Human Reasoning Module

Enkhbold Nyamsuren (e.nyamsuren@rug.nl)
Department of Artificial Intelligence, University of Groningen,
Nijenborgh 9, 9747 AG Groningen, Netherlands

Niels A. Taatgen (n.a.taatgen@rug.nl)
Department of Artificial Intelligence, University of Groningen,
Nijenborgh 9, 9747 AG Groningen, Netherlands

Correspondence concerning this article should be addressed to Enkhbold Nyamsuren, Department of Artificial Intelligence, University of Groningen, Nijenborgh 9, 9747 AG Groningen, Netherlands.
E-mail: e.nyamsuren@rug.nl
Phone: +31 50 363 7450

Abstract

This paper introduces a framework of human reasoning and its ACT-R based implementation called the Human Reasoning Module (HRM). Inspired by the human mind, the framework seeks to explain how a single system can exhibit different forms of reasoning ranging from deduction to induction, from deterministic to probabilistic inference, from rules to mental-models. The HRM attempts to unify previously mentioned forms of reasoning into a single coherent system rather than treating them as loosely connected separate subsystems. The validity of the HRM is tested with cognitive models of three tasks involving simple casual deduction, reasoning on spatial relations and Bayesian-like inference of cause/effect. The first model explains why people use an inductive, probabilistic reasoning process even when using ostensibly deductive arguments such as modus ponens and modus tollens. The second model argues that visual bottom-up processes can do fast and efficient semantic processing. Based on this argument, the model explains why people perform worse in a spatial relation problem with ambiguous solutions than in a problem with a single solution. The third model demonstrates that statistics of Bayesian-like reasoning can be reproduced using a combination of a rule-based reasoning and probabilistic declarative retrievals. All three models were validated successfully against human data. The HRM demonstrates that a single system can express different facets of reasoning exhibited by the human mind. As a part of a cognitive architecture, the HRM is promising to be a useful and accessible tool for exploring deeps of human mind and modeling biologically inspired agents.

Keywords: reasoning, ACT-R, casual, spatial, Bayesian.

Introduction

In this paper, we introduce a framework that attempts to unify various approaches to human reasoning. The Human Reasoning Module, or HRM, is an implementation of this framework developed as a part of the ACT-R cognitive architecture (Anderson, 2007). As opposed to ACT-R's core modules that represent specific types of cognitive resources such as vision or memory, the HRM does not add a new type of cognitive resource. The HRM extends the theoretical frameworks and corresponding computational functionalities of the existing modules of ACT-R. Therefore, the HRM is both a theory and a tool for modeling. As a theory, it advocates for a specific structure of knowledge organization in our declarative memory. The structure is still based on knowledge chunks, but adds specific requirements on chunk types and its slots. Furthermore, the HRM advocates the existence of task-general procedural knowledge that gives us the ability to reason and solve problems based on real-time information and previous experience. The proposed structures of declarative and procedural knowledge define grammar, axiom schemata and inference rules of human logic. As a tool, the HRM both extends and constrains the functionality of ACT-R's declarative module and also adds a set of task-general production rules to ACT-R's procedural module. Ideally, if the HRM is a valid model of human reasoning it should be able to tackle any form of reasoning process. However, the HRM's current unification attempt is limited to two dimensions depicted in Fig. 1. The next subsection discusses in details these dimensions.

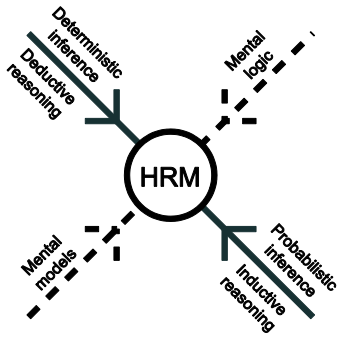


Fig. 1 Two dimensions of human reasoning that the HRM attempts to unify.

Inductive and deductive reasoning

At the core of the HRM, there is an assumption that the human general reasoning skill is inherently probabilistic or inductive. Any true form of classical deductive reasoning requires a *closed world assumption* stating that what is not currently known to be true is false. This is an extremely unpractical assumption in the real world full of uncertainties (Rajasekar, Lobo, Minker, 1989), and we subconsciously or consciously recognize this fact. Cummins (1995) demonstrated that even when someone is reasoning with ostensibly deductive arguments one still uses an inductive, probabilistic reasoning process. Further uncertainty arises due to limitations of our cognitive resources: our perception of the world can be noisy or limited and our memory may be forgetful. With such uncertainties, any deductive system will fail the tests of *validity* and *soundness*, necessary requirements for any formal deductive inference (Jeffrey, 1981). Furthermore, we do not often try to satisfy both of these requirements in our reasoning process (Thompson, 1996). Therefore, the HRM operates under the *open world assumption*, what is not proven is not necessarily false, and tries to prove truthfulness rather than falsity of knowledge.

However, the HRM does not exclude a possibility that deductive reasoning occurs within the context of specific tasks. Let us assume a specific problem that eliminates environmental uncertainties by clearly and unambiguously specifying contextual boundaries, constraints and rules. We can further assume that the problem is tractable within capacities and limitations of our cognitive resources, and there is no interference to the solution from our past knowledge outside of the problem's context. Such context will follow the closed world assumption, and, hence, deductive reasoning may be used. Therefore, in the HRM, there are no two separate processes for deductive or inductive reasoning. Instead, the HRM assumes that deductive reasoning is an instance of inductive reasoning over a specific domain of discourse with a near-zero uncertainty. A degree of uncertainty is the common dimension that implicitly unifies inductive and deductive reasoning in the HRM.

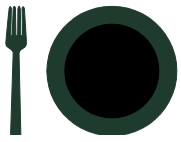
Mental logic, mental models and bottom-up reasoning

Next, the HRM further argues that general human reasoning does not necessarily rely on formal propositional forms and is not strictly top-down (conscious). There is a long history of debate over the theories of mental models and mental logic. The mental logic theory argues that a set of inference rules is applied to logical forms abstracted from stimuli (Rips, 1983). A commonly agreed interpretation of mental models theory dictates that stimuli are abstracted into a form of mental diagram where configuration information reflects the relationship between entities (Banks & Millward, 2009; Johnson-Laird, 1983). In the HRM, the two theories are part of the same reasoning process. It is based on the assumption that these two are not mutually exclusive strategies. Roberts (1993) rightfully pointed to the fact that there are no obvious reasons why the two types of theories should be incompatible. Coney (1988) argued for individual differences based on a study showing that some people are better at spatial reasoning while others prefer reasoning based on formal propositions. Johnson-Laird (2004), a chief proponent of the mental models theory, admitted that the model theory does not imply that reasoners never rely on rules of inference.

The HRM consolidates the two theories by assuming that a mental model is a form of working memory that allows convenient representation and storage of knowledge required for reasoning. New premises, including ones not explicitly stated by the problem context, are assumed to be extracted on demand from the mental model during a rule-based inference similar to the mental logic. The mental model as a working memory simplifies a manipulation and retrieval of knowledge that otherwise has to be stored in a less efficient long-term memory. For example, items in the existing model can be easily reconfigured to produce an alternative model. The smaller amount of cognitive effort required by the mental model can explain why people prefer it over direct inference on given propositional forms. This interpretation of the mental model implies that it is not the main tool of reasoning by itself. This is a major distinction from Johnson-Laird's (2004) interpretation arguing that the probability of a conclusion is estimated based on the proportion of equipossible models in which it holds. Certainly, our interpretation of the mental model is more parsimonious.

At this point, we need to map a metal model onto a specific cognitive resource. Johnson-Laird (2004) provided three functional requirements for the mental model: 1. A mental model should have an imagery capability to abstract meaning of premises into a mental diagram; 2. A mental model should be iconic; 3. Mental models should represent what is true, but not what is false. The cognitive resource that matches all above requirements is visual short-term memory (VSTM). It is specialized visuo-spatial mechanism in working memory for storing visual iconic information for a short duration (Logie, Zucco, Baddeley, 1990). VSTM stores a factual representation of the current state of affairs and, therefore, implies that information in it is assumed true. Arguably, one of the most important roles of VSTM is to retain and combine information gathered across successive fixations to construct dynamically a high-level internal representation of the outside world (Henderson & Hollingworth, 2003; Rensink 2000a, 2000b). The same process of retaining and combining information is likely to be necessary for building a mental model. Furthermore, VSTM is likely to have at least some imagery capability (Phillips, 1983; Wintermute, 2012). Phillips (1983), one of the first to introduce the concept of VSTM, emphasized that VSTM facilitates our ability to visualize problem space and is not just a sensory store. Jiang, Olson, and Chun (2000) reported that spatial information stored in VSTM includes not only object's location but also its relationship to other objects in VSTM. Based on these studies, we can conclude that VSTM is a suitable candidate for storing a mental model.

Now, we should discuss whether VSTM is distinct from long-term declarative memory. Unlike declarative memory, VSTM needs to provide a fast and reliable access to information to allow the scene representation to be constructed dynamically across rapid fixations. Thus, VSTM is functionally different from declarative memory. Furthermore, Phillips (1983) made a clear distinction between VSTM and long-term visual memory noting that head injuries affecting long-term memory do not affect visualization. Baddeley (2003) argued for distinction between long-term memory and the multi-component working memory that includes visuospatial sketchpad with imagery capability, a functional analogue to VSTM. Neuroimaging studies suggest that the short-term memory responsible for storing visuo-spatial information is located in parietal lobe (Baddeley, 2003; Lum, Conti-Ramsden, Page, & Ullman, 2012; Xu & Chun, 2005) and not in the hippocampus that is commonly associated with declarative memory. Finally, Formisano, Linden, Di Salle, Trojano, Esposito, Sack, Grossi, Zanella, & Goebel (2002) showed that parietal lobe also performs distinct functions of mental imagery. All these factors together support our assumption that VSTM is a distinct memory suitable for building a mental model.



(r-left-of fork plate)

Fig. 2 The image on the left contains an implicit knowledge that the fork is on the left side of the plate. Such knowledge can be extracted to form explicit proposition on the right.

