A cognitive model of switching between reflective and reactive decision making in the Wason task

Othalia Larue (othalia.larue@wright.edu), Alexander Hough (hough.15@wright.edu), Ion Juvina (ion.juvina@wright.edu)

Department of Psychology, Wright State University, ASTECCA laboratory
Dayton, OH 45402 USA

Abstract

Using the Wason card selection task, we model how humans decide to engage in further deliberation after generating an intuitive response. Central to our model is the Feeling of Rightness which is the fluency with which one makes a decision, and how good one feels about it. The model is implemented in a cognitive architecture, ACT-R. A core affect mechanism and the Feeling of Rightness component are used to drive the decision. A training procedure was used to simulate individual differences in heuristic and analytical behavior. Different degrees of reinforcement were then used to allow the model to learn logical skills, and acquire differences in FOR, which determined the degree of further deliberation. By relying on ACT-R's different memory components and the core affect mechanism we were able to reproduce the variation between analytical and reactive behavior.

Keywords: cognitive decoupling, cognitive architecture, Wason task, deliberation, Feeling of Rightness, heuristic.

Introduction

The abstract Wason card selection task (Wason, 1968) is a reasoning task that involves two types of answers, which reflect different types of processing. In this context, fast (heuristic) processing results in an incorrect answer, while slower and more analytical processing is more likely to result in a correct answer.

In previous work (Larue et al, 2013), we explored this phenomenon by implementing a specific theory (the tripartite framework, Stanovich, 2009) in a computational model. Stanovich's tripartite framework provides an explanation of how reflective and reactive human behavior emerges from the interaction of three distinct cognitive levels (or minds): autonomous, algorithmic, and reflective. We showed how the three levels contribute to the different types of answers in the Wason card selection task. However, we did not explain what triggered each of the three levels to become involved in the decision process. Here, we will do so using the same task.

We will briefly review the extant literature about human performance on the Wason card selection task, what functions are involved, and how they are triggered to improve decision-making. For the latter we will specifically go over the notion of Feeling of Rightness (FOR; Thompson, Evans, & Campbell, 2013), which describes how good we feel about the answer we produced and helps us decide if we want to engage in further deliberation. Then, we will present a computational cognitive model that attempts to explain how humans decide to engage in deliberation in the Wason task by integrating both Stanovich's tripartite framework (Stanovich, 2009) and a complementary notion, the FOR (Thompson, Evans, & Campbell, 2013) into the ACT-R cognitive architecture (Anderson, 2007). Finally, we will present qualitative results of our computational model on the Wason task.

Related work

In the abstract Wason card selection task (Wason, 1968), four cards with a letter on one side and a number on the other side are presented. The visible sides are "A", "D", "3" and "7". The following rule is also presented: "If there is an A on one side of the card, then there is a 3 on the other side of the card". Then, the participants are asked: "Which card(s) should you turn over in order to test the truth of the rule ?"

The rules of inference that should be used for deliberation are: *modus ponens* (if P then Q, P: therefore, Q) and *modus tollens* (If P then Q. Not Q: therefore, not P). However, these rules do not seem to be systematically applied by participants as evidenced by the variety of answers: ("A" and "3"), ("A"), ("A" and "7"). The most frequent answers are "A" and ("A" and "3").

Different theories have attempted to explain these results, such as the mental model theory (Johnson-Laird, 1980) and the heuristic analytic theory (Evans, 2006), but have not provided a complete explanation for the variety of observed behaviors. Stanovich's tripartite framework (Stanovich, 2009) is a dual process theory (i.e., a theory distinguishing two processing styles: type 1, that is fast and automatic, and type 2 that is slow and deliberative) that fills this gap. It provides an explanation of how both reflective (i.e., type 2 processing characterized by sequential processing) and adaptive (i.e., type 1 processing characterized by reactivity) behaviors emerge from the interaction of three distinct levels: (1) an autonomous mind responsible for fast context-sensitive behaviors (type 1), (2) an algorithmic mind responsible for control (type 2), and (3) a reflective mind responsible for deliberative processing (type 2). Engaging in deliberation involves a process called cognitive decoupling. It is launched by the reflective level and carried out at the algorithmic level. Cognitive decoupling is an off-line simulation: modules fitting its description can be found in the psychology research (Leslie, 1987) and in cognitive architectures (Larue, Poirier, & Nkambou, 2013; Sun, 2015).

