But does it really do that? Using formal analysis to ensure desirable ACT-R model behaviour.

Vincent Langenfeld (langenfv@tf.uni-freiburg.de)
Department of Computer Science, Albert-Ludwigs-Universität Freiburg

Bernd Westphal (westphal@tf.uni-freiburg.de)
Department of Computer Science, Albert-Ludwigs-Universität Freiburg

Rebecca Albrecht (rebecca.albrecht@unibas.ch)
Department of Psychology, University of Basel

Andreas Podelski (podelski@tf.uni-freiburg.de)
Department of Computer Science, Albert-Ludwigs-Universität Freiburg

Abstract

Cognitive modelling uses computer models to investigate psychological theories. To conclude from executions of a cognitive model to the theory, the model needs to be a correct implementation of the theory since a defective cognitive model may yield wrong statistical figures. We consider three common reasons for a model to be incorrect wrt. a theory: situations which unintentionally do not enable any production rule, rules which erroneously construct undesired declarative knowledge, and wrongly chosen architecture parameters. Defects of these kinds are hard to detect since repeated execution and observation of the model does not guarantee to uncover these defects.

In this work, we give formal definitions of the three kinds of defects in terms of an existing abstract formal semantics of the hybrid architecture ACT-R. We demonstrate the application of formal analysis techniques to ACT-R models to reliably detect the considered defects and to thereby increase the confidence that the model behaves according to the psychological theory.

Keywords: ACT-R; Formal Methods; Model Analysis; SMT; Model Checking

Introduction

In cognitive modelling, computer models are used to describe human cognitive processes wrt. psychological assumptions. Unified theories of cognition and their implementations (called cognitive architectures) provide means for cognitive modelling. A widely used cognitive architecture is ACT-R (Anderson, 1983, 2007).

One goal of cognitive modelling in ACT-R is the validation of psychological theories, i.e., a hypothesis on how a given task is principally solved by humans. The psychological theory can be validated by constructing an ACT-R model which (a) *correctly* implements the psychological theory and (b) predicts experimental data with sufficient precision. Practically, figures like average error rates or response times are derived from several executions of the ACT-R model and compared to statistical data collected in experiments. A crucial aspect of this approach is Aspect (a): the correct implementation of the psychological theory. The simulation of an *incorrect* implementation of a *valid* psychological theory may yield statistical figures which do not resemble experimental data (false negative). For an *invalid* theory, simulation may yield resembling figures due to the incorrectness of the implementation (false positive). The psychological theory can only be rejected if the implementation is correct and the figures from the simulation do not resemble the experimental data (true negative). In the latter case, the reason of the invalidity needs to be identified within the psychological theory.

ACT-R is a so-called hybrid architecture with a symbolic and a subsymbolic layer. Major parts of the symbolic layer are declarative knowledge (chunks) and procedural knowledge (production rules). The production rules of ACT-R syntactically consist of a precondition part (which is matched against the current cognitive state) and an action part (which, e.g., requests modules). The interface between the symbolic and the subsymbolic layer in ACT-R is given by so-called modules. Modules are requested by production rules to process declarative information and make them accessible through associated buffers. The subsymbolic layer is defined by the behaviour of modules, i.e. the responses of modules for given requests. For some modules, these responses depend on numerical parameters, e.g. the decay rate for the implementation of base-level learning as part of the declarative module.

A *defect* in general denotes any kind of programming errors in production rules, like simple typing errors, or forgotten conditions or requests, or unnecessary conditions or requests. In this work, we consider the following three particular defects. Firstly, we distinguish *precondition* and *action defects* following the syntactical structure of ACT-R production rules, Precondition defects may have the effect that a deadlock occurs. A deadlock is a cognitive state where no production rule is able to fire while the end of the modelled behaviour has not been reached. Since human participants do not "get stuck" during typical experiments, a deadlock may cause a mismatch between human data and figures from model simulation. Action defects may cause cognitive states which are not reachable according to the psychological theory. For these cognitive states, unintended production rules may fire and thereby cause mismatches between human data and figures from model simulation. Thirdly, we consider inappropriate choices of global model parameters (like decay rate) to be model defects. They may unintentionally affect figures from model simulation. Today, the common way to examine an ACT-R model for the presence of such (clearly

undesired) defects is tedious repeated execution and observation of the model. This approach does not give any guarantees for finding an existing defect due to the non-deterministic nature of the subsymbolic layer.