The HRM treats the content of VSTM as a mental model unless it is irrelevant to the task. When available, the HRM extracts premises necessary for inference from the iconic content of VSTM. This process assumes that implicit semantic information is converted into explicit information. As an example, imagine that the VSTM contains visual objects as shown on left side of Fig. 2. Each object has set of features describing it such as color, shape, spatial position, etc. There is also relative spatial information, such as, the fork being on the left side of the plate. This information was not encoded as part of any object. However, it implicitly exists inside VSTM even though we may not be consciously aware of it until it is parsed. The relative spatial position can be quickly extracted on demand and converted into explicit propositional form shown on the right side of Fig. 2. Rensink (2007) indicated bottom-up visual processes may be able to process information at a semantic level subconsciously and even pre-attentively. It is feasible to assume that the same bottom-up processes are responsible for extracting explicit knowledge from implicit knowledge. Within the HRM, we refer to such process as *visual bottom-up reasoning* mechanism (not to be confused with inductive reasoning). We will further explore the mental logic and the mental model using an example task and a cognitive model based on the HRM.

Deterministic and probabilistic inferences

In the previous section, we have mentioned that the HRM uses rule-based inference that is inherently deterministic. This determinism relies on the assumption that the knowledge source is consistent and reliable. We also discussed that visual short-term memory is a source of knowledge for reasoning. As a form of working memory, VSTM provides a reliable access to reasonably consistent knowledge and does not violate above-mentioned assumption. Therefore, when the reasoning process relies on VSTM only it can be deterministic and deductive.

However, there is a second source of knowledge, a long-term declarative memory. The HRM uses ACT-R's declarative memory (DM). As a proper model of human long-term memory, DM has inherited its peculiarities as well. DM can contain inconsistent and often competing knowledge. Knowledge chunk retrieval is governed by probabilities based on activation values. As a result, retrieved knowledge may not match completely what is requested, or retrieval may even fail. It has been already suggested that DM plays a central role in casual reasoning (Drewitz & Brandenburg, 2012). The uncertainty over retrieved knowledge from DM transforms the HRM's rule-based inference into probabilistic inference. Based on example models, this paper describes how the HRM is used to simulate casual deduction, pragmatic reasoning and even inductive Bayesian inference.

Finally, little is known about the form of cognitive processes that provide meta-control over reasoning strategies. For example, how do we decide whether to use as a source of knowledge the mental model in a form of visual short-term memory or declarative memory? Not every problem context can be converted into an iconic form, and in such cases, there is no other choice but to use knowledge in declarative memory. However, what if both VSTM and DM contain relevant or even conflicting knowledge? The HRM introduces a simple, but effective cognitive construct referred to as a *reasoning pipeline* that addresses these issues. A reasoning pipeline assumes a sequential process where alternative strategies are used one by one in increasing order of cognitive effort required until a conclusion is reached. For example, access to VSTM requires less time than a declarative retrieval. Thus, the HRM prefers reasoning based on VSTM knowledge to reasoning on declarative knowledge.

Architecture of the HRM

Knowledge representation in declarative memory

Chunk types and chunks, instances of chunk types, represent factual knowledge in ACT-R. A chunk type defines a set of slots its instance chunks can inherit. Those slots can contain values describing chunk's properties. Those values can be either other chunks or atomic values such as strings of characters or numeric values. ACT-R provides no restrictions on chunk types and chunks that can be defined by a modeler. The HRM restricts a modeler to a predefined set of chunk types thereby encouraging a commitment to a common knowledge structure that is not model specific. The core set of chunk types in the HRM are ones describing concepts, triples and inference rules.

Concepts and triples

The atomic unit of knowledge in the HRM is a *concept*. Any unit of knowledge that has distinct semantic meaning can be a concept. There are two types of concepts in the HRM: property instance and class instance. Property instance is any concept that is used to relate two other concepts semantically. As such, the knowledge organization

inside the HRM revolves around a predicate construct referred to as a *triple*: (*property subject object*). Inside a triple, *property* establishes a semantic connection between *subject* and *object*. The following is an example of a triple: (*r-left-of fork plate*). In the HRM, *r-left-of* is a property instance that is used to represent a spatial relation between two class concepts. In example above, the meaning of the triple is equivalent to "*a fork is in left side of a plate*".

A property instance can also be used as triple's subject or object. For example, the HRM has two different property instances, *r-left-of* and *r-dir-left-of*, for expressing a similar spatial relation between two class instances. *r-dir-left-of* expresses semantically more restrictive spatial relation implying that subject is to the left of an object, and both subject and object are aligned vertically. Therefore, triple (*r-dir-left-of fork plate*) entails triple (*r-left-of fork plate*). One way to express such one-way relation is to have another triple (*entails r-dir-left-of r-left-of*). Here, property instance *entails* semantically connects two other property instances instead of class instances. Otherwise, *entails* is no more special from other property instances such as *r-left-of* or *r-dir-left-of*.

Most of the studies of human mental logic advocate for some form of predicate construct as a way of knowledge organization. We have chosen the triple form because it closely resembles a linguistic predicate typology consisting of subject, verb and object. It is the most common sentence structure found across different languages. Such commonality strongly indicates that underlying knowledge from which a sentence is constructed may also be organized in the same form consisting of subject, object and verb (Crystal, 1997).

The HRM has a limited notion of time. A triple can be assigned a specific timestamp. For example, the sentence "*John ate sandwiches yesterday and today*" can be expressed with two triples with the same structure but different timestamps:

(*eat John sandwich* (*ts "yesterday"*))
(*eat John sandwich* (*ts "today"*))

A special slot named *ts* is used to assign a timestamp. When necessary, the above two triples can be differentiated by timestamps, otherwise they are semantically similar. In current implementation of the HRM any value can be used as a timestamp. This implementation required the least amount of effort, but it is not a realistic representation of human temporal cognition. Ideally, there should be restrictions on what kind of values can be used to represent time. On the one hand, it can be an explicit class instance to represent our high level understanding of time and data. On the other hand, timestamp value can be more implicit estimations of time intervals done by our internal biological clock. ACT-R already provides a temporal module (Taatgen, Van Rijn & Anderson, 2007) that provides such time interval estimations. Future updates of the HRM should include more restrictions on time values as well as integration with the temporal module.

Statements

In the HRM, *statement* is a type of triple that represents factual knowledge. It is a statement of a fact that is true or was true. The example triples from the preceding subsection are all valid *statements*. The HRM provides several ways to create a *statement*. Firstly, a modeler can explicitly define custom *statements*, as model's background knowledge. Secondly, the model itself can create *statements* in real-time via production rule calls to a special *reasoner* buffer. This option simulates the ability to obtain new explicit knowledge through external input, such as stimuli from the outside world. Finally, a model can generate a new statement by inferring it from existing statements using top-down reasoning, or by deriving it from an implicit connection between concepts using bottom-up reasoning.

Implicit and explicit knowledge

The HRM makes a distinction between explicit and implicit knowledge. *Statements* are explicit knowledge, a form of a knowledge that is known consciously. Implicit knowledge is knowledge that is represented by slot values of concept chunks. Such knowledge is implicit because it is assumed that ACT-R is not consciously aware of its presence, but subconsciously can extract it to form explicit *statements* using bottom-up processes. Following the previous example, there may not be any *statement* such as (*r-left-of fork plate*). However, concepts chunks for *fork* and *plate* may have slot values with *x* and *y* coordinates implicitly indicating relative spatial positions of two concepts. Those values then can be converted into explicit concepts such as *r-left-of* when necessary.

Inference rules

In the HRM, rules describe how a new *statement* can be inferred from existing *statements*. The HRM assumes that rules reflect our past experience and are formed as a result of our observations of relations among real-world entities such as cause/effect, pre-condition/action, action/post-condition observations, etc. Rules use special triples called

rule-statements. Semantically, a rule-statement is not a fact, but either a condition or an implication of a possibility. Any rule consists of left- and right-hand sides. A left-hand side must have one or more rule-statements (antecedent), and the right hand-side should have exactly one rule-statement (consequent). In order for a consequent to be true, all antecedent rule-statements should also be true. For example, the rule below states "*if the fork is on the left of the plate then the plate is on the right of the fork*":

(*r-left-of fork plate*) ==> (*r-right-of plate fork*)

Unlike ordinary *statements*, rule-statements can use variables as one of the entities in the triple. The previous example rule can be rewritten as:

(*r-left-of "@item" plate*) ==> (*r-right-of plate "@item"*)

Above rule states "*if any item is on the left of the plate then plate is on the right of that item*". In this rule, "*@item*" is a variable, not a *concept*. The HRM recognizes as a variable any string value that starts with "@". It can be replaced by any valid concept that is factually on the left side of the plate. Variables provide a possibility to generalize rules beyond a scope of a particular concept or even an entire model. It also introduces a possibility to reuse the same rules across different ACT-R models, at least partially, addressing one of the major reusability challenges in ACT-R.

Assertion

Assertion is another type of triple used by the HRM. Assertion represents a query questioning the HRM whether a triple is true. For example, the assertion (*r-right-of plate fork*) represents the query: "*Is the plate on the right side of the fork?*" Similar to rule-statements, assertions can have variables. The assertion (*r-right-of plate "@item"*) asks the HRM to find any class instance that is on the right side of the *plate*. In ACT-R, the HRM can be queried with an assertion via *reasoner* buffer. Upon receiving an assertion, the HRM starts a reasoning process called a *backward reasoning pipeline*. The task of reasoning pipeline is to check if assertion can be proven to be true or to find/prove any *statement* that matches the assertion if assertion contains variables. If assertion is true then it is converted into a *statement* and placed inside *reasoner* buffer. If a matching *statement* is found then that *statement* is put inside *reasoner* buffer.

