fits with the premise, and the second figure being constructed according to the free fit (ff) strategy which inserts “C” between the two other tokens of “A” and “B”.

The preferred model theory argues that when individuals encounter such indeterminate problems, they construct preferred mental models that are biased toward specific reasoning strategies (Knauff et al., 1995; Ragni & Knauff, 2013). These preferences and possible variations will be covered more in the following section.

2.3) Computational Models of Spatial Reasoning

Spatial and visual properties are often intertwined in mental representations, and so the field of computational imagery is conceptualised as incorporating both of these properties through the ability to represent, retrieve, and reason about such information (Glasgow &

Papadias, 1992). Many visuospatial reasoning accounts employ various types of representations; some of these attempts are implemented within cognitive architectures, while others are not, but many fall within the categories of propositional structures (Jahn, 2003), array-based structures (Ragni, & Knauff, 2008), diagrammatic reasoning (Chandrasekaran, 2004) or image-based reasoning (Bertel et al., 2006).

The concern of this thesis, however, exclusively deals with spatial representations of which there have been multiple attempts to simulate and provide computational accounts of human memory and reasoning - see Madl et al., (2015) for a review. One popular method with which computational accounts of spatial reasoning have been conceptualised is array-based representations (Glasgow & Papadias, 1992; Hummel & Holyoak, 2001, 2005; Kunda et al., 2013; Tabachneck-Schijf et al., 1997). Array theory is the mathematics of nested, rectangularly arranged data objects (More, 1979) and will be the primary form of representation for constructing the models of this thesis.

While mental model theory continues to be one of the most preferred frameworks for understanding human spatial reasoning, the theory itself makes quite general claims and assumptions about what makes some reasoning problems more difficult than others. The previously mentioned PRISM is one example of an attempt to model relational reasoning problems as more computationally explicit. PRISM is a computational description (Ragni & Knauff, 2013) of the preferred mental model theory which can be implemented (Ragni et al., 2021) to simulate and explain how preferences are found in the construction, inspection, and variation phases of human reasoning with mental models. Two descriptions can broadly characterise the construction and inspection phases of PRISM models. First, spatial working memory contains representations of objects from premises as tokens with their relations

conceptualised within a two-dimensional spatial array. Second, all operations of the reasoning process on this spatial array are considered to be moves of a spatial focus of which conclusions may either be generated or verified.

PRISM is a successor to spatial reasoning by models (SRM) (Ragni & Knauff, 2008; Ragni et al., 2005). The differences between SRM and PRISM predominantly reside in the final third stage of variation, which is primarily intended for problems of indetermination. Since our models will focus on determinate reasoning problems to keep consistency between them, this final stage will not need to be fully implemented in our models. This is not of concern since, as Ragni and Knauff (2013, p. 568) claim, the model variation phase only rarely takes place and is a major departure from the classical mental model theory. The following explanation of the construction and inspection phases of mental models holds true for both the operations of PRISM and SRM but will be primarily referred to by the former going forward.

PRISM reasons via binary relations with each binary relation provided by a premise defined as a triple (X, r, Y) in which X is the to be located object (LO), r is the binary relation, and Y is the relatum, or reference object (RO) (Miller, & Johnson-Laird, 1976). In logic, a relatum is the second or one of the succeeding terms of a logical relation. Given a premise such as "The car is left of the hydrant", it is argued (Oberauer & Wilhelm, 2000; Logan, 1994) that the relatum, hydrant, functions as a reference frame that we use to locate the object car. PRISM follows this same line of reasoning with the exception of the first premise provided in a reasoning task where the theory assumes that individuals prefer to change the roles of the reference object and to be located object in favour of incremental model construction - a phenomenon demonstrated by (Oberauer and Wilhelm, 2000). Suppose a premise is encountered where the reference object as the second object has not been encountered, but the to be located

object has as the first object in a premise. In that case, PRISM will place its focus on what has already been inserted in the model.

At the beginning of the construction phase of a reasoning problem, PRISM's spatial focus begins at the coordinate position (0,0) and has four possible directions in which it can be moved. These operations of directional movement are right, left, forward, and backward, as well as a possible no-move operation. When reading a premise, PRISM distinguishes between four types; see figure 1 for the logic of the construction phase of PRISM.

Type 1-Initial premise: this is the first premise encountered in a reasoning problem and provides the starting point of a model's construction.

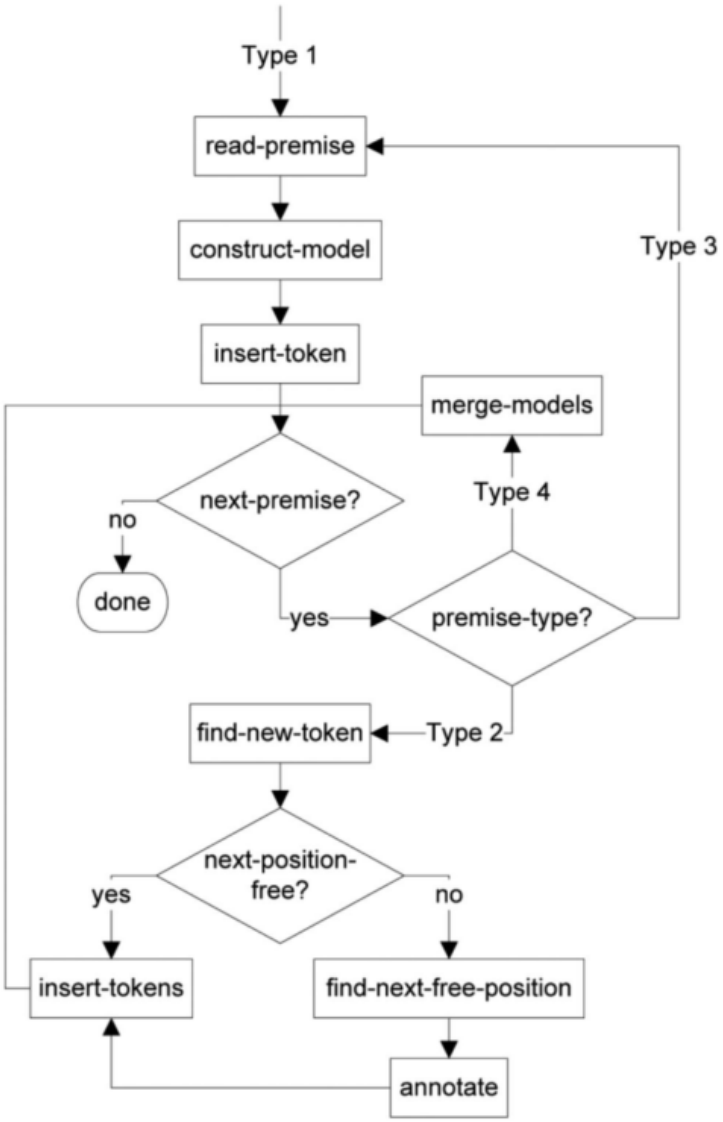
Type 2-One-new-token-premise: a type 2 premise consists of two tokens, of which one token has already appeared in a previous premise and has already been inserted into the model. To account for the difference between determinate and indeterminate problems, Type 2 premises may either be of type 2d (determinate) or type 2i (indeterminate). If a 2d premise is encountered, a token is placed at the appropriate coordinate. However, suppose a type 2i premise is encountered where there is more than one possible position that agrees with the spatial relation. In that case, the preferred model theory follows the first-free-fit principle, as well as annotates that coordinate for later inspection and variation phases.

Type 3-Two-new-tokens-premise: in a type 3 premise, both the reference object and to be located object have not yet been encountered. This type of premise, which is found in discontinuous type premise orders, necessitates the construction of separate models which do not have any tokens in common but are often merged into a single model in the following premise.

Type 4 -Connecting-sub-models-premise: a type 4 premise often follows a type 3 premise and occurs when a token appears that connects two separate models.

Figure 1.

Control Structure Logic of PRISMS Construction Phase



Note. The model construction phase in PRISM which inserts tokens into mental models according to different types of premises. Reprinted from "A Theory and a Computational Model of Spatial Reasoning With Preferred Mental Models", by M. Ragni and M. Knauff, 2013, *Psychological Review, 120* (3), p. 568. Copyright 2013 by the American Psychological Association

To better elucidate the construction phase of PRISM, consider the following three premises one-dimension relational reasoning problem:

A is to the left of B
C is to the right of B
D is to the right of A

At the beginning of a reasoning problem, PRISM is in its construction phase with the spatial focus beginning at coordinate (0,0) and receives a premise of type 1. Since this is a type 1 premise, PRISM inserts the first token of the first premise (A) at coordinate cell (0,0). Using object A as the reference object, the spatial focus moves one cell to the right and inserts object B as the to be located object at cell (1,0). The "parser", which is not part of PRISM theory, assumes the meaning of premises as already understood and reads the next premise, which is of type 2d. Since the spatial focus is already on object B, this is used as the reference object, and the spatial focus moves one step to the right and inserts a token for object C at coordinates (2,0). The final premise is then read by the parser, "D is to the right of A".

PRISM is based on a theory of preferred models and so constructs a single model to reason over. Since this final premise would be of type 2i, though, it would first require the spatial focus to locate object A as the reference object, with the spatial focus performing a series of move left operations. Now that the spatial focus is on the coordinate (0,0) with object A found, it proceeds to integrate the rest of the premise into the mental model by first moving one step to the right. Since moving one step right, lands the spatial focus on object B, it performs its first free fit (fff) strategy and writes an annotation of the sort "D is right of B" on object B so as to use it if model variation would ever be needed. The spatial focus now continues to move one more step

right to the right, landing on object C at cell (2,0), which is also annotated. The spatial focus moves one more step to the right and finally encounters an empty cell to which object D is inserted at cell (3,0).

Now that the model has been constructed, if PRISM was required to verify the conclusion "A is to the left of D", it would enter its second phase of inspection. After construction, the spatial focus is on object D at coordinate (3,0), which now serves as the inspection phases' starting point. Using D as a reference object, PRISM begins shifting its spatial focus incrementally one step to the left until object A is located. Once object A is located, PRISM verifies the conclusion as true - if object A is not located, it is verified as false.

If PRISM were required to build a variation of the constructed model in the case of a type 2i premise example, PRISM would start inspection in a similar way at the last point it entertained during its construction phase. If an annotation is processed at the beginning or at any time while shifting coordinates, PRISM signals that an alternative model may exist and applies *a principle of minimal change* so that the varied set of models after the variation is as similar as possible to the initial preferred model through local transformations. What this entails is an insertion of D between A and B so as to be in accordance with the first fit (ff) strategy, thereby producing the model “A – D – B – C”, which importantly may still verify the conclusion, “A is to the left of D”, as true.

A large motivation behind PRISM is its theoretical contribution to mental model theory through providing a computational account of which micro-strategies are preferred for how mental models are constructed, inspected, and varied as found in empirical data. In the standard account of mental model theory, the difficulty of a problem is assessed by the number of alternative models which are possible, so that the more models an individual must consider, the

higher the load on working memory. In contrast, because PRISM's spatial array functions as its spatial working memory, it offers the accumulated number of necessary spatial focus operations, which follow a *principle of minimal directional change,* as a measure of the difficulty of an inference. Despite these measures of difficulty, much like how mental model theory remains general as to the computational implementation of mental model reasoning, many computational accounts of spatial reasoning leave much to be desired in terms of how working memory interacts with mental models. One way in which working memory effects may be tested on mental models is through cognitive architectures.

A cognitive architecture that has successfully implemented PRISM's predecessor, spatial reasoning by models (SRM), has been ACT-R (Boeddinghaus et al., 2006). This ACT-R implementation was able to model several effects of spatial reasoning, such as the premise order effect and the difficulty of indeterminate problems with complexity predicted through processing times. However, while these processing times take longer for more difficult problems, they were still far below empirical findings. The models built for this thesis will be based on this previous work but will focus on another form of evaluation, accuracy rates of relational reasoning problems. Boeddinghaus et al. (2006) touched on accuracy rates by explaining that alternative models might be more difficult due to the decay of ACT-R chunks in the variation phase of reasoning. However, the models for this thesis will focus on only determinate problems and so will not require a variation phase.

2.4) ACT-R Cognitive Architecture

Cognitive architectures belong to a research paradigm conducted within the field of artificial intelligence, which seeks to provide a unified theory of cognition to better model the human mind (Lieto et al.,2018; Kotseruba & Tsotsos, 2020). The concept of cognitive

architectures originated from work on unified theories of cognition, which argued that the behavioural sciences, through their methodologies, have largely remained fragmented and unable to reassemble into a well integrated theory of the mind (Newell 1973; 1982; 1990; Anderson 2004). ACT-R is one of the first computationally functional implemented cognitive architectures and continues to be one of the most popularly used (Anderson 1983, 1990, 2004, 2007; Anderson & Lebiere, 1998). The ACT-R cognitive architecture will be used in this thesis to model the spatial relational reasoning tasks examined in the next section.

The history of ACT-R originally began with the Human Associative Memory (HAM) model of memory, in which information processing operated through a series of buffers that encoded information from outside stimuli into useful chunks (Anderson & Bower, 1973). ACT-R has evolved over the years, but its core theory has largely remained stable from iteration to iteration and is currently on its seventh edition with ACT-R 7. See Ritter et al. (2019) for history and summary of ACT-R.

ACT-R is often considered to be a hybrid architecture in that it incorporates symbolic as well as sub-symbolic representations in its operations. At the core of ACT-R is a modular theory of mind. A theoretical module in this context will be understood as some cognitive faculty that is particularly ascribed to a particular brain region or network.

There are five core modules to ACT-R. The central production system module stores all the productions required to model a task and operates via productions in a 'match-select-fire' or "if-then" fashion. The Declarative Memory Module is intended to store and retrieve information in the form of symbolic chunks which are affected by sub-symbolic parameters. This hybrid architectural nature of ACT-R allows us to better model declarative memory phenomena through threshold, decay, and partial matching parameters. Over the years, ACT-R researchers have

identified a particular set of “canonical” parameter values that should be used when constructing models (Stewart & West, 2007). The rate of memory chunk decay and productions firing at 50ms for example are some of the most well established of these parameter values and should only be altered with justified reason. There are however other parameters such as latency, activation noise, partial matching, and threshold which are allowed to vary more from model to model. Our models for this thesis will only alter the threshold parameter with an increase for each move of the PRISM spatial focus.

The third module, the intentional module sometimes referred to as the imaginal module, is intended to hold the current mental representation of a problem as well as to maintain a chunk for context-relevant information in service of the intentional task goal. The last two modules are the vision and motor modules, which make up a perceptual-motor system intended to encode and manipulate an environment. Communication between these modules is facilitated through the exchange of "chunks" of information (i.e., attribute-value matrices) through buffers of each module. The contents of the chunks held in the buffers collectively make up the working memory of the agent.

In every ACT-R model, there are generally two major theoretical commitments, the first being the architectural theory being implemented which was just discussed - a pattern matching production system which operated via chunk-based storage and communication through a network of buffers. The second theoretical commitment of an ACT-R model is then how the knowledge represented within the architecture drives the agent with ACT-R containing the capabilities to alter this knowledge over time by means of different utility learning methods. PRISM theory can contribute to this second theoretical commitment well and serves as the primary theory of knowledge representation for our ACT-R models.

In general, there are three major ways to analyze how well an ACT-R agent models human data of a task: response times, error rates, and learning rates. Two commonly reported memory errors in experimental psychology are those of omission, failures to remember, and errors of commission, in which a participant performs an incorrect or additional action. ACT-R allows us to model both of these types of errors (Kelly et al., 2000; Lebiere, et al., 1994) and, in doing so, provides us with an avenue for examining the effects of different memory errors on accuracy rates of relational reasoning problems. This thesis develops on previous attempts to model spatial relational reasoning through modelling the effects of different working memory errors on accuracy rates for relational reasoning problems through the construction of four ACT-R models. These four ACT-R models will provide accuracy rates of four determinate, continuous-premise ordered relational reasoning problems taken from the Multidimensional Relational Reasoning Task (Cortes et al., 2021).

