Cognitive Psychology 74 (2014) 1–34

Contents lists available at ScienceDirect

Cognitive Psychology

journal homepage: www.elsevier.com/locate/cogpsych

Extending problem-solving procedures through reflection ☆

John R. Anderson *, Jon M. Fincham

Department of Psychology, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:
Accepted 18 June 2014
Available online xxxx

Keywords:
Mathematical problem solving
Reflection
Cognitive modeling
Multivoxel pattern analysis
Hidden Markov models

ABSTRACT

A large-sample ($n = 75$) fMRI study guided the development of a theory of how people extend their problem-solving procedures by reflecting on them. Both children and adults were trained on a new mathematical procedure and then were challenged with novel problems that required them to change and extend their procedure to solve these problems. The fMRI data were analyzed using a combination of hidden Markov models (HMMs) and multi-voxel pattern analysis (MVPA). This HMM–MVPA analysis revealed the existence of 4 stages: Encoding, Planning, Solving, and Responding. Using this analysis as a guide, an ACT-R model was developed that improved the performance of the HMM–MVPA and explained the variation in the durations of the stages across 128 different problems. The model assumes that participants can reflect on declarative representations of the steps of their problem-solving procedures. A Metacognitive module can hold these steps, modify them, create new declarative steps, and rehearse them. The Metacognitive module is associated with activity in the rostrolateral prefrontal cortex (RLPFC). The ACT-R model predicts the activity in the RLPFC and other regions associated with its other cognitive modules (e.g., vision, retrieval). Differences between children and adults seemed related to differences in background knowledge and computational fluency, but not to the differences in their capability to modify procedures.

☆ The analyses and models in this paper can be obtained at http://act-r.psy.cmu.edu/?post_type=publications&p=16145.

* Corresponding author.
E-mail addresses: ja@cmu.edu (J.R. Anderson), fincham@cmu.edu (J.M. Fincham).

http://dx.doi.org/10.1016/j.cogpsych.2014.06.002

1. Introduction

While some instruction has as its goal that the learner become skilled at just what is being taught, in many cases the goal is for the learner to be able to transfer what is learned to new situations. The literature abounds with demonstrations of both failed transfer (e.g., Bassok, 1990; Detterman, 1993; Gick & Holyoak, 1980) and near total transfer (e.g., Bovair et al., 1980; Singley & Anderson, 1989). Educators properly anguish over the implications of these apparently contradictory results (e.g., Bransford & Schwartz, 1999; Carraher & Schliemann, 2002).

One of the reasons for the different perspectives on transfer is the wide variety of things that can transfer. They can range from transfer of highly proceduralized skills such as from one kind of manual transmission to another to what might better be called discovery such as the connection made between the structure of the solar system and the structure of the atom. This paper will focus on a particular type of transfer – where one derives new solution procedures by extending problem-solving procedures that one already knows. It is particularly important in mathematics learning, which is the content focus of this paper. To take a modest example, children who learn the basic principles for solving equations need to apply them successfully to an infinite space of equations. To take a more ambitious example, mathematics education hopes that students will transfer what they learn in the classroom to being successful workers and informed citizens.

More specifically, this paper will consider situations where participants need to reflect on a known procedure and modify and replace parts of it. For instance, people often face such a situation when a favorite piece of software is upgraded. It is an explicit goal of the National Council of Teachers of Mathematics (NCTM) standards (Romberg, 1992) that students should be able to "generate new procedures and extend or modify familiar ones."

This paper will develop a theory of procedural extension within the ACT-R theory (Anderson, 2007; Anderson et al., 2004; Salvucci, 2013; Taatgen, Huss, Dickison, & Anderson, 2008) of procedure following. The ACT-R theory holds that both verbal procedural instructions and examples of procedures are initially encoded as declarative representations of problem-solving steps, which are retrieved and interpreted in solving a problem. Note that declarative encodings of procedures are not the sort of unconscious "procedures" that occupy much of the discussion about the procedural–declarative distinction in psychology (e.g., Cohen, Poldrack, & Eichenbaum, 1997; Willingham, Nissen, & Bullemer, 1989). With enough practice such declarative knowledge can be compiled into production rules in ACT-R, which are one form of unconscious procedures.

Recently, Taatgen (2013) has produced an ACT-R theory of transfer in which steps from one procedure automatically transfer to another procedure. This is not the reflective transfer considered here. This paper is concerned with situations where one consciously reflects on what one knows and how to extend that knowledge. A classic example would be Wertheimer's (1945/1959) study of how children could use what they know of the area of rectangles to find the area of a parallelogram.

2. ACT-R, procedure following, and fMRI

As background for the current research, we will briefly review the ACT-R theory, how procedure following is modeled, and how the activity of components in the ACT-R theory have been related to fMRI measures. ACT-R 6.0 (Anderson, 2007) consists of a set of different modules whose interactions are controlled by a production system. Different modules are specialized to achieve specific goals. Of relevance to this paper, the Manual module programs the hands, the Visual module encodes visual input, the Retrieval module accesses declarative information, and the Imaginal module manipulates mental representations. These modules put products into module-specific buffers – for instance, representation of a visual object into the Visual buffer or a retrieved memory into a Retrieval buffer. Productions can detect information in these buffers and make requests of modules. For instance, a production can detect an object in the Visual buffer and request the Retrieval module find declarative information relevant to that object.

The ACT-R theory of declarative procedure following has had considerable success in modeling the learning of a number of procedures including simple algebra (e.g., Anderson, 2005). According to

ACT-R (Anderson, 2007; Salvucci, 2013; Taatgen et al., 2008), learners start out with declarative representations of procedures. The initial declarative representations of procedures, called **operators**, consist of preconditions, simple actions that ACT-R can perform, arguments for these actions, and post-actions (see Appendix B for more detailed discussion). These operators are retrieved interpreted by the general production rules.

During the execution of a skill, multiple modules will be engaged according to the requirements of the skill. Activities in each of these modules reflect demands on corresponding brain regions supporting the functions of each module. There are the following associations between brain regions and ACT-R modules (see Fig. 1):

1. *Visual module:* While large portions of the brain are devoted to processing visual information, researchers have found that the fusiform gyrus plays a critical role in the recognition of visual objects (Grill-Spector, Knouf, & Kanwisher, 2004; McCandliss, Cohen, & Dehaene, 2003). While different small areas in this region respond maximally to different material, this aggregate region has consistently tracked the engagement of the ACT-R Visual module across multiple tasks.
2. *Imaginal module:* The Imaginal module is associated with a region of posterior parietal cortex (PPC). This association is roughly consistent with the research of others who have found that this area is involved in spatial processing (e.g., Dehaene, Piazza, Pinel, & Cohen, 2003; Reichle, Carpenter, & Just, 2000) and verbal encoding (Clark & Wagner, 2003; Davachi, Maril, & Wagner, 2001). This module is responsible for transformations of problem representations.