Schema and inference types

Conditional proof schema

The HRM uses the same conditional proof schema defined by Braine & O'Brien (1991): to derive or evaluate *if p then q*, first suppose *p*; when *q* follows from the supposition of *p* together with other information assumed, one may assert *if p then q*. This schema together with the open world assumption has several implications that make the HRM's inference different from an inference based on material conditionals of a classical logic:

1. The HRM does not follow the closed world assumption unless it is explicitly required. Therefore, what the HRM cannot prove is not necessarily false.
2. There can be two or more competing or conflicting inference rules that can be true at different instances: e.g. *if p then q*; *if p then k*. For example, the agent may build following two inference rules through observations of rolling dice: *If throw dice then get 6*; *If throw dice then get 3*.
3. The *sufficiency* requirement will not necessarily hold: the antecedent *p* is not necessarily a sufficient condition for a consequent *q* because other information may be assumed to assert *if p then q*. Consider following common sense rule: *If brakes are pressed then car stops*. Most of the times, the rule is true. However, there it is assumed that, for example, the brakes are not broken.
4. The *validity* requirement of deductive reasoning will not necessarily hold: the conclusion may not be true even if the premises are true. For example, the HRM may fail to assert *if p then k* because it already asserted *if p then q*. Consider the dice example from the implication 2. If a dice is thrown then the HRM may assume that result is 6. The second possible conclusion of 3 remains untrue even though its premise of dice being thrown is true. Furthermore, the validity requirement cannot hold if the sufficiency requirement is not met.
5. The law of *contrapositive*, or *Modus Tollens (if p then q, therefore if ¬q then ¬p*), also does not necessarily hold. Consider the contrapositive of the example from the implication 3: *If a car hasn't stopped then the brakes were not pressed*. Because of violation of the sufficiency requirement, the contrapositive argument may not be true: *The brakes were pressed, but the car hasn't stopped because the brakes were broken*. In this case, the assumed information that the brakes are not broken is not true. Therefore, the HRM does not automatically

generate contrapositives from inference rules. The HRM assumes that a contrapositive should be observed and memorized as an inference rule of its own right.

The *law of syllogism* (*if p then q, if q then k, therefore if p then k*) is at the center of the HRM's capability for complex reasoning. Consider following example: *If the sun sets then a night comes. If a night comes then a temperature drops. Therefore, if the sun sets then a temperature drops*. There is no explicit relation between the sun setting and the temperature dropping in two rules. However, it can be inferred using of law of syllogism. The ability to chain the inference rules together allows the HRM to explore different reasoning strategies with the same inference process.

Reasoning types

The inference rules can be used for two types of reasoning in the HRM: backward and forward. Backward reasoning is used to determine whether a specific conclusion can be reached. Forward reasoning is used to determine what kind of conclusion can be reached given set of evidences. Backward reasoning retrieves an inference rule by matching its consequent, while forward reasoning retrieves the inference rule by matching its antecedent. For further explanation, let us assume that there is the following *Rule 1*:

Rule 1: *(have-state brake pressed)* *(NOT-have-state brake broken)* *==>* *(decrease car speed)*	Interpretation: *If a brake is pressed, and it is not broken then car speed decreases.*

With *Rule 1*, the HRM can answer two types of questions. The first question is "*Is car speed decreasing?*". It is a question answerable by backward reasoning. The HRM's equivalent of this question will be an assertion *(decrease car speed)* sent to a *reasoner* buffer with an expected conclusion that it is true or not true. The assertion will be true if there is a rule that (1) has a consequent matching the assertion and (2) has an antecedent where all rule-statements are true or inferred to be true via the law of syllogism. In this case, the HRM will use the *Rule 1* because its consequent matches the assertion. However, to infer that the assertion is true the HRM will also have to infer that *Rule 1*'s antecedent is also true. We will discuss later various strategies used for such inference.

The second question is "*What happens if the brake is pressed, and it is not broken?*". It is a question answerable by forward reasoning. The HRM's equivalent of this question will be supplying two facts, *(have-state brake pressed) and (NOT-have-state brake broken),* to the *reasoner* buffer and expecting some or no conclusion. The conclusion will be reached if there is a rule that (1) has an antecedent matching the given facts in the *reasoner* buffer and (2) has an antecedent where all rule-statements are true or inferred to be true due the law of syllogism. The facts in the *reasoner* buffer can be used to assert truth-values of the antecedent. In this case, the HRM concludes that the car speed should decrease (*decrease car speed*) because of the *Rule 1*. It is possible to ask another question such as "*What happens if a brake is pressed?*". The HRM's equivalent of this question will be supplying only single fact *(have-state brake pressed)* to a *reasoner* buffer. However, according to the *Rule 1*, the second fact, *(NOT-have-state brake broken),* is required to reach a conclusion. In such case, the HRM will try to prove the second fact using backward reasoning.

A *reasoning pipeline* provides a meta-cognitive control over reasoning processes. The HRM uses two reasoning pipelines for backward and forward reasoning respectively. In ACT-R, reasoning pipelines are implemented as a series of automated calls to production rules built into the HRM. These production rules are task-general reasoning rules and are part of the cognitive architecture. This approach differs from traditional ACT-R modeling practices that treat all production rules as part of a model. On the other hand, the declarative inference rules are often treated (but not necessarily always) as being task-specific. The inference rules together with statements of facts provide a problem context within which the task-general production rules can reason and derive conclusions.

Following the threaded cognition theory (Salvucci & Taatgen, 2008, 2011), reasoning pipelines are contained within the HRM's own cognitive thread that runs in parallel with other (model-specific) cognitive threads. This means that model-specific production rules irrelevant to reasoning pipelines can fire in-between production rules belonging to the HRM. It opens the possibility that declarative retrievals requested by other threads can interfere with the HRM's reasoning that relies heavily on declarative memory. Such interference is possible despite the fact that ACT-R locks access to declarative memory during individual retrieval instances (it is not possible to recognize a thread that initiated retrieval). Therefore, the HRM uses a stricter control that locks declarative memory through entirety of the reasoning pipeline.

Backward reasoning pipeline

As it was discussed earlier, new knowledge can be generated from existing knowledge using one of several different strategies. The backward reasoning pipeline establishes priority among those strategies and organizes them into series of consecutive steps. The highest priority strategy receives an assertion first and tries to prove it. If it fails then the assertion is passed to the next highest priority strategy. The HRM triggers calls to backward reasoning pipeline as soon as it receives an assertion request inside *reasoner* buffer. The backward reasoning pipeline recursively calls itself (the law of *syllogism*) until either the assertion is proven or it is decided that the assertion cannot be proven.

Currently, backward reasoning pipeline supports three different strategies: bottom-up reasoning, declarative retrieval and top-down reasoning. Fig. 3 shows the prioritization of those strategies. Bottom-up reasoning is preferred requiring the least amount of cognitive effort. Bottom-up reasoning is followed by declarative retrieval and top-down reasoning in decreasing order of priority.

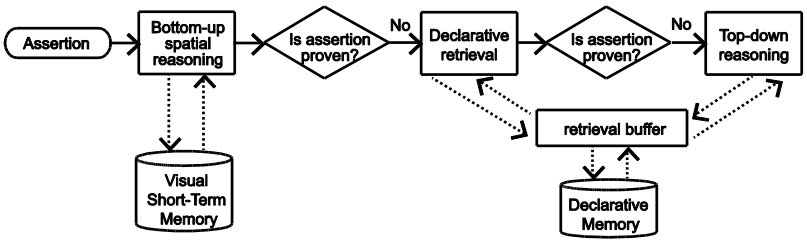


Fig. 3 A simplified workflow of an HRM reasoning pipeline in ACT-R.

Bottom-up reasoning

The current implementation of the HRM's visual bottom-up reasoning supports only spatial reasoning. As with other forms of reasoning, spatial reasoning requires a source of knowledge based on which it can derive a new knowledge. In the HRM, such knowledge source is a visual short-term memory (VSTM). VSTM was introduced by the newer version of the Pre-Attentive and Attentive Vision module (Nyamsuren & Taatgen, 2013), an extension to ACT-R's default vision module. VSTM is a high resolution, but low capacity visual memory. Every visual object encoded from the external world is temporarily stored inside VSTM until it decays out or is deleted due to capacity limitations. Unlike declarative memory, VSTM is considered as a visual analog of a working memory. Hence, objects inside VSTM can be accessed by the HRM with no cognitive cost, and explicit knowledge can be derived with little effort.

The HRM can take advantage of VSTM whenever it receives an assertion about spatial relation between two concepts such as (*r-right-of plate fork*). VSTM contains detailed information about each visual object currently in its store, including the object's original position in real world. In ACT-R, those are two-dimensional spatial coordinates. The HRM can use such implicit knowledge to quickly derive explicit *statements* about spatial relations between concepts inside VSTM. If one of those derived statements supports the assertion then the assertion is proven.

Declarative retrieval

If bottom-up reasoning fails then the HRM tries to retrieve from declarative memory any *statement* that can directly confirm the assertion. In ACT-R, a declarative retrieval can be a time-costly process. Furthermore, there is a chance that retrieval will fail even if a matching *statement* exists. Those are the reasons why bottom-up reasoning takes priority over declarative retrieval as a more reliable and faster process.

Top-down reasoning

Top-down reasoning is only invoked if declarative retrieval fails. It involves rule-based reasoning where a chain of inference rules is used to prove an assertion.

The current implementation of the HRM supports a fully functional backward-chaining algorithm implemented as a set of ACT-R production rules. The first production retrieves from declarative memory any consequent rule-statement that matches the assertion. If the retrieval of a rule's consequent is successful then the next production retrieves the first antecedent rule-statement of the same rule. The retrieved antecedent rule-statement is converted into an assertion and fed back to the HRM. This starts a new recursive call with a new reasoning pipeline. If recursive call was able to prove that current antecedent rule-statement is true then the next antecedent rule-statement is retrieved, converted into assertion and fed back to the HRM. This process continues until all antecedent rule-

statements are proven. In such a case, the consequent rule-statement is also true, and, hence, the original assertion is true as well. If any of the antecedent rule-statements cannot be proven then the HRM stops the reasoning process and sets the *reasoner* buffer to an error state.

The top-down reasoning consists of a series of production calls coupled with frequent declarative retrievals. Not only it is a hugely time-consuming process, but also it is very costly in terms of cognitive resources. Since ACT-R allows only one production call at the time, it creates a bottleneck for other task-specific productions. Furthermore, declarative memory is locked through entirety of the time the HRM uses it to prove an assertion. Hence, other cognitive processes cannot access declarative memory. The overall high cost puts top-down reasoning in the lowest priority position.