The tripartite theory explains three different behaviors observed in the Wason task: (1) a complete reliance on the autonomous mind leads to the frequently observed error (i.e., selecting two cards ("A" and "3"), one with the correct answer ("A") and one with the incorrect answer that are found in the wording of the task), (2) a partial deliberation leads to selecting only one of the correct answers (*modus ponens*), the one containing the antecedent, and (3) a complete deliberation allows for the activation of the counter-information rule (*modus tollens*) which leads to a response that is less frequently observed ("A" and "7").

The initial heuristic judgment ("A" and "3") is cued and automatic (matching to the wording). Why is this judgment accepted without more analysis? According to Thompson et al. (2013), we are tempted to accept this initial response as our final one because it is compelling; it has the feeling of rightness (FOR). It is a metacognitive experience that determines if one proceeds to further analysis; it is essentially the perceived confidence that an initial answer is correct.

In previous work (Juvina, Larue, & Hough, 2017), we showed that an affective modulation of memory helps to make better decisions in complex tasks that exceed human limited memory capacities. In this paper, we will use the same mechanism to explain people's judgment of their initial answer based on its fluency. We hypothesized that the fluency of the initial answer creates the gut feeling that leads to being confident in this first answer. In our model, we assume that FOR is the component deciding if one has more of an analytical profile or a reactive profile. It reflects how fast it was to produce a first non-analytical "gut" answer and influences the extent of subsequent processing.

Model

The model[1] is implemented in ACT-R (Adaptive Control of Thought – Rational; Anderson, 2007), a unified theory of human cognition used to develop computational models of various cognitive tasks. ACT-R is composed of various modules: goal, imaginal, visual, aural, manual, vocal, and two memory modules: declarative memory (facts) and procedural memory (know-how). It is a hybrid cognitive architecture. Symbolic components are combined with subsymbolic components: the retrieval of a fact from declarative memory depends on subsymbolic retrieval equations (pondering the context and history of retrieval of the fact), and, the selection of a rule depends on a subsymbolic quantity called utility, which computes costs and benefits associated with the rule. Learning processes act at both subsymbolic and symbolic levels. Additionally, we use the core affect mechanism (Juvina, Larue & Hough, 2017): the memory elements (chunks) are reinforced through reward patterns that occur within the environment.

Model elements

Memory representation of the "mental models" Chunks are elements of the declarative memory in ACT-R. As shown in Figure 1, the representation of a mental model includes two rule structure chunks and four card chunks. The first line of Figure 1 means that the chunk with the name rule1 is a chunk of the type rule with a slot "if" that has the value "p", a slot "then" with the value "q", the slot "notif" with the value 0 (no negation), the slot "notthen" with a value 0 (no negation). This chunk represents the logical structure of modus ponens and will be used by the procedure modus ponens. The third line of Figure 1 means that the chunk with the name card1 is a chunk of the type card with a slot "letter" that has the value "A", a slot "meaning" with an empty value (""), the slot "not" with a value 0 (no negation), no number (nil), and the slot "deco" with a value 0 (to signify that it is not part of the decoupling inner simulation). As the model processes information, the card1, card2, card3, card4 chunks will be modified.

```
(rule1 isa rule if "p" then "q" notif 0 notthen 0)
(rule2 isa rule if "q" then "p" notif 1 notthen 1)

(card1 isa card letter "A" meaning "" not 0 number nil deco 0)
(card2 isa card letter "D" meaning "" not 0 number nil deco 0)
(card3 isa card number "3" meaning "" not 0 letter nil deco 0)
(card4 isa card number "7" meaning "" not 0 letter nil deco 0)
```

Figure 1: Memory representations in ACT-R.

When the decoupling starts, the model's inner simulation is populated by the card elements that could be retrieved at that time ('card1', 'card2', 'card3', 'card4' in Figure 1). To mark the element in declarative memory as part of the decoupling operation, its "deco" slot is switched to 1. *Modus ponens* and *modus tollens* rules are only applied on an element with a "deco" slot of 1. When decoupling ends, the "deco" slot is switched to 0.

When the *modus ponens* rule (which uses the *modus ponens* structure chunk) is applied, it modifies the "meaning" slot as "p" and 'card2' "meaning" slot as "q". The *modus tollens* rule will lead to the modification of card4 meaning slot as "notQ".