Ch 3) Methodology & Models Built

3.1) Overview

Four Python ACT-R models of spatial relational reasoning were created based on *preferred inferences in reasoning with spatial mental models* (PRISM) (Boeddinghaus et al., 2006; Ragni & Knauff, 2013; Ragni et al., 2021). Each model is based on a single question taken from the Multidimensional Relational Reasoning Task (MRRT) Cortes et al., (2021). Since there are only four conditions of premise and dimension complexity in the MRRT, two premises one dimension, two premises two dimensions, three premises one dimensions, and three premises two dimensions, only four determinate and continuous questions were determined to be sufficient to model the complexity conditions. For each one of the four models, however, four

variations were also run, increasing the ACT-R *threshold* parameter per move of PRISM's theoretical spatial focus to better investigate optimal parameter settings. In summary, this means four ACT-R models based on four questions of the MRRT, with four variations per model.

These four Python ACT-R models and each of their four variations demonstrate that differing memory errors of omission and commission due to increased working memory demands can produce a similar relationship of accuracy rates between spatial relational reasoning problems of varying complexity, as found in Cortes et al. (2021).

3.2) Multidimensional Relational Reasoning Task (MRRT)

Despite multiple studies having found different stimulus properties of relational reasoning problems to affect the difficulty between problem variations, it has only been recently that a study has investigated a collection of the different spatial properties on the same set of stimuli. The following section is a review of Cortes et al. (2021) as the study provides the relational reasoning problems that are to be modelled. Cortes et al., (2021) data and materials can be found at the Open Science Framework: https://osf.io/qfvp2/ .

Cortes et al. (2021) present the Multidimensional Relational Reasoning Task (MRRT), a task that consists of 90 reasoning problems that vary stimulus properties within a single set of relational reasoning problems. These varied stimuli properties of the MRRT are a number of premises (2 or 3), a number of dimensions (1 or 2), relation type (spatial or non-spatial), solution (true, false, or indeterminate), premise order (continuous or discontinuous), and conclusion phrasing ("A first" or "A second"). Each problem presented to the participants either consisted of two or three premises and a conclusion which was intended to be verified with "True" if participants believed the conclusion was a consequence of the previous premises or a "False" if they believed the conclusion did not follow the premises or if the solution was indeterminate.

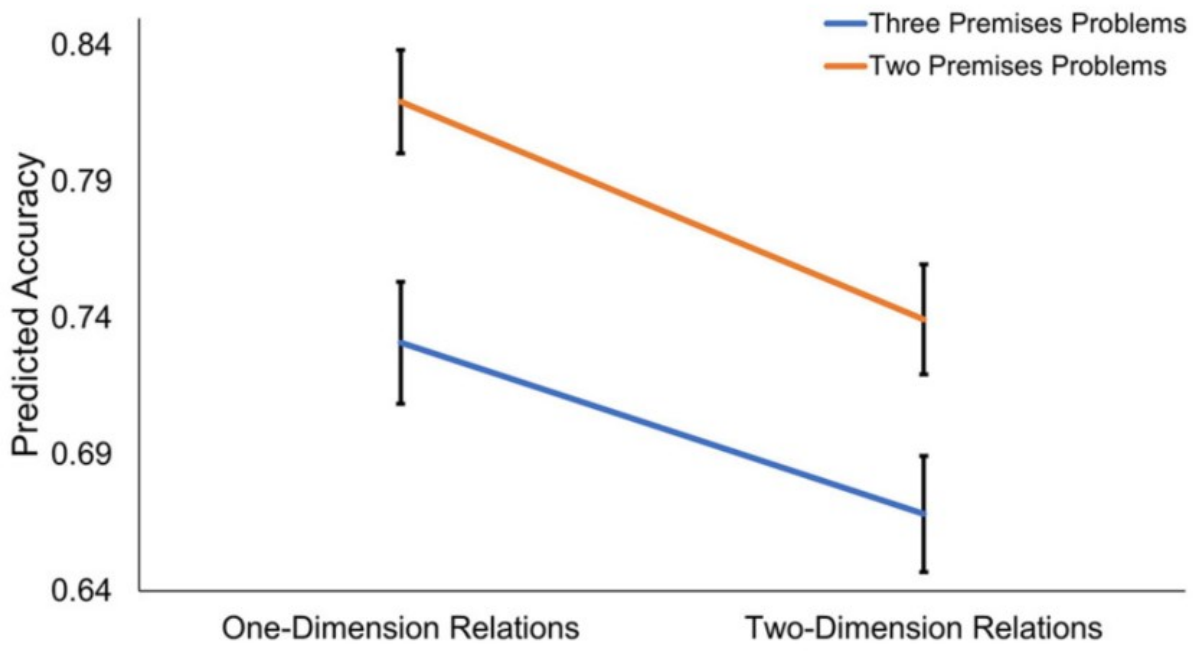
Participants were instructed to solve each problem in their head without the assistance of a pen and paper, use of their fingers, or any other external aid. Participants solved the problems in three separate blocks, each separated by three-minute breaks. One objective of this study was to be one of the first to provide normative reaction time data on these varying problems and therefore permitted unlimited time for answering. Cortes et al. (2021) utilised a three-form design (Graham et al., 1996; 2006), such that 90 problems were divided into four different sets of 22-23 problems (X, A, B, C), with each set having the same number of problem types. All groups completed the X set of problems and then each of the three groups completed two of the three A, B, or C sets, with each participant completing a total of 67 problems. Group 1 (N= 105) completed X, A, B; Group 2 (N = 102) completed X, A, C; and Group 3 (N=103) completed X,B,C. Therefore, each problem was completed by at least 200 participants, with the X set being completed by all 310 participants.

To assess the differences between stimulus properties on reaction time and correct accuracy of conclusion performance, Cortes et al. (2021) conducted a series of mixed-effects models. The first model assessed accuracy rates through a mixed-effects logistical regression, while response time was assessed through a second model via a mixed-effects linear regression model. The data analysed by Cortes et al. (2021) examined both multiple stimulus properties in relation to and controlling for each other. What was found lined up well with previous literature findings on relational reasoning. The higher the number of both premises (two or three) and dimensions (one or two) had a large effect on both accuracy and reaction time. The number of dimensions and premises for both spatial and non-spatial problems was additionally found to have an interaction effect with Cortes et al. (2021, p. 6), explaining that the necessitation of creating more complex models may increase working memory load – see Figure 2. Analyses

revealed problems with non-spatial relations were found to be more difficult than problems with spatial relations and indeterminate problems were found to be associated with lower accuracy rates. Regarding premise order or conclusion phrasing, no effects were found on accuracy or reaction times.

Figure 2.

Cortes et al., (2021) Predicted Marginal Means from Dimension X Premise Interaction.



Note. Predicted marginal means of Dimension X Premise interaction results from mixed-effects models. Reprinted from "What Makes Mental Modeling Difficult? Normative Data for the Multidimensional Relational Reasoning Task", by R. Cortes et al., 2021, *Frontiers in Psychology,* p. 7. Copyright 2021 by the Frontiers in Psychology

3.3) A Python ACT-R Implementation of PRISM

PRISM is an implementation of the preferred mental model theory and is intended to simulate human performance on spatial reasoning tasks (Ragni, & Knauff, 2013; Ragni et al.,

2021). The constructed ACT-R models are based on PRISM's construction phase and inspection phase, which conceptualizes spatial memory representations as two-dimensional arrays. All operations and reasoning processes conducted on this spatial array are facilitated by and considered to be moves of a spatial focus identical to its predecessor spatial reasoning by models (SRM) (Knauff et al., 2005). This thesis will focus on determinate, continuous problems to model results from Cortes et al. (2021), so PRISM's third stage of variation will not need to be fully implemented.

Mental models are constructed based on information found in premises and through actions of the spatial focus. The spatial focus may move in the four directions of left, right, above, and below, in addition to a no move operator, and is responsible for the insertion of objects found in premises in the appropriate cell of an array. These spatial arrays are then inspected by the spatial focus to validate or generate a conclusion based on the relations between objects. Each binary relation provided by a premise is defined as a triple (X, r, Y) in which X is the referent or to be located object (LO), r is the binary relation, and Y is the relatum, or reference object (RO). (Miller, & Johnson-Laird, 1976).

To examine the effects of a theory of working memory on relational reasoning problems, PRISM's first two phases are implemented in cognitive architecture ACT-R. Specifically, these models are constructed in Python ACT-R (Stewart & West, 2007), the use of which highlights an important distinction between theory and implementation. Python ACT-R uses similar formulas and modular organisation as regular ACT-R version 7 but has the benefit of operating with a more current and commonly taught programming language - Python rather than the traditional LISP.

To implement the construction and inspection phases of PRISM in ACT-R, a similar strategy will be adopted as in Boeddinghaus et al. (2006), where PRISMS' spatial focus is instantiated as a buffer with slots-value pairs for the name of the token object it is currently on in the array, the direction the spatial focus points to, and the position values of that location. When a token object is placed at a location, a new chunk of information is created and stored in the declarative memory module with slots for the object's name, its position values, and a slot for annotations primarily used in problems of indeterminacy.

To maintain similar construction and inspection processes across models, our PRISM ACT-R agents will use 14 different production types for each relational reasoning problem held in a second buffer, a goal-focus buffer. Productions used to facilitate the reasoning process include 'read_premise' and 'read_conclusion', which stand-in for a required semantic parser that is not part of ACT-R or PRISM theory. PRISM's five-movement types are represented by the productions: 'move_above', 'move_'below, 'move_right', 'move_left', and 'move_none'. To interact with the declarative memory module, productions 'request_name' and 'recall_name' allow for a branching of solutions depending on if a memory error occurred. Therefore, the productions of 'recall_correct', recall_omission', and 'recall_commission' are included to provide better clarity for the modeller and clarify what solution is being implemented. The final production that then produces an answer is referred to as 'confirm_conclusion'.

3.4) Models Built

This thesis seeks to model memory errors that may occur in varying spatial relational reasoning problems to examine what effect these errors of omission and commission may have on reasoning processes and accuracy rates. To do so, four models were constructed to attempt to model the decrease in accuracy rates due to the interaction effect of increased premises and

dimensions found in Cortes et al. (2021). Four questions were selected from the Multi-Dimensional Reasoning Task to model such interaction effects that differed in the number of premises and dimensions. To preserve consistency outside the manipulated variables of the number of premises and dimensions, all problems are determinate and have a continuous premise order. These four relational reasoning questions of the multi-dimensional reasoning task are illustrated below.

Question: 6 (Two Premise, One Dimension)

P1) Edward is to the left of Derek.

P2) Derek is to the left of Travis.

C) EDWARD IS TO THE LEFT OF TRAVIS. (solution = True)

Question: 36 (Two Premise, Two Dimension)

P1) Edward is below and to the left of Derek.

P2) Derek is below and to the left of Travis.

C) EDWARD IS BELOW AND TO THE LEFT OF TRAVIS. (solution = True)

Question: 21 (Three Premise, One Dimension)

P1) Edward is to the left of Derek.

P2) Derek is to the left of Travis.

P3) Travis is to the left of Brian.

C) EDWARD IS TO THE LEFT OF BRIAN. (solution = True)

Question: 66 (Three Premise, Two Dimension)

P1) Edward is below and to the left of Derek.

P2) Derek is below and to the left of Travis.

P3) Travis is below and to the left of Brian.

C) EDWARD IS BELOW AND TO THE LEFT OF BRIAN. (solution = True)

All ACT-R models for this thesis will begin their construction phases at default parameter settings, these are: production_time = 0.05, production_sd=0.01, latency=0.05, threshold=0.3, maximum_time=10.0, finst_size=0, finst_time=3.0, noise=0.3, base_noise=0.3, partial matching strength =0.0, partial matching limit = 0.0, DM spread_strength=1. Four variations of each spatial relational reasoning problem will be run for 16 model variations. The first variation will run all four relational reasoning problems (2P1D, 2P2P, 3P1D, and 3P2D) beginning at default parameter settings and will not alter them during the running of the models. The three subsequent variations will increase the threshold parameter setting of the ACT-R agent by 0.1, 0.2, or 0.3 for each move of the spatial focus. According to PRISM theory, the accumulated number of movements required by the spatial focus can serve as a proxy measurement for the difficulty of a problem. Therefore, the reason for this increase in threshold is to model the increased stress on the memory system of the cognitive agent due to working memory load in accordance with PRISM theory of movements of the spatial focus as a measurement for a problem's difficulty.

These models, therefore, argue for an existing relationship between the difficulty of a problem as a function of movements of the spatial focus and an increase in the threshold parameter. Threshold increases were chosen to increase with moves of the spatial focus as it best reflects the increase in errors of omission that comes with increased demand on working memory and is one of the more theoretically justifiable parameters to be manipulated in ACT-R models as compared to decay rates and production times which are often argued as remaining constant. Additionally, partial matching strength was set to the minimum of 0.0 since there is no theoretical reason in these models as to why one position or object may be biased for an error of commission compared to the others found in the mental model. To give a better understanding,

the following illustrates how question 66 of the multidimensional relational reasoning task would be processed in our ACT-R model with a constant threshold increase of 0.2 per move of the spatial focus.

To begin the construction phase, the goal_buffer is set to the first production, ‘read_premise’, fires, and prints "Edward is to the left of Derek". The goal_buffer then processes the production 'insert_edward', and the spatial_focus buffer, which resides at location (0,0), inserts Edward, thereby creating a declarative memory chunk in the form *object: Edward, location: 0,0, annotation: na.* Now that Edward has been placed in the array, the production 'move_right' moves the spatial focus over one step at coordinates (1,0). Production 'insert_derek' then inserts token object Derek at its current coordinates and creates a chunk in declarative memory in the same fashion as Edward. The second premise of "Derek is to the left of Travis” is then processed in the same way. The spatial focus at this time is still on Derek, and so fires a 'move_right' production followed by an 'insert_travis' production, along with a declarative memory addition at coordinate (2,0) to conclude the construction phase.

With Travis functioning as the reference object, the to be located object of Edward must now be remembered, and so a 'request_edward' production fires to request the token's location from declarative memory. Every declarative memory request introduces a branching of logic as to what the next possible production may be. In these ACT-R models, there exist three possibilities after a declarative memory request is made. The first possibility is that Edward’s location is correctly recalled. The second possibility is a memory error of omission occurs in which Edward's location cannot be remembered. The third possibility is a memory error of commission in which Edward's location is misremembered for another - in this case; it would be Brain's since the spatial focus is still on Trevor. Because the spatial focus must move left from

Trevor's position (2,0), all objects within the array left of the spatial focus are candidates for the declarative memory retrieval. How the construction phase is concluded based on which possible outcome occurs will now be covered.

If the declarative memory request is successful, a 'recall_correct' production would fire printing Edwards location of (0,0). The spatial focus now moves towards coordinate (0,0) through the firing of two 'move_left' production sequences until Edward is found. Once the spatial focus is on Edward, a 'confirm_conclusion' production validates the conclusion as "True, Edward is Left of Travis!".

The second possibility that may occur after a declarative memory request is a memory error of omission by a "recall_omission" production. Now what happens after the agent realizes they cannot remember is difficult to ascertain. In the multidimensional relational reasoning task, because one of the goals of Cortes et al. (2021) was to provide normalized data reaction time across varied stimulus properties, participants were allowed unlimited time per problem. This serves as problematic for our models as participants could have implemented a wide variety of strategies to overcome such an error of omission. Because the current literature lacks theories as to what happens in instances of memory errors of omission during relational reasoning problems, it was decided that during the inspection phase, if such memory errors were to occur, all possible subsequent productions involving conclusions would have an equal chance of being selected, so in the case of a choosing a conclusion this results in a random equal chance of selecting a true or false verification.