Forward reasoning pipeline

The simplified workflow of a forward reasoning pipeline is shown in Fig. 4.Given statements of facts as a query, the HRM retrieves from declarative memory any rule that has antecedent rule-statements matching the statements in the query. A rule selection is governed by several criteria. Firstly, a rule must have rule-statements matching all query statements. Secondly, the order of rule-statements must be the same as the order of corresponding query statements. Thirdly, irrelevant rules that may not lead to a desired conclusion can be ignored. One of the unique aspects of human reasoning is that we can do it with an intention of achieving a particular conclusion. It is also possible to do the same in the HRM. If a target concept is specified in a query then the HRM ignores all rules that do not mention that concept in its consequent rule-statements. All three criteria applied to rule retrieval are based on principles of memory retrieval during decision-making under uncertainty. It was suggested that memory chunks are evaluated during retrieval with respect to relevancy, availability and accessibility (Kahnemann, 2003) as well as cross compared with alternative retrieval candidates (Schooler & Hertwig, 2005).

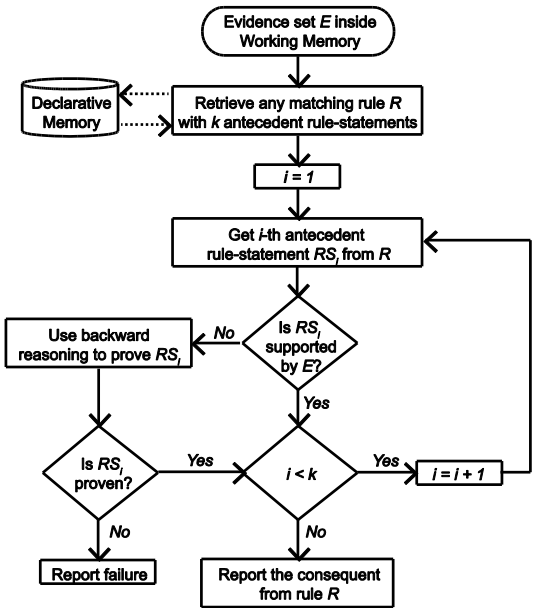


Fig. 4 A simplified workflow of a forward reasoning pipeline.

If forward reasoning uses a rule that has antecedent rule-statements that were not specified in the query then those rule-statements are verified for truthfulness by invoking a backward reasoning. As such, forward reasoning is not purely forward and can have series of nested backward reasoning calls. This is different from traditional view where backward and forward reasoning are considered two distinct processes. The heterogeneous nature of the reasoning pipeline significantly increases a range of inference problems that the HRM can solve. The power of mixed forward and backward reasoning will be explored via example model of blicket categorization task (Griffiths, Sobel, Tenenbaum & Gopnik, 2011).

Validation Models

This section introduces three models of different experimental tasks. Each model is used to replicate human behavioral and validated against human performance data.

The model of a casual deduction task is used to demonstrate the HRM's basic reasoning abilities based on inference rules in declarative memory. It is the simplest of the validation models described in this study. It uses only declarative knowledge and does not require other modules such as vision. The reasoning strategy used by this model is limited to declarative retrieval of rules. The model demonstrates how competing and conflicting declarative knowledge can affect outcomes of even simple reasoning. It shows the importance of considering uncertainty in declarative retrieval results during any logical reasoning task.

The model of a spatial reasoning task demonstrates the full potential of the HRM's backward reasoning ability. It uses all three backward reasoning strategies: bottom-up reasoning, declarative retrievals and top-down reasoning with recursive calls to the backward reasoning pipeline. The reasoning in this model uses knowledge in declarative memory as well as in visual short-term memory.

This final model based on a blicket task is a demonstration of the HRM's ability to use both inductive and deductive reasoning approaches to solve problem of inferring cause and effect relationship from series of observations. The model mainly uses forward reasoning with series of nested calls to backward reasoning. In addition to declarative knowledge, the model is presented with new knowledge during the progress of the trial. As such, it is a good demonstration of how reasoning outcomes can change based on dynamic events even if the underlying set of inference rules remains the same.

Model of Casual Deduction Task

Cummins, Lubart, Alksnis and Rist (1991) and Cummins (1995) extensively studied the process of casual deduction. Subjects are provided with a sentence describing a cause/effect in a form of "*If <cause>, then <effect>*". The sentence is followed by four different forms of arguments: Modus Ponens (MP), Affirming the Consequent (AC), Modus Tollens (MT) and Denying the Antecedent (DA). Each argument consists of a fact and an implication. Subjects are asked to evaluate how likely it is that the implication is true given a cause/effect sentence and the argument's fact. Here is an original example from Cummins et al. (1991) of a cause/effect sentence: "*If the brake was depressed, then the car slowed down.*" The four arguments with respect to this sentence are: "*The brake was depressed. Therefore the car slowed down.*" for MP; "*The car slowed down. Therefore the brake was pressed.*" for AC; "*The car did not slow down. Therefore, the brake was not depressed.*" for MT; and "*The brake was not depressed. Therefore, the car did not slow down.*" for DA.

The study revealed that acceptance of arguments is influenced significantly by subjects' previous experience. The casual deduction was sensitive to two factors: alternative causes and disabling conditions (Cummins et al., 1991). An alternative cause is a cause that is different from one given in a sentence but still can result in the same effect. A disabling condition is a condition that prevents the effect from occurring despite the presence of a cause. Fig. 5 shows the acceptance ratings of the four conditions gathered from two separate studies. Firstly, there is a robust effect of disabling conditions on acceptance of MP and MT arguments. When there are many possible disabling conditions, subjects are less likely to accept truthfulness of these two types of arguments. Secondly, there is a persistent effect of alternative causes on acceptance of DA and AC arguments. When there are many possible alternative causes of the effect, subjects are less likely to accept DA and AC arguments. Thirdly, it is not surprising that the acceptance rating varies a lot between two studies. The nature of the task is extremely subjective and participants' previous experiences vary a lot. It is likely that the rating further depend on the specific materials used in two experiments.

Using an ACT-R model that uses the HRM's knowledge structure, we explore the nature of effects invoked by alternative causes and disabling conditions on our ability of casual deduction.

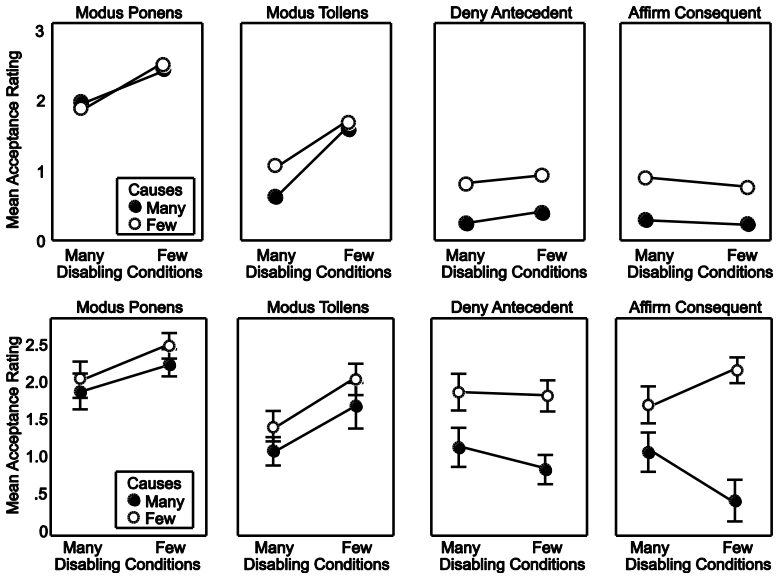


Fig. 5 Mean acceptance ratings of four argument forms in casual deduction experiments conducted in (a) Cummins et al. (1991) and (b) Cummins (1995).

Model's knowledge structure

In this experiment, the model used the same 16 cause/effect sentences described in Cummins (1995). The model stored both affirmative and negatives versions of all 16 sentences in its declarative memory in form of rules. For example, the previously mentioned example cause/effect sentence was converted to the following two rules:

Rule 1:
(*have-state brake pressed*)
==>
(*decrease car speed*)

Rule 2:
(*NOT-decrease car speed*)
==>
(*NOT-have-state brake pressed*)

Inside declarative memory, the model also had alternative causes and disabling conditions for each sentence. They were also stored in form of rules. Here is an example of affirmative and negative rules for an alternative cause:

Rule 3:
(have-state car go-uphill)
==>
(decrease car speed)

Rule 4:
(NOT-decrease car speed t)
==>
(NOT-have-state car go-uphill)

An affirmative version of the same disabling condition can be written as two following rule forms:

Rule 5:
(have-state brake pressed)
(have-state brake broken)
==>
(NOT-decrease car speed)

Rule 6:
(have-state brake pressed)
(NOT-decrease car speed)
==>
(have-state brake broken)

Both forms were stored in declarative memory. Finally, an example of a negative version of a disabling condition would be as following:

Rule 7:
(have-state brake pressed)
(NOT-have-state brake broken)
==>
(decrease car speed)

Sentences were divided into four groups. In Many/Many group, a sentence had three disabling conditions and three alternative causes. In Many/Few group, there were three disabling conditions and one alternative cause. Similarly, the other two groups were Few/Many and Few/Few.

Model's reasoning strategy

With each sentence, the model had to do four trials, one for each argument form. The model's general strategy was simple: given an argument, retrieve any matching rule from declarative memory and verify if the rule supports the argument. The workflow of the strategy is shown in Fig. 6. Depending on the argument form, the model used different forms of reasoning. For MP arguments, the model did forward reasoning with fact. It retrieved any rule that had antecedent rule-statement matching the fact and checked if retrieved rule's consequent matched the implication. If a match was found, then the argument was accepted. For AC arguments, the model did backward reasoning with fact: it retrieved any rule that had consequent matching the fact and checked if any of the antecedent rule-statements matched the implication. If match was found then argument was accepted. In a similar manner, the model did forward reasoning with fact for MT arguments and forward reasoning with implication for DA arguments.