The resulting mental representation in declarative memory is the mental model of the task at a given time. The merging with declarative memory that occurs at the end of the cognitive decoupling reinforces the chosen mental model in memory.

How FOR is modeled? Cards are represented as chunks in the declarative memory (see Figure 1). Rethinking times, answer changes, and fluency are a function of the value of FOR. FOR depends on the time required to achieve the retrieval of "A" and "3" through an initial priming rule: the higher the FOR, the less you engage in deliberation. The FOR-inverse variable is a measure of how fast the model retrieved those two first cards/answers, as measured by the temporal module. FOR-inverse acts as a gateway for further processing when it is above a threshold. Thus, the time required to complete the initial retrievals is assigned to the FOR-inverse variable; when the value of this variable is below threshold, the model goes with the initial answer (i.e., type 1 processing); when FOR-inverse is above threshold, cognitive decoupling is launched and the model engages in further processing (i.e., type 2 processing). Every reward sent in the system is a function of FOR. including the reward sent

[1] Model code available: thttp://psych-scholar.wright.edu/astecca/

at the end of decoupling (see analytical thinking column of Figure 2): it will determine the extent to which a decoupling result (i.e., wrong, partial, and complete) is taken into account during answer selection in our model.

Answer selection The model produces an answer when the valuation of a representation is above a certain threshold. The valuation and arousal values, which help to define the core affect, are sub-symbolic quantities added to the current sub-symbolic equations of ACT-R. It allows for a specification of how emotion meshes with other cognitive processes. A detailed explanation of the core affect model is out of the scope of this paper and can be found in Juvina et al. (2017).

Core affect augments ACT-R's general equation.
The augmented core-affect activation equation is:

$$A_i = B_i + S_i + P_i + V_i + Ar_i + \varepsilon_i \qquad (1)$$

As part of the general activation equation, A_i is the activation of the chunk i, Bi the base-level term reflecting its recency and frequency, Si is the spreading term (context effect) and Pi is the partial matching term. This general equation is augmented with the valuation which is learned from rewards. Vi is the valuation term and reflects the rewards received by the model after referencing chunk i. Ari is the arousal term which reflects the importance of chunk i and is computed as the absolute magnitude of the valuation term.

$$V_i(\text{n}) = V_i(\text{n-1}) + \alpha v[R_i(\text{n}) - V_i(\text{n-1})] \qquad (2)$$

Vi(n) is the valuation of chunk i after its nth update. Vi(n-1) is the valuation of chunk i prior to its nth update. αv is the learning rate for valuations. Ri(n) is the effective reward value received by chunk i before its nth valuation update.

When a reward is triggered, valuations are updated. All chunks that have been referenced within a time window are updated. The time window is controlled by the parameter :vtw (valuation time window).

Rewards are a function of the initial FOR (negative factor in the case of negative reward). It will affect the answer selection ("yes" or "no" answers will be produced according to how the model "feels" about the answer). A reward is sent after the initial priming so that the priming of the first answer is salient. When the model engages in cognitive decoupling, a negative reward attenuates the priming of that first answer (but does not erase it). Rewards also reinforce the chunks of the cards retrieved from memory during *modus ponens* and *modus tollens*.

Rules involved in cognitive decoupling

These rules will be used if the FOR-inverse is above a certain threshold (Parameters of the model can be seen in Table 1). Successive retrievals of chunks of the type "card" populate the representations used by cognitive decoupling. Their "deco" slot is modified from 0 to 1. Rules are applied on chunks from the declarative memory which "deco" slot has been marked as 1.

Modus ponens Before *modus ponens* is applied, a procedure inhibits the previous answer (from priming) by triggering a reward, which will decrease its valuation. This procedure is applied during the cognitive decoupling. Through an exploratory procedure, "A" is identified as P (the meaning slot of the chunk with the letter slot with value "A" is filled with "p"). *Modus ponens* only retrieves the "A", which will be reinforced by the retrieval and the positive reward sent after the end of the decoupling, leading the model's instance to answer "A". If the *modus ponens* rule structure can't be retrieved, *modus ponens* can't be applied, resulting in a cognitive decoupling with a wrong answer ("A" and "3"). The chunks with the "A" and "3" slot values are inhibited at the beginning of decoupling; however, the positive reward sent at the end of the cognitive decoupling will still reinforce the "A" and "3" who were the last element retrieved before decoupling (as they are still in the valuation window).