In (Albrecht & Westphal, 2014b), we presented a formal semantics for ACT-R which enables us to develop automatic and (semi-)complete procedures for formal defect analyses of ACT-R models. A first step into this direction consists of the line of work of (Gall & Frühwirth, 2017), which presents analyses for confluence in cognitive models based on their adaptation (Gall & Frühwirth, 2014) of our formal semantics.

In the following we present new formal definitions of the model defects discussed above and their reduction to logical satisfiability problems. We demonstrate the feasibility of formal analyses of ACT-R models for the presence of defects and discuss the potential of this approach to impact the overall process of cognitive modelling. Our approach is demonstrated on (but not limited to) a model of the preferred mental model theory (Johnson-Laird, 1980).

Motivating Example

Experimental Setting. An important and active field of research is the domain of spatial cognition, and especially spatial relational reasoning, because humans have to constantly plan and reason in space (e.g. Ragni & Knauff, 2013). A typical psychological experiment in the domain of relational spatial reasoning is the following. Verbal statements that spatially relate two objects to each other are subsequently presented on a computer screen. All statements but the last one are called premises, they describe valid spatial arrangements of the mentioned objects. The last relational statement is the so-called conclusion and participants should state whether the premises and the conclusion are contradictory.

Table 1 shows an example of a spatial reasoning task. Here, the conclusion is not contradictory because objects a, b, and c can be arranged in an order which satisfies both premises and the conclusion. In the following, we write, e.g., *aRb* (and *aLb*) to abbreviate premise or conclusion "a is to the right of b" (and "a is to the left of b").

Premise 1:	a is to the left of b.
Premise 2:	c is to the right of b.
Conclusion:	Is a to the left of c?

Table 1: Spatial reasoning task.

The Theory of Preferred Mental Models and ACT-R. An established theory that aims to explain human spatial reasoning is the mental model theory (MMT; Johnson-Laird, 1980). The most recent refinement of the MMT in the spatial context is the preferred mental model theory (PMMT; Ragni, Knauff, & Nebel, 2005; Ragni & Knauff, 2013). In the PMMT it is assumed that participants construct a mental spatial array of dynamic size which integrates information given by the premises. Whether a conclusion contradicts the given premises is checked by comparing the positions of the symbols in the conclusion with the positions of those symbols in the spatial array.

An ACT-R implementation of the PMMT is complicated because the rather stiff structure given by the mental array needs to be defined in terms of the dynamic structure of ACT-R's declarative memory and production rule system. Our straight-forward ACT-R implementation of the PMMT assumes the following cognitive process. First, a premise is read by the visual module. A production rule constructs a mental representation of the premise in the imaginal buffer (b_I). For example, for the premise *aLb* the chunk $\{l \mapsto a, r \mapsto b, sym \mapsto L\}^{prem}$ of type '*prem*' is constructed. If the premise is the first premise, a chunk $\{p_1 \mapsto a, p_2 \mapsto b\}^{mm}$ of type '*mm*' (mental model chunk) is constructed which assigns the two objects to positions. For the second premise, an existing mental model chunk is requested from the declarative module. If it can be recalled successfully, and the recalled mental model chunk and new premise share a common object, the new object can be assigned to a position in the mental spatial array as well. For example, the second premise *cRb* and the mental model chunk $\{p_1 \mapsto a, p_2 \mapsto b\}^{mm}$ have object b in common and constrain the position of object c to the right of b. As a consequence, c can only be assigned to position p_3, resulting in the mental model chunk $\{p_2 \mapsto b, p_3 \mapsto c\}^{mm}$.