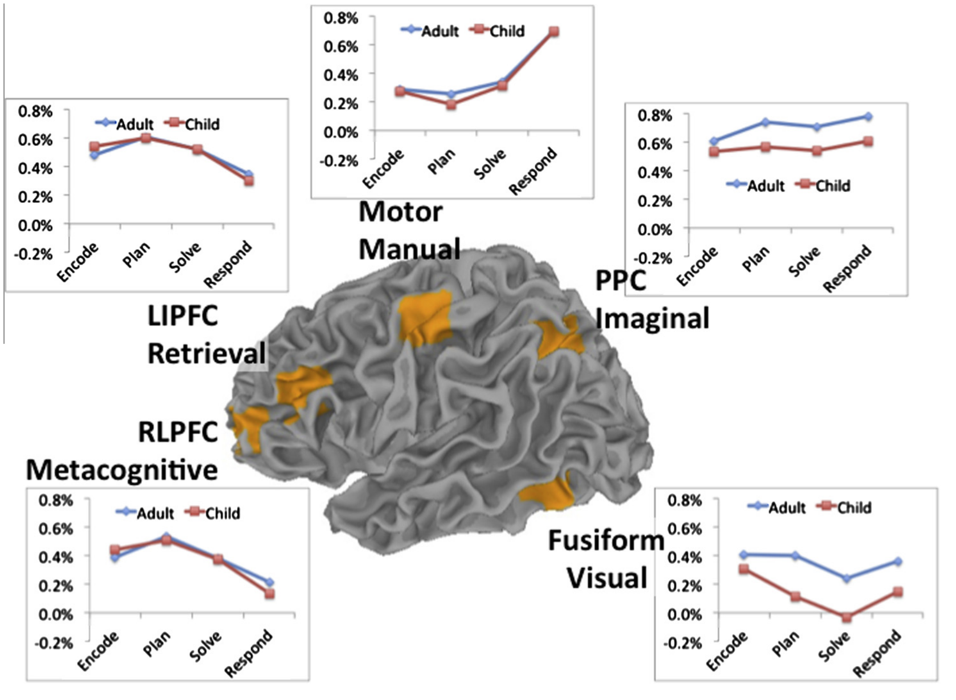

Fig. 1. The brain regions associated with modules in ACT-R: *Rostrolateral prefrontal cortex (RLPFC – Metacognitive module):* a 12.8 mm (high) by 15.6 × 15.6 mm^2 region centered at Talairach coordinates −34,47,8; *lateral inferior prefrontal cortex (LIPFC – Retrieval module):* a 12.8 mm (high) by 15.6 × 15.6 mm^2 region centered at Talairach coordinates −43,23,24 spanning Brodmann Areas 9 and 46; *posterior parietal cortex (PPC – Imaginal module):* a 12.8 mm (high) by 15.6 × 15.6 mm^2 region centered at Talairach coordinates −23,−63,40 spanning Brodmann Areas 9 and 46; *motor (Manual module):* a 12.8 mm (high) by 15.6 × 15.6 mm^2 region centered at Talairach coordinates −42,−19,50, involving Brodmann Areas 2 and 4 at the central sulcus; *fusiform (Visual module):* a 9.6 mm (high) by 12.5 × 12.5 mm^2 region centered at Talairach coordinates −42,−60,−8 in Brodmann Area 37. Also displayed are the activation patterns for children and adults.

3. *Manual module:* Hand movements are reflected in the activity of the region along the central sulcus where the hand is represented. This includes parts of both the motor and sensory cortex.
4. *Retrieval module:* While many regions, including the hippocampus, are crucial in the storage and retrieval of memories, substantial evidence indicates that the lateral inferior prefrontal cortex (LIPFC) plays a critical role in the effortful retrieval of learned information (e.g., Badre & Wagner, 2007; Thompson-Schill, D'Esposito, & Kan, 1999; Wheeler & Buckner, 2004).
5. *Metacognitive module:* This paper introduces a new Metacognitive module and associates its activity with the rostrolateral prefrontal cortex (RLPFC). Considerable other research relates the RLPFC to reflective functions. For instance, the RLPFC is engaged upon feedback in episodic memory experiments (Reynolds, McDermott, & Braver, 2006; Rugg, Henson, & Robb, 2003), when considering intentions in prospective memory (Benoit et al., 2012), when reasoning about higher-order relationships and analogies (Bunge, Helskog, & Wendelken, 2009; Bunge, Wendelken, Badre, & Wagner, 2005; Volle, Gilbert, Benoit, & Burgess, 2010; Wendelken, Nakhabenko, Donohue, Carter, & Bunge, 2008), and when reflecting on task performance (Fleming, Huijgen, & Dolan, 2012). These findings suggest a general capacity of this region consistent with the function of ACT-R's Metacognitive module, which is to reflect on declarative representations of cognitive procedures.

As illustrated in Fig. 1, we will focus on left hemisphere regions. Except for the motor region, similar (if weaker) patterns are typically found in the right-lateralized homologues. While the functionality of these modules is probably not just achieved in these regions, having predefined regions does allow for tests of predictions of an ACT-R model about module activity. As we will describe, the activity of ACT-R's modules can be converted into predictions about the temporal patterns of activations in these regions.

3. The challenge of modeling procedural extension

There has been a considerable history of ACT-R models successfully predicting activity in the regions of Fig. 1 (other than the RLPFC; see Anderson, 2007; Anderson, Carter, Fincham, Ravizza, & Rosenberg-Lee, 2008 for reviews of the work). This comfortable picture of research success was upset when we decided to explore what happens when participants were asked to extend what they had been taught to do. One such task involves what are called **pyramid problems** which are presented with a dollar symbol as the operator – e.g., $4\$3 = X$. Pyramid problems involve a **base** ("4" in this example) that is the first term in an additive sequence and a **height** ("3" in this example) that determines the number of terms to add. Each term in the sequence is one less than the previous – so $4\$3 = 4 + 3 + 2$. As students work with pyramid problems they quickly master the procedure of iterative addition for solving these Regular problems. A Regular problem is one that involves positive single digits for base and height. Participants typically require about an hour of practice to become proficient in solving such **Regular** problems.

Interest focuses on their performance on the following day, when participants are presented with **Exception** problems that require they extend their knowledge. For instance:

$-9\$4 = X$
$X\$4 = X.$

Perhaps the most significant observation about Exception problems is that every student we have tested is able to solve some of them. If one views the students as having learned some fixed procedure for solving Regular problems, they should not be able to solve the Exception problems (or for some Exceptions, e.g., $100\$201 = X$, not be able to solve them in the allotted time). Thus, these students must have some ability to change their procedures on the fly. However, simply observing an individual student's pattern of success or failure offers little insight into how they are modifying their procedures. We have gathered verbal protocols and while they can indicate the method the student uses to solve the problem, they do not indicate what the mental processes are. Consider examples of protocols that accompanied the two problems above:

"So wait it's one less so it's minus ten...minus nine plus minus ten plus minus eleven is three and then negative – minus 42"

"different... Oh, that's interesting – x, 3x + 6"

In both cases the "..." reflect long, pregnant pauses in which "something" is happening.

There were two earlier efforts at modeling pyramid problems within ACT-R. The first model (available at the website associated with Anderson, 2007) fit latency data from a few Exception problems. That model made predictions about brain activation that did not correspond to the data that we later obtained. A second model (available at the website associated with Wintermute, Betts, Ferris, Fincham, & Anderson, 2012) successfully fit both the behavioral and activation data from Regular problems but did not address Exception problems at all. To deal with Exception problems and the imaging data associated with them, we have had to develop new analyses of the imaging data, extend the capabilities of the ACT-R architecture, and enrich our understanding of how ACT-R relates to imaging data. Fig. 2 illustrates our approach to coming up with a satisfactory model and also indicates much of the ground that this paper will cover:

1. *Discover mental states:* We combined multi-voxel pattern analysis (MVPA) and hidden Markov models (HMMs) to identify the mental states that one goes through in a problem-solving task. MVPA identifies the stable patterns of brain activity and HMMs identify the temporal patterns formed by these states. We call the combined methodology HMM–MVPA. We refer to the stable pattern of brain activity associated with a state as its **brain signature**. While the brain signatures for a state are assumed to be constant across problems, the duration of the state can vary from problem to problem. The methodology provides estimates of how long the states last in different conditions and how individual problems vary from the mean state times for their condition.
2. *Guide ACT-R modeling:* The information about the brain signatures and state durations proved to be informative in determining how different problems are solved. This served to guide the construction of an ACT-R model of the information-processing steps in solving these problems.

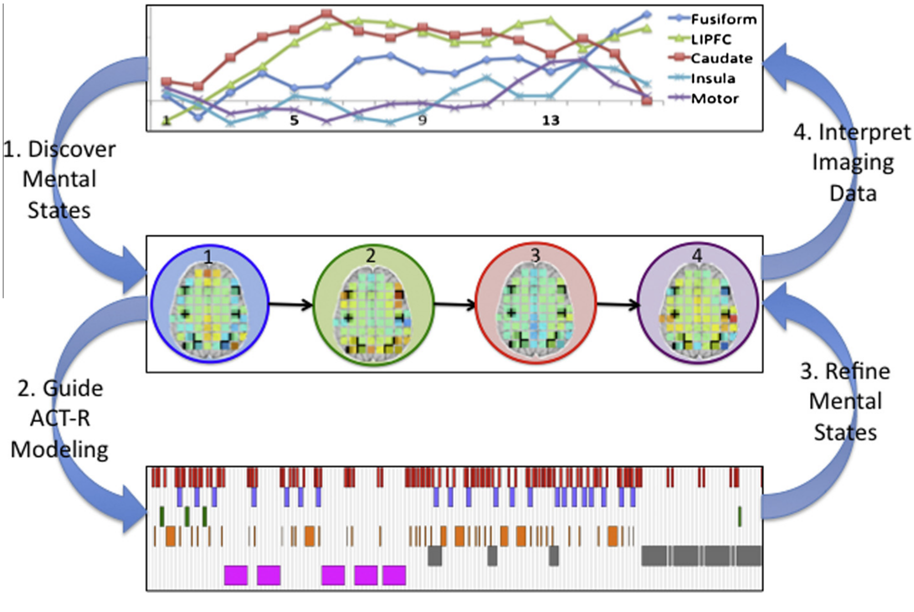

Fig. 2. The overall plan for discovering mental states, guiding modeling, further constraining the states, and then interpreting the data.