The third possibility that may occur instead of a successful memory request or memory error of omission is a memory error of commission in which the ACT-R agent would believe that Edward's location is incorrectly Brian's at coordinate (1,0). Therefore, beginning with a

'recall_comission' production, the ACT-R agent prints "Edward is at location (1,0)". This is then followed by the productions of 'move_left' and 'confirm_conclusion', which interestingly results in true conclusion verification despite the incorrect location of the to be located object. The occurrence of a memory error of commission, yet still inferring a correct conclusion, highlights an overlooked problem in the literature of how different memory errors may result in different responses or solutions. In the case of our example, this issue is especially pertinent and possibly problematic, as our models also predict that current accuracy rates of spatial relational reasoning problems may be overinflated due to false positives due to errors of commission.

Future work may be conducted to create types of relational reasoning problems to delineate between the types of memory errors which may occur during the reasoning process. For example, a reasoning problem's conclusion that requires the spatial focus to be moved to the middle of the mental model in establishing a reference object may allow for errors of commission to be identified depending on the position of the to be located object.

Ch 4) Results

Four ACT-R models were created to replicate the findings in Cortes et al. (2021) of decreased accuracy rates in spatial relational reasoning for more complex conditions of increased premises and dimension problems. All models began with the same default parameter values; however, four variations of these models were also run for a total of 16 models - four models based on four questions of the MRRT, with four variations per model. The first set of variations were run with no alteration of parameters; the subsequent three variations each increased the parameter of threshold either by 0.1, 0.2, or 0.3 for each move of the spatial focus. This increase in threshold was done with each move of the spatial focus so as to be in accordance with PRISM

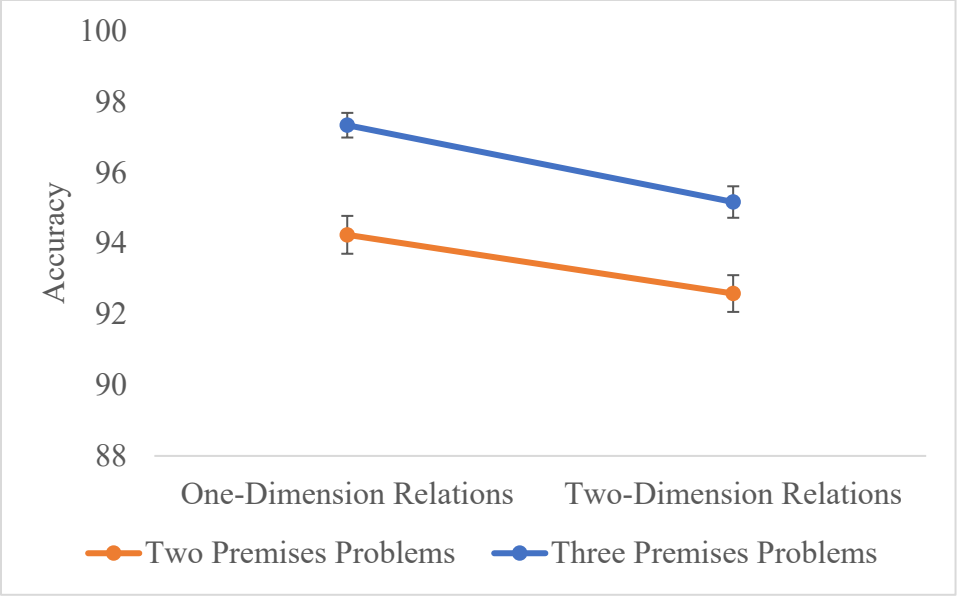
theory's of use of accumulated moves of the spatial focus as being a measurement of task difficult. In doing so, our ACT-R models attempt to demonstrate the stress of increased working memory load on the cognitive agent and the subsequent memory errors that would occur as a consequence.

For each relational reasoning problem (2P1D, 2P2D, 3P1D, 3P2D), each model variation was run equivalent to the number of participants in Cortes et al.'s (2021) X condition (n=310), which gave an average proportion of accuracy per model. To determine 95% confidence intervals, an n=30 was collected per model, requiring 9,300 iterations per model variation, for a total of 37,200 iterations per relational reasoning problem condition. A student's t-distribution was then used to determine confidence intervals.

All relational reasoning problems of our first set of ACT-R models with a 0.0 threshold increase per move of spatial focus produced similar results. At a threshold increase of 0.0 relational problems produced the following accuracy rates: 2P1D (n=30, M=94.24, SD=1.43. CI=0.53), 2P2D (n=30, M=92.59, SD=1.39, CI=0.52), 3P1D (n=30. M=97.34, SD=0.93, CI=0.35), 3P2D (n=30, M=95.17, SD=1.19, CI=0.44) - see figure 3.

Figure 3

Accuracy Rates for ACT-R Models with Threshold Increase Set to 0.0

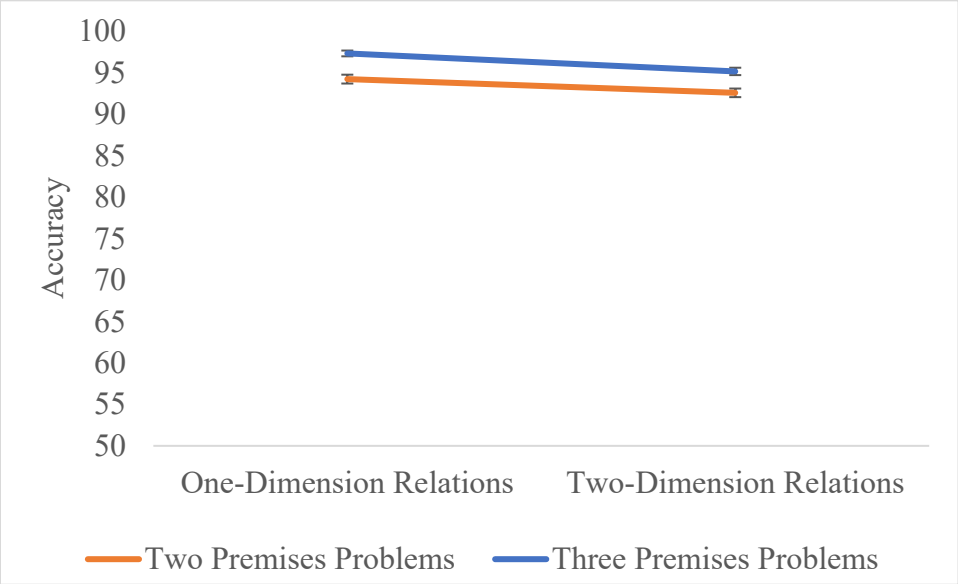


Note. ACT-R models with a threshold increase of 0.0 per move of the spatial focus.

At first glance, these results seem quite surprising, with three premise problems having a higher degree of accuracy. However, with a change of the Y-axis to a broader range of 50% to 100% accuracy rates, it becomes evident that these models all perform similarly to one another and produce a little correlation between moves of the spatial focus and accuracy rates – see Figure 4 and Figure 5.

Figure 4

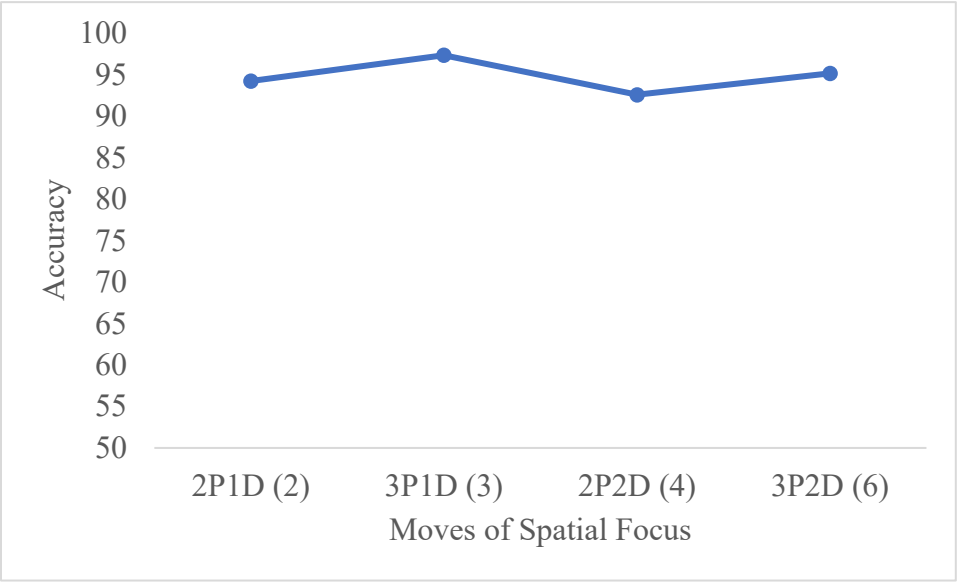
Accuracy Rates of ACT-R Models with Threshold Increase Set to 0.0 with Expanded Y-Axis



Note. ACT-R models with a threshold increase of 0.0 per move of the spatial focus and an expanded Y-axis.

Figure 5

Correlation of Accuracy Rates and Total Moves of Spatial Focus with Threshold Increase Set to 0.0



Note. Y-axis is the accuracy rate of ACT-R models with a threshold increase of 0.0 per move of the spatial focus. The X-axis is the total moves of spatial focus per problem placed in ascending order of moves of the spatial focus.

A potential reason for the similar output of these models might be, as Boeddinghaus et al. (2006) mentioned, that default ACT-R mechanisms may be too fast for such complex spatial reasoning tasks, especially since there is no simulation of the reading and interpretation of the premises. Due to this issue of speed of the models, not enough time may have passed, so that ACT-R's decay parameter has an impact on chunks not being able to reach the required threshold for a correct memory recall, thereby leading to an increase in errors of omission.

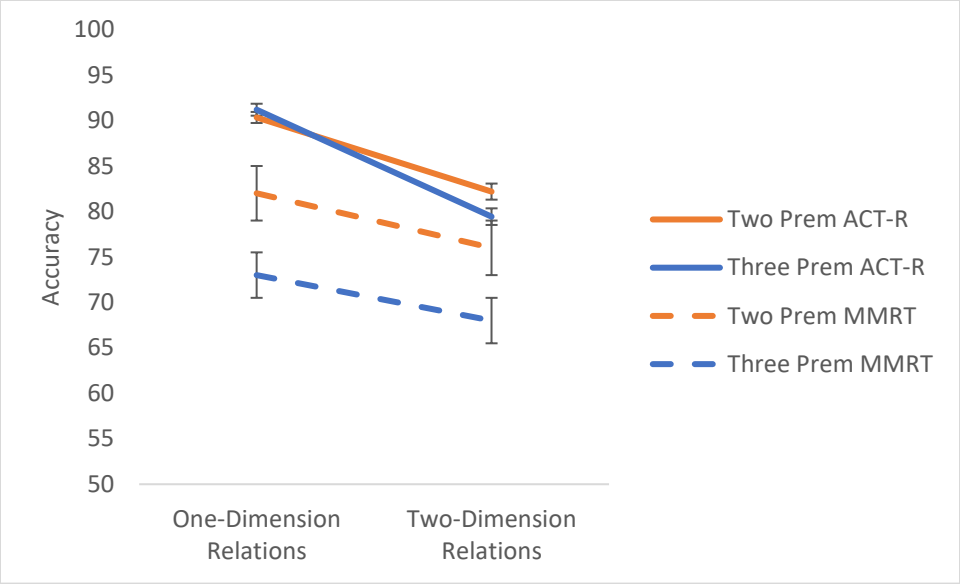
Cortes et al. (2021) reported average reaction times in seconds per stimulus property, and these were two premise problems (M=27.56, SD=15.18), three premise problems (M=41.21, SD=25.88), one-dimension problems (M=30.77, SD=19.15), and two-dimension problems (M= 22.48, SD=22.48). Like in Boeddinghaus et al. (2006), our models replicated more complex

questions containing more dimensions and premises, resulting in longer processing times. However, these were drastically below empirical reaction times. Our ACT-R models with a threshold increase at 0.0 resulted in the following mean production times in seconds: 2P1D (M=0.628), 2P2D (M=0.796), 3P1D (M=0.813), 3P2D (M=1.087). The subsequent models attempt to account for and make up for these equivalent accuracy rates this with a systematic increase of threshold parameter per move of the spatial focus to reflect increased working memory demand.

As the threshold increase manipulation is implemented, an immediate effect is seen in decreased accuracy rates corresponding to moves of the spatial focus. At a threshold increase of 0.1 per move of the spatial focus, relational problems produced the following accuracy rates: 2P1D (n=30, M=90.34, SD=1.63 CI=0.61), 2P2D (n=30, M=82.18, SD=2.37, CI=0.88), 3P1D (n=30. M=91.19 , SD=1.76, CI=0.66), 3P2D (n=30, M=79.43, SD=2.45, CI=0.91). These results do not match participant data, see Figure 6, but do begin showing a correlation between decreased accuracy rates and increased moves of the spatial focus, see Figure 7.

Figure 6

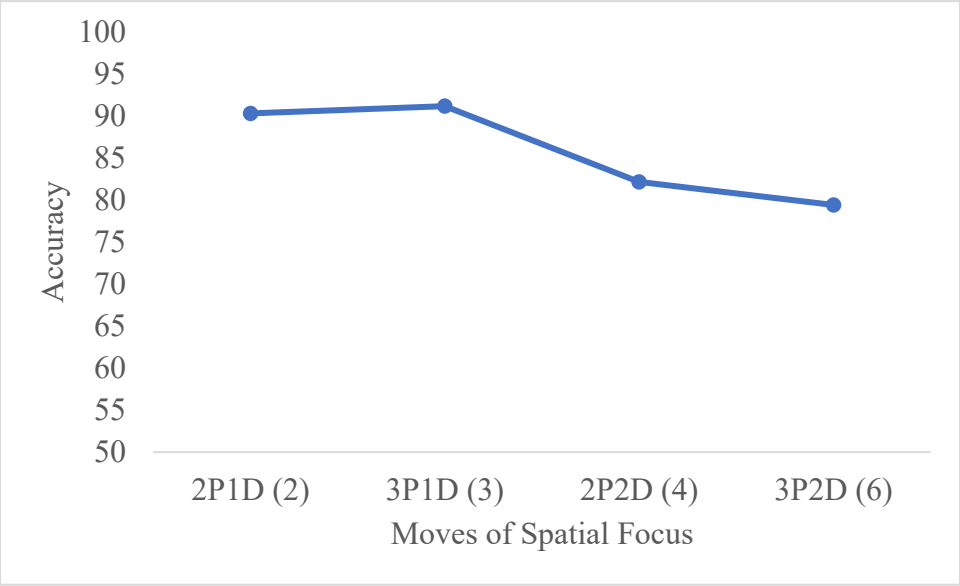
Accuracy Rates Comparison of ACT-R Models and MRRT results with Threshold Increase Set to 0.1



Note. Accuracy rates of ACT-R models with a threshold increase of 0.1 per move of the spatial focus along with MRRT results of predicted marginal means of dimension x premise reported by Cortes et al. (2021).

Figure 7.