A conclusion is verified by requesting the related mental model chunks from declarative memory and comparing the according positions. For example, for conclusion *aLc* both mental model chunks are recalled and positions $p_1 \mapsto a$ and $p_3 \mapsto c$ are compared. Here, the conclusion does not contradict the premises because p_1 is smaller than p_3.

Formal Analysis of ACT-R Models

In the following sections we give formal definitions of the three kinds of defects described in the introduction in terms of our formal ACT-R semantics (Albrecht & Westphal, 2014b). We provide illustrative examples for the three kinds of defects considered here in a PMMT model, propose a logical encoding of conditions which characterise a presence of the defect, and report on first experience from using tools to automatically analyse the satisfiability of the resulting formulae (and thus of the presence of defects).

Deadlock-freedom

We call an ACT-R model *deadlock-free* if and only if it cannot get stuck during simulation except when it has reached a final state. A final state is a cognitive state in which the model is designated to end (wrt. the cognitive theory), usually by making an output on console or via the motor system.

In the motivating example, the final states correspond to solving the given task, that is, giving the answer whether the conclusion is contradictory or not. For an example of a highly undesirable and hard to spot model defect, assume Rule `upd-mm-p1-cRb` would read '`posR =p2`' instead of '`posR =p1`' (which would be well-formed and could be

```
(p upd-mm-p1-cRb
 =goal>
    state     r-mm
  > premise   1
 =retrieval>
    type      mm
    p1        =p1
    p2        =p2
 =imaginal>
    type      prem
    sym       "r"
    posL      =l
    posR      =p1
==>
 +imaginal>
    type     mm
    p0       =l
    p3       =p1
 =goal>
    state    start
)
```

```
(p upd-mm-p2-cRb
 =goal>
    state     r-mm
  > premise   1
 =retrieval>
    type      mm
    p1        =p1
    p2        =p2
 =imaginal>
    type      prem
    sym       "r"
    posL      =l
    posR      =p2
==>
 +imaginal>
    type     mm
    p2       =p2
    p3       =l
 =goal>
    state    start
)
```

Figure 1: Two production rules integrating the second premise into the mental model. The rules differ only slightly thus there is a high risk for e.g. typos and hence for defects.

$$
\begin{array}{lll}
(goal.state_3 = recall-mm & \wedge goal.premise_3 > 1 & \wedge \\
retrieval.type_3 = mm & \wedge retrieval.p1_3 = p1_3 & \wedge \\
retrieval.p2_3 = p2_3 & \wedge imaginal.sym_3 = r & \wedge \\
imaginal.r_3 = p2_3 & \wedge imaginal.l_3 = l_3) & \\
\Rightarrow & & \\
(imaginal.sym_4 = mm & \wedge imaginal.p2_4 = p2_3 & \wedge \\
imaginal.p3_4 = l_3 & \wedge goal.state_4 = start & \wedge \\
rule_4 = 2) & &
\end{array}
$$

Figure 2: BMC-encoding of the production rule `upd-mm-p2-cRb` (cf. Figure 1). The formula states that if the cognitive state after the third transition of the model satisfies the premise of `upd-mm-p2-cRb`, then the fourth cognitive state may correspond to the action of `upd-mm-p2-cRb`.

the result of a simple typo). Then the model would run into a deadlock situation if premise *cRb* is presented after *aLb*. Hence, the model would not store the information from premise *cRb* at all, and thus be unable to solve the task with conclusion *aLc* (human participants easily solve that one).

Definition. In order to formally define deadlock-freedom, we need to recall essential notions from our formal ACT-R semantics (Albrecht & Westphal, 2014b). A production rule is a pair $r = (\rho, \alpha)$ which comprises a precondition ρ and an action α. The precondition ρ is a set of propositions over buffers and module states. An ACT-R model is a finite set $R = \{r_1, \ldots, r_n\}$ of production rules. A cognitive state (γ, t) consists of a mapping γ from buffers to chunks (or to the symbol *nil*), and a time-stamp $t \in \mathbb{R}_0^+$. A (formal) ACT-R model, i.e., a set of production rules, induces a timed transition system on cognitive states as follows. Two cognitive states (γ, t) and (γ', t') are in transition relation, denoted by $(\gamma, t) \xrightarrow{r} (\gamma', t')$, if there is a rule $r = (\rho, \alpha)$ such that precondition ρ is satisfied in γ, γ' is obtained by applying α to γ, and $t' - t$ is the time needed to execute action α.