Modus tollens This rule is applied during the cognitive decoupling after the *modus ponens* rule. Before this rule can be applied, an exploratory rule is applied; "7" is identified as notQ (the meaning slot of the chunk with the number slot with value "7" is filled with "notQ"). If declarative memory retrieval requests fail during the exploratory rule, *modus tollens* can't be applied. As the chunk with letter slot "A" and number slot "7" are retrieved by the exploratory rule, they are reinforced by the retrieval and the positive reward sent after the end of the decoupling, leading to answer "A" and "7".

Temporal module ACT-R has a temporal cognition module (Taatgen, Rijn, & Anderson, 2007), allowing it to account for how one estimates time that passed. If the model has access to the temporal module, it can estimate how much time has passed. In the "fast" condition, this value is factored in when the model evaluates its current FOR. As more time passes, the FOR-inverse variable increases and FOR decreases. In the fast condition, to account for time pressure, a factored value of this time is added to the FOR-inverse-threshold that will lead to a deeper analysis of the problem to be done. This is what leads instances to stop "thinking" more often in this condition.

Table 1. Training and model parameters

Parameter	Value
Baseline heuristic behavior training iterations	3
Additional training iterations	1
Proportion of formal logic in additional training	20%
Proportion of heuristic in additional training	80%
:rt	-2.4
For-inverse-threshold	15
:vw	0.8

Training procedure

A training procedure allowed us to create individual differences in our simulation: differences in personal logic and impersonal logic.

Personal logic: creation of the "if" heuristic First, as a baseline the heuristic behavior (also referred to as personal logic) in each individual instance of the model is reinforced. Consistent with Evans (1998) and Stanovich (2009), the "if" heuristic directs the participant's attention to the "if" statement and leads to an initial bias in the building of a mental representation of a task (where both elements are identified as true). In order to represent this heuristic, we placed a structural representation of the "if" statement in declarative memory, which is retrieved for application in a rule (matches-if-form) and applied by another rule (matching).

To simulate the frequent use of the "if" heuristic we make the model fire the "if-form" rule repeatedly leading to an increased utility of this rule. Additionally, we used production compilation (Taatgen and Lee, 2003); a mechanism to model skill acquisition in ACT-R. Because the two rules retrieving the two elements of the "if" statement ("A" and "3") fire in sequence and often, the production compilation mechanism creates a new rule out of those two rules.

Furthermore, through the production compilation mechanism, if one of the rules retrieved an element from memory (here the "if" structural statement), that element is directly integrated into the new rule. Therefore, according to the degree to which our model is trained, it is more sensitive (answers faster) to "if" statements. The formed new rule will initially have a low utility but through training will increase its utility and be selected more often (and differently in each instance of the model) representing how much the rule is expected to contribute to the model's goal.

Impersonal logic: advanced logical aptitudes After initial training, an additional training is administered, which simulates individual differences. This training reflects the lesser proportion of people who received a strong formal logic training (impersonal logic). This is accomplished by training 20% of the simulated population with formal logic, while the other 80% received additional personal logic training.

To represent that participants are naturally more exposed and trained to apply the *modus ponens* (compared to the *modus tollens*), the training procedure involved more retrieval of the *modus ponens* representation in declarative memory than for the *modus tollens*; leading to their reinforcement in declarative memory. Differences in chunk retrieval create varying activations between the chunks and modulates if and how fast they are retrieved. Failure in the retrieval of the *modus tollens* representation is more frequent, leading to its less frequent application.

Additionally, the rules involved in the analytical processing in the model were activated, which increased their utility in procedural memory. Since the production compilation was active for the whole training procedure, in training where the *modus tollens* rule was correctly retrieved, the ending rule of the *modus ponens* is combined with the beginning of the application of the *modus tollens* guaranteeing the application of *modus tollens* after *modus ponens*. Without rule compilation, the transition between *modus ponens* and *modus tollens* would not be automatic. In those model instances, the application of *modus tollens* after *modus ponens* was automatic since the production compilation mechanism has eliminated one retrieval operation. The frequent retrieval of the *modus tollens* representation, followed eventually by its application and the compilation of rules, is how some simulated participants developed advanced logical skills.