Table 1
Ten Types of problems used (number of distinct problems in parentheses), mean accuracies, and correct latencies.

Type	Example	Accuracy		Latency	
		Adults (%)	Children (%)	Adults (s)	Children (s)
Regular (24)	8$4 = *X*	89.1	82.8	8.3	9.6
Negative height (8)	4$−3 = *X*	83.9	67.9	10.8	13.9
Negative base (8)	−2$4 = *X*	70.1	45.0	13.7	15.6
Large base value? (8)	208$3 = *X*	62.3	32.0	16.1	18.1
Unknown height (24)	5$*X* = 12	84.5	71.3	9.4	10.7
Unknown base (24)	*X*$4 = 30	80.1	62.7	10.9	12.1
Double X (8)	*X*$*X* = 15	64.4	46.4	13.3	13.4
LBUH (8)	110$*X* = 434	90.5	74.3	9.0	9.4
Fractional height (8)	5$2⅓ = *X*	72.2	42.6	12.1	14.5
Mirror (8)	200$401 = *X*	61.2	31.6	9.3	8.0

Example feedback given for each problem:
8$4 = X: 8$4 = 8 + 7 + 6 + 5 = 26.
4$−3 = X: 4$−3 = 4 + 5 + 6 = 15 (or 5 + 6 + 7 if participant had chosen that solution).
−2$4 = *X*: −2$4 = −2 + −3 + −4 + −5 = −14.
208$3 = *X*: 208$3 = 208 + 207 + 206 = 621.
5$*X* = 12: 5$3 = 5 + 4 + 3 = 12.
***X*$4 = 30:** 9$4 = 9 + 8 + 7 + 6 = 30.
***X*$*X* = 15:** 5$5 = 5 + 4 + 3 + 2 + 1 = 15.
110$*X* = 434: 110$4 = 110 + 109 + 108 + 107 = 434.
5$21/3 = *X*: 5$21/3 = 5 + 4 + 1/3(3) = 10.
200$401 = *X*: 200$401 = 200 + 199 + . . .−199 + −200 = 0.

3. *Refine mental states:* The ACT-R model allowed us to bring top-down computational constraints to bear in more accurately estimating the states initially identified bottom-up in Step 1.
4. *Interpret imaging data:* These constrained mental states allow us to understand differences between two participant populations, adults and young adolescents.

Before going through these four steps we will describe in more detail the pyramid experiments.

4. Pyramid experiments

We have collected data from 40 adults (ages 19–35) and 35 young adolescents (ages 12–14) solving these problems. Although the adults were more successful and somewhat faster than children (see Table 1), the two populations overlap (see Fig. 20). Their data are pooled but the results do not substantially change if the two populations are analyzed separately. A data set this large provides a basis for application of the state discovery procedures. The end of this paper will address the question of what this methodology says about the differences between the two populations. Participants practiced solving a large number of Regular problems (adults solved 81 problems with heights from 1 to 9, children solved 108 problems with heights from 1 to 6) on a prior day outside of the scanner. On the second day, they were tested in an fMRI scanner with Exception problems mixed in with Regular problems. In each scanner block, they solved 3 Regular problems and one instance of each of the 9 Exception types in Table 1. The first problem was always a Regular problem and was excluded from analysis as a warm-up problem. Otherwise the problems were tested in a random order. There were 10 problem types consisting of the Regulars and 9 exception types. The 10 types could be further broken out into 128 different specific problems, but no participant saw all 128 problems. Our interest is in how participants solved these problems and so we will focus on correct problems. This resulted in anywhere from 19 to 87 solutions per participant with a mean number of 53.7.

The procedures were described in detail in the original empirical report on the adult data (Wintermute et al., 2012).[1] The only difference between the populations during the scanning session

[1] The children data has not been previously published.

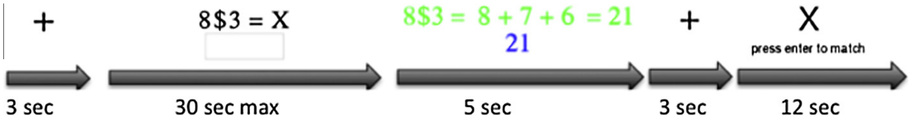

Fig. 3. The experimental procedure on a trial. Each trial consists of a fixation period, a problem-solving period, a feedback period, a fixation period, and a repetition detection task.

is that most children only completed 6 blocks in the allotted time while adults completed 8 blocks. Fig. 3 illustrates the basic procedure on a trial. The problem was presented after a 3 s fixation. Participants had 30 s to input an answer. Participants entered their answers with a physical keypad that they had been trained on. After their response or 30 s expired, feedback was presented for 5 s. Examples of the feedback are shown in Table 1. The feedback was in green if the participant's answer was correct and in red if the answer was wrong or the problem had timed out. A fixation cross was again presented for 3 s. Then participants performed a repetition detection task for 12 s. During repetition detection, letters appeared on the screen at a rate of 1 per 1.25 s, and participants were instructed to press Enter on the keypad when a letter repeated. This task served to distract the participants from the main pyramid task and return brain activity to a relatively constant level.

Images were acquired using gradient echo-echo planar image (EPI) acquisition on a 3T Verio Scanner using a 32-channel RF head coil, with 2 s repetition time (TR), 30 ms echo time (TE), 79 degree flip angle, and 20 cm field of view (FOV). Both experiments acquired 34 axial slices on each TR using a 3.2 mm thick, 64×64 matrix. This produces voxels that are 3.2 mm high and $3.125 \times 3.125\ mm^2$. The anterior commissure–posterior commissure (AC–PC) line was on the 11th slice from the bottom scan slice. Acquired images were pre-processed and analyzed using AFNI (Cox, 1996; Cox & Hyde, 1997). Functional images were motion-corrected using 6-parameter 3D registration. All images were then slice-time centered at 1 s and co-registered to a common reference structural MRI by means of a 12-parameter 3D registration and smoothed with a 6 mm full-width-at-half-maximum 3D Gaussian filter to accommodate individual differences in anatomy.

In complex tasks like this, it is useful to perform MVPA on whole brain activity. However, as a step of dimension reduction and to accommodate variations in anatomy over participants that may not be dealt with in co-registration, we aggregated the voxels in each slice into larger 2×2 voxel regions. There are 12,481 such regions.[2] Some of these regions show an excess of extreme values for some participants, probably reflecting differences in anatomy. These were regions mainly on the top and bottom slices as well as some regions around the edge of the brain. Eliminating these regions resulted in keeping 8365 regions.

The BOLD response is calculated as the percent change from a linear baseline defined from first scan (beginning of fixation before problem onset) to last scan (beginning of fixation before next problem). This is deconvolved with a hemodynamic response function to produce an estimate of the underlying activity signal. The hemodynamic function is the SPM difference of gammas (Friston, Ashburner, Kiebel, Nichols, & Penny, 2011: gamma(6,1) – gamma(16,1)/6). A Wiener filter (Glover, 1999) with a noise parameter of .1 was used to deconvolve the BOLD response into an inferred activity signal on each scan.

The output of this process is an estimation of activity in 8365 regions during every 2-s scan. However, because of high correlation among regions this reflects nothing like 8365 independent pieces of information. We perform a spatial principal component analysis (PCA) of the voxel activity where each voxel is treated as a variable that varies over scans, trials, and subjects. The first 20 components of the PCA capture 67% of the total variance in the data and we have had success working with just these. These 20 component scores are approximately distributed as 20 independent normally distributed variables. We standardize them to have mean 0 and variance 1 (performing a PCA and standardizing the data is sometimes called "whitening").

[2] This contrasts with Anderson and Fincham (2014) where we used 408 larger $4 \times 4 \times 4$ voxel regions and kept 290 to avoid extreme values.