Formally, the ACT-R model $R = \{r_1, \ldots, r_n\}$ is called *deadlock-free* given a set of final cognitive states F (wrt. the cognitive theory) if and only if R does not have any transition sequence $(\gamma_0, t_0) \xrightarrow{r_1} (\gamma_1, t_1) \xrightarrow{r_2} \ldots \xrightarrow{r_n} (\gamma_n, t_n)$, where (γ_0, t_0) is an initial cognitive state, $(\gamma_n, t_n) \notin F$ is not a final state, and there is no rule $r = (\rho, \alpha) \in R$ such that the precondition ρ is satisfied by γ_n. In other words, in an ACT-R model with a deadlock there exists a transition sequence such that no rule can be applied anymore although the model has not reached an outcome described in the cognitive theory.

Analysis Procedure. In the following, we show how to reduce the problem whether an ACT-R model R is deadlock-free within the first n transitions to satisfiability checking of logical formulae. Firstly, note that our definition of deadlock-freedom is independent of the sub-symbolic layer. That is, for all matching chunks in the declarative memory, retrieval can be successful. Thus model R has a deadlock in the sense of the definition if there is a symbolic execution of the model that has a deadlock.

Our approach is based on the well-known idea of bounded model checking (BMC) (Biere, Cimatti, Clarke, & Zhu, 1999). A BMC-encoding of the first n transitions of a transition system uses $n+1$ copies of system variables. If the transition system is induced by a set of rules, one uses n encodings $e(r,1), \ldots, e(r,n)$ of each rule, where $e(r,i)$ characterises the effect of rule r if cognitive state (γ_{i-1}, t_{i-1}) matches the premise of r. Thus, encoding all possible computation paths up to a length of n. BMC-encoding can be used to analyse bounded reachability problems as follows. Any state characterised by a formula φ is reachable in exactly n transitions if and only if the conjunction of the encoded production rules conjoined with φ is satisfiable. For details we refer to (Biere et al., 1999).

We can formulate deadlock-freedom within the first n transitions as a BMC problem as follows. We represent a cognitive state γ after the i-th transition, $i \in \{0, \ldots, n\}$, by the logical variables $b.s_i$ for each slot s of buffer b accessed in the ACT-R model. In addition, we assume an appropriate set of logical variables which represent the chunks in the declarative memory. In the following we explain the BMC encoding of production rules. Given production rule $r = (\rho, \alpha)$, the encoding $e(r,i)$ is obtained as follows. The precondition ρ is a set of module and buffers tests. The latter consist of slot tests of the form $s \sim v$ with slot name s, comparative symbol $\sim \in \{=, <, >, \neq\}$, and a constant or variable name v. If v is a constant, the slot test is encoded by the logical term $b.s_i \sim v$. If v is a variable, we introduce a fresh logical variable v_i and encode the slot test as the logical term $b.s_i \sim v_i$.

The action α of a production rule is a tuple $(\gamma_{part}, \lambda, C)$ consisting of a partial cognitive state γ_{part}, a valuation λ of module queries, and a set C of chunks. Each assignment in the partial state γ_{part} is encoded similar to the buffer test in the precondition. For all buffers b and slots s in γ_{part}, the encoding is the equation $b.s_i = \gamma_{part}(b,s)$. The set C in the action of a production rule consists of chunks that must be updated in declarative memory. These updates are encoded by equations using the logical variables which represent the declarative memory. If $e(\rho,i)$ and $e(\alpha,i)$ are the encodings of precondition and action of rule $r = (\rho,\alpha)$ for the i-th transition, then the encoding of r is $e(r,i) = e(\rho,i) \Rightarrow e(\alpha,i)$. The i-th transitions of ACT-R model $R = \{r_1,\ldots,r_n\}$ are encoded by the disjunction over the $e(r_j,i)$, $j \in \{1,\ldots,n\}$. The initial goal chunk is assigned to the according slots in γ_0.