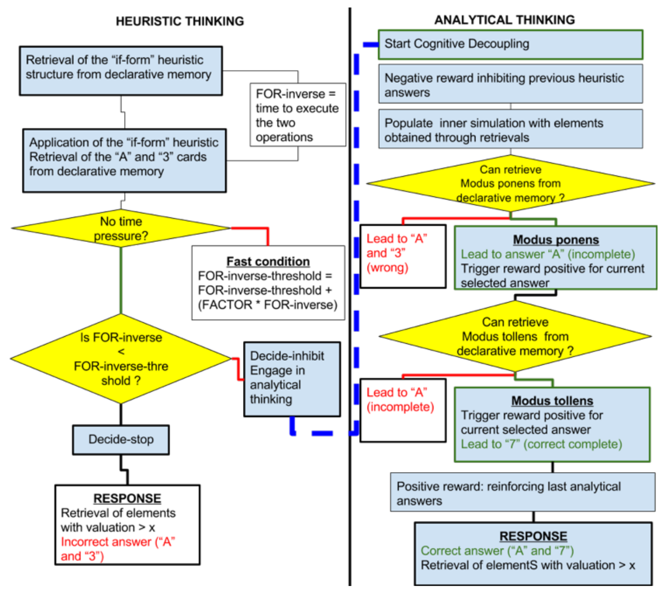

Figure 2: Model dynamics

Model dynamics

We describe here the processing in a model instance where the model starts out thinking heuristically and moves towards analytical thinking, depending on the FOR. (this is further explained in the following sections). In the declarative memory of the model, the instruction "if 'A' then '3'" is present. A first rule will retrieve the "if-form" from declarative memory (shown in the heuristic thinking column of Figure 2). Another rule will match the retrieved if-form to the instruction. This leads to the retrieval of card "A" and "3" representations. The retrieval of the two cards leads to their reinforcement in declarative memory in ACT-R. At this point, time pressure and the value of FOR-inverse variable are calculated. In the fast condition, a factored value of this time is added to the FOR-inverse-threshold to account for time pressure. Depending on time pressure and the value of FOR-inverse, two rules can then fire: decide-inhibit or decide-stop. If the variable is under the FOR-inverse-threshold (the answer was fluent) then the model's instance will stop and provide an answer (i.e., decide-stop). Otherwise, the instance will engage in analytical thinking to further analyze the problem before it answers (shown in the right column, transition between the left column and the right column is shown by the dashed blue line in Figure 2). Cognitive decoupling starts with a negative reward inhibiting the previous answers activated at the heuristic phase (retrieved cards "A" and "3"). Then a retrieval of 'card' chunk types (see Figure 1) provides the elements to populate the inner-simulation of the problem. Note that, if the elements in declarative memory are under a certain retrieval threshold they won't be included in the inner-simulation. This is the focal bias of the model; if the elements that would lead to an activation of *modus tollens* (here the "7" card) are not in the inner simulation then this rule can't be applied. When elements (chunks) are retrieved for the inner simulation, their deco (decoupling) slot is changed from 0 to 1.

The *modus ponens* rule is the first one that the instance will try to apply by retrieving it from memory, if this retrieval fails, the decoupling will end and the first answer will be left unchanged (i.e., "A" and "3"). If the retrieval succeeds, it leads to the incomplete answer of "A", which receives a positive reward. A rule then tries to match the structure of *modus ponens* to the elements of the inner simulation. If the rule does not match, it won't fire and *modus ponens* won't be applied. The *modus tollens* rule is the second one to be applied. The *modus tollens* structure is retrieved from memory, but if this retrieval fails, the only answer produced comes from the application of *modus ponens* (The incomplete answer of "A"). If the retrieval succeeds, a rule will try to match the structure of *modus tollens* to the elements of the inner simulation. Again, if the rule does not match, it won't fire and *modus ponens* won't be applied. At the end of cognitive decoupling, the elements retrieved by the both the *modus ponens* and *modus tollens* rules are placed sequentially in the imaginal module and their deco (decoupling) slot is changed back from 1 to 0, and reentered in declarative memory (reentrance in declarative memory leads to reinforcement). Additionally, a positive reward is produced, reinforcing the most recently retrieved elements. The final answer of the system is produced based on the valuation of each card: each card is retrieved and if it is under a valuation threshold the instance answers "no" for this card, otherwise it answers "yes". The operation is repeated for each card.