Correlation of ACT-R Models Accuracy Rates and Total Moves of Spatial Focus with Threshold Increase Set to 0.1

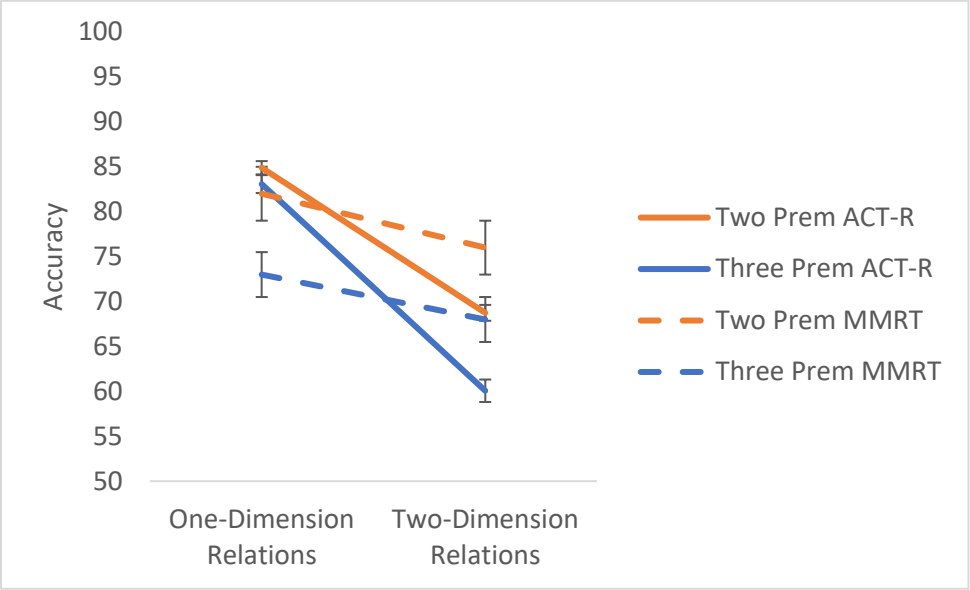


Note. Y-axis is the accuracy rate of ACT-R models with a threshold increase of 0.1 per move of the spatial focus. The X-axis is the total moves of spatial focus per problem placed in ascending order of moves of the spatial focus.

At a threshold increase of 0.2 per move of the spatial focus, relational problems produced the following accuracy rates: 2P1D (n=30, M=84.89, SD=1.98 CI=0.74), 2P2D (n=30, M=68.74, SD=2.37, CI=0.88), 3P1D (n=30. M=83.07 , SD=2.67, CI=1.0), 3P2D (n=30, M=60.07, SD=3.35, CI=1.25). Models approach closer to participant data trends, see Figure 8, and correlation between decreased accuracy rates and increased moves of the spatial focus continues, see Figure 9.

Figure 8

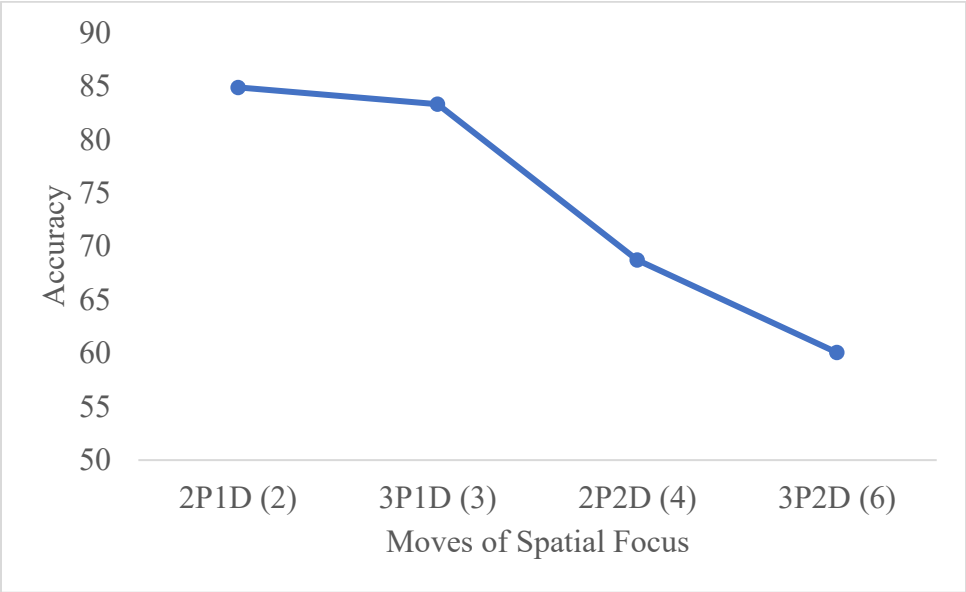
Accuracy Rates Comparison of ACT-R Models and MRRT results with Threshold Increase Set to 0.2



Note. Accuracy rates of ACT-R models with a threshold increase of 0.2 per move of the spatial focus along with MRRT results of predicted marginal means of dimension x premise reported by Cortes et al. (2021).

Figure 9

Correlation of Accuracy Rates and Total Moves of Spatial Focus with Threshold Increase Set to 0.2

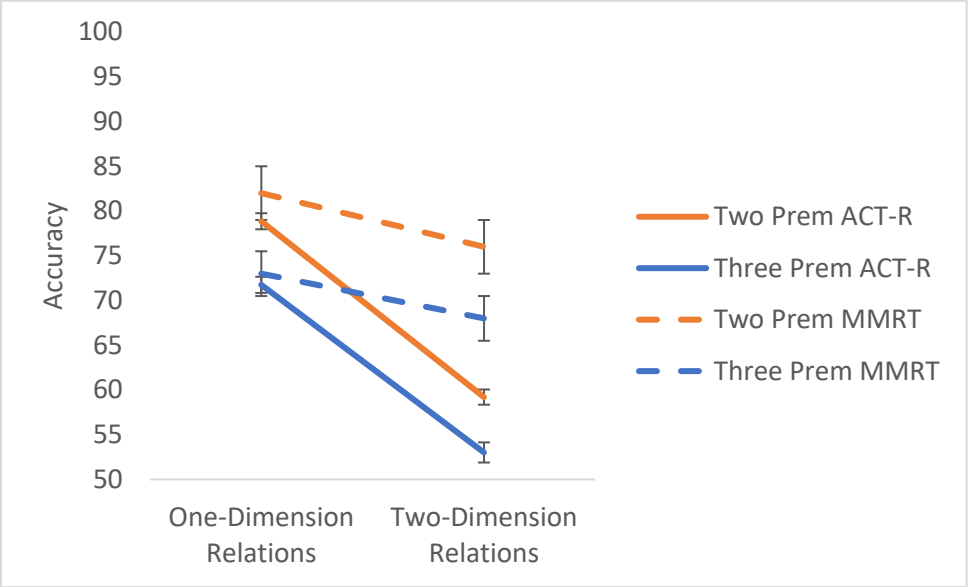


Note. Y-axis is the accuracy rates of ACT-R models with a threshold increase of 0.2 per move of the spatial focus. The X-axis is the total moves of spatial focus per problem placed in ascending order of moves of the spatial focus.

At a threshold increase of 0.3 per move of the spatial focus, relational problems produced the following accuracy rates: 2P1D (n=30, M=78.86, SD=2.40 CI=0.90), 2P2D (n=30, M=59.21, SD=2.27, CI=0.85), 3P1D (n=30. M=71.75 , SD=2.42, CI=0.91), 3P2D (n=30, M=53.02, SD=3.02, CI=1.13). A fifth variation was not created because that would entail a more than doubling of default parameters per move of the spatial focus. However, compared to the other variations of the models produced, a threshold increase of 0.3 per move of the spatial focus provides the closest fit to human data, see Figure 10, and continues to show a correlation between decreased accuracy rates and increased moves of the spatial focus continues, see Figure 11.

Figure 10.

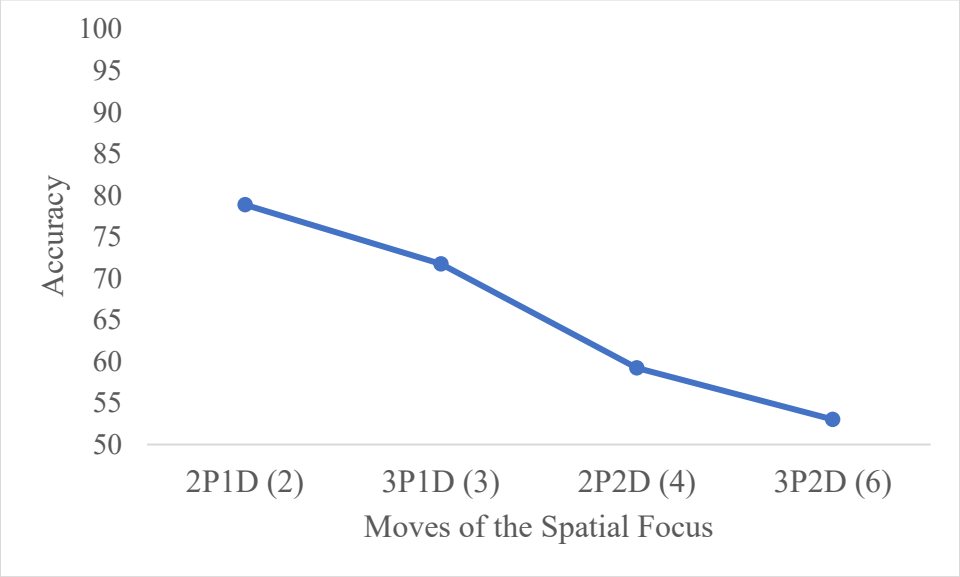
Accuracy Rates Comparison of ACT-R Models and MRRT results with Threshold Increase Set to 0.3



Note. Accuracy rates of ACT-R models with a threshold increase of 0.3 per move of the spatial focus along with MRRT results of predicted marginal means of dimension x premise reported by Cortes et al. (2021).

Figure 11.

Correlation of Accuracy Rates and Total Moves of Spatial Focus with Threshold set to 0.3



Note. *Y-axis is the* accuracy rates of ACT-R models with a threshold increase of 0.3 per move of the spatial focus. The X-axis is the total moves of spatial focus per problem placed in ascending order of moves of the spatial focus.

For question six of the MMRT (2P1D), Cortes et al. (2021) reported an average accuracy rate of M=83.5. Continuous and discontinuous determinate spatial problems of two premises and one dimension produced an overall average of M=82.62 and a range of 73.56% - 90.97% accuracy. Both our models run at 0.3 threshold increase (M=78.86), and 0.2 threshold increase (M=84.89) were within the range of empirical data. Both continuous and discontinuous premise orders of determinate problem averages are used here for a comparative range of values despite our models only being of determinate continuous problems since Cortes et al. (2021) reported that there was no effect found for premise order. Additionally, the MRRT only contains a small number of continuous spatial relational reasoning problems: one for 2P1D, two for 2P2D, two for 3P1D, and five for 3P2D.

For question 36 of the MRRT (2P2D), Cortes et al. (2021) reported an average accuracy rate of M=81.25. Continuous and discontinuous determinate spatial problems of two premises and two dimensions produced an overall average of M=74.86 and a range of 64.88% - 81.25% accuracy. Our models running at a 0.3 threshold increase (M=59.21) did not fall within this aforementioned range of values, but our models running at a 0.2 threshold increase (M=68.74) were within the range of empirical data suggesting that a better threshold increase for modelling empirical data in Python ACT-R may be somewhere between 0.2 and 0.3.

For question twenty-one of the MMRT (3P1D), Cortes et al. (2021) reported an average accuracy rate of M=79.03. Continuous and discontinuous determinate spatial problems of three premises and one dimension produced an overall average of M=76.99 and a range of 67.63% - 84.84% accuracy. Our models running at a 0.3 threshold increase (M=71.75) and 0.2 threshold increase (M=83.07) both fell within the empirical accuracy rates. Notice, however, how close in value the accumulated averages of Cortes et al (2021) two premise two-dimension problems (M=74.86) are to three premise one-dimension problems (M=79.03).

For question sixty-six of the MRRT (3P2D), Cortes et al. (2021) reported an average accuracy rate of M=73.91. Continuous and discontinuous determinate spatial problems of three premises and two dimensions produced an overall average of M=68.21% and a range of 58.39% - 75.16% accuracy. Our models running at a 0.3 threshold increase (M=53.02) did not fall within this empirical range, but our models running at a 0.2 threshold increase did (M=60.07).

While our ACT-R models were able to produce accuracy rates that fell within the ranges of empirical participant data and were able to model the difference in accuracy found in Cortes et al. (2021), especially in the most complex conditions of three premise-two-dimension problems compared to the least complex two premise-one dimension problems, our models primarily did

this through means of dimensions. The reliance on increased dimensions resulting in increased moves of a spatial focus to model the difficulty of relational reasoning problems is in accordance with PRISMS use of accumulated moves of the spatial focus as a proxy for difficulty measurement. However, it falls short at modelling the impact of premise increase for the interaction effect reported by Cortes et al. (2021). These discrepancies are not surprising given how fast our models process each problem, so that the effect ACT-R's theory of memory decay may have on errors of omission is very weak. Future work may explore how to better model the effects of premise understanding, something which PRISM theory offloads to an assumed "semantic parser".

Ch 5) Discussion

This thesis sought to investigate how the different types of memory errors of omission and commission may be computationally modelled to provide a theoretically unifying account of qualitatively different cognitive reasoning processes and quantitatively different accuracy rates on experimental measurements of spatial relational reasoning problems.

Four ACT-R models were created, each based on a different question of complexity from the multidimensional relational reasoning task (MRRT) (Cortes et al., 2021). Preferred inference in reasoning with spatial mental models (PRISM) was used to facilitate the organization and representation of knowledge used by these ACT-R models. Four additional variations per model were also analyzed to determine a good fit for a systematic threshold parameter increase based on PRISM's measurement of problem difficulty - the accumulated moves of a spatial focus.

Our ACT-R model's results replicated similar trends found in the empirical participant data provided by Cortes et al (2021) of a negative relationship between accuracy rates and increased premise and dimensions complexity conditions per spatial relational reasoning

problem. Accuracy rates were found to decrease with each increase of premises and dimensionality complexity.

In doing so, this thesis was successful in demonstrating how different memory errors of omission and commission for spatial relational reasoning problems due to increased working memory demands may lead to qualitatively different cognitive reasoning processes and quantitatively different accuracy rates. Our models argue for a possible existing relationship for the difficulty of a problem as a function of movements of a cognitive spatial focus and an increase of the threshold required to be passed for a successful declarative memory recall.

Modelling of empirical results primarily based on moves of a spatial focus, however, did show to be limited in capturing the effects of additional premises. Due to 50-millisecond production firings of our current ACT-R models, it may be that premises were processed too quickly to model participant data and consequently did not have much of an impact on accuracy rates by means of ACT-R's theory of memory decay. Future work may seek how to better simulate the reading and interpretation of premises for spatial relational reasoning problems to better model such tasks, something which is not part of PRISM theory which takes as assumed knowledge and delegates such tasks to a separate semantic parser. Additionally, the dependency on a strictly additive increase in threshold parameter based on moves of a spatial focus may be considered problematic.

This thesis attempted to demonstrate findings not by highlighting the significant differences between conditions, as is often done in hypothesis testing and experimental psychology, but rather attempts to argue for a *significant sameness* between our computational models and participant data based on specific theories of cognitive architectures and spatial reasoning. While our models are largely assessed through similar trends demonstrated on

graphical representations and ranges of accuracy rates per condition, more advanced methods for tests of equivalency for cognitive modelling exist that future work may incorporate, see West & Stewart (2010). The investigation conducted for this thesis did however also reveal several other worthwhile insights into cognitive modelling of relational reasoning problems and how current experiment measures may be improved.

The most pertinent concern in this thesis regarding current methods of measuring spatial relational reasoning problems is that they do not account for how differences in working memory error types may lead to qualitatively different reasoning processes. In particular, our models demonstrated how errors of commission may lead to false positives when the conclusion to be verified begins at the last location an object was inserted, and that location is edge bound in the mental model - something which all spatial questions of the MRRT contained. This is especially problematic when assessing accuracy rates in aggregate participant responses, an old criticism demonstrating the importance of identifying individual differences in micros strategies and memory errors (Newell, 1973).