To analyse deadlock-freedom within the first n steps, we add logical variables $rule_i$, $i \in \{0,\ldots,n\}$, and extend $e(\alpha,i)$ by the equation $rule_i = j$. Intuitively, the value of $rule_i$ denotes the index of the rule from R which was considered for the i-th transition; $rule_i$ being 0 encodes that no rule was enabled after $i-1$ steps. The formula for the reachability analysis is the equation $\varphi \equiv (rule_n = 0)$. The BMC-formula for $n \in \mathbb{N}_0$ and φ is satisfiable if and only if ACT-R model R has a deadlock after n transitions.

Example. Figure 2 shows the encoding of the production rule `upd-mm-p2-cRb` of the ACT-R model of the mental model theory. Note that buffer actions are resolved directly in the encoding of the production rules' actions, i.e., contents of imaginal and retrieval buffer are encoded as immediately being part of the successor state. This abstraction is possible as every transition of the model requires the imaginal and declarative system to have finished all requests. Module tests are not encoded as part of the precondition because it is possible to resolve all module tests for this model during encoding.

The ACT-R model of the mental model theory does only request chunks of the *mm* (mental model) type from declarative memory. During a run of the ACT-R model only two chunks of the type *mm* are cleared from a buffer and are at some transition part of C, thus an appropriate representation of the declarative memory consists of the logical variables $dm_j.p0, dm_j.p1, \ldots, dm_j.p3$ for $j \in \{1,2\}$, which represent the declarative memory after the first and second premise.

The SMT solver SMTInterpol (Christ, Hoenicke, & Nutz, 2012) is able to prove in a fraction of a second that our BMC-encoding of the PMMT model is unsatisfiable , thus we obtain a proof that this ACT-R model is deadlock-free in the first 6 transitions. That is, the production rules of the ACT-R model are able to process every combination of mental model and premise.

Correctness of the Mental Model

From many psychological theories, one can derive requirements on the content of the declarative memory at particular points in time. For example, PMMT implies that presenting the premises *aLb* and *bRc* (in this order) yields a declarative memory which contains exactly the chunks $\{p_1 \mapsto a, p_2 \mapsto b\}^{mm}$ and $\{p_2 \mapsto b, p_3 \mapsto c\}^{mm}$. A defect in the production rules which assemble the mental model could have a severe influence on the validity of an ACT-R model wrt. its psychological theory. A defect might not lead to obviously faulty behaviour (e.g. a deadlock) but it might influence the statistical data obtained from the cognitive model. Spotting such errors by simulation is in general tedious and time consuming.

Definition. Let Γ_{part} be a sequence of partial cognitive states of length $n \in \mathbb{N}$ and $M = \{c_1,\ldots,c_m\}$ a finite set of chunks. We call an ACT-R model R *mental model correct* wrt. (Γ_{part}, M) if and only if, for each transition sequence $(\gamma_0,t_0) \xrightarrow{r_1} \ldots \xrightarrow{r_n} (\gamma_n,t_n)$, where $r_i = (\rho_i, (\gamma_{part,i}, \lambda_i, C_i))$, we have $C_1 \cup \cdots \cup C_n = M$ and the i-th partial cognitive state $\Gamma_{part}(i)$ matches γ_i. If Γ_{part} (of length n) represents input, then mental model correctness intuitively requires that after input Γ_{part}, declarative memory is equal to M.

Analysis Procedure. Checking mental model correctness can be reduced to a BMC-problem as discussed above with additional formulae to encode the inputs Γ_{part} and with a goal formula φ which characterises M. We obtain a proper formula since both Γ_{part} and M are finite.