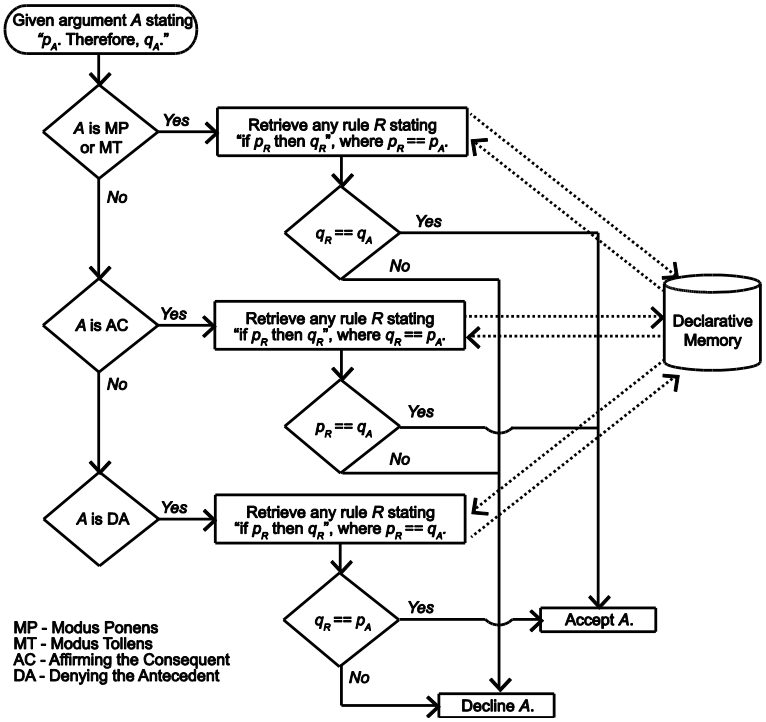


Fig. 6 A workflow of the strategy used by the model of the Casual Deduction task.

Results

The model repeated the same experiment 50 times, accounting to a total of 3200 trials. Fig. 7 shows proportions of trials where arguments were accepted. The proportions were calculated separately for each combination of four argument forms and sentence groups. The model shows the same behavior as human subjects. The model is more likely to accept MP and MT arguments for cause/effect rules that have few disabling conditions. Next, the model is more likely to accept DA and AC arguments for cause/effect rules that have few alternative causes.

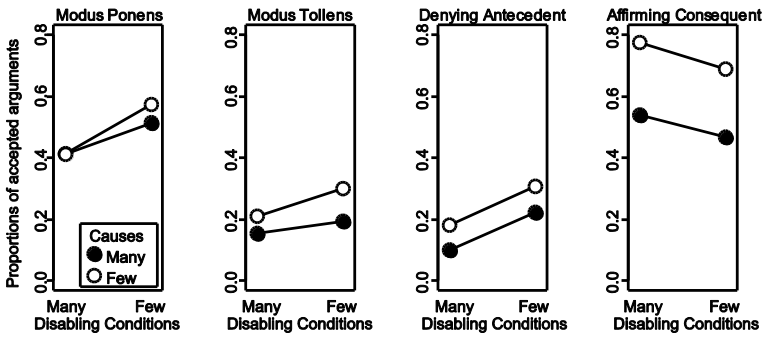


Fig. 7 Proportions of arguments accepted by the model in four forms of arguments.

The effects are explained by a mutual interference among rules during the step when the model tries to retrieve a proper rule that can support an argument. For example, let us assume that the model received following MP argument:

Fact: (have-state brake pressed)
Implication: (decrease car speed)

In this scenario, the model will use the fact (*have-state brake pressed*) to retrieve any rule with matching antecedent rule-statement. These include not only the original cause/effect rule 1, but also the affirmative and negative disabling condition rules 5, 6 and 7. In presence of several matching chunks during a declarative retrieval, ACT-R randomly picks one. The rules 5 and 6 have consequents that are different from the argument's implication. Therefore, if either rule 5 or 6 is retrieved then the model will not accept the argument's implication. It is quite easy to see that as the number of disabling conditions increases, the model will be less likely to retrieve a rule that supports the argument and, hence, more likely to reject it. This rule interference mechanism is also responsible for the effects observed in other three argument forms.

One aspect that should be considered is that the rules have the same activation values in the model. Hence, the rules have the same probability of retrieval. This is an unlikely scenario with human subjects. Firstly, an activation value for the same rule may differ between subjects. Secondly, different rules may have different activation values within a subject. For example, despite leading to the same conclusion, rule 7 is likely to have less activation than rule 1 because people do not worry often about state of the brakes. Use of equal activation values for all rules may have affected model fit. The model's acceptance rate of AC arguments is a bit higher than subjects' rate. Assigning lower activation values to negative versions of disabling conditions, such as rule 7, can decrease the acceptance rate of AC arguments and result in better fit. However, the current simpler version of the model serves better for the purpose of demonstrating the HRM's basic reasoning capabilities.

It is certainly possible that other computational models can explain the same effects. However, in case of our model the main explanatory power comes not from the model built for this specific task, but rather from the aspects of the cognitive architecture: a combination of ACT-R's activation-based declarative memory and the HRM's conditional proof schema (Braine & O'Brien, 1991).

Model of Spatial Relations Task

This task is used to study people's fundamental ability to derive a spatial relation from a set of premises. Three problems below are examples of such task. In each problem, subjects are given four premises and then queried about the spatial relation between two items that were not explicitly connected in any of the premises.

The studies showed that people prefer to use strategy of mental states rather than formal representations (Byrne & Johnson-Laird, 1989). In such strategy, people build mental states or imagery using abstract objects representing items in the premises. Such mental state is built iteratively as premises are processed one by one (Carreiras & Santamaria, 1997). With such mental states, the spatial relation between two query items can be derived directly. Examples of mental states are shown below. Problem 1 results in one mental state. Problems 2 and 3 result in two possible mental states. Furthermore, the same studies have shown that one-state problems are easier than two-state problems.

Problem 1:
1. A is on the right of B
2. C is on the left of B
3. D is below C
4. E is below B
What is the relation between D and E?

Possible mental state:
C B A
D E

Problem 2:
1. B is on the right of A
2. C is on the left of B
3. D is below C
4. E is below B
What is the relation between D and E?

Possible mental states:
C A B A C B
D E D E

Problem 3:
1. B is on the right of A
2. C is on the left of B
3. D is below C
4. E is below A
What is the relation between D and E?

Possible mental states:
C A B A C B
D E E D

Byrne and Johnson-Laird (1989) reported 61% and 50% correct responses in one- and two-state problems respectively. Similarly, Carreiras and Santamaria (1997) reported 99% and 89% correct responses in one- and two-state problems. There are also two-state problems that have no valid conclusion. In those problems, mental states resulted in contradicting relations between two query items, and subjects were required to report that there is no single solution. For example, Problem 3 results in two possible mental states contradicting each other. Problems with no valid conclusion result in the lowest proportion of correct responses. Two separate experiments by Byrne and Johnson-Laird resulted in 18% and 15% of correct responses in problems with no valid conclusions.

It is assumed that a two-state problem is more difficult because it requires higher working memory load than a one-state problem. However, it does not explain why accuracy drops even lower in a two-state problem with no valid conclusion. Both types of two-state problems have equal numbers of mental states, premises and items. Furthermore, both types of problems require two swaps to derive the second mental state from the first one. Therefore, the working memory load should be the same in both types of problems. As result, an explanation based on a working memory load is not sufficient to explain subjects' performance. Our ACT-R model that uses the HRM module provides a possible explanation for this effect.

Model's design

The model's strategy can be divided into five steps:

1. The model constructs a mental state of the problem inside VSTM. The mental state is built iteratively by processing premises one at a time and updating VSTM on each iteration. Items from a premise are converted into abstract visual objects and given (*x, y*) coordinates based on positions relative to the items already existing inside VSTM. A premise is also converted into a logical *statement* stored inside declarative memory, but it is done only after VSTM is updated (Fig. 8a). The model can handle two-state problems. For example, while processing the second premise in Problem 2, the model uses *assertion* (*r-dir-left-of "@item" B*) to check if there is already another item present to left of *B*. This assertion triggers bottom-up spatial reasoning and the HRM returns any visual object that is to the left of *B*. In case of Problem 2, *A* is returned. Then the model stores both *C* and *A* inside its working memory as items to be swapped positions in a second mental state (Fig. 8b). In Problem 2, the mental state inside VSTM will be as following at the end of step 1:

C A B
D E

2. After all premises are processed and a mental state is built inside VSTM, the model sends an assertion to the HRM to try to answer a query. The assertion is in form of (*"@property" D E*). To answer the assertion, the HRM uses bottom-up spatial reasoning to evaluate relative positions of *D* and *E* inside VSTM. In case of Problem 2, the model's answer will be either (*r-left-of D E*) or (*r-dir-left-of D E*).

3. If it is a one-state problem then the model does nothing else. However, if there are two possible mental states then, after answering the query and storing it in declarative memory, the model creates the second possible mental

state inside VSTM. This is done by swapping positions of the two objects previously stored inside working memory. In case of Problem 2, *C* is placed at the position of *A*, and *A* is placed at the former position of *C* changing the mental state inside VSTM into following:

A C B
D E

4. At this step, the model checks if any visual object was positioned relative to the swapped objects. If that is the case then the model verifies if relations still hold, if not then positions of those objects are corrected as well.

5. After creating the second mental state, the model sends to the HRM the same assertion as in step 2. The answer for this assertion is compared to the answer from step 2 that is retrieved from declarative memory. If answers are not the same then the model assumes that problem does not have valid conclusion and reports the inconsistency.