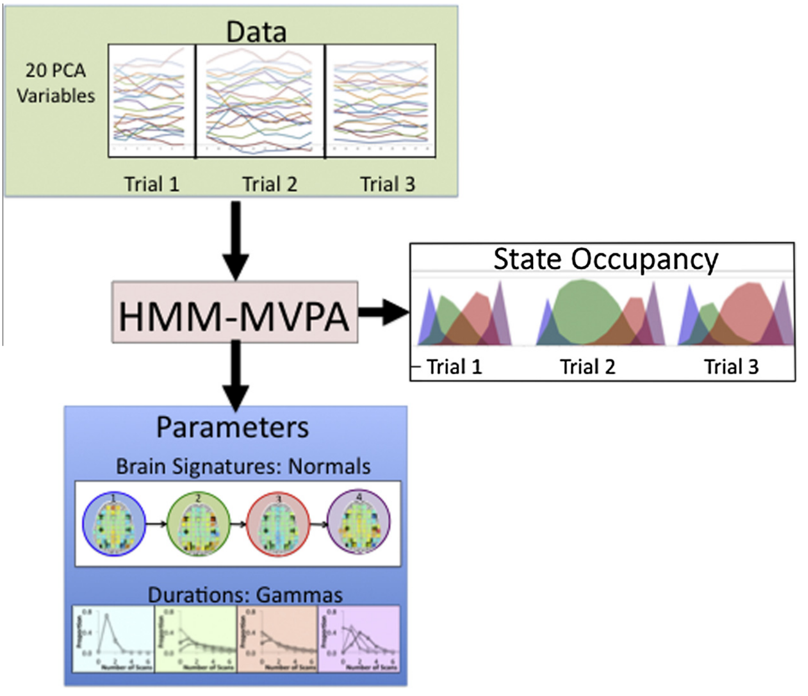

Fig. 4. An illustration of HMM–MVPA producing a 4-state solution with 3 conditions. Input to the HMM–MVPA are scans organized within trials. Each scan consists of the 20 PCA scores. Parameters are estimated for each state: 20 PCA means that define the brain signature and distribution parameters for each condition-state combination. Also calculated are the state occupancies (probabilities that the participant is in each state on a particular scan).

The majority of this paper will focus on correct trials only because these are the trials where we can be most confident of the procedures that the participant was using. However, we will also provide an analysis of what this methodology can say about incorrect trials at the end of the paper.

5. Step 1. Discovering mental states

This section describes an updated version of the model discovery process described in Anderson and Fincham (2014). This procedure is purely data-driven and is in no way specific to the ACT-R theory. Fig. 4 provides an overview of the state discovery procedure. The inputs to this state discovery procedure are the 20 PCA scores for each scan. The outputs are a set of parameters that describe the states and a description of these trials in terms of their **state occupancy** (probability of being in a state on a scan).

We conceive of the participant as going through a sequence of mental states during the solution of one of these problems. These states are "hidden" in the sense that we do not see them but only see their consequences in the brain signals. Because of the poor temporal resolution of fMRI, these mental states capture many seconds of brain activity. A mental state is a period of time with a constant activation pattern, that state's brain signature. The brain signature for a state is estimated as a set of 20 means for the 20 PCA dimensions. We assume that individual scans in the state form a normal distribution around this point in a 20-dimensional space. The dimensions of this distribution for each state are treated as orthogonal with standard deviations of 1, just like the full set of all PCA scores.[3]

[3] The distributions around the state means do appear orthogonal and normal but they do have somewhat smaller standard deviations (about .9). However, we choose not to estimate standard deviations to reduce the number of parameters.

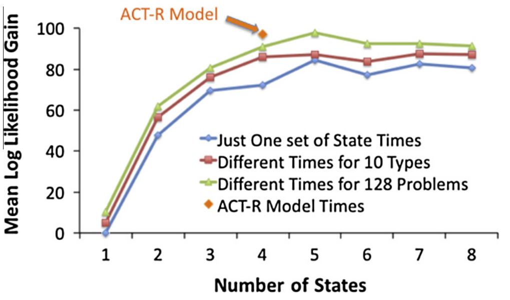

Fig. 5. A search for the best HMM–MVPA solution through a space defined by different numbers of states and different numbers of time distributions within a state.

The duration of each state will vary from trial to trial. Some of this might be random variation, but some of it might also reflect differences in the problems being solved. For instance, states could have different mean durations for each of the 10 problem types in Table 1. Within a condition, there will be trial-to-trial variation in the duration of the state. We characterize this distribution of state times in a condition as a gamma distribution defined by an index parameter v and a scale parameter a (the mean is va and the variance va^2). For purposes of avoiding overfitting, we have found it better to estimate a single scale parameter a per state and just allow the index v to vary with condition.[4]

To summarize, a state is defined by a single brain signature for all conditions (defined by means for the 20 PCA dimensions), but within a state durations will vary with condition (defined by the index parameter for the condition). Identifying a state involves estimating the means of the 20 PCA dimensions, the single scale parameter a, and the index parameters for each condition. They are estimated to maximize the probability of data (i.e., the 20 PCA scores for all scans for all problems). In approaching a novel domain like the pyramid problems we do not know the number of states or how the problems break up into meaningful conditions. Thus, in addition to estimating the parameters that define the states and conditions within states, we need to discover the states and conditions themselves.

Allowing for more states or more conditions will mean more parameters and therefore a better fit to the data, but this might just be overfitting (i.e., fitting noise rather than systematic variation). Although there are metrics for penalizing models for their extra parameters like BIC (Kass & Raftery, 1995), they do not extend in a straightforward manner to situations with so many parameters (Berger, Ghosh, & Mukhopadhyay, 2003) or where observations are not independent as is true of fMRI data (Jones, 2011). In contrast, cross-validation methods offer an effective way to assess models and identify when the extra model complexity is justified. This paper uses simple leave-one-out cross validation (LOOCV). Our application of LOOCV estimates the maximum-likelihood parameters for all but one of the participants and then uses these to calculate the likelihood of the data for the remaining participant. In essence, this is estimating parameters from $k-1$ participants and predicting the kth participant. LOOCV rotates this process through all k participants. One model is to be preferred over another if it better predicts the kth participant. Note that this approach focuses on predictions that generalize across participants rather than predictions that generalize within participants, which has been more typical of MVPA approaches. This focus and the decision to use LOOCV rather than other cross-validation methods reflects our goal of using what we learn from a set of participants to understand/predict a new individual. This goal is central to the educational applications that set the general context for our research.

A more complex model (more states or more conditions) is to be preferred over a less complex model if it results in better predictions for significantly more participants. The probability that the

[4] This achieves better performance than allowing both index and scale to vary as in Anderson and Fincham (2014). If we interpret the gamma distribution as a convolution of exponential distributions, then allowing only the index parameter to vary can be interpreted as a state varying in number of steps as a function of condition. However, only slightly worse results are obtained if we assume a single index parameter and allow the scale parameter to vary with condition.

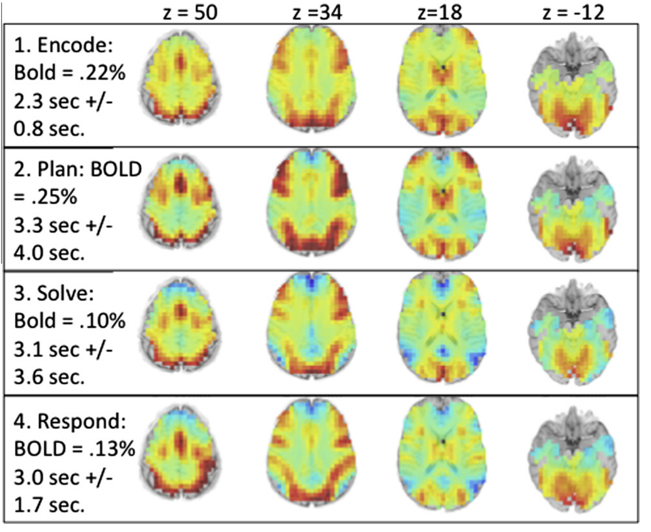

Fig. 6. The brain signatures of the 4 states calculated from the 20 principal component scores. Dark red reflects 1% above baseline while dark blue reflects 1% below baseline activity. The z coordinates for a brain slice (radiological convention: image left = participant's right) is at $x = y = 0$ in Talairach coordinates.

more complex model will better predict x of k participants by chance is upper-bounded by the probability of getting x out of k successes in a binomial process with probability .5. Thus, we have used a simple sign test for determining when the model has gotten complex enough.[5]

Fig. 5 summarizes a search through a space of models varying from 1 to 8 states. We considered models that assumed all problems had the same distribution of times in a state, models that had different distributions of times for each of the 10 conditions in Table 1, and models that had different distributions of times for each of the 128 problems. Fig. 5 shows the results of the search measured in terms of the mean log-likelihood of the left-out participant. These are all expressed in terms of average improvement over the simplest case (a single state and a single condition). For any of the choices for number of states, the best model involves estimating a different distribution of times for each of the 128 problems. As we will see, there are not 128 distinct distributions of times within each state, but rather distinctions within the 10 problem types in Table 1 for some of the states.