Example. To illustrate the complexity of the incremental construction of mental models, Figure 3 shows a part of the transition graph of the PMMT model. The initial state γ_0 has a transition to γ_1 with the production rule *attend*. Depending on the premise presented to the model the successor state may either be γ_{2l} or γ_{2r} with production rule *read-relation*. Production rule *read-relation* modifies the imaginal buffer b_I so that γ_{2l} contains $\{l \mapsto a, r \mapsto b, sym \mapsto L\}^{prem}$ in buffer b_I. Altogether there are ten rules assembling the mental model: Two rules processing the first premise and eight rules processing the second premise (see time steps t_2,t_3 and t_5,t_6 in Figure 1). The differences between the rules are subtle and errors are hard to spot; simulation is tedious and time consuming, since already this simple example has a total of sixteen possible combinations of premises and therefore as many simulations have to be executed.

For the formal analysis of mental model correctness, premises are encoded. For example premise *aLb* and *bRc* are encoded by the formulae $imaginal.l_2 = a \wedge imaginal.r_2 = b \wedge imaginal.sym_2 = L$ and $imaginal.l_4 = c \wedge imaginal.r_4 = b \wedge imaginal.sym_4 = R$. We check if the cognitive state after the input is combined in memory (see time step t_6 in Figure 1) with $\varphi = \neg(dm1.p1 = a \wedge dm1.p2 = b \wedge dm2.p2 = b \wedge dm2.p3 = c)$. The encoding can only be satisfied if there is a computation path that assembles a wrong representation of the inputs in the declarative memory.

SMTInterpol (Christ et al., 2012) checks the resulting 16 BMC-encodings for satisfiability again in a fraction of a second. During these analyses, the SMT solver found a satisfying valuation of the logical variables for one combination of premises. Thus there is an input sequence on which the con-

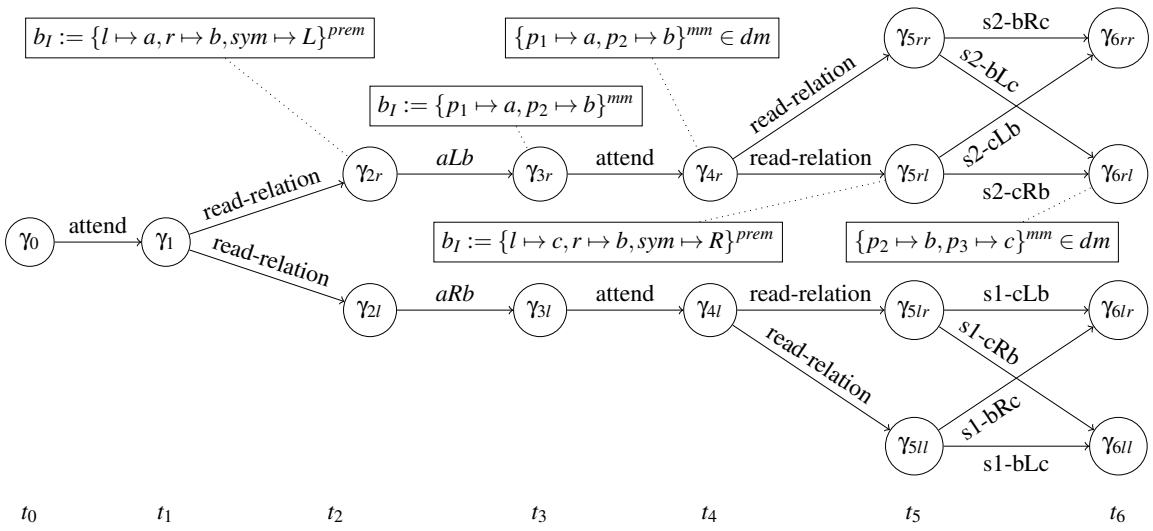

Figure 3: Transition graph of the first six transitions of the ACT-R model of the preferred mental model theory. Cognitive states are depicted as nodes, transitions are labelled with the production rules. Cognitive states for the sequence of transitions $\gamma_0 \xrightarrow{attend} \ldots \xrightarrow{s2-blc} \gamma_{6rl}$ are annotated with parts of their cognitive state. The states illustrate the construction of a mental model from the premises *aLb* and *cRb*. The imaginal buffer is depicted as b_I, declarative memory as dm.

sidered ACT-R model did not assemble the expected mental model. This shows that the original ACT-R model in fact had a defect that was not found during implementation. The incorrect model resulted from assigning an incorrect variable to slot p_2, probably a nasty copy-and-paste error.