The incorporation of individual differences in modelling has become a forefront issue with current work in computational cognitive modelling (Ragni, Brand & Riesterer, 2021), arguing that future models should focus more on predicting any individuals' behaviour. Preferred model theory and PRISM touch on individual differences in the construction and variation stages of mental model reasoning by use of either the first fit strategy or first-free-fit strategy of object placement. However, as our models demonstrate, current theories remain unsatisfactory as to what occurs after an error of omission occurs in which an agent can not recall a piece of information. When such an error occurs, participants may either begin the problem anew, guess

true or false – such as in the case with our models, or incorporate a different micro-strategy entirely.

Future experimental work may be altered to address such concerns presented. For example, Cortes et al. (2021) allotted participants as much time required to solve each problem of the MRRT; this could be considered problematic as participants could have implemented a multitude of various strategies or number of attempts to solve each problem until they believed they had achieved the correct answer. Ways in which future work may gain a better understanding of individual differences may be through talk aloud protocols, the presentation of premises one at a time, and the inclusion of questions addressing errors of commission which, as previously stated, current measures do not delineate between these errors and may have an inflated accuracy rate due to false positives. A future manipulation of spatial relational reasoning questions may be constructed, which requires the reference object of a mental model to be shifted away from the edge-bound position of the object last inserted so as to increase the likelihood of an error of commission delineating between an incorrect response and a false positive due to the conclusion required to be generated or verified.

Appendix

Appendix A: Python ACT-R Model for Two Premises-One Dimension Problem

```
import python_actr
from python_actr import *
import csv

#-----------------------------------------------------

# 2P1D - MRRT Q.6

# P1) Edward is to the left of Derek.
# P2) Derek is to the left of Travis.
# C) EDWARD IS TO THE LEFT OF TRAVIS.

# Avg_Acc : 0.835748792
# Avg_RT : 26.36728502
# Relation: Spatial
# Premises: Two Premises
# Dimensions: One Dimension
# Answer: True
# Solution: True
# Premise Order: Continuous
# CPhrasing: Afirst

# Accumulated number of moves of spatial focus for construction : 2

# -----------------------------------------------------

class MyEnvironment(python_actr.Model):
    pass


class MyAgent(ACTR):

    spatial_focus=Buffer()
    goal_focus=Buffer()
    DMbuffer=Buffer()

    production_time=0.05
    production_sd=0.01

    DM=Memory(DMbuffer, latency=0.05, threshold=0.3, maximum_time=10.0, finst_size=0, finst_time=3.0)
    DMN=DMNoise(DM, noise=0.3, baseNoise=0.3)
    DMBaseLevel(DM, decay=0.5, limit=None)

    dm_spread=DMSpreading(DM,goal_focus)
    dm_spread.strength=1
    dm_spread.weight[goal_focus]=0.0

    partial=Partial(DM,strength= 0.0, limit= 0.0)
    partial.similarity('edward', "derek", 0)
    partial.similarity('travis', 'derek', 0)
    partial.similarity('edward', 'travis', 0)



#####----------------------CONSTRUCTION-----------------------------------#####

    goal_focus.set('read_premise_1')
    spatial_focus.set('object:na location:na direction:na')

    # 1-construction-read_premise
    def read_premise_1 (goal_focus='read_premise_1', spatial_focus='object:na location:na direction:na'):
        print('Premise 1: Edward is to the left of Derek.')
        spatial_focus.set('object:na location:na direction:na')
        goal_focus.set('insert_edward_2')

    # 2-construction-insert_edward
    def insert_edward_2 (goal_focus='insert_edward_2', spatial_focus='object:na location:na direction:na'):
        print("Insert Edward")
        DM.add('object:edward location:0.0 annotation:na')
        spatial_focus.set('object:edward location:0.0 direction:na') # (E)
        goal_focus.set('move_right_3')

```

```
    # 3-construction-move_right
    def move_right_3 (goal_focus='move_right_3', spatial_focus='object:edward location:0.0 direction:na'):
        print("Move right")
        DM.threshold+=.3
        spatial_focus.set('object:na location:1.0 direction:right') # E ()
        goal_focus.set('insert_derek_4')

    # 4-construction-insert_derek
    def insert_derek_4 (goal_focus='insert_derek_4', spatial_focus='object:na location:1.0 direction:right'):
        print("Insert Derek")
        DM.add('object:derek location:1.0 annotation:na')
        spatial_focus.set('object:derek location:1.0 direction:right') # E (D)
        goal_focus.set('read_premise_5')

    # 5-construction-read_premise
    def read_premise_5 (goal_focus='read_premise_5', spatial_focus='object:derek location:1.0 direction:right'):
        print('Premise 2: Derek is to the left of Travis')
        spatial_focus.set('object:derek location:1.0 direction:right')
        goal_focus.set('move_right_6')

    # 6-construction-move_right
    def move_right_6 (goal_focus='move_right_6', spatial_focus='object:derek location:1.0 direction:right'):
        print("Move right")
        DM.threshold+=.3
        spatial_focus.set('object:na location:2.0 direction:right') # E D ()
        goal_focus.set("insert_travis_7")

    # 7-construction-insert_travis
    def insert_travis_7 (goal_focus='insert_travis_7', spatial_focus='object:na location:2.0 direction:right'):
        print("Insert Travis")
        DM.add('object:travis location:2.0 annotation:na')
        spatial_focus.set('object:travis location:2.0 direction:right') # E D (T)
        goal_focus.set('read_conclusion_8')


######--------------------INSPECT----------------------------#######

    # This function is used to write the conclusions True(1)/False(0) to a csv file
    def decision (self,answer):
        myfile=open("01-2P1D_MMRT_DATA.csv", "a", newline="")
        wr=csv.writer(myfile)
        wr.writerow(answer)
        myfile.close()

    # 8-inspection-read_conclusion
    def read_conclusion_8 (goal_focus='read_conclusion_8', spatial_focus='object:travis location:2.0 direction:right'):
        print('Conclusion: Edward is to the left of Travis')
        spatial_focus.set('object:travis location:2.0 direction:right') # spatial_focus still on Travis after construction : E D (T)
        goal_focus.set('request_edward_9')

    # 9-inspection-request_edward
    def request_edward_9 (goal_focus='request_edward_9', spatial_focus='object:travis location:2.0 direction:right'):
        print("Recalling Edward's location...(9)")
        DM.request('object:edward location:!2.0') #!2.0 because spatial spatial_focus is already on this location
        spatial_focus.set('object:travis location:2.0 direction:right')
        goal_focus.set('recall_edward_10A1B1C1')
        # to allow branching, all possible productions have the same value in their buffers which then change to continue down the selected branch path

### Branch ###


###-----Correct Recall of Edwards Position-----###

    # 10A1-inspection-recall_correct
    def recall_correct_10A1 (goal_focus='recall_edward_10A1B1C1', spatial_focus='object:travis location:2.0 direction:right',  DMbuffer='location:0.0'):
        print('Edward is at location 0.0 (recall_correct) (10A1).')
        spatial_focus.set('object:travis location:2.0 direction:right')
        goal_focus.set('move_left_10A2')

    # 10A2-inspection-move_left
    def move_left_10A2 (goal_focus='move_left_10A2', spatial_focus='object:travis location:2.0 direction:right'):
        print("Move left, spatial focus on Derek (recall_correct) (10A2).")
        DM.threshold+=.3
        spatial_focus.set('object:derek location:1.0 direction:left') # E (D) T
        goal_focus.set('move_left_10A3')

    # 10A3-inspection-move_left
    def move_left_10A3 (goal_focus='move_left_10A3', spatial_focus='object:derek location:1.0 direction:left'):
        print('Move left, spatial focus on Travis. (recall_correct) (10A3)')
        DM.threshold+=.3
        spatial_focus.set('object:edward location:0.0 direction:left') # (E) D T
        goal_focus.set('confirm_conclusion_10A4')

    # 10A4-inspection-confirm_conclusion (TRUE)
    def confirm_conclusion_10A4 (goal_focus='confirm_conclusion_10A4', spatial_focus='object:edward location:0.0 direction:left'):
        print("True, Edward is to the left of Travis (recall_correct) (10A4).")
        self.decision('1')
        self.stop()
```

```
162 ###-----Error of Omission of Edwards Position-----###
163
164     # 10B1-inspection-recall_omission
165     # spatial_focus still on Travis after construction : E D (T)
166     def recall_omission_10B1 (goal_focus='recall_edward_10A1B1C1', spatial_focus='object:travis location:2.0 direction:right', DM='error:True', DMbuffer=None):
167         print("I don't remember the Edward's location (recall_omission)") # introduces need for theories about micro-strategies, we assume a 50/50 chance decision.
168         spatial_focus.set('object:travis location:2.0 direction:right')
169         goal_focus.set('confirm_conclusion_10B2A1B1') # to allow branching
170
171 ### Branch ###
172
173     # 10B1A1-inspection-confirm_conclusion (TRUE)
174     def confirm_conclusion_10B2A1 (goal_focus='confirm_conclusion_10B2A1B1', spatial_focus='object:travis location:2.0 direction:right'):
175         print('True, Edward is to the left of Travis (recall_omission-10B2A1)')
176         self.decision('1')
177         self.stop()
178
179
180     # 10B1B1-inspection-confirm_conclusion (FALSE)
181     def confirm_conclusion_10B2B1 (goal_focus='confirm_conclusion_10B2A1B1', spatial_focus='object:travis location:2.0 direction:right'):
182         print('False, Edward is not to the left of Travis (recall_omission-10B2B1)')
183         self.decision('0')
184         self.stop()
185
186
187 ###-----Error of Commission of Edwards Position for Derek-----###
188
189     # 10C1-inspection-recall_commission
190     def recall_commission_C1 (goal_focus='recall_edward_10A1B1C1', spatial_focus='object:travis location:2.0 direction:right',  DMbuffer='location:1.0'):
191         print('I recall Edwards location to be 1.0 (recall_commission) (Derek location)')
192         spatial_focus.set('object:travis location:2.0 direction:right')
193         goal_focus.set('move_left_10C2')
194
195     # 10C2-inspection-move_left
196     def move_left_10C2 (goal_focus='move_left_10C2', spatial_focus='object:travis location:2.0 direction:right'):
197         print('Move left')
198         DM.threshold+=.3
199         spatial_focus.set('object:derek location:1.0 direction:left') # E (D) T
200         goal_focus.set('confirm_conclusion_10C3')
201
202     # 10C3-inspection-confirm_conclusion
203     def confirm_conclusion_10C3 (goal_focus='confirm_conclusion_10C3', spatial_focus='object:derek location:1.0 direction:left'):
204         print('True, Edward is Left of Travis (recall_commission-10C3)') # E (D) T
205         self.decision('1')
206         self.stop()
207
208
209 ##################################################################
210
211
212 nico = MyAgent()
213 thesis = MyEnvironment()
214 thesis.agent=nico
215 python_actr.log_everything(thesis)
216 thesis.run()
217 python_actr.finished()
218
```