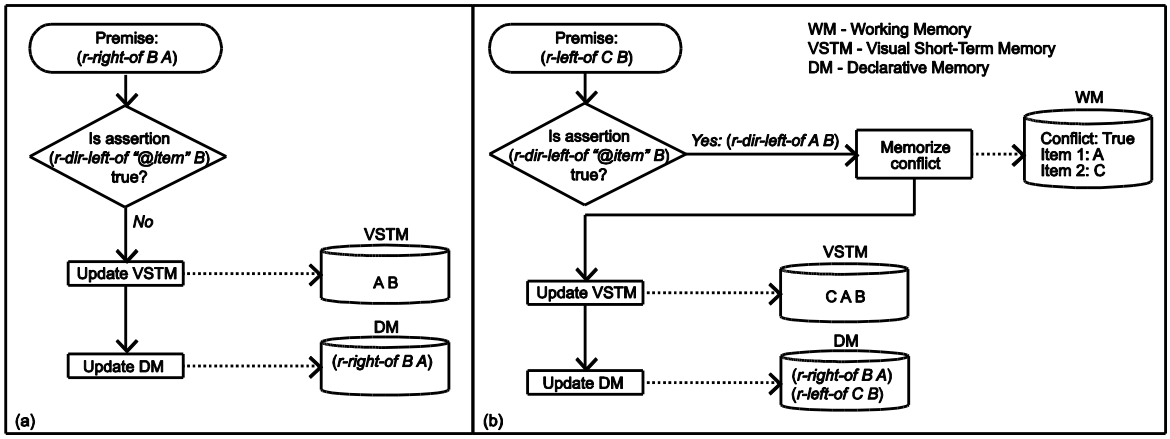


Fig. 8 Two diagrams are describing how the model processes the first (a) and the second (b) premises of the Problem 2 during step 1 of the strategy.

Results

Model's proportions of correct responses in one-state problems, two-state problems with valid conclusion and two-state problems with no valid conclusion are 100%, 74% and 31% respectively. The model always gives correct answers in one-state problems. However, it starts making mistakes in two-state problems. Furthermore, the model shows lowest accuracy in two-state problems with no valid conclusion. The cause of mistakes is model's confusion between similar spatial properties such as *r-below* and *r-dir-below*.

The first mistake can be made during step 4. Consider following example from Problem 3 where the model just finished step 3 by swapping positions of *A* and *C*:

C A B ==> A C B
D E D E

During step 4, the model has to verify whether the spatial relation between *D* and *C* still holds. One of two possible assertions can be used for such verification: *(r-below D C)* or *(r-dir-below D C)*. The model's choice is random in this case. However, if *r-below* is used then the assertion will be evaluated to be true since bottom-up reasoning with *r-below* does not check for vertical alignment. This leads the model to a wrong conclusion that *D*'s position does not need to be corrected. Such mistake can lead to a situation where, for example, in Problem 3, the relation between *D* and *E* is still the same in both mental states. The second mistake can be made during comparison in step 5. Let us consider the case where, in Problem 2, the answers to the assertions in step 2 and 5 were *(r-left-of D E)* and *(r-dir-left-of D E)* respectively. These two statements, although similar, are not the same. Hence, if no explicit top-down reasoning is used to prove that one entails the other, the two answers are considered different. The model decides randomly whether to invoke top-down verification since it is not always necessary.

The model makes more mistakes in two-state problems with no valid conclusion because it is vulnerable to both types of mistakes in those problems. However, only second mistake is possible in two-state problems with valid conclusion. In one-state problems, the model is not susceptible to any of those mistakes.

Model of Bayesian-like Inference in Blicket Task

We focused on a simulation of the first experiment conducted by Griffiths et al. (2011). The task context consists of ordinary pencils (blocks) and super pencils (blickets). We further refer to ordinary and super pencils as blocks and blickets. Subjects were asked to rate on a scale of 1-7 how likely a block was to be a blicket. Subjects' ratings were based on observations that consisted of one or two blocks placed on a special detector. The detector activated when at least one blicket was placed on it. Griffiths et al. used this task to study people's ability to infer casual relations based on number of observations and prior knowledge.

The experiment consisted of three consecutive phases: a training phase, AB-event phase and A-event phase. Subjects were divided into five groups that received different trainings during the first phase. During the training, each subject was shown ten blocks placed individually on the detector one after another. Some blocks activated the detector (Fig. 9a) others did not (Fig. 9b). A subject's group determined the frequency of blickets among the ten blocks. In group 1/6, only one of ten blocks was a blicket. In group 1/3, three of ten were blickets. Similarly, subjects in groups 1/2, 2/3 and 5/6 observed five, seven and nine blickets respectively.

After the training phase, the subjects were shown two new blocks, A and B. At this point, subjects were asked to provide initial ratings of how likely each was to be a blicket. Following the initial ratings, both A and B blocks were simultaneously placed on the detector causing it to detect a blicket (Fig. 9c). This phase is referred to as AB-event. After AB-event, subjects were asked to rate both blocks again. Finally, block A was placed alone on the detector activating it (Fig. 9d). This phase is referred to as A-event. Subjects were asked to rate A and B blocks after A-event as well.

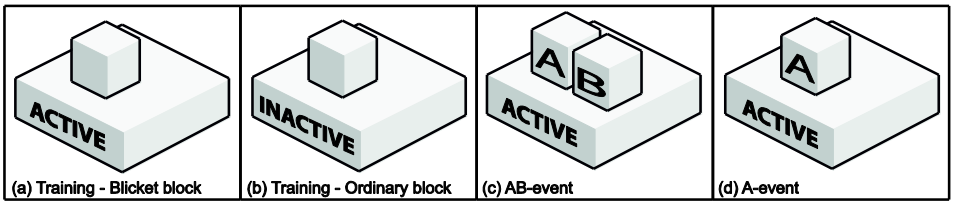


Fig. 9 (a) A blicket activates the detector during the training phase. (b) The detector remains inactive when ordinary block is placed on it during the training phase. (c) Two blocks, A and B, are placed on the detector activating it during AB-event. (a) During A-event, only block A is placed on the detector activating it.

Before conducting the experiment, Griffiths at al. (2011) created a Bayesian model predicting the probabilities of objects A and B being rated as blickets. Fig. 10a shows those predictions for all five groups. According to the model predictions, the initial ratings reflect prior probabilities of encountering a blicket established by a training phase. Those ratings are higher in groups that observe a higher number of blickets during the training phase. After AB-event, the mean ratings increase above baseline level. However, such increase gets smaller as baseline prior probability gets higher. After A-event, the object A is given a maximum rating. However, the rating of object B goes down. As shown in Fig. 10b, subjects' mean ratings closely follow predicted Bayesian probabilities.

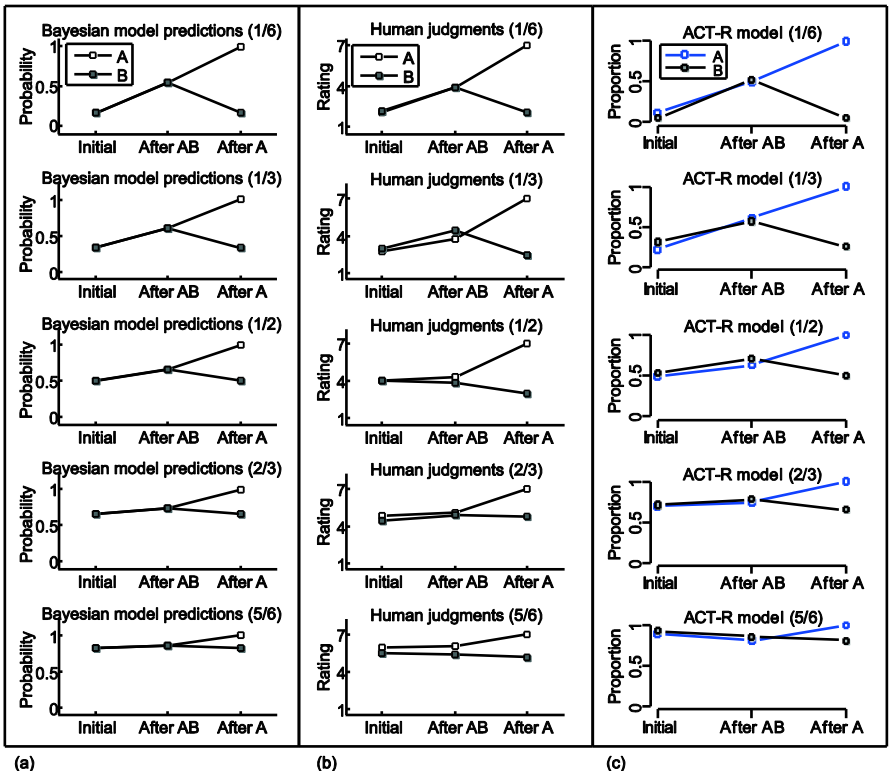


Fig. 10 (a) Probability predictions of the Bayesian model created by Griffiths et al. (2011). (b) Human mean ratings of A and B pencils at initial stage, after AB event and after A events. (c) Probabilities produced by our ACT-R model.

Model's knowledge structure

In addition to the basic set of concepts describing elements of the task, the model started with a core set of inference rules that are used to reason based on both previous experience and real-time evidence. Those rules are described on Table 1.

Rules 0 and 1 reflect the commonly reported simple inductive strategy of solving a problem by analogy (Gentner, Holyoak, & Kokinov, 2001; Winston, 1980). Analogies are the basis for any integrated cognitive systems (Gust, Krumnack, Kühnberger, & Schwering, 2008). Therefore, it is reasonable to assume that subjects have rules to classify blocks by analogy given uncertainty.

Rules 2-7 reflect the task structure and instructions subjects receive during the introduction to the experiment. Subjects were given demonstrations of blocks and blickets and their interactions with the detector. Subjects were shown cases with one and two blocks blocks placed on the detector simultaneously. The demonstrations were given to ensure that subjects clearly understood the activation laws. Rules 2 and 3 reflect laws of activation when only one block is placed on the detector. Rules 4-7 reflect laws of activation when two blocks are placed on the detector at the same time.

Rule 8 is based on a *backward blocking* paradigm (Chapman, 1991; Miller & Matute, 1996; Shanks, 1985). When subjects are shown two cues (A and B) occurring with an outcome, subjects associate both cues with the outcome. Next, if subjects are shown only one of those cues (A) occurring with the outcome then subjects associate only the latter cue (A) with the outcome. The diminished association between the second cue (B) and the outcome in light of latter evidence is backward blocking effect. Furthermore, Sobel, Tenenbaum and Gopnik (2004) found that the degree of cue-outcome association in backward blocking is affected by the base rates of blickets. Similarly, Rule 8 considers the order of evidence and base rate of blickets to re-evaluate chances of a block being a blicket.