Timing Feasibility

When modelling PMMT in ACT-R, a crucial aspect is the use of the declarative memory. In the ACT-R theory, the time and probability for retrieving a chunk from declarative memory depend on the *activation* of chunks. Activation in turn depends on different assumptions on human memory processing, e.g. spreading activation, where the similarity between the buffer contents and the chunks in the declarative memory is considered and base level learning, where the history of chunk constructions and retrievals is considered. In an ACT-R cognitive architecture where only base level learning is enabled, the activation is calculated based on two global parameters: the decay rate δ which determines how fast the activation of a chunk decays over time and the threshold τ which defines a lower bound on activation values for successful chunk retrieval.

A fundamental assumption of the PMMT is that the preferred mental model for the two premises is constructed *before* the conclusion is presented. That is, the behaviour of the environment imposes hard deadlines on the timing of the model: any valid ACT-R model for the PMMT must complete the processing of all rules needed to construct the preferred mental model before the next stimulus is presented.

During a task, stimuli are presented to the participants at fixed points in time. For example, let E_1 denote the onset premise and E_2 denote the onset of the conclusion shown at times t_2 and t_6, respectively (cf. Figure 3). This is the interval where the second premise has to be processed. Then, according to the assumption stated above, processing of the second premise has to be completed within $t_b := t_2 - t_6$ time units. An ACT-R model for this task in particular needs to model successful solutions of the task. That is, in an ACT-R model which is valid wrt. experimental data, the execution of all rules which are involved in constructing the mental model must complete in at most t_b time units (cf. the top row of Figure 4). The lower row of Figure 4 shows an abstract transition graph of a corresponding ACT-R model. Here, we assume that there is only one request to the declarative module by rule r, i.e. a request for the already constructed mental model comprising premise 1, which has two qualitatively different outcomes: a correct reply ('✔'), and a wrong reply or no reply at all ('✘'). Now given corresponding rules, if it is *impossible* to choose the decay rate δ and the threshold τ such that $t_2 - t_i \leq t_b$, then the considered rules definitely *do not* constitute a valid (partial) ACT-R model for the PMMT.

Definition. An ACT-R model R is called *timing feasible* wrt. cognitive states E_1 and E_2 and duration $t_b \in \mathbb{R}_0^+$ if and only if there exist values for decay rate δ and the threshold τ such that there is a transition sequence $(\gamma_0, t_0) \xrightarrow{r_1} \ldots \xrightarrow{r_n} (\gamma_n, t_n)$ of R and $0 \leq i, j \leq n$ such that $\gamma_i = E_1$ and $\gamma_j = E_2$, and $t_j - t_i \leq t_b$.

Analysis Procedure. Timing feasibility can be encoded as a BMC-formula as discussed above with the goal formula $\varphi = t_6 - t_2 \leq t_b$. Timing is tracked by additional timestamp variables added to the encoding of the cognitive states. The recall time of a chunk from declarative memory depends on the activation of the chunk. We replaced the non-

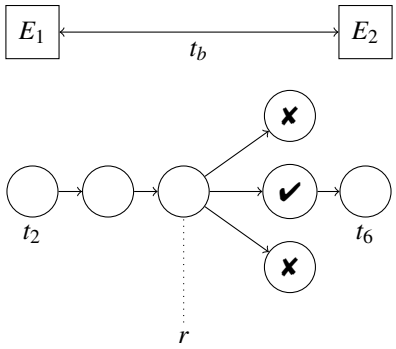

Figure 4: Example sequence of cognitive states (circles) in between environment event E_1 and E_2 (rectangles). A cognitive state which leads to a correct reply is denoted by '✔', and a state which leads to a wrong reply or no reply at all as '✘'. Label r indicates a state where a retrieval request is posed to the declarative module by rule r.

deterministic subsymbolic layer used in the previous sections by a linear approximation of base level learning: $A(c,t) = ln(2) - ln(1-\delta) - \delta \cdot (t - t_c)$ where t_c refers to the first time a chunk c was presented. This demonstrates that our analysis approach is not only feasible for the analysis of the symbolic aspects of ACT-R but integrates well with the analysis of aspects of the subsymbolic layer such as activation and retrieval latency.