Results

Experimental procedure

The experimental procedure is similar to the one described in Thompson et al. (2013). We qualitatively compared the results of our computational models to the results obtained in Thompson et al. (2013). The simulation consisted of 100 trials (100 model "participants" with 1 trial for each model "participant"). For each trial, the model was run twice: once in the fast condition and once in the complete condition (without time pressure). At the beginning of each trial, the training procedure was implemented to create individual differences between subjects. Each subject received a strong training in what we called "personal logic" (making deduction from the wording of the task according to linguistic cues). Subjects received varying training (2/10 ratio) in impersonal logic (logic rules *modus ponens* and *modus tollens*). The model was reset between each trial, after completion of both the fast and complete conditions, to keep the same individual profile of that model in both conditions and transfer resulting activations from the previous fast condition into the complete condition.

Response times and answer changes

The FOR shown on the graph was computed by inversing FOR-inverse and dividing its value between the maximum FOR-inverse and the minimum FOR-inverse by 8 (assigning FOR from 1 to 8 depending on which interval the FOR-inverse belonged to). Human participants in the study of Thompson et al. (2013) provided a Likert scale evaluation of their FOR. In our model, we use the internal value of the model's FOR (computed with the time it takes the model to finish applying the first "matching" heuristic).

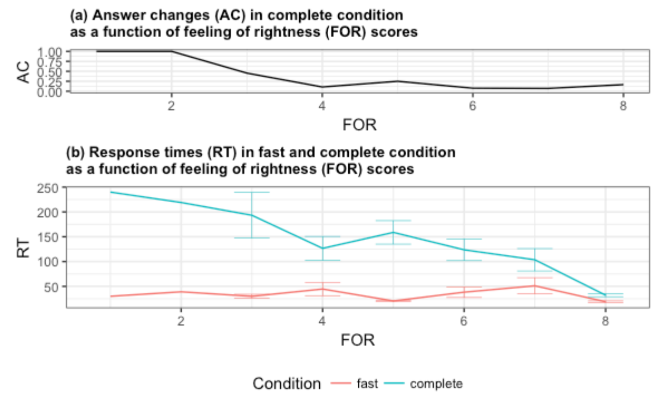

Figure 3: (a) Answer changes (AC) as a function of FOR. (b) Response Times (RT) as a function of FOR

Figure 3 presents the Response Times and answer changes according to the FOR grouped in seven intervals. Note that, while Thompson's study is similar to ours, it included a variety of wordings of the task we did not reproduce here. We were able to produce results qualitatively similar to the original study (see Figure 4 for original study's results). The second response times in the slow condition and the complete condition correlate with the answer changes indicating that answer changes result when further analytical processing occurs. There also is less individual variability in the response times as we reach the extreme values of the FOR (higher and lower). There is a significant difference in response times between the two conditions ($t(137.54)=7.32$, $p < 0.001$). Response time for higher FOR in the complete condition are also closer to values of the fast condition; indicating that the model adequately reproduced participants tendency to not engage in analytical processing when they felt right about their first answer.

The standard error observed in Figure 3.b comes from the variation between individuals we introduced to training. There is no significant difference between the two conditions when FOR is higher than six. The instances where the model was trained more in personal logic are the ones with the highest FOR as the retrieval of the "if-form" structure and the application of the if-form rule (after production compilation) goes faster. Even when offered a chance to reconsider their answer (with a For-inverse-threshold that is not affected by the time pressure factor), the model's instances keep their first answer. Training in personal logic was provided to a higher proportion of simulated participants (models) to account for "real-life" experience. Therefore, there are more instances with a higher FOR (e.g. more heuristic thinking). Furthermore, fewer instances received a higher training in personal logic and even if they received the training, they still have more chances to not complete a cognitive decoupling (inner simulation populated with an incomplete representation of the problem, *modus ponens* structure not retrieved, *modus tollens* structure not retrieved, *modus ponens* or tollens can't be applied because they don't match with any element from the incompletely populated inner simulation). The difference between the two conditions is significant from FOR values less than 6 (t (107.36)=-7.20, $p < 0.001$).