Appendix B: Python ACT-R Model for Two Premises-Two Dimension Problem

```python
import python_actr
from python_actr import *
import csv

#-----------------------------------------------------

### 2P2D - MRRT Q.36

# P1) Edward is below and to the left of Derek.
# P2) Derek is below and to the left of Travis.
# C) EDWARD IS BELOW AND TO THE LEFT OF TRAVIS.


# Relation: Spatial
# Premise: Two
# Dimensions: Two
# Answer: True
# Solution: True
# Premise Order: Continuous
# CPhrasing: Afirst

# Avg_Acc: 0.8125
# Avg_RT: 27.92271635

# Accumulated number of moves of spatial focus for construction : 4

# -----------------------------------------------------

class MyEnvironment(python_actr.Model):
    pass


class MyAgent(ACTR):

    spatial_focus=Buffer()
    goal_focus=Buffer()
    DMbuffer=Buffer()

    production_time=0.05
    production_sd=0.05

    DM=Memory(DMbuffer, latency=0.05, threshold=0.3, maximum_time=10.0, finst_size=0, finst_time=3.0)
    DMN=DMNoise(DM, noise=0.3, baseNoise=0.3)
    DMBaseLevel(DM, decay=0.5, limit=None)

    dm_spread=DMSpreading(DM,goal_focus)
    dm_spread.strength=1
    dm_spread.weight[goal_focus]=0.0

    partial=Partial(DM,strength= 0.0, limit= 0.0)
    partial.similarity('edward','derek', 0)
    partial.similarity('edward','travis', 0)

    partial.similarity('derek','travis', 0)
```

```
#####----------------------CONSTRUCTION-----------------------------------#####

    goal_focus.set('read_premise_1')
    spatial_focus.set('object:na location:na direction:na')

    # 1-construction-read_premise
    def read_premise_1(goal_focus='read_premise_1', spatial_focus='object:na location:na direction:na'):
        print('Premise 1: Edward is below and to the left of Derek. (1)')
        spatial_focus.set('object:na location:0.0 direction:na')
        goal_focus.set('insert_edward_2')

    # 2-construction-insert_edward
    def insert_edward_2(goal_focus='insert_edward_2' , spatial_focus='object:na location:0.0 direction:na'):
        print('Insert Edward (2)')
        DM.add('object:edward location:0.0 annotation:na')
        spatial_focus.set('object:edward location:0.0 direction:na')
        goal_focus.set('move_above_3')

    # 3-construction-move_above
    def move_above_3(goal_focus='move_above_3', spatial_focus='object:edward location:0.0 direction:na'):
        print('Move above (3)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:0.1 direction:above')
        goal_focus.set('move_right_4')

    # 4-construction-move_right
    def move_right_4(goal_focus='move_right_4', spatial_focus='object:na location:0.1 direction:above'):
        print('Move right (4)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:1.1 direction:right')
        goal_focus.set('insert_derek_5')

    # 5-construction-insert_derek
    def insert_derek_5(goal_focus='insert_derek_5', spatial_focus='object:na location:1.1 direction:right'):
        print('Insert Derek (5)')
        DM.add('object:derek location:1.1 annotation:na')
        spatial_focus.set('object:derek location:1.1 direction:right')
        goal_focus.set('read_premise_6')


    # 6-construction-read_premise
    def read_premise_6(goal_focus='read_premise_6', spatial_focus='object:derek location:1.1 direction:right'):
        print('Premise 2: Derek is below and to the left of Travis.')
        spatial_focus.set('object:derek location:1.1 direction:right')
        goal_focus.set('move_above_7')

    # 7-construction-move_above
    def move_above_7(goal_focus='move_above_7', spatial_focus='object:derek location:1.1 direction:right'):
        print('Move above (7)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:1.2 direction:above')
        goal_focus.set('move_right_8')

    # 8-construction-move_right
    def move_right_8(goal_focus='move_right_8', spatial_focus='object:na location:1.2 direction:above'):
        print('Move right (8)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:2.2 direction:right')
        goal_focus.set('insert_travis_9')

    # 9-construction-insert_travis
    def insert_travis_9(goal_focus='insert_travis_9', spatial_focus='object:na location:2.2 direction:right'):
        print('Insert Travis (9)')
        DM.add('object:travis location:2.2 annotation:na')
        spatial_focus.set('object:travis location:2.2 direction:right')
        goal_focus.set('read_conclusion_10')
```

```
126 ######--------------------INSPECT----------------------------#######
127
128     # This function is used to write the conclusions True(1)/False(0) to a csv file
129     def decision (self,answer):
130         myfile=open("02-2P2D_MMRT_DATA.csv", "a", newline="")
131         wr=csv.writer(myfile)
132         wr.writerow(answer)
133         myfile.close()
134
135     #10-inspection-read_conclusion
136     def read_conclusion_10(goal_focus='read_conclusion_10',spatial_focus='object:travis location:2.2 direction:right'):
137         print('Conclusion: Edward is below and to the left of Travis (10).')
138         spatial_focus.set('object:travis location:2.2 direction:right')
139         goal_focus.set('request_edward_11')
140
141     #11-inspection-request_edward
142     def request_edward_11(goal_focus='request_edward_11', spatial_focus='object:travis location:2.2 direction:right'):
143         print('I am recalling Edward location...(11)')
144         DM.request('object:edward location:!2.2')
145         spatial_focus.set('object:travis location:2.2 direction:right')
146         goal_focus.set('recall_edward_12A1B1C1')
147
148 ##########################################################################################
149 #-------------------BRANCH-12-A1-B1-C1----------------------------------------------
150 ##########################################################################################
151
152 ###-----------12A1-Recall of Edward location Correct----------###
153
154     #12A1-inspection-recall_correct
155     def recall_correct_12A1(goal_focus='recall_edward_12A1B1C1', spatial_focus='object:travis location:2.2 direction:right', DMbuffer='location:0.0'):
156         print('I recall Edward location to be (0.0) (recall_correct) (12A1)')
157         spatial_focus.set('object:travis location:2.2 direction:right')
158         goal_focus.set('move_below_12A2')
159
160     #12A2-inspection-move_below
161     def move_below_12A2(goal_focus='move_below_12A2', spatial_focus='object:travis location:2.2 direction:right'):
162         print('Move below (12A2)')
163         DM.threshold+=.3
164         spatial_focus.set('object:na location:2.1 direction:below')
165         goal_focus.set('move_left_12A3')
166
167     #12A3-inspection-move_left
168     def move_left_12A3(goal_focus='move_left_12A3', spatial_focus='object:na location:2.1 direction:below'):
169         print('Move left (12A3)')
170         DM.threshold+=.3
171         spatial_focus.set('object:derek location:1.1 direction:left')
172         goal_focus.set('move_below_12A4')
173
174     #12A4-inspection-move_below
175     def move_below_12A4(goal_focus='move_below_12A4', spatial_focus='object:derek location:1.1 direction:left'):
176         print('Move below (12A4)')
177         DM.threshold+=.3
178         spatial_focus.set('object:na location:1.0 direction:below')
179         goal_focus.set('move_left_12A5')
180
181     #12A5-inspection-move_left
182     def move_left_12A5(goal_focus='move_left_12A5', spatial_focus='object:na location:1.0 direction:below'):
183         print('Move left (12A5)')
184         DM.threshold+=.3
185         spatial_focus.set('object:edward location:0.0 direction:left')
186         goal_focus.set('confirm_conclusion_12A6')
187
188     #12A6-inspection-confirm_conclusion
189     def confirm_conclusion_12A6(goal_focus='confirm_conclusion_12A6', spatial_focus='object:edward location:0.0 direction:left'):
190         print('True, Edward is below and to the left of Travis! (12A6) (recall_correct)')
191         self.decision('1')
192         self.stop()
193
194 ##############################################################################################
195
```

```
###-------12B1-Error of Omission-------------####

    #12B1-inspection-recall_omission
    def recall_omission_12B1(goal_focus='recall_edward_12A1B1C1', spatial_focus='object:travis location:2.2 direction:right', DM='error:True', DMbuffer=None):
        print('I can not recall Edward location (12B1)')
        spatial_focus.set('object:travis location:2.2 direction:right')
        goal_focus.set('confirm_conclusion_12B2A1B1')

    #12B2A1-inspection-confirm_conclusion
    def confirm_conclusion_12B2A1(goal_focus='confirm_conclusion_12B2A1B1', spatial_focus='object:travis location:2.2 direction:right'):
        print('True, Edward is below and to the left of Travis! (12B2A1) (recall_omission)')
        self.decision('1')
        self.stop()

    #12B2B1-inspection-confirm_conclusion
    def confirm_conclusion_12B2B1(goal_focus='confirm_conclusion_12B2A1B1', spatial_focus='object:travis location:2.2 direction:right'):
        print('False, Edward is NOT below and to the left of Travis! (12B2B1) (recall_omission)')
        self.decision('0')
        self.stop()



##########################################################################################

###-------12C1-Error of Commission-------------####

    #12C1-recall_commission...derek
    def recall_commission_12C1(goal_focus='recall_edward_12A1B1C1', spatial_focus='object:travis location:2.2 direction:right', DMbuffer='location:1.1'):
        print('I recall Edward to be at location (1.1) (recall_commission) (12C1)')
        spatial_focus.set('object:travis location:2.2 direction:right')
        goal_focus.set('move_below_12C2')

    #12C2-move_below
    def move_below_12C2(goal_focus='move_below_12C2', spatial_focus='object:travis location:2.2 direction:right'):
        print('Move below (12C2)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:2.1 direction:below')
        goal_focus.set('move_left_12C3')

    #12C3-move_left
    def move_left_12C3(goal_focus='move_left_12C3', spatial_focus='object:na location:2.1 direction:below'):
        print('Move left (12C3)')
        DM.threshold+=.3
        spatial_focus.set('object:derek location:1.1 direction:left')
        goal_focus.set('confirm_conclusion_12C4')

    #12C4-confirm_conclusion
    def confirm_conclusion_12C4(goal_focus='confirm_conclusion_12C4', spatial_focus='object:derek location:1.1 direction:left'):
        print('True, Edward is below and to the left of Travis! (12C4) (recall_commission)')
        self.decision('1')
        self.stop()


##############################################################


nico = MyAgent()
thesis = MyEnvironment()
thesis.agent=nico
python_actr.log_everything(thesis)
thesis.run()
python_actr.finished()
```

Appendix C: Python ACT-R Model for Three Premises-One Dimension Problem

```python
import python_actr
from python_actr import *
import csv

#-----------------------------------------------------

### 3P1D - MRRT Q.21 ###

# P1) Edward is to the left of Derek.
# P2) Derek is to the left of Travis.
# P3) Travis is to the left of Brian.
# C) EDWARD IS TO THE LEFT OF BRIAN.


# Relation: Spatial
# Premise: Three
# Dimensions: One
# Answer: True
# Solution: True
# Premise Order: Continuous
# CPhrasing: Afirst
# Avg_Acc: 0.790322581
# Avg_RT: 29.98980645

# Accumulated number of moves of spatial focus for construction : 3

# -----------------------------------------------------

class MyEnvironment(python_actr.Model):
    pass


class MyAgent(ACTR):

    spatial_focus=Buffer()
    goal_focus=Buffer()
    DMbuffer=Buffer()

    production_time=0.05
    production_sd=0.01

    DM=Memory(DMbuffer, latency=0.05, threshold=0.3, maximum_time=10.0, finst_size=0, finst_time=3.0)
    DMN=DMNoise(DM, noise=0.3, baseNoise=0.3)
    DMBaseLevel(DM, decay=0.5, limit=None)

    dm_spread=DMSpreading(DM,goal_focus)
    dm_spread.strength=1
    dm_spread.weight[goal_focus]=0.0

    partial=Partial(DM,strength= 0.0, limit= 0.0)
    partial.similarity('edward','derek', 0)
    partial.similarity('edward','travis', 0)
    partial.similarity('edward','brian', 0 )

    partial.similarity('derek','travis', 0)
    partial.similarity('derek','brian', 0)

    partial.similarity('travis','brian', 0)
```

```
#####----------------------CONSTRUCTION-----------------------------------#####

    goal_focus.set('read_premise_1')
    spatial_focus.set('object:na location:na direction:na')

    # 1-construction-read_premise
    def read_premise_1(goal_focus='read_premise_1', spatial_focus='object:na location:na direction:na'):
        print('Premise 1: Edward is to the left of Derek.')
        spatial_focus.set('object:na location:na direction:na')
        goal_focus.set('insert_edward_2')

    # 2-construction-insert_edward
    def insert_edward_2(goal_focus='insert_edward_2', spatial_focus='object:na location:na direction:na'):
        print('Insert Edward (2)')
        DM.add('object:edward location:0.0 annotation:na')
        spatial_focus.set('object:edward location:0.0 direction:na')
        goal_focus.set('move_right_3')

    # 3-construction-move_right
    def move_right_3(goal_focus='move_right_3', spatial_focus='object:edward location:0.0 direction:na'):
        print('Move right (3)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:1.0 direction:right')
        goal_focus.set('insert_derek_4')

    # 4-construction-insert_derek
    def insert_derek_4(goal_focus='insert_derek_4', spatial_focus='object:na location:1.0 direction:right'):
        print('Insert Derek (4)')
        DM.add('object:derek location:1.0 annotation:na')
        spatial_focus.set('object:derek location:1.0 direction:right')
        goal_focus.set('read_premise_5')

    # 5-construction-read_premise
    def read_premise_5(goal_focus='read_premise_5', spatial_focus='object:derek location:1.0 direction:right'):
        print('Premise two: Derek is to the left of Travis.')
        spatial_focus.set('object:derek location:1.0 direction:right')
        goal_focus.set('move_right_6')

    # 6-construction-move_right
    def move_right_6(goal_focus='move_right_6', spatial_focus='object:derek location:1.0 direction:right'):
        print('Move right (6)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:2.0 direction:right')
        goal_focus.set('insert_travis_7')

    # 7-construction-insert_travis
    def insert_travis_7(goal_focus='insert_travis_7', spatial_focus='object:na location:2.0 direction:right'):
        print('Insert Travis (7)')
        DM.add('object:travis location:2.0 annotation:na')
        spatial_focus.set('object:travis location:2.0 direction:right')
        goal_focus.set('read_premise_8')

    # 8-construction-read_premise_8
    def read_premise_8(goal_focus='read_premise_8', spatial_focus='object:travis location:2.0 direction:right'):
        print('Premise three: Travis is to the left of Brian.')
        spatial_focus.set('object:travis location:2.0 direction:right')
        goal_focus.set('move_right_9')

    # 9-construction-move_right_9
    def move_right_9(goal_focus='move_right_9', spatial_focus='object:travis location:2.0 direction:right'):
        print('move_right_9')
        DM.threshold+=.3
        spatial_focus.set('object:na location:3.0 direction:right')
        goal_focus.set('insert_brian_10')

    # 10-construction-insert_brian_10
    def insert_brian_10(goal_focus='insert_brian_10', spatial_focus='object:na location:3.0 direction:right'):
        print('Insert Brian (10)')
        DM.add('object:brian location:3.0 annotation:na')
        spatial_focus.set('object:brian location:3.0 direction:right')
        goal_focus.set('read_conclusion_11')
```

```
135  ######--------------------INSPECT----------------------------#######
136
137      # This function is used to write the conclusions True(1)/False(0) to a csv file
138      def decision (self,answer):
139          myfile=open("03-3P1D_MMRT_DATA.csv", "a", newline="")
140          wr=csv.writer(myfile)
141          wr.writerow(answer)
142          myfile.close()
143
144
145      # 11-inspection-read_conclusion
146      def read_conclusion_11(goal_focus='read_conclusion_11', spatial_focus='object:brian location:3.0 direction:right'):
147          print('Conclusion: Edward is to the left of Brian. (11)')
148          spatial_focus.set('object:brian location:3.0 direction:right')
149          goal_focus.set('request_edward_12')
150
151      # 12-inspection-request_edward
152      def request_edward_12(goal_focus='request_edward_12', spatial_focus='object:brian location:3.0 direction:right'):
153          print('Recalling Edward location...(12)')
154          DM.request('object:edward location:!3.0')
155          spatial_focus.set('object:brian location:3.0 direction:right')
156          goal_focus.set('recall_edward_13A1B1C1D1')
157
158  ##########################################################################################
159  #--------------------BRANCH-13-A1-B1-C1-D1---------------------------------------------
160  ##########################################################################################
161
162  ###------------13A1-Recall of Edward location Correct-----------###
163
164      # 13A1-recall_correct
165      def recall_correct_13(goal_focus='recall_edward_13A1B1C1D1', spatial_focus='object:brian location:3.0 direction:right', DMbuffer='location:0.0'):
166          print('I recall Edward location to be (0.0) (recall_correct) (13A1)')
167          spatial_focus.set('object:brian location:3.0 direction:right')
168          goal_focus.set('move_left_13A2')
169
170      def move_left_13A2 (goal_focus='move_left_13A2', spatial_focus='object:brian location:3.0 direction:right'):
171          print('Move left (13A2)')
172          DM.threshold+=.3
173          spatial_focus.set('object:travis location:2.0 direction:left')
174          goal_focus.set('move_left_13A3')
175
176      def move_left_13A3 (goal_focus='move_left_13A3', spatial_focus='object:travis location:2.0 direction:left'):
177          print('Move left (13A3)')
178          DM.threshold+=.3
179          spatial_focus.set('object:derek location:1.0 direction:left')
180          goal_focus.set('move_left_13A4')
181
182      def move_left_13A4 (goal_focus='move_left_13A4', spatial_focus='object:derek location:1.0 direction:left'):
183          print('Move left (13A4)')
184          DM.threshold+=.3
185          spatial_focus.set('object:edward location:0.0 direction:left')
186          goal_focus.set('confirm_conclusion_13A5')
187
188      def confirm_conclusion_13A5 (goal_focus='confirm_conclusion_13A5', spatial_focus='object:edward location:0.0 direction:left'):
189          print('True, Edward is to the left of Brian!  (13A5)')
190          self.decision('1')
191          self.stop()
192
193
194  ###------------13B1-Error of Omission----------###
195
196      # 13B1-inspection-recall_omission
197      def recall_omission_13B1(goal_focus='recall_edward_13A1B1C1D1', spatial_focus='object:brian location:3.0 direction:right', DM='error:True', DMbuffer=None):
198          print('I can not recall Edward location (recall_omission) (13B1)')
199          spatial_focus.set('object:brian location:3.0 direction:right')
200          goal_focus.set('confirm_conclusion_13B2A1B1')
201
202      # 13B2A1-inspection-confirm_conclusion
203      def confirm_conclusion_13B2A1(goal_focus='confirm_conclusion_13B2A1B1', spatial_focus='object:brian location:3.0 direction:right'):
204          print('True, Edward is to the left of Brian! (13B2A1)')
205          self.decision('1')
206          self.stop()
207
208      # 13B2B1-inspection-confirm_conclusion
209      def confirm_conclusion_13B2B1(goal_focus='confirm_conclusion_13B2A1B1', spatial_focus='object:brian location:3.0 direction:right'):
210          print('False, Edward is NOT to the left of Brian! (13B2B1)')
211          self.decision('0')
212          self.stop()
213
214
```

```
215 ###------------13C1-Error of Commission...travis----------###
216
217     # 13C1-recall_commission...travis
218     def recall_commission_13C1 (goal_focus='recall_edward_13A1B1C1D1', spatial_focus='object:brian location:3.0 direction:right', DMbuffer='location:2.0'):
219         print('I recall Edward location to be (2.0) (recall_commission...travis) (13C1)')
220         spatial_focus.set ('object:brian location:3.0 direction:right')
221         goal_focus.set ('move_left_13C2')
222
223     # 13C2-move_left
224     def move_left_13C2 (goal_focus='move_left_13C2', spatial_focus='object:brian location:3.0 direction:right'):
225         print('Move left (13C2)')
226         DM.threshold+=.3
227         spatial_focus.set('object:travis location:2.0 direction:left')
228         goal_focus.set('confirm_conclusion_13C3')
229
230     # 13C3-confirm_conclusion
231     def confirm_conclusion_13C3 (goal_focus='confirm_conclusion_13C3', spatial_focus='object:travis location:2.0 direction:left'):
232         print('True, Edward is to the left of Brian! (13C3)')
233         self.decision('1')
234         self.stop()
235
236
237 ###------------13D1-Error of Commission...derek----------###
238
239     # 13D1-recall_commission
240     def recall_commission_13D1(goal_focus='recall_edward_13A1B1C1D1', spatial_focus='object:brian location:3.0 direction:right', DMbuffer='location:1.0'):
241         print('I recall Edward location to be (1.0) (recall_commission...derek) (13D1)')
242         spatial_focus.set('object:brian location:3.0 direction:right')
243         goal_focus.set('move_left_13D2')
244
245     def move_left_13D2(goal_focus='move_left_13D2', spatial_focus='object:brian location:3.0 direction:right'):
246         print('Move left (13D2)')
247         DM.threshold+=.3
248         spatial_focus.set('object:travis location:2.0 direction:left')
249         goal_focus.set('move_left_13D3')
250
251     def move_left_13D3(goal_focus='move_left_13D3', spatial_focus='object:travis location:2.0 direction:left'):
252         print('Move left (13D3)')
253         DM.threshold+=.3
254         spatial_focus.set('object:derek location:1.0 direction:left')
255         goal_focus.set('confirm_conclusion_13D4')
256
257     # 13D4-confirm_conclusion
258     def confirm_conclusion_13D4(goal_focus='confirm_conclusion_13D4', spatial_focus='object:derek location:1.0 direction:left'):
259         print('True, Edward is to the left of Brian! (13D4)')
260         self.decision('1')
261         self.stop()
262
263
264 ################################################################
265
266
267 nico = MyAgent()
268 thesis = MyEnvironment()
269 thesis.agent=nico
270 python_actr.log_everything(thesis)
271 thesis.run()
272 python_actr.finished()
273
```