Example. Consider the phase of the PMMT shown in Figure 4. More specifically, consider a rule r which requests the declarative module for the first mental model chunk when the second premise is presented at time t_2. The ACT-R model has only one nondeterministic rule r which is ever enabled between t_2 and t_6. Then the time to execute the model only varies wrt. the time for executing r.

SMTInterpol (Christ et al., 2012) checked the resulting BMC-encoding for timing feasibility again in a fraction of a second. The satisfying assignment of the reported solution provides variables for δ and τ that allow a timing feasible execution of the model. If we choose the initial activation of the chunk in memory at time E_1 too low, SMTinterpol proves the example to be timing infeasible.

Note that our approach is not limited to the analysis of *single* rules. Given an upper bound n on the number of rules possibly executed between two points in time, a similar construction of a BMC-formula is possible.

Validity. For a model with critical timing the ACT-R model validity problem basically reduces to checking whether, given a start cognitive state (γ, t) and a goal state (γ', t') there exist values for δ and τ such that there is a sequence of transitions. By adding, e.g., constraints on τ and δ to, we can use the same procedure whether their values lie within a range accepted by the community. Therefore it may not even be necessary to write a whole ACT-R model as validity of the model can be rejected by analysing timing feasibility.

Conclusion

We propose and formally define three new properties of ACT-R models which, if not satisfied, indicate that the model does not correctly implement its psychological theory. The properties characterise the model defects deadlock, incorrectness of the mental model, and timing infeasibility. We present a method to automatically check a given ACT-R model for the three properties and thus the presence of defects. Our method is based on encoding the properties as a satisfiability problem, which can be analysed by an SMT solver. A proof-of-concept implementation of the new analysis methods showed deadlock-freedom and timing feasibility and interestingly uncovered a previously unknown defect wrt. mental model construction in our running example, a variant of the preferred mental model theory (Ragni & Knauff, 2013).

We expect the modelling task, that is, the creation of a *correct* implementation for a given psychological theory to become much more efficient using our approach as compared to simulation-based approaches.

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – PO 279/2-1. This work extends results presented at KogWis 2014, Tübingen, Germany.

References

Albrecht, R., & Westphal, B. (2014a). Analysing Psychological Theories with F-ACT-R. *Cognitive Processing*, *15*, 77–79.

Albrecht, R., & Westphal, B. (2014b). F-ACT-R: Defining the Architectural Space. *Cognitive Processing*, *15*, 79-81.

Anderson, J. R. (1983). *The Architecture of Cognition*. Psychology Press.

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press.

Biere, A., Cimatti, A., Clarke, E. M., & Zhu, Y. (1999). Symbolic model checking without BDDs. In *TACAS* (Vol. 1579, pp. 193–207).

Christ, J., Hoenicke, J., & Nutz, A. (2012). SMTInterpol: An Interpolating SMT Solver. In A. F. Donaldson & D. Parker (Eds.), *SPIN* (Vol. 7385, p. 248-254). Springer.

Gall, D., & Frühwirth, T. W. (2014). A formal semantics for the cognitive architecture ACT-R. In *LOPSTR* (Vol. 8981, pp. 74–91). Springer.

Gall, D., & Frühwirth, T. W. (2017). A decidable confluence test for cognitive models in ACT-R. In *RuleML+RR* (Vol. 10364, pp. 119–134). Springer.

Johnson-Laird, P. (1980). Mental models in cognitive science. *Cognitive Science*, *4*, 71–115.

Ragni, M., & Knauff, M. (2013). A theory and a computational model of spatial reasoning with preferred mental models. *Psychological review*(3), 561–588.

Ragni, M., Knauff, M., & Nebel, B. (2005). A Computational Model for Spatial Reasoning with Mental Models. In *Proc. of the 27th annual Cog. Sci. Conf.* (pp. 1064–1070).