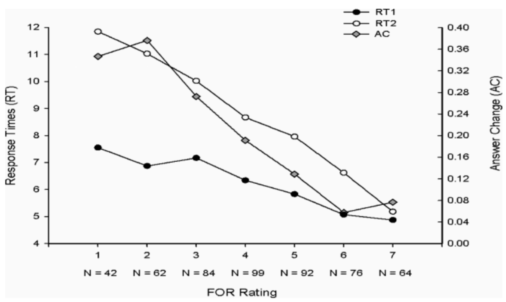

Figure 4. RT1 (first response time), RT2 (second response time), AC (Answer change) from Thompson et al. (2013)

We were able to reproduce the variation tendency of response times. However, our response times were not at the same time scale as Thompson's study (see Figure 4): our model did not parse the instructions like a human participant would. The purpose of this paper was not to reproduce the time participants spent reading the task material and processing the visual aspects of task (looking at the card). Here, we were mainly concerned with the processing that occurred after the instructions were read and participants processed the visual presentation of the cards.

Different types of reasoning

Figure 2 presents the different types of reasoning the system engages in to produce the "A" and "3" (incorrect answer), "A" (correct but incomplete answer – *modus ponens*), "A" and "7" (complete correct answer – *modus ponens* + *modus tollens*). Figure 2 shows the model can reproduce those three types of reasoning that correspond to different level of involvements of analytical processing. "A" and "3" is produced when no cognitive decoupling happens. *Modus ponens* and *modus tollens* answers happen with cognitive decoupling. If only *modus ponens* is applied however, it means that the cognitive decoupling was incomplete. The *modus ponens* and *modus tollens* rule structures are present in the declarative memory of the system, depending on the activation noise set in the system. An instance of the model might fail to retrieve a rule while another instance of the model won't. This results in individual differences in reasoning for each instance of the model.

Conclusion

In humans, the tendency to engage in analytical thinking and cognitive decoupling varies individually, and in this paper, we focused on reproducing the mechanism that determines if one will or will not engage in further analytical processing. In our model, we assume that FOR is the component necessary to decide if one will have more of an analytical profile or a reactive profile. The FOR is similar to the fluency of the model, how fast it was able to produce a first non-analytical "gut" answer, which influenced the extent of subsequent processing.

While we were able to qualitatively reproduce the same type of behavior as the human participants from Thompson et al. (2013) in our model, we left elements out of the current model that would have allowed fitting the data quantitatively (i.e. instructions reading). We could also refine the model and specifically the training procedure to simulate other versions of the tasks (drinking-age problem, negated version, etc.). Finally, the Wason task is limited; we would like to extend this framework to more complex real-life tasks where both automatic and reflective behaviors are required.

References

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

Evans, J. S. B. (2006). The heuristic-analytic theory of reasoning: Extension and evaluation. Psychonomic Bulletin & Review, 13(3), 378-395.

Johnson-Laird, P. N. (1980). Mental models in cognitive science. Cognitive science, 4(1), 71-115.

Juvina, I., Larue, O., & Hough, A. (2017). Modeling valuation and core affect in a cognitive architecture: The impact of valence and arousal on memory and decision-making. Cognitive Systems Research.

Larue, O., Poirier, P., & Nkambou, R. (2013). Hypothetical-thinking based on cognitive decoupling and thinking dispositions in a dual cognitive agent. Biologically Inspired Cognitive Architectures, 6, 67-75.

Leslie, A. M. (1987). Pretense and representation: The origins of "theory of mind.". Psychological rev, 94(4), 412.

Sun, R. (2015). Interpreting psychological notions: A dual-process computational theory. Journal of Applied Research in Memory and Cognition, 4(3), 191-196.

Taatgen, N. A., & Lee, F. J. (2003). Production compilation: A simple mechanism to model complex skill acquisition. Human Factors, 45(1), 61-76.

Taatgen, N. A., Rijn, H. van, & Anderson, J. (2007). An integrated theory of prospective time interval estimation. Psychological Review, 114(3), 577–598.

Thompson, V. A., Evans, J. S. B., & Campbell, J. I. (2013). Matching bias on the selection task: It's fast and feels good. *Thinking & Reasoning*, *19*(3-4), 431-452.

Wason, P. C. (1968). Reasoning about a rule. The Quarterly journal of experimental psychology, 20(3), 273-281.