The best model involves a sequence of 5 states with 128 time distributions. It is better than the closest competitor (4 states with 128 distributions) for 57 of the 75 participants. We were surprised to find 5 states because earlier work on these problems (Anderson & Fincham, 2014) had indicated that a 4-state solution was optimal. We attribute the finding of 5 states to more precise parameter estimation resulting from the greater number of participants and the finer brain grid. However, as discussed in Appendix A, this 5-state model just creates a new transition state to better fit the scan of transition between states 3 and 4 in the 4-state model and reflects no theoretically interesting variation.[6] Therefore, we will take the 4-state 128-distribution as the reference solution for further discussion (which is significantly better than any simpler model in Fig. 5 and only inferior to the 5-state 128-distribution model). Following the labels established in Anderson and Fincham (2014) we call these 4 states the Encoding State, the Planning State, the Solving State, and the Responding State, although the ACT-R model to follow will provide a more precise interpretation of these states than implied by the labels.

[5] Note that this claim about a binomial test and LOOCV is not general. It does apply when normal data from a single distribution is randomly divided into two groups and we ask whether LOOCV within groups predicts better than LOOCV means estimated from both groups.

[6] Also, as noted in Fig. 2, the 5-state 128-condition solution l is not superior to a 4-state solution based ACT-R model that we will describe.

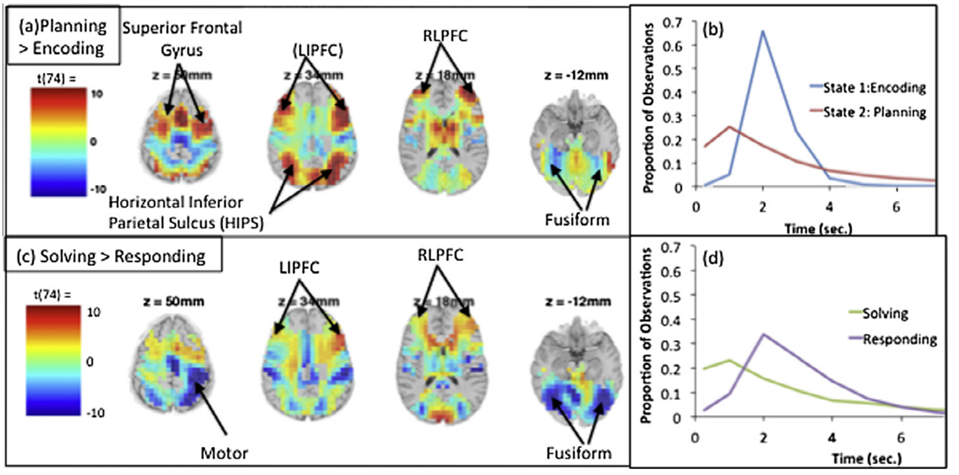

Fig. 7. (a) Values of t's contrasting activity in the Planning State minus the Encoding State. The z coordinates for a brain slice (radiological convention: image left = participant's right) is at $x = y = 0$ in Talairach coordinates. (b) Proportion of trials with different durations of the Encoding and Planning State. (c) Values of t's contrasting activity in the Solving State minus the Responding State. (d) Proportion of trials with different durations of the Solving and Responding State.

Each state is defined by its mean scores on the 20 PCA dimensions encoding brain activity patterns. We can reconstruct whole brain activation from these 20 means. Fig. 6 shows the activation pattern associated with each of these 4 states in 4 representative slices. While the 20 PCA factor means have a mean inter-state correlation of −.13, these whole brain reconstructions of 8365 regions have an inter-state correlation of +.82. Nonetheless, one can perceive some differences among states in Fig. 6. One that is fairly apparent is that the mean level of activation is higher in the first two states (.22% and .25% above the fixation baseline) than the last two states (.10% and .13% above baseline). Fig. 7a focuses on the difference between the first two, higher-activation states and Fig. 7c on the difference between the last two, lower-activation states. Fig. 7a and c presents only those regions whose mean activity over the whole problem-solving period is significantly (at the .01 level) above baseline. For these regions Fig. 7a plots t-values for the activation difference of State 2 minus State 1, while Fig. 7c plots the t-values for activation difference of State 3 minus State 4.

Regions showing greater activation for the Planning State in Fig. 7a are those that past empirical reports (Anderson, Betts, Ferris, & Fincham, 2011; Wintermute et al., 2012) identified as the "metacognitive network" – the superior frontal gyrus, LIPFC, RLPFC, and the horizontal intraparietal sulcus. Darker blue in Fig. 7a denotes regions significantly more active in the Encoding State than the Planning State. They include regions in the vicinity of the fusiform, which would be expected because participants need to parse the pyramid expression that they have been shown. In Fig. 7c, the Solving State shows greater activation in the LIPFC and RLPFC reflecting the calculations that are being performed.[7] In contrast, the Responding State shows greater activation in the left motor region controlling the right hand, which types the response, and the fusiform, which is monitoring this observable output.

Along with estimating the various parameters, the HMM–MVPA estimates a probability that each scan is in each state (State Occupancies in Fig. 4). One can sum these probabilities over the scans to get an estimate of the time in each state for that trial. Fig. 7b plots the distribution of durations of the Encoding and the Planning States averaged over all problems and Fig. 7d plots the durations of Solving

[7] While more active than the Responding State, these prefrontal activations in the Solving State are significantly lower than in the Planning State.

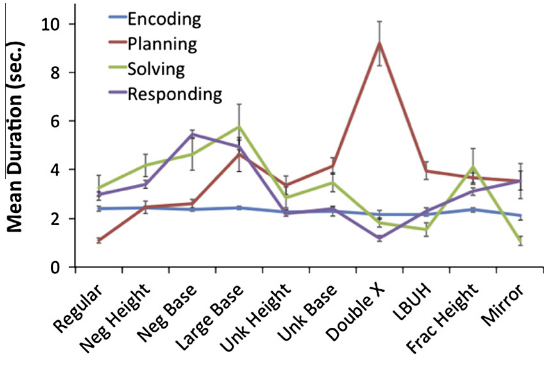

Fig. 8. Mean state durations for the 10 conditions in Table 1. Standard deviations are calculated over different problems in a condition.

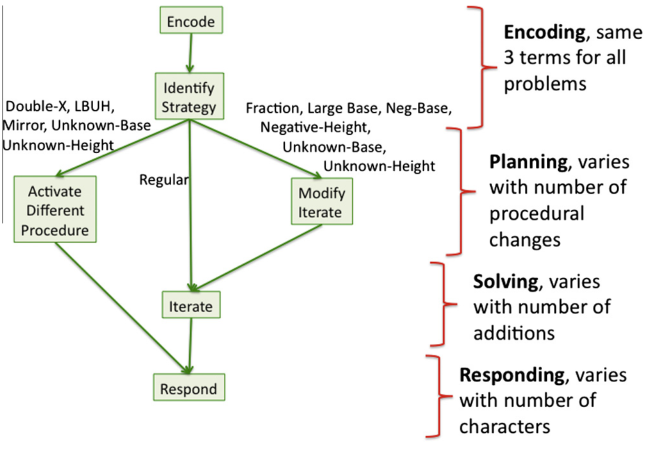

Fig. 9. The overall flow of control in the ACT-R model for solving pyramid problems.

and Responding states. The distributions vary considerably in their spread with the Encoding times being narrowest and the Solving times widest.

While Fig. 7b and d show one distribution for each state, the HMM–MVPA actually estimates a different distribution for each of the 128 problems. Getting separate estimates for each problem might seem disconcerting for purposes of making theoretical generalizations, but it enables a test of whether differences among the 10 conditions (see Table 1 for these conditions) generalize over individual problems. Fig. 8 displays the mean times for these conditions and the standard errors of these means, calculated over the individual problems that constitute a type. The average standard errors of the means are relatively small compared to the differences among the means – .11 s for Encoding, .39 s. for Planning, .48 s. for Solving, and .21 s. for Responding. While the Encoding times show strikingly little difference among conditions, the other states show substantial variability. The Planning State is particularly brief for Regular problems and particularly long for Double X problems. The Solving States are longest for Large Base and shortest for Double X, LBUH, and Mirror. The Responding State is longest for Negative Base and Large Base, which involve keying the most characters (including the negative sign) and shortest for conditions that require keying only a single character (Unknown Height, Unknown Base, Double X, and LBUH).

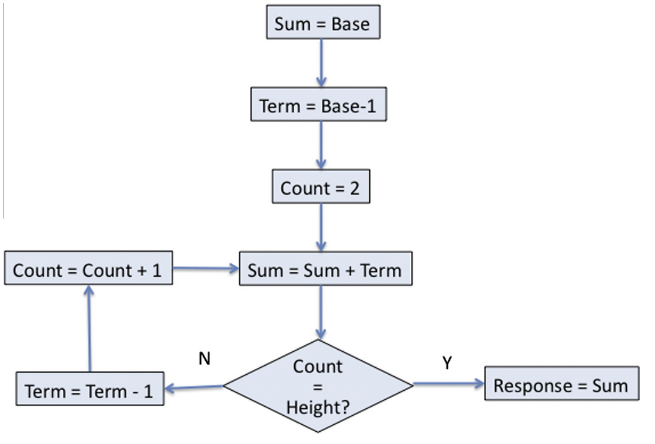

Fig. 10. The flow of control in the ACT-R model of the iterative addition procedure for pyramid problems. Every box corresponds to a declarative instruction.