Appendix D: Python ACT-R Model for Three Premises-Two Dimension Problem

```
import python_actr
from python_actr import *
import csv

#-----------------------------------------------------

### 3P2D - MRRT Q.66 ###

# P1) Edwards is below and to the left of Derek
# P2) Derek is below and to the left of Travis
# P3) Travis is below and to the left of Brian
# C) Edward is below and to the left of Brian

# Avg_Acc: 0.739130435
# Avg_RT: 36.07855072
# Relation: Spatial
# Premise: Three
# Dimensions: Two
# Answer: True
# Solution: True
# Premise Order: Continuous
# CPhrasing: Afirst

# Accumulated number of moves of spatial focus for construction : 6

# -----------------------------------------------------

class MyEnvironment(python_actr.Model):
    pass


class MyAgent(ACTR):

    spatial_focus=Buffer()
    goal_focus=Buffer()
    DMbuffer=Buffer()

    production_time=0.05
    production_sd=0.01


    DM=Memory(DMbuffer, latency=0.05, threshold=0.3, maximum_time=10.0, finst_size=0, finst_time=3.0)
    DMN=DMNoise(DM, noise=0.3, baseNoise=0.3)
    DMD=DMBaseLevel(DM, decay=0.5, limit=None)

    dm_spread=DMSpreading(DM,goal_focus)
    dm_spread.strength=1
    dm_spread.weight[goal_focus]=0.0

    partial=Partial(DM,strength= 0.0, limit= 0.0)
    partial.similarity('edward','derek', 0)
    partial.similarity('edward','travis', 0)
    partial.similarity('edward','brian', 0 )

    partial.similarity('derek','travis', 0)
    partial.similarity('derek','brian', 0)

    partial.similarity('travis','brian', 0)

```

```
#####----------------------CONSTRUCTION-----------------------------------#####

    goal_focus.set('read_premise_1')
    spatial_focus.set('object:na location:na direction:na')

    # 1-construction-read_premise
    def read_premise_1(goal_focus='read_premise_1',spatial_focus='object:na location:na direction:na'):
        print('Premise 1: Edwards is below and to the left of Derek (1)')
        spatial_focus.set('object:na location:0.0 direction:na')
        goal_focus.set('insert_edward_2')

    # 2-construction-insert_edward
    def insert_edward_2(goal_focus='insert_edward_2',spatial_focus='object:na location:0.0 direction:na'):
        print('Insert Edward (2)')
        DM.add('object:edward location:0.0 annotation:na')
        spatial_focus.set('object:edward location:0.0 direction:na')
        goal_focus.set('move_above_3')

    # 3-construction-move_above
    def move_above_3(goal_focus='move_above_3',spatial_focus='object:edward location:0.0 direction:na'):
        print('Move above (3)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:0.1 direction:above')
        goal_focus.set('move_right_4')

    # 4-construction-move_right
    def move_right_4(goal_focus='move_right_4',spatial_focus='object:na location:0.1 direction:above'):
        print('Move right (4)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:1.1 direction:right')
        goal_focus.set('insert_derek_5')

    # 5-construction-insert_derek
    def insert_derek_5(goal_focus='insert_derek_5',spatial_focus='object:na location:1.1 direction:right'):
        print('Insert Derek (5)')
        DM.add('object:derek location:1.1 annotation:na')
        spatial_focus.set('object:derek location:1.1 direction:right')
        goal_focus.set('read_premise_6')

    # 6-construction-read_premise
    def read_premise_6(goal_focus='read_premise_6',spatial_focus='object:derek location:1.1 direction:right'):
        print('Premise 2: Derek is below and to the left of Travis (6)')
        spatial_focus.set('object:derek location:1.1 direction:right')
        goal_focus.set('move_above_7')

    # 7-construction-move_above
    def move_above_7(goal_focus='move_above_7',spatial_focus='object:derek location:1.1 direction:right'):
        print('Move above (7)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:2.1 direction:above')
        goal_focus.set('move_right_8')

    # 8-construction-move_right
    def move_right_8(goal_focus='move_right_8',spatial_focus='object:na location:2.1 direction:above'):
        print('Move right (8)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:2.2 direction:right')
        goal_focus.set('insert_travis_9')

    # 9-construction-insert_travis
    def insert_travis_9(goal_focus='insert_travis_9',spatial_focus='object:na location:2.2 direction:right'):
        print('Insert Travis (9)')
        DM.add('object:travis location:2.2 annotation:na')
        spatial_focus.set('object:travis location:2.2 direction:right')
        goal_focus.set('read_premise_10')
```

```
    # 10-construction-read_premise
    def read_premise_10(goal_focus='read_premise_10',spatial_focus='object:travis location:2.2 direction:right'):
        print('Premise 3: Travis is below and to the left of Brian (10)')
        spatial_focus.set('object:travis location:2.2 direction:right')
        goal_focus.set('move_above_11')

    # 11-construction-move_above
    def move_above_11(goal_focus='move_above_11',spatial_focus='object:travis location:2.2 direction:right'):
        print('Move above (11)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:3.2 direction:above')
        goal_focus.set('move_right_12')

    # 12-construction-move_right
    def move_right_12(goal_focus='move_right_12',spatial_focus='object:na location:3.2 direction:above'):
        print('Move right (12)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:3.3 direction:right')
        goal_focus.set('insert_brian_13')

    # 13-construction-insert_brian
    def insert_brian_13(goal_focus='insert_brian_13',spatial_focus='object:na location:3.3 direction:right'):
        print('Insert Brian (13)')
        DM.add('object:brian location:3.3 annotation:na')
        spatial_focus.set('object:brian location:3.3 direction:right')
        goal_focus.set('read_conclusion_14')


######--------------------INSPECT----------------------------#######

    # This function is used to write the conclusions True(1)/False(0) to a csv file
    def decision (self,answer):
        myfile=open("04-3P2D_MMRT_DATA.csv", "a", newline="")
        wr=csv.writer(myfile)
        wr.writerow(answer)
        myfile.close()

    #14-inspection-read_conclusion
    def read_conclusion_14(goal_focus='read_conclusion_14',spatial_focus='object:brian location:3.3 direction:right'):
        print('Conclusion: Edward is below and to the left of Brian (14)')
        spatial_focus.set('object:brian location:3.3 direction:right')
        goal_focus.set('request_edward_15')


    #15-inspection-request_edward
    def request_edward_15(goal_focus='request_edward_15',spatial_focus='object:brian location:3.3 direction:right'):
        print('Recalling Edwards location...(15)')
        DM.request('object:edward location:!3.3') # !3.3 because spatial_focus is already on this location
        spatial_focus.set('object:brian location:3.3 direction:right') # spatial_focus still on Brian after construction
        goal_focus.set('recall_edward_16A1B1C1D1')


##########################################################################

#----------------BRANCH-16-A1-B1-C1-D1-------------------------------

##########################################################################

#--------16A1-Correct Recall of Edwards Position-------------

    #16A1-inspection-recall_correct
    def recall_correct_16A1(goal_focus='recall_edward_16A1B1C1D1',spatial_focus='object:brian location:3.3 direction:right', DMbuffer='location:0.0'):
        print('I recall Edward is at location 0.0 (recall_correct) (16A1)')
        spatial_focus.set('object:brian location:3.3 direction:right')
        goal_focus.set('move_below_16A2')

    #16A2-inspection-move_below
    def move_below_16A2(goal_focus='move_below_16A2',spatial_focus='object:brian location:3.3 direction:right'):
        print('Move below (16A2)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:3.2 direction:below')
        goal_focus.set('move_left_16A3')

    #16A3-inspection-move_left
    def move_left_16A3(goal_focus='move_left_16A3',spatial_focus='object:na location:3.2 direction:below'):
        print('Move left (16A3)')
        DM.threshold+=.3
        spatial_focus.set('object:travis location:2.2 direction:left')
        goal_focus.set('move_below_16A4')

    #16A4-inspection-move_below
    def move_below_16A4(goal_focus='move_below_16A4',spatial_focus='object:travis location:2.2 direction:left'):
        print('Move below (16A4)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:2.1 direction:below')
        goal_focus.set('move_left_16A5')
```

```
    #16A5-inspection-move_left
    def move_left_16A5(goal_focus='move_left_16A5',spatial_focus='object:na location:2.1 direction:below'):
        print('Move left (16A5)')
        DM.threshold+=.3
        spatial_focus.set('object:travis location:1.1 direction:left')
        goal_focus.set('move_below_16A6')

    #16A6-inspection-move_below
    def move_below_16A6(goal_focus='move_below_16A6',spatial_focus='object:travis location:1.1 direction:left'):
        print('Move below (16A6)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:1.0 direction:below')
        goal_focus.set('move_left_16A7')

    #16A7-inspection-move_left
    def move_left_16A7(goal_focus='move_left_16A7',spatial_focus='object:na location:1.0 direction:below'):
        print('Move left (16A7)')
        DM.threshold+=.3
        spatial_focus.set('object:edward location:0.0 direction:left')
        goal_focus.set('confirm_conclusion_16A8')

    #16A8-inspection-confirm_conclusion
    def confirm_conclusion_16A8(goal_focus='confirm_conclusion_16A8',spatial_focus='object:edward location:0.0 direction:left'):
        print('True, Edward is below and to the left of Brian! (recall_correct) (16A8)')
        self.decision('1')
        self.stop()

#######################################################################

###---------16B1-Omission Recall Error of Edwards Position----------###

    #16B1-inspection-recall_omission
    def recall_omission_16B1(goal_focus='recall_edward_16A1B1C1D1', spatial_focus='object:brian location:3.3 direction:right', DM='error:True',DMbuffer=None):
        print('I can not recall Edwards location (recall_omission) (16B1)')
        spatial_focus.set('object:brian location:3.3 direction:right')
        goal_focus.set('confirm_conclusion_16B2A1B1')


#---------16B1-A1-B1-------------------------------------

    #16B1A1-inspection-confirm_conclusion
    def confirm_conclusion_16B2A1(goal_focus='confirm_conclusion_16B2A1B1',spatial_focus='object:brian location:3.3 direction:right'):
        print('True, Edward is below and to the left of Brian! (recall_omission) (16B2A1)')
        self.decision('1')
        self.stop()


    #16B1B1-inspection-confirm_conclusion
    def confirm_conclusion_16B2B1(goal_focus='confirm_conclusion_16B2A1B1',spatial_focus='object:brian location:3.3 direction:right'):
        print('False, Edward is NOT below and to the left of Brian! (recall_omission) (16B2B1)')
        self.decision('0')
        self.stop()

#######################################################################


###-----16C1-Commission Recall Error of Edwards Position as Travis-----###

    #16C1-inspection-recall_commission...travis
    def recall_commission_16C1 (goal_focus='recall_edward_16A1B1C1D1',spatial_focus='object:brian location:3.3 direction:right',DMbuffer='location:2.2'):
        print('I recall Edward is at location 2.2 (recall_commission) (16C1)')
        spatial_focus.set('object:brian location:3.3 direction:right')
        goal_focus.set('move_below_16C2')

    #16C1A2-inspection-move_below
    def move_below_16C2(goal_focus='move_below_16C2',spatial_focus='object:brian location:3.3 direction:right'):
        print('Move below (16C2)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:3.2 direction:below')
        goal_focus.set('move_left_16C3')

    #16C1A3-inspection-move_left
    def move_left_16C3(goal_focus='move_left_16C3',spatial_focus='object:na location:3.2 direction:below'):
        print('Move left (16C3)')
        DM.threshold+=.3
        spatial_focus.set('object:travis location:2.2 direction:left')
        goal_focus.set('confirm_conclusion_16C4')

    #16C1A4-confirm_conclusion
    def confirm_conclusion_16C4(goal_focus='confirm_conclusion_16C4',spatial_focus='object:travis location:2.2 direction:left'):
        print('True, Edward is below and to the left of Brian! (recall_commission...travis) (16C4)')
        spatial_focus.set('object:travis location:2.2 direction:left')
        self.decision('1')
        self.stop()
```

```
#------16D1-Commission Recall Error of Edwards position as Dereks---------

    #16D1-inspection-recall_commission...derek
    def recall_commission_16D1(goal_focus='recall_edward_16A1B1C1D1',spatial_focus='object:brian location:3.3 direction:right',DMbuffer='location:1.1'):
        print('I recall Edward is at location 1.1 (recall_commission) (16D1)')
        spatial_focus.set('object:brian location:3.3 direction:right')
        goal_focus.set('move_below_16D2')

    #16D2-inspection-move_below
    def move_below_16D2(goal_focus='move_below_16D2',spatial_focus='object:brian location:3.3 direction:right'):
        print('Move below (16D2)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:3.2 direction:below')
        goal_focus.set('move_left_16D3')

    #16D3-inspection-move_left
    def move_left_16D3(goal_focus='move_left_16D3',spatial_focus='object:na location:3.2 direction:below'):
        print('Move left (16D3)')
        DM.threshold+=.3
        spatial_focus.set('object:travis location:2.2 direction:left')
        goal_focus.set('move_below_16D4')

    #16D4-inspection-move_below
    def move_below_16D4(goal_focus='move_below_16D4', spatial_focus='object:travis location:2.2 direction:left'):
        print('Move below (16D4)')
        DM.threshold+=.3
        spatial_focus.set('object:na location:2.1 direction:below')
        goal_focus.set('move_left_16D5')

    #16C1B5-inspection-move_left
    def move_left_16D5(goal_focus='move_left_16D5',spatial_focus='object:na location:2.1 direction:below'):
        print('Move left (16D5)')
        DM.threshold+=.3
        spatial_focus.set('object:derek location:1.1 direction:left')
        goal_focus.set('confirm_conclusion_16D6')

    #16C1B6-inspection-confirm_conclusion
    def confirm_conclusion_16D6(goal_focus='confirm_conclusion_16D6',spatial_focus='object:derek location:1.1 direction:left'):
        print('True, Edward is below and to the left of Brian! (recall_commission...derek) (16D6)')
        spatial_focus.set('object:derek location:1.1 direction:left')
        self.decision('1')
        self.stop()


################################################################


nico = MyAgent()
thesis = MyEnvironment()
thesis.agent=nico
python_actr.log_everything(thesis)
thesis.run()
python_actr.finished()
```

References

Anderson, J. R., & Bower, G. H. (1973). Human associative memory. Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.