Model's overall strategy

The overall strategy consists of two major steps. The first step is evaluating presented evidence. This step is done every time the model is presented with one or more blocks placed on the detector. The model categorizes each block

based on the detector's state and prior knowledge. Such evidence evaluation is done via forward reasoning using rules on Table 1. The resulting categorizations of blocks are stored in model's declarative memory.

The second step is a query response. When a query asking to categorize a block is received, the model tries to retrieve from declarative memory the recent categorization of the queried block. If the retrieval is successful then the retrieved categorization is reported. Otherwise, the model uses an analogy-based induction to decide block's category. The model retrieves another block that was already categorized and assigns its category to the queried block. The second step was implemented as two sequential backward reasoning calls for retrieval and analogy-based induction respectively.

Table 1 Core set of rules used by the model to categorize A and B blocks.

Rules	Descriptions
Rule 0: *(have-role"@block1" "@role" (ts "base"))* *==>* *(have-role "@block2" "@role" (ts "init"))*	If a block on the antecedent has some category then assign the same category to the block in the consequent.
Rule 1: *(have-role "@block" "@role" (ts "@t1"))* *==>* *(have-role "@block" "@role" (ts "@t2"))*	If a block had some category at some time *t1* then it has the same category at some time *t2*.
Rule 2: *(alone-on "@block" Detector (ts "@t1"))* *(have-state Detector Active (ts "@t1"))* *==>* *(have-role "@block" Blicket)*	If a block is alone on the active Detector then it is a blicket.
Rule 3: *(alone-on "@block" Detector (ts "@t1"))* *(have-state Detector Inactive (ts "@t1"))* *==>* *(have-role "@block" NON-Blicket)*	If a block is alone on the inactive Detector then it is not a blicket.
Rule 4: *(on "@block1" Detector (ts "@t1"))* *(on "@block2" Detector (ts "@t1"))* *(have-state Detector Active (ts "@t1"))* *==>* *(have-role "@block1" Blicket)*	If, at the same time, two blocks are on the active Detector then the first block is a blicket. *Rule 5* is similar to *Rule 4*, but concludes that the second block is a blicket.
Rule 6: *(on "@block1" Detector (ts "@t1"))* *(on "@block2" Detector (ts "@t1"))* *(have-state Detector Active (ts "@t1"))* *(have-role "@block1" Blicket (ts "@t1"))* *==>* *(have-role "@block2" NON-Blicket)*	If, at the same time, two blocks are on the active Detector, and one of the blocks is a blicket then the other block is not a blicket. The *Rule 7* is similar to *Rule 6*, but concludes that the first block is not a blicket.
Rule 8: *(alone-on "@block1" Detector (ts "@t1"))* *(have-state Detector Active (ts "@t1"))* *(on "@block1" Detector (ts "@t2"))* *(on "@block2" Detector (ts "@t2"))* *(have-state Detector Active (ts "@t2"))* *(have-role "@block2" NON-Blicket)* *==>* *(have-role "@block2" NON-Blicket)*	If there are two possible blocks that can activate Detector, and one was observed to activate the Detector alone, and the other one is likely not to be a blicket then the latter is not a blicket.

Model's strategy for training phase

During the training phase, ten blocks are sequentially presented to the model. For example, the evidence presented to the model for the first block is:

(alone-on Block1 Detector (ts "base"))
(have-state Detector Active (ts "base"))

The model uses forward reasoning to evaluate evidence and categorize ten blocks. Only Rules 2 and 3 are used because those rules always provided the best match to the presented evidence. The resulting categorization stored in the model's declarative memory can be as following: *(have-role Block1 Blicket (ts "base"))*. The two rules represent typical instructions human subjects would receive during the task.

Next, the model receives an initial request to categorize A and B blocks. Since the model has no existing categorization of the two blocks in its declarative memory, it has to use analogy-based induction to categorize each block. A backward reasoning with an example assertion *(have-role BlockA "@role" (ts "init"))* invokes Rule 0 from Table 1. Antecedent from Rule 0 triggers retrieval of any category statement belonging to one of ten blocks categorized during the training phase. Because all ten blocks have equal probabilities of retrieval, the probability of block A being categorized as blicket is equal to a prior probability of blickets established during the training phase. For example, if the model retrieves *(have-role Block2 NON-Blicket (ts "base"))* then block A will be also categorized as non blicket: *(have-role BlockA NON-Blicket (ts "init"))*.

Model's strategy for AB-event

The evidence for AB-event is presented to the model as:

(on BlockA Detector (ts "AB"))
(on BlockB Detector (ts "AB"))
(have-state Detector Active (ts "AB"))

The order of the first two statements in the evidence is random. Given such evidence, the model uses a forward reasoning to categorize both A and B blocks during AB-event. Rules 4-7 have equal match to provided evidence. Rules 4 and 5 result in a block being categorized as blicket, while Rules 6 and 7 can result in a negative categorization. Four rules allow the model to guess based on the notion that at least one of the blocks should be a blicket without excluding the probability that the other one may not be a blicket. Because of Rules 6 and 7, positive categorization for one block can result in negative categorization of another block.

After evidence evaluation, the model is queried about A and B. The models reports with categories it has inferred during AB-event. The model may fail to categorize a block if either Rule 6 or 7 is used and the model does not have any fact supporting the last antecedent rule-statement *(have-role "@block1" Blicket (ts "@t1"))*. In such cases the model reports category inferred during the training phase. This analogy-based induction is governed by Rule 1 and allows the model to fall back to prior decision if it is confused by ambiguous evidence such as in AB-event.

Model's strategy for A-event

The evidence for A-event is presented to the model as:

(alone-on BlockA Detector (ts "A"))
(have-state Detector Active (ts "A"))

Given this evidence, the model again has to infer the categories for A and B. Inferring A's category is straightforward since Rule 2 is always the best match to infer A's category given the evidence above. Correspondingly, block A is always categorized as blicket.

Inferring B's category is trickier since above evidence does not provide any information about B. The model uses Rule 8 to infer B's updated category. This rule was introduced to the model based on the effect of *backward blocking*. Backward blocking is observed in a task with two potential causes (A and B). It was found that subjects who observe that A alone can cause the outcome are less likely to accept B as a second cause than subjects that only observed A and B causing the outcome together (Shanks, 1985). Rule 8 allows the model to backward block B if it was previously observed together with A. According to the rule, if at any time B was categorized as non-blicket then that decision will be reinforced given the positive evidence about A.

Model results

The model repeated the experiment 50 times. Proportions of times the model reported A and B as blickets are shown in Fig. 10c. The model's good fit supports the hypothesis that the casual learning in blicket tasks is not simply associative (Griffiths et al., 2011). Furthermore, our model provides a detailed account of underlying cognitive processes happening in human brain. The original Bayesian model by Griffiths et al. lacks such explanatory power. In addition to reflecting a knowledge structure required for the task, rules also govern how the knowledge should be evaluated and updated.

The most intriguing aspect of our model is its ability to simulate Bayesian-like inference despite using an inherently deterministic rule-based inference. Just like the Bayesian model, our model is able to incorporate not only the immediate knowledge, but also prior knowledge that is being constantly updated throughout the task. Such behavior is facilitated by the fact that outcome of new inference is dependent on outcome of the previous inference. Furthermore, there are multiple competing rules that can be used for the same inference, and probabilistic nature of DM's retrieval is the defining factor over which rule is chosen.

Discussion and Conclusion

In this study we have proposed a computational module of human reasoning system called the HRM. We have also described three models of different reasoning tasks. These models tested and validated individual cognitive functionalities of the HRM based on a fit to human data.

The model of spatial relations task shows an in depth view of how rule- and mental model-based reasoning strategies are used together in the same task. It is imperative for success that both strategies complement each other. The core of model-based reasoning is bottom-up reasoning, an ability to derive explicit knowledge from an implicit knowledge. Although fast and efficient, bottom-up reasoning has limitations on the complexity of semantics it can operate. Those limitations make the model-based reasoning prone to mistakes if not corrected by rule-based reasoning. On the other hand, top-down rule-based reasoning is a slow and costly process not feasible for real-time interactive tasks. It has to rely on a model-based reasoning to speed up the reasoning process. When a reasoning pipeline recursively calls itself, it blurs the boundary between rule- and model-based strategies since both of them may be used during the same reasoning process.

The first casual deduction model is a demonstration of how a triple-based knowledge structure can help to explain how complex background knowledge can influence an outcome of even simple deductive reasoning. As such, it is no longer a deductive reasoning, but rather a pragmatic reasoning, a reasoning based on both a given propositional form and its content, previous knowledge (Braine & O'Brien, 1991). It is interesting to see a rise of the pragmatic reasoning in the HRM since it does not incorporate any dedicated controls for it. The very dependency of the HRM's deductive reasoning on ACT-R's declarative mechanisms gives rise to a quite natural pragmatic reasoning. As such, there is a possibility that a pragmatic reasoning is not a different logical process, but a deductive reasoning bound by properties and limitations of our long-term declarative memory.

The model of blicket task further extends the notion of pragmatic reasoning and steps into a territory of Bayesian probabilistic inference. The model's good fit challenges the traditional view of vertical division between deterministic and probabilistic inferences of human reasoning. Instead, the model shows that given an inconsistent nature of human memory and uncertainty of its recall the deterministic inference can become probabilistic.

One of the unexpected outcomes of this study is a seamless unification of similarity- and rule-based reasoning within the HRM. Earlier studies suggested that both rule- and similarity-based processing may emerge from application of a single learning rule (Pothos, 2005; Verguts & Fias, 2009). In the blicket model, Rules 0 and 1 are used for similarity-based reasoning while others are rules defined by the task. Both types of rules are handled by the HRM's reasoning pipelines, and transition from one form of reasoning to another is seamless and on-demand.

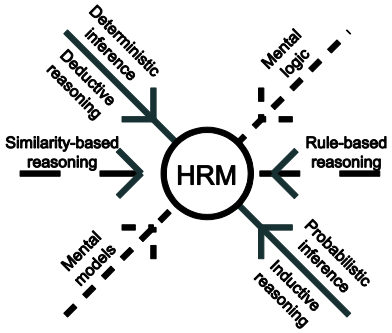


Fig. 11 Current forms of reasoning that were used by three models based on HRM.