6. Step 2. Guiding an ACT-R Model[8]

Treating each of 128 problems separately leaves open the question of what can be concluded generally about what participants are doing. To address this question, we developed an ACT-R model, guided by the differences in state durations for individual problems. Like other ACT-R models, this is a "full-task" model that addresses the visual encoding and motor processing as well as the cognitive aspects of the task. This is critical for explaining whole brain patterns of activation, because perceptual and motor activities drive a lot of that activation.

Fig. 9 provides a general overview of the model with different branches for different problem types. The model begins by encoding the terms of the problem and identifying the strategy for solving the problem. The strategy identification involves classifying the base, height, and value as to whether they are negative, their number of digits, presence of a fraction, or a variable. Once the three terms have been classified, the model tries to retrieve a strategy that might be appropriate for that problem. As we will discuss later, this strategy identification is critical to successful solution and errors can result when no strategy is identified or a wrong strategy is chosen. However, the ACT-R model only simulates correct solutions. Thus, it always succeeds in finding a strategy that will solve the problem. The strategy selection process is common to all problems and engages the Metacognitive module. In the case of Regular problems, the strategy selected is simply to use the existing iterative addition procedure that had been practiced the prior day and the Planning state is bypassed. However, for Exception problems there can be a significant Planning state in which that strategy is converted into a procedure that can be used to solve the problem. Finally, all problems end with the generation of the response, which we associate with the Responding State.

Anderson and Fincham (2014) called the third state the Solving State but it has become apparent that it captures a much more specific activity than "solving" in any general sense. In this state participants are engaged in an iterative process of repeatedly adding terms. It is the process involved in solving Regular problems and is part of the process in solving a number of the Exception problems. We will first discuss Regular problems and this iterative addition procedure.

6.1. Regular problems and the iterative addition procedure

The iterative addition procedure in Fig. 10 requires keeping track of the *Term* to add, which is decremented on each iteration, the in-progress *Sum* to which *Term* is added, and the *Count* that keeps

[8] The ACT-R model and instructions on how to run it is available as part of the material available on the website associated with this paper: http://act-r.psy.cmu.edu/?post_type=publications&p=16145.

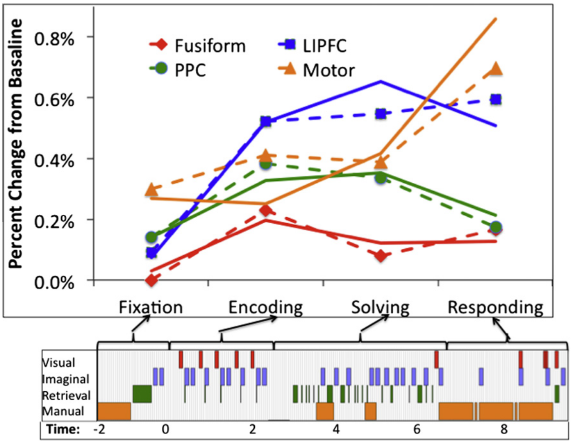

Fig. 11. Top: Average activation (dotted lines connecting data points) for Regular problems and the predictions (in solid lines) based on module activity. The activation values are obtained from the regions as designated by the ACT-R HMM–MVPA analysis. Bottom: The ACT-R swimlane representation of module activity during the solution of a pyramid problem 7$4 = *X*.

track of the number of iterations. Many participants report using finger counting to keep track of the count and this is how it is modeled in ACT-R. This off-loads holding the count from the Imaginal buffer (ACT-R's working memory). The current model uses finger counting for *Count* and so only needs the Imaginal buffer to track *Term* and *Sum*. Each of the boxes in Fig. 10 reflects a declarative representation of a step of the procedure that the model retrieves and interprets as discussed. This contrasts with Fig. 9 where each of the boxes just describes the overall control and does not reflect a specific object in the theory. Appendix B provides a discussion of some of the detail of the representation of the operators and they way they are interpreted by the model.

Fig. 11 illustrates, for Regular problems, how module activity corresponds to activity in associated brain regions. We postpone discussion of the Metacognitive module and the RLPFC until the next subsection on Exception problems where they play a much larger role. The bottom of the figure provides a "swimlane" representation of activity in four modules during the solution of 7$3 = *X* which took the model 8.9 s (participants average 9.1 s). Colored boxes in each lane represent the activity of one of the ACT-R modules. Time moves to the right and each box reflects a period of time when the module was engaged. The lengths of the boxes reflect the duration of the engagement. The Visual module is engaged in encoding of the problem and monitoring of the output. The Imaginal module holds a representation of the problem and keeps track of the accumulating sum and the current term being added. The Retrieval module is engaged in retrieving instructions and retrieving arithmetic facts. The Manual module is engaged in finger counting to track the *Count* and to generate the response. Fig. 11 also shows the 2 s before problem onset. As preparation for the next trial, in this time the model orients the hand to respond, retrieves the pyramid task definition to replace that of the inter-trial letter task, and initializes a template to represent the upcoming problem.

The swimlane representation in Fig. 11 brackets off periods of time corresponding to different states in the HMM–MVPA analysis. These periods correspond to points where the goal changes in the model from encoding to solving and then to responding. Note that within one of these goals, the relative mix of module activity is fairly constant. It is this stable mix of module activity that produces the constant brain signature of a state. Note also that this analysis identifies no period of time corresponding to planning. Thus, while the 128-condition HMM–MVPA estimated brief but non-zero times for planning regular problems (see Fig. 8), the ACT-R model predicts no Planning time. We will see that the ACT-R model leads to better predictions in LOOCV, suggesting that the HMM–MVPA estimate of non-zero Planning time for Regulars may be a matter of overfitting in parameter estimation.[9]

[9] Because the time for a state is lower-bounded by zero, all errors in estimation would be greater than the predicted zero time.

Table 2
Correlations between brain regions and module activity.

(Fixation + 3 states)	ACT-R module			
	Visual	Imaginal	Retrieval	Manual
(a) Regular problems				
Brain regions				
Fusiform	0.948	0.623	0.533	0.214
PPC	0.822	0.949	0.722	0.551
LIPFC	0.797	0.705	0.945	−0.384
Motor	0.356	0.428	−0.033	0.930

(10 states)	ACT-R module				
	Visual	Imaginal	Retrieval	Manual	Metacognitive
(b) All problems					
Brain regions					
Fusiform	0.893	0.473	0.563	−0.104	0.624
PPC	0.676	0.816	0.693	0.151	0.636
LIPFC	0.726	0.701	0.844	−0.580	0.839
Motor	−0.187	−0.084	−0.277	0.942	−0.326
RLPFC	0.758	0.391	0.870	−0.472	0.924

On the other hand, it is possible that some model like the current one but which also had some planning would do even better.

Fig. 11 shows activity in four regions associated with these modules. The left fusiform is associated with the Visual module, the left posterior parietal cortex (PPC) with the Imaginal module, the left lateral inferior prefrontal cortex (LIPFC) with the Retrieval module, and the left motor region with the Manual module. The 4 points on the *x*-axis are the fixation period and the 3 states. In earlier research (see Anderson, 2007, for a review) we had plotted scans along the *x*-axis. However, because of the high variability in how long it takes to solve problems, this raised severe challenges for averaging trials (Anderson et al., 2008). These challenges are avoided if one averages states of constant activity.

Fig. 11 also presents predictions of the ACT-R theory. These predictions are obtained by taking the module activity for a problem and generating a BOLD response assuming the SPM hemodynamic response function. Then, just as with the actual BOLD response from participants, this was deconvolved with a Wiener filter to produce activity associated with the various periods of time. These activities were then multiplied by a parameter (1 parameter for each region) to provide a best fit to the data. Note there is no motor activity during the Encoding period and no visual activity during the solving period. Despite this, this procedure produces non-zero values in these states reflecting averaging of adjacent states. There are two reasons for such averaging: First, state transitions do not line up with scan boundaries, creating border scans. These border scans are a mix of activity of the two states and contribute some of that mix to the estimate of each state. Second, effects spread into adjacent states because of the non-zero noise values used in the Wiener filter to avoid ringing effects (Gonzalez & Woods, 2002). By inspecting the swimlane traces, we could get pure estimates of module engagement in each state, but it is not possible to do the same with our participants. To compare model and imaging data, we subject the model to the same stream of processing.[10]

While the ACT-R predictions involve estimating a multiplicative parameter for each region to scale the ACT-R activity, the basic shape of the predicted activity is parameter free.