Anderson, J. R. (1990). The adaptive character of thought. Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Lebiere, C. (1998). The Atomic Components of Thought. Mahwah, NJ: Erlbaum.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, *111*(4), 1036–1060. https://doi.org/10.1037/0033-295X.111.4.1036

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195324259.001.0001

Chandrasekaran, B., Kurup, U., Banerjee, B., Josephson, J. R., & Winkler, R. (2004). An Architecture for Problem Solving with Diagrams. In A. Blackwell, K. Marriott, & A. Shimojima (eds) Proceedings of Diagrams 2004, 151—165, Berlin: Springer.

Bertel, S., Barkowsky, T., Engel, D., & Freksa, C. (2006). Computational modeling of reasoning with mental images: basic requirements. In: D. Fum, d. F. Missier, & A. Stocco (Eds.), Proceedings of ICCM 2006, Trieste , pp. 50-55. Edizioni Goliardiche, Trieste.

Beth EW, Piaget J (1966) Mathematical Epistemology and Psychology (Reidel, Dordrecht, The Netherlands).

Boeddinghaus, J., Ragni, M., Knauff, M., & Nebel, B. (2006). *Simulating Spatial Reasoning Using ACT-R*. Proceedings of the ICCM 06. 62-67.

Braine, M. D. (1978). On the relation between the natural logic of reasoning and standard logic. *Psychological Review, 85*(1), 1–21. https://doi.org/10.1037/0033-295X.85.1.1

Braine, M. D. S., & O'Brien, D. P. (Eds.). (1998). *Mental logic.* Lawrence Erlbaum Associates Publishers.

Byrne, R. M. J., & Johnson-Laird, P. N. (1989). Spatial reasoning. *Journal of Memory and Language*, *28*(5), 564–575. https://doi.org/10.1016/0749-596x(89)90013-2

Cortes, R. A., Weinberger, A. B., Colaizzi, G. A., Porter, G. F., Dyke, E. L., Keaton, H. O., Walker, D. L., & Green, A. E. (2021). What Makes Mental Modeling Difficult? Normative Data for the Multidimensional Relational Reasoning Task. *Frontiers in psychology*, *12*, 668256. https://doi.org/10.3389/fpsyg.2021.668256

Craik, K. J. W. (1943). *The nature of explanation.* University Press, Macmillan.

de Soto, C. B., London, M., & Handel, S. (1965). Social Reasoning and spatial paralogic. *Journal of Personality and Social Psychology*, *2*(4), 513–521. https://doi.org/10.1037/h0022492

Ehrlich, K., & Johnson-Laird, P. N. (1982). Spatial descriptions and referential continuity. _Journal of Verbal Learning & Verbal Behavior, 21_(3), 296–306. [https://doi.org/10.1016/S0022-5371(82)90626-0](https://psycnet.apa.org/doi/10.1016/S0022-5371(82)90626-0)

Glasgow, J., & Papadias, D. (1992). Computational imagery. *Cognitive Science*, 16(3), 355–394.

Goel, V., & Dolan, R. J. (2001). Functional neuroanatomy of three-term relational reasoning. *Neuropsychologia*, *39*(9), 901–909. https://doi.org/10.1016/s0028-3932(01)00024-0

Goodwin, G. P., & Johnson-Laird, P. N. (2005). Reasoning about relations. *Psychological Review*, 112_(2), 468–493. [https://doi.org/10.1037/0033 295X.112.2.468]

Graham, J. W., Hofer, S. M., & MacKinnon, D. P. (1996). Maximizing the Usefulness of Data Obtained with Planned Missing Value Patterns: An Application of Maximum Likelihood Procedures. *Multivariate behavioral research*, *31*(2), 197–218. https://doi.org/10.1207/s15327906mbr3102_3

Graham, J. W., Taylor, B. J., Olchowski, A. E., & Cumsille, P. E. (2006). Planned missing data designs in psychological research. *Psychological methods*, *11*(4), 323–343. https://doi.org/10.1037/1082-989X.11.4.323

Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds matter: an introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of experimental psychology. Applied*, *6*(4), 322–335. https://doi.org/10.1037//1076-898x.6.4.322

Held, C., Vosgerau, G. M., & Knauff, M. (Eds.). (2006). *Mental models and the mind: Current developments in cognitive psychology, neuroscience and philosophy of mind*. Elsevier Science & Technology.

Hummel, John & Holyoak, Keith. (2001). A process model of human transitive inference. 279-305.

Hummel, J. E., & Holyoak, K. J. (2005). Relational Reasoning in a Neurally Plausible Cognitive Architecture: An Overview of the LISA Project. *Current Directions in Psychological Science, 14(3)*, 153–157. https://doi.org/10.1111/j.0963-7214.2005.00350.x

Hunter, I. M. L. (1957). The solving of three-term series problems. *British Journal of Psychology, 48,* 286–298. https://doi.org/10.1111/j.2044-8295.1957.tb00627.x

Huttenlocher, J. (1968). Constructing spatial images: A strategy in reasoning. *Psychological Review, 75*(6), 550–560. https://doi.org/10.1037/h0026748

Jahn, G. (2003). Hybrid representation of spatial descriptions. International workshop Spatial and Visual Components in Mental Reasoning about Large-Scale Spaces. 01-02 Sept 2003, Bad Zwischenahn, Germany.

Jahn, G., Knauff, M., & Johnson-Laird, P. (2008). Preferred mental models in reasoning about spatial relations. Memory & cognition. 35. 2075-87. 10.3758/BF03192939.

Johnson-Laird, P. N. (1972). The three-term series problem. *Cognition,* 1(1), 57–82. https://doi.org/10.1016/0010-0277(72)90045-5

Johnson-Laird, P. N. (1980). *Mental Models in Cognitive Science*. 45.

Johnson-Laird, P. N. (1983). Mental Models. *Towards a cognitive science of language, inference and consciousness*. Cambridge, UK: Cambridge University Press.

Johnson-Laird, P. N., & Bara, B. G. (1984). Syllogistic inference. *Cognition, 16*(1), 1–61. https://doi.org/10.1016/0010-0277(84)90035-0

Johnson-Laird, P. N. (1989). "Mental models," in Foundations of Cognitive Science, ed. M. I. Posner (Cambridge, MA: MIT Press), 469–499.

Johnson-Laird, P. N., & Byrne, R. M. J. (1991). *Deduction.* Lawrence Erlbaum Associates, Inc.

Johnson-Laird, P. N. (1998). Imagery, visualization, and thinking, in *Perception and Cognition at Century's End*, ed. J. Hochberg (San Diego: Academic Press), 441–467.

Johnson-Laird, P. N. (2001). Mental models and deduction. *Trends in Cognitive Sciences*, *5*(10), 434–442. https://doi.org/10.1016/s1364-6613(00)01751-4

Johnson-Laird, P. N. (2004). The history of mental models. In K. Manktelow & M. C. Chung (Eds.), Psychology of reasoning: Theoretical and historical perspectives (pp. 179212). New York, NY: Psychology Press.

Johnson-Laird, P. N. (2006). How we reason. New York, NY: Oxford University Press.

Johnson-Laird, P. N. (2010). Mental models and human reasoning. _Proceedings of the National Academy of Sciences_, _107_(43), 18243–18250 (https://doi.org/10.1073/pnas.1012933107)

Johnson-Laird P. N. (2013) Mental models and cognitive change, Journal of Cognitive Psychology, 25:2, 131-138, (https://doi.org/10.1080/20445911.2012.759935)

Kelley, T. D., Lee, F. J., & Wiley, P. W. (2000). Developing an ACT-R model of mental manipulation. https://doi.org/10.21236/ada378078

Knauff, M., Rauh, R., & Schlieder, C. (1995). 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). Mahwah, NJ: Lawrence Erlbaum Associates.

Knauff, M., Rauh, R., Schlieder, C., & Strube, G. (1998). Mental models in spatial reasoning. In C. Freksa, C. Habel, & K. F. Wender (Eds.), Spatial cognition: An interdisciplinary approach to representation and processing of spatial knowledge (pp. 267–291). Berlin, Germany: Springer.

Knauff, M., & Johnson-Laird, P. N. (2002). Visual imagery can impede reasoning. *Memory & Cognition, 30*(3), 363–371. https://doi.org/10.3758/BF03194937

Knauff, M., Fangmeier, T., Ruff, Ch., & Johnson-Laird, P. (2003). Reasoning, Models, and Images: Behavioral Measures and Cortical Activity. Journal of cognitive neuroscience. 15. 559-73. 10.1162/089892903321662949.

Knauff, M., Nebel, B., & Ragni, M. (2005). A Computational Model for Spatial Reasoning with Mental Models. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 27. Retrieved from https://escholarship.org/uc/item/3w8662hq

Kosslyn, Stephen M. (1980). *Image and Mind.* Harvard University Press.

Kosslyn, S. M. (1994). *Image and brain: The resolution of the imagery debate*. The MIT Press.

Kosslyn, S. M., Thompson, W. L., & Ganis, G. (2006). *The case for mental imagery.* Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195179088.001.0001

Kotseruba, I., & Tsotsos, J. K. (2020). 40 years of cognitive architectures: Core cognitive abilities and practical applications. *Artificial Intelligence Review*, *53*(1), 17–94

Kunda, M., McGreggor, K., & Goel, A. K. (2013). A computational model for solving problems from the Raven's Progressive Matrices intelligence test using iconic visual representations. Cognitive Systems Research, 22, 47–66.

Lebiere, C., Anderson, J., & Reder, L. (1994). Error modeling in the ACT-R production system.

Lieto, A., Bhatt, M., Oltramari, A., & Vernon, D. (2018). The role of cognitive architectures in General Artificial Intelligence. *Cognitive Systems Research*, *48*, 1–3. https://doi.org/10.1016/j.cogsys.2017.08.003

Logan G. D. (1994). Spatial attention and the apprehension of spatial relations. *Journal of experimental psychology. Human perception and performance*, *20*(5), 1015–1036. https://doi.org/10.1037//0096-1523.20.5.1015

Madl, T., Chen, K., Montaldi, D., & Trappl, R. (2015). Computational cognitive models of spatial memory in navigation space: A review. *Neural Networks, 65,* 18–43. https://doi.org/10.1016/j.neunet.2015.01.002

Miller, G. A., & Johnson-Laird, P. N. (1976). *Language and perception.* Belknap Press.

More, T., (1979). The nested rectangular array as a model of data. *SIGAPL APL Quote Quad 9*, 4-P1 (June 1979), 55–73. DOI:https://doi.org/10.1145/390009.804440

Nejasmic, J., Krumnack, A., Bucher, L., & Knauff, M. (2011). Cognitive processes underlying the continuity effect in spatial reasoning. Proceedings of the 33rd Annual Conference of the Cognitive Science Society.

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. *Visual Information Processing*, 283–308. https://doi.org/10.1016/b978-0-12-170150-5.50012-3

Newell, A. (1982). The knowledge level. *Artificial Intelligence*, *18*, 87–127. https://doi.org/10.21236/ada106556

Newell, A (1990), Unified theories of cognition. Cambridge, MA: Harvard University Press

Oberauer, K., & Wilhelm, O. (2000). Effects of directionality in deductive reasoning: I. The comprehension of single relational premises. *Journal of experimental psychology. Learning, memory, and cognition*, *26*(6), 1702–1712. https://doi.org/10.1037//0278-7393.26.6.1702

O'Brien, D., & Manfrinati, A. (2010). The mental logic theory of conditional propositions. In Cognition and Conditionals: Probability and Logic in Human Thinking. : Oxford University Press.

Pearson, J., & Kosslyn, S. M. (2015). The heterogeneity of mental representation: Ending the imagery debate. *Proceedings of the National Academy of Sciences of the United States of America, 112(33),* 10089–10092. https://doi.org/10.1073/pnas.1504933112

Peirce, C. S. (1931-1958). Collected papers of Charles Sanders Peirce (Vol. 4, C. Hartshorne, P. Weiss, & A. Burks (Eds.). Cambridge, MA: Harvard University Press.

Pylyshyn, Z. W. (1981). The imagery debate: Analogue media versus tacit knowledge. *Psychological Review, 88*(1), 16–45. https://doi.org/10.1037/0033-295X.88.1.16

Pylyshyn, Z. (2001). Is the imagery debate over? If so, what was it about? In E. Dupoux (Ed.), *Language, brain, and cognitive development: Essays in honor of Jacques Mehler (pp. 59–83*). The MIT Press.

Pylyshyn, Z. (2002). Mental imagery: In search of a theory. *Behavioral and Brain Sciences, 25(2)*, 157-182. doi:10.1017/S0140525X02000043

Pylyshyn, Z. (2003). Return of the mental image: Are there really pictures in the brain? *Trends in Cognitive Sciences, 7(3),* 113–118. https://doi.org/10.1016/S1364-6613(03)00003-2

Ragni, M., Knauff, M., & Nebel, B. (2005). *A Computational Model for Spatial Reasoning with Mental Models. 7.*

Ragni, M., Knauff, M. (2008). Deductive Spatial Reasoning: A Computational and Cognitive Perspective. *KI: Themenheft KI und Kognition*, 8; 1/08: 13-17.

Ragni, M., & Knauff, M. (2013). A Theory and a Computational Model of Spatial Reasoning with Preferred Mental Models. *Psychological Review*, *120*(3), 561–588. https://doi.org/10.1037/a0032460

Ragni, M., Brand, D., & Riesterer, N. (2021). The Predictive Power of Spatial Relational Reasoning Models: A New Evaluation Approach. *Frontiers in psychology*, *12*, 626292. https://doi.org/10.3389/fpsyg.2021.626292

Rips, L. J. (1994). *The psychology of proof: Deductive reasoning in human thinking.* The MIT Press.

Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 10(3), e1488. https://doi.org/10.1002/wcs.1488

Schultheis, H., Bertel, S., Barkowsky, T., & Seifert, I. (2007). The Spatial and the Visual in Mental Spatial Reasoning: An Ill-Posed Distinction. In T. Barkowsky, M. Knauff, G. Ligozat, & D. R. Montello (Eds.), Spatial Cognition V Reasoning, Action, Interaction (Vol. 4387, pp. 191–209). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-75666-8_12

Sima, J. F., Schultheis, H., & Barkowsky, T. (2013). Differences between Spatial and Visual Mental Representations. *Frontiers in Psychology, 4,* 240.https://doi.org/10.3389/fpsyg.2013.00240

Stewart, T. C., & West, R. L. (2007). Deconstructing and reconstructing act-r: Exploring the architectural space. *Cognitive Systems Research*, *8*(3), 227–236. https://doi.org/10.1016/j.cogsys.2007.06.006

Stewart, T., & West, R. (2010). Testing for equivalence: A methodology for computational cognitive modelling. *Journal of Artificial General Intelligence*, *2*(2), 69–87. https://doi.org/10.2478/v10229-011-0010-8

Tabachneck-Schijf, H.J., Leonardo, A.M., & Simon, H.A. (1997). CaMeRa: A Computational Model of Multiple Representations. *Cogn. Sci.,* 21, 305-350.

Tversky, B. (1993). "Cognitive maps, cognitive collages, and spatial mental models," in Spatial Information Theory: A Theoretical Basis for GIS – Proceedings of COSIT'93, eds A. U. Frank, and I. Campari (Berlin: Springer), 14–24

Van der Henst, J. Mental model theory versus the inference rule approach in relational reasoning. Thinking and Reasoning 8 (2002), 193–205.

Waller, D., & Nadel, L. (2013). INTRODUCTION: FRAMEWORKS FOR UNDERSTANDING SPATIAL THOUGHT (OR WRAPPING OUR HEADS AROUND SPACE). In D. Waller & L. Nadel (Eds.), *Handbook of Spatial Cognition* (pp. 3–12). American Psychological Association. http://www.jstor.org/stable/j.ctv1chs7vk.5

West, R. L.; Provnost, S. (2009). "Modeling SGOMS in ACT-R: Linking Macro- and Microcognition". *Journal of Cognitive Engineering and Decision Making*. 3 (2): 194–207. doi:10.1518/155534309X441853.