The eventual goal of developing Human Reasoning Module is to create a unified theory of human reasoning and a practical tool for simulating it. As such, the HRM was designed to be general and task-independent. It is not constrained to specific formal system of logic. These properties make the HRM potentially suitable for modeling wide variety of tasks. However, the same properties raise concerns whether the module can reliably simulate human behavior in specific reasoning tasks. We tried to mitigate those concerns by modeling three different tasks that address human reasoning from very different perspectives. We are still in the process of elaborating what the unified model of human reasoning should be. However, the HRM is promising to be a step in the right direction. Fig. 11 shows the six forms of reasoning used by the three models. The first dimension unifies two popular theories of mental logic and mental models. The HRM assumes that a mental model is a form of working memory, Visual Short Term Memory, which has the capability to extract basic semantic relations from its content using fast and efficient bottom-up cognitive processes. Then, these semantic relations can be used by mental logic to perform more complex semantic processing. Therefore, the HRM argues that human reasoning is not strictly top-down and can rely on subconscious bottom-up processes to evaluate semantic relations. The second dimension unifies probabilistic inductive reasoning and deterministic deductive reasoning. The HRM suggests that the human general reasoning skill is likely to be inherently probabilistic and inductive due to stochastic nature of knowledge access and retrieval. However, deterministic deductive reasoning is still possible when knowledge-related uncertainty is minimized. Ideally, deductive reasoning is an instance of inductive reasoning with zero uncertainty. Therefore, the amount of uncertainty is the common dimension that unifies inductive and deductive reasoning. Furthermore, a degree of uncertainty may be one of the main factors defining reasoning strategy. Inductive reasoning can be viewed as an instance of probabilistic reasoning with a strong prior toward a particular conclusion. Probabilistic reasoning is inference based on significant past experiences defined by strengths of cause/effect, pre-condition/action, action/post-condition observations. Inference without prior knowledge about the given instance is either reasoning by analogy or simply guessing. In the HRM, reasoning by analogy is still done via rule-based reasoning. This unification of similarity-based and rule-based reasoning is the final dimension depicted in Fig. 11.

Many open questions still remain. One of them is still how inference rules are constructed. For example, verbal instructions given to subjects in blicket task should be somehow translated into set of rules shown in Table 1. On the one hand, it is possible that we have set of general rules that serve as templates and are translated into task specific forms. On the other hand, there might be set of meta-rules similar to schema that govern how inference rules should be constructed based on the perceived information.

The source code and related data for the HRM module and validation models can be downloaded from here: http://www.ai.rug.nl/~n_egii/models/ or http://www.bcogs.net. The current implementation of the HRM is in the prototype phase, and its features may change with future revisions.

References

Anderson, J. R. (2007). *How can human mind occur in the physical universe?* New York: Oxford University Press.

Baddeley, A. (2003). Working memory: Looking back and looking forward. *Nature Reviews Neuroscience*, 4(10), 829-839.

Banks, A. P., & Millward, L. J. (2009). Distributed mental models: Mental models in distributed cognitive systems. *The Journal of Mind and Behavior , 30*, 249-266.

Braine, M. D., & O'Brien, D. P. (1991). A Theory of "If": A Lexical Entry, Reasoning Program, and Pragmatic Principles. *Psychological Review , 98* (2), 182-203.

Byrne, R. M., & Johnson-Laird, P. N. (1989). Spatial Reasoning. *Journal of Memory and Language , 28*, 564-575.

Carreiras, M., & Santamaria, C. (1997). Reasoning About Relations: Spatial and Nonspatial Problems. *Thinking and Reasoning , 3* (3), 191-208.

Chapman, G. B. (1991). Trial order affects cue interaction in contingency judgments. *Journal of Experimental Psychology: Learning, Memory, & Cognition , 17*, 837–854.

Coney, J. (1988). Individual differences and task format in sentence verification. *Current Psychology , 7* (2), 122-135.

Crystal, D. (1997). *The Cambridge Encyclopedia of Language.* Cambridge: Cambridge University Press.

Cummins, D. D. (1995). Naive theories and causal deduction. *Memory & Cognition , 23* (5), 646-658.

Cummins, D. D. (1996a). Dominance hierarchies and the evolution of human reasoning. *Minds and Machines , 6* (4), 463-480.

Cummins, D. D. (1996b). Evidence for the innateness of deontic reasoning. *Mind & Language , 11* (2), 160-190.

Cummins, D. D., Lubart, T., Alksnis, O., & Rist, R. (1991). Conditional reasoning and causation. *Memory & Cognition , 19* (3), 274-282.

Drewitz, U., & Brandenburg, S. (2012). Memory and Contextual Change in Causal Learning. *Proceedings of the 11th International Conference on Cognitive Modeling* (pp. 265-270). Berlin: TU-Berlin Academic Press.

Formisano, E., Linden, D. E., Di Salle, F., Trojano, L., Esposito, F., Sack, A. T., Grossi, D., Zanella F. E., & Goebel, R. (2002). Tracking the mind's image in the brain I: time-resolved fMRI during visuospatial mental imagery. *Neuron*, 35 (1), 185-194.

Gentner, D., Holyoak, K. J., & Kokinov, B. N. (Eds.). (2001). *The analogical mind: Perspectives from cognitive science.* The MIT Press.

Griffiths, T. L., Sobel, D. M., Tenenbaum, J. B., & Gopnik, A. (2011). Bayes and Blickets: Effects of Knowledge on Causal Induction in Children and Adults. *Cognitive Science , 35*, 1407–1455.

Gust, H., Krumnack, U., Kühnberger, K. U., & Schwering, A. (2008). Analogical reasoning: A core of cognition. *Zeitschrift für Künstliche Intelligenz (KI), Themenheft KI und Kognition , 1*, 8-12.

Henderson, J. M., & Hollingworth, A. (2003). Eye movements and visual memory: Detecting changes to saccade targets in scenes. *Perception & Psychophysics*, 65(1), 58-71.

Jeffrey, R. (1981). *Formal logic: Its scope and limits (2nd ed.).* New York, NY: McGraw-Hill.

Jiang, Y., Olson, I. R., & Chun, M. M. (2000). Organization of visual short-term memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 26(3), 683-702.

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

Johnson-Laird, P. N. (2004). The history of mental models. *Psychology of reasoning: Theoretical and historical perspectives* , 179-212.

Kahneman, D. (2003). A perspective on judgment and choice: Mapping bounded rationality. *American Psychologist , 58* (9), 697–720.

Logie, R. H., Zucco, G. M., & Baddeley, A. D. (1990). Interference with visual short-term memory. *ACTA Psychologica , 75* (1), 55-74.

Lum, J. A., Conti-Ramsden, G., Page, D., & Ullman, M. T. (2012). Working, declarative and procedural memory in specific language impairment. *Cortex*, 48(9), 1138-1154.

Miller, R. R., & Matute, H. (1996). Biological significance in forward and backward blocking: Resolution of a discrepancy between animal conditioning and human causal judgment. *Journal of Experimental Psychology: General , 125*, 370–386.

Nyamsuren, E., & Taatgen, N. A. (2013). Pre-attentive and Attentive Vision Module. *Cognitive Systems Research , 24*, 62-71.

Phillips, W. A. (1983). Short-Term Visual Memory. *Royal Society of London Philosophical Transactions Series B*, 302, 295-308.

Pothos, E. (2005). The rules versus similarity distinction. *Behavioral and Brain Sciences , 28*, 1–49.

Rajasekar, A., Lobo, J., & Minker, J. (1989). Weak generalized closed world assumption. *Journal of automated reasoning , 5* (3), 293-307.

Rensink, R. A. (2007). The modeling and control of visual perception. In W. D. Gray, *Integrated Models of Cognitive Systems* (pp. 132-148). New York: Oxford.

Rensink, R. A. (2000a). The dynamic representation of scenes. *Visual cognition*, 7 (1-3), 17-42.

Rensink, R. A. (2000b). Seeing, sensing, and scrutinizing. *Vision research*, 40 (10), 1469-1487.

Rips, L. J. (1983). Cognitive processes in propositional reasoning. *Psychological Review , 90* (1), 38-71.

Roberts, M. J. (1993). Human Reasoning: Deduction Rules or Mental Models, or Both? *The Quarterly Journal of Experimental Psychology , 46A* (4), 569-589.

Salvucci, D. D., & Taatgen, N. A. (2011). *The multitasking mind*. New York: Oxford University Press.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychological Review*, 1, 101–130.

Schooler, L. J., & Hertwig, R. (2005). How Forgetting Aids Heuristic Inference. *Psychological Review , 112* (3), 610-628.

Sobel, D. M., Tenenbaum, J. B., & Gopnik, A. (2004). Children's casual inferences from direct evidence: Backward blocking and Bayesian reasoning in preschoolers. *Cognitive Science , 28*, 303-333.

Shanks, D. R. (1985). Forward and backward blocking in human contingency judgment. *Quarterly Journal of Experimental Psychology , 37B*, 1–21.

Taatgen, N. A., Van Rijn, H., & Anderson, J. R. (2007). An integrated theory of prospective time interval estimation: The role of cognition, attention, and learning. *Psychological Review , 114* (3), 577-598.

Thompson, V. A. (1996). Reasoning from false premises: The role of soundness in making logical deductions. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale , 50* (3), 315-319.

Verguts, T., & Fias, W. (2009). Similarity and Rules United: Similarity- and Rule-based Processing in a Single Neural Network. *Cognitive Science , 33*, 243-259.

Winston, P. H. (1980). Learning and reasoning by analogy. *Communications of the ACM , 23* (12), 689-703.

Wintermute, S. (2012). Imagery in cognitive architecture: Representation and control at multiple levels of abstraction. *Cognitive Systems Research , 19-20*, 1-29.

Xu, Y., & Chun, M. M. (2005). Dissociable neural mechanisms supporting visual short-term memory for objects. *Nature*, 440(7080), 91-95.

Abbreviations

HRM - Human Reasoning Module
DM - Decalarative Memory
VSTM - Visual Short-Term Memory
MP - Modus Ponens
MT - Modus Tollens
DA - Denying the Antecedent
AC - Affirming the Consequent