Table 2a reports the correlations between all the regions and all the modules. The correlations are much stronger down the main diagonal, confirming the apparent correspondence between region and

[10] It would be nice if we could directly compare deconvolved activity from the data with proportion module engagement from the model. However, because of bleeding between states in the estimation process, deconvolved activity from the model would not match the sharp state boundaries that one can get with module engagement from the model. Therefore, one would not expect deconvolved activity from participants to match well with module engagement.

assigned modules. We will report a stronger test of the correspondence at the end of the next subsection when we include Exception problems and the Metacognitive module.

6.2. Exception problems

The strategies identified for Exception problems either required some change to the standard procedure or a completely different procedure. We used verbal protocols (from pilot participants, not in the scanner) and informal participant interviews to identify some of the procedures that were being used for each of the problem types. There was some variation in how the problems were solved both within and between participants. Some of the alternative procedures involved modest variations on the exception procedure that the ACT-R model implemented, and others were substantially different. It was neither feasible nor possible to represent the complete mixture of procedures used. We believe the implemented procedures represented common procedures, but we do not really know how each of the 75 participants were solving each of their problems. The model assessment in this section involves data averaged over multiple problems that the model solves by different procedures. By averaging these different procedures, we are capturing some of the impact of procedural variation.

As Fig. 9 indicates, for some problems the model modified the well-practiced iterative procedure. This modification involved retrieving the declarative representations of the steps in the procedure that had to be modified, placing them into the Metacognitive buffer, and changing them. The Metacognitive module also raises the activation of these modified steps. The model rehearses the step to increase its activity so that when it gets to where a change has been made in the algorithm it retrieves the modified step rather than the original step (similar to Altmann & Gray, 2008). It was also often necessary to add some additional steps to set up the iterative procedure or to use the results that it produced. Declarative representations of these steps were similarly constructed and rehearsed by the Metacognitive module.

Fig. 12 illustrates the procedure for negative height problems like 7$-4 = *X*. There are a number of possible interpretations that have been proposed for negative height and the experimental software was programmed to recognize all of them. However, in all but one instance in the 408 correct solutions, participants interpreted negative height as counting up – thus, 7$-4 = 7 + 8 + 9 + 10 = 34. Three edits are required to set up this procedure. The first edit, inserted before the start of the iterative

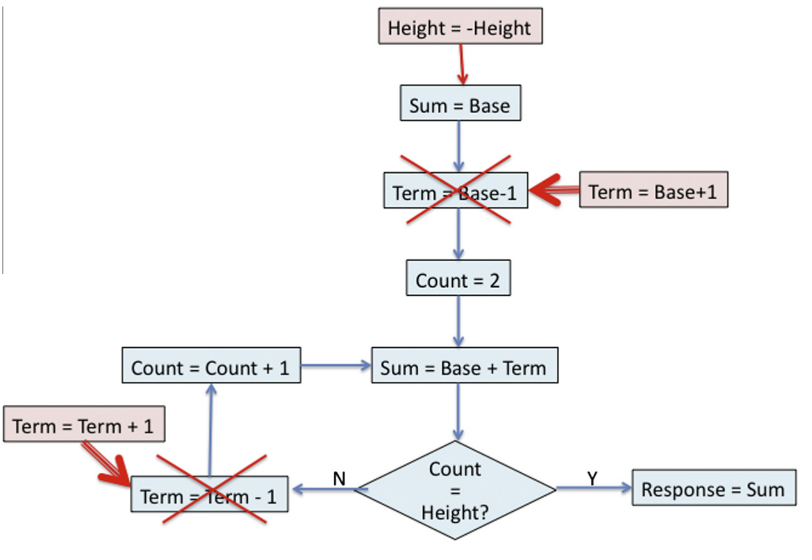

Fig. 12. The edits to the flow of control for negative-height problems. The changes need to be created and their declarative representations rehearsed by the Metacognitive module.

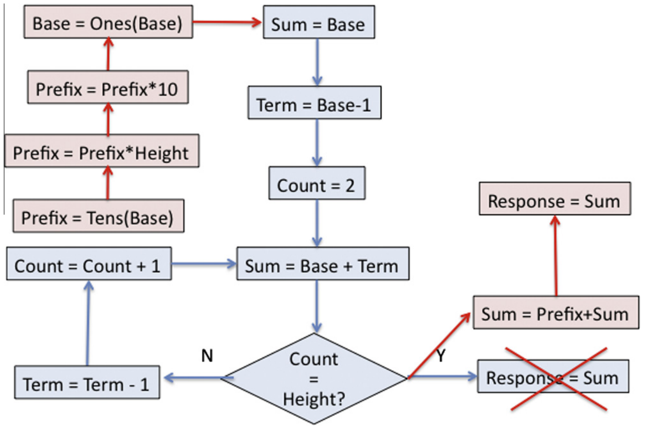

Fig. 13. The edits to the flow of control for large-base problems. The changes need to be created and their declarative representations rehearsed by the Metacognitive module.

procedure, involves changing the sign of the height so that it represents the number of terms to add. The other two edits are to change the step of decrementing the term into a step of incrementing the term. Appendix B discusses some of the detail behind performing such edits. One thing that we hope is made clear in the Appendix is that there is nothing special or privileged about such activity – it is just like any other process in ACT-R except that it involves an additional module.

Fig. 13 illustrates the modifications for large-base problems like 257$3 = *X*. To avoid performing 3-digit addition the model decomposed the calculation as 257$3 = 250 * 3 + 7$3. It calculates 250 * 3 by just multiplying 25 by 3 to get 75 and then appending a 0. It uses the regular iterative addition procedure to calculate 7$3. This problem involves six edits to the basic procedure in contrast to the three changes in Fig. 12 for negative-height. Planning time is largely a function of the number of changes that are needed. Thus, Planning time should be greater for large-base problems than negative-height problems, as is the case in Fig. 8.

The other possibility in Fig. 9 is that the participant will come up with a new procedure that does not involve iterative addition. Fig. 14 illustrates the procedure for large-base unknown height (LBUH) problems like 207$*X* = 618. This procedure just focuses on the hundreds digits and divides them (dividing 6 by 2 in this example) to produce the answer (3 here). Some of the LBUH problems involved a base with that was not close to some multiple of 100 (e.g., 157$*X* = 468) and an extra step of planning is required for these.[11] As Table 1 indicates, LBUH problems have high accuracy and short latency, in contrast to Large-Base problems (Fig. 13). Each of the boxes in Fig. 14 reflects a step of the procedure, which is constructed in the Metacognitive buffer and rehearsed. Planning time is a function of the number of declarative steps, which are either 5 or 6. Since this problem is solved without iterative additions, there is no Solving State. Fig. 8 shows that the 128-condition HMM–MVPA estimates a brief but non-zero Solving time. We will see that the ACT-R assumption of 0 Solving time leads to better predictions in LOOCV.

It is the strategy that is identified at the end of the Encoding stage that guides the construction of a procedure in the Planning stage. ACT-R uses the strategy identified for a problem to index declarative knowledge that describes how to make the changes and additions that are required to construct a successful procedure. We think of these strategies as general schemes that people have developed for adapting procedures. For example, the procedures in Figs. 13 and 14 reflect the result of strategies that many people use with multi-digit arithmetic problems when the numbers allow simplifying decompositions. As another example, a number of the other exception procedures (see Appendix A) reflect

[11] Rounding to the nearest hundred, 157$*X* = 468 is initially solved as 5/2 = 2 and the adjustment involves incrementing by 1.

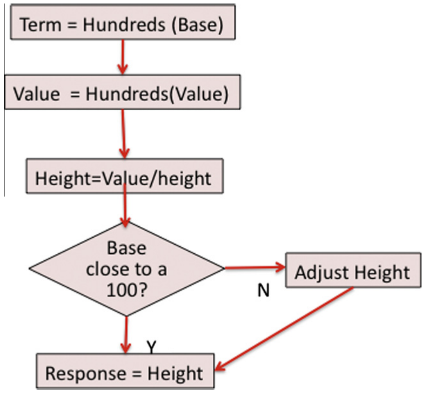

Fig. 14. The procedure for large-base unknown height (LBUH) problems. All the steps of this procedure need to be created and their declarative representations rehearsed by the Metacognitive module.

application of a guess-and-test strategy. While we think of these strategies as general purpose, the actual model specifies the application of these strategies to pyramid problems only.

Mirror problems (e.g., 200$401 = *X*) are the one class of problems for which most students do not have a prior strategy for constructing a solution procedure. The insight to solving these is to recognize that they involve a sequence of positive and negative numbers that will cancel each other out (e.g. 200$401 = 200 + 199 + . . . + −199 + −200 = 0).[12] In a paper and pencil protocol study with 12 participants, Anderson (2007) found 9 could solve their first mirror problem but required over a minute to figure out a solution plan (and so would have timed out in this experiment). Only 20% of our 75 participants correctly answered their first mirror problem in this experiment, compared to 48% first-time correct answers for other Exception problems. The feedback on this problem displays this cancelation and success on mirror problems jumps to over 50% on later blocks. Our analysis is restricted to correct answers and for these the latencies are quite short (less than 9 s on average) reflecting the rather simple procedure in the Appendix that takes advantage of this cancellation strategy. We believe that for this problem, the majority of the solutions involved retrieving a strategy based on the feedback and producing a procedure based on that.

6.2.1. Fitting predefined regions

The bottom of Fig. 15 shows the ACT-R swimlane representation for the large-base problem 105$4 = *X*. The swimlane covers the 2 s of fixation and the 15.6 s of problem solving (participants average 17.4 s to solve this problem). This is a problem that involves all of the Encoding, Planning, Solving, and Responding states. 80 of the 128 problems involve all 4 states in the ACT-R model. There are 24 problems that skip the iterative Solving State as in Fig. 14. Finally, there are the 24 Regular problems, already considered in Fig. 10 that skip the Planning State. Thus, with respect to inclusion of states there are three problem types: Plan&Iterate, Plan only, and Iterate Only. Analyzing separately the non-zero states for the three types of problems produces 10 activity measures. In addition, we calculated the average activation for the fixation period. Fig. 15 gives these activity measures for each region of interest, including the RLPFC, which we associate with the activity of the Metacognitive module. Comparing the observed and predicted activity across the fixation and these 10 states provides a stronger test of the model. Table 2b presents the correlation coefficients between modules and regions. Again, each region is best matched by its associated module. However, it is worth noting that the Metacognitive module almost fits the LIPFC as well as the Retrieval module. These two modules are highly correlated in their predictions ($r = .962$, the next highest inter-module correlation is .703

[12] Not all mirror problems had 0 as an answer. Some stopped one number short of complete cancelation (e.g. 200$400 = 200) and some went one number beyond (e.g., 200$402 = −201).

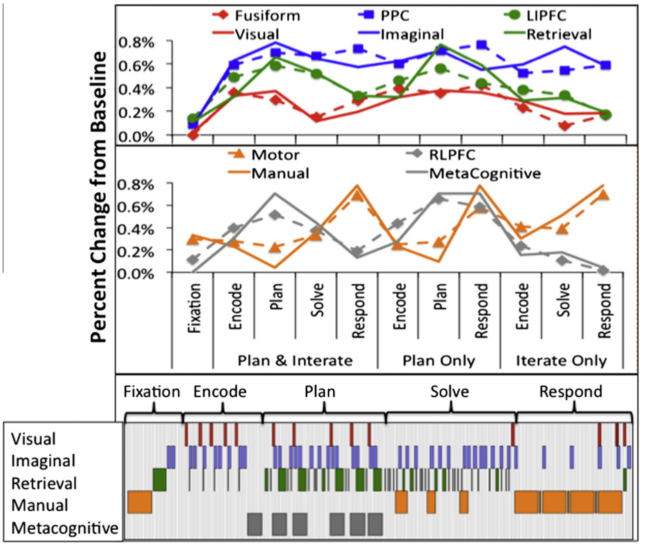

Fig. 15. Average activation for all problems in predefined brain regions as a function of state and the predictions based on module activity. Bottom Right: The ACT-R swimlane representation of module activity during the solution of a pyramid problem 105$4 = *X*.

between Visual and Metacognitive). This is not surprising given that Metacognitive module activity drives a significant amount of Retrieval module activity.

In summary, following the trail marked by the HMM states, we have been able to construct an ACT-R model that is capable of solving the 128 problems. This model is based on ACT-R instruction-following models that have been applied to simpler mathematical problem-solving tasks (e.g., Anderson, 2005). However, the current model contains a major new architectural component, the Metacognitive Module, which plays a significant role in the planning process by enabling the model to create new instructions on the fly. Planning for Exception problems requires modifying or replacing steps in the established procedure for Regulars. The Metacognitive module is responsible for constructing the representation of these new steps. In addition to assure that they will be retrieved rather than the well-practiced steps of the standard procedure, the Metacognitive module is responsible for the rehearsal of these steps. While the Imaginal buffer is responsible for holding the problem representation, the Metacognitive buffer is responsible for holding the procedural representation.

7. Step 3. Refining the mental states

From the ACT-R model, we can obtain time estimates for each state for each problem by noting when the goal associated with that state was active. Fig. 16 presents a comparison of these predictions of the ACT-R model and the estimated state times from the 128-condition HMM. There is 0 correlation for the Encoding State because the ACT-R model predicts no variation in its duration across problems. Correspondingly, the variation is least for this state in the HMM times (standard deviations of .55 s among the 128 means for Encoding, 2.31 s. for Planning, 2.28 s. for Solving, and 1.23 s. for Responding). For the other states there are large positive correlations between the state durations and the ACT-R times although in no case do the correlations approach unity.

Both the ACT-R times and the HMM times were estimated to match the mean time of the different problems. In the case of the HMM times, there were 129 (128 index parameters and 1 scale parameter) $\times$ 4 (states) = 516 parameters. These are more than enough to enable a near perfect correlation (r = .999) with the mean total times to solve the 128 problems. In the case of the ACT-R times, as detailed in Appendix B, there were 5 parameters set to control the retrieval times for different

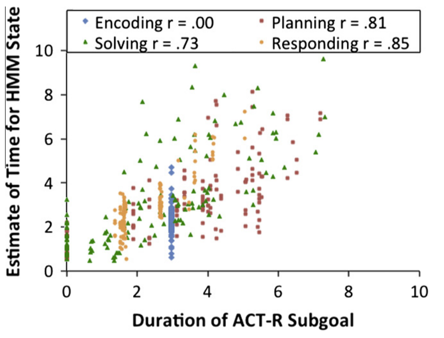

Fig. 16. Plot of the times of the 128-time HMM estimates for the 128 problems against the times in the ACT-R model.

information, a 500 ms estimate for metacognitive updates, a 100 ms estimate for Imaginal updates, and 150% scaling of motor times to reflect the problems of using the keypad in the scanner. As these round values indicate, these parameters were not carefully optimized. Nonetheless, these 8 time parameters served to produce a .89 correlation with mean total times for the 128 problems. ACT-R's average total time over the 128 problems (11.5 s) is closer to the true average (11.3 s) than the HMM (12.4 s). The HMM over-estimate occurs because it fits whole scans and the last scan has some "dead time" after the response is complete. While the ACT-R total times correlate .88 with the HMM total times, it need not follow that particular pairs of state times will correlate. In contrast to the strong correlations displayed in Fig. 16, the average correlation between non-corresponding states is .01.

If the states identified in the HMM analysis do reflect the periods identified by the ACT-R model, the estimates of the state durations for individual problems should be improved by informing the HMM analysis of ACT-R state times for each problem type. So, instead of estimating 128 gamma index parameters per state, one for each problem, we set the index parameters for problems to be a linear function of the ACT-R times: $v = b + c$ * ACT-R-time. We further constrained the gamma scale parameter a such that $a * c = 1$ so that if the index intercept b estimated to be 0, the HMM state times would be exactly the ACT-R times. This is a considerable reduction from the 129 parameters per state in the 128-state HMM. In LOOCV the 128-state HMM necessarily fits the 74 participants better, but the ACT-R HMM more often predicted the 75th participant better (53 of the 75 participants, $p < .001$, with a mean log likelihood advantage of 4.9). The massive reduction in the time parameters appears to avoid overfitting, while capturing systematic variation.

Table 3 reports the estimated values of the a, b, and c parameters for each state. Note that for all but the Responding State the intercept parameter b is estimated to be 0, meaning that the average times for the first three states are the ACT-R times. The non-zero intercept in the Responding State means that its duration is $a * b = .75$ s longer than the ACT-R time. On average, the participants' response completes with .84 s of the last scan remaining. The 0.75 intercept likely reflects this 0.84 s. idle time. Since we only estimate an intercept for the Responding State, this final ACT-R HMM has reduced the 129 * 4 = 516 time parameters for the 128-state HMM to 5 (plus the 8 that control the behavior of the ACT-R model).

Table 3
The ACT-R parameters estimated to infer the distribution of brain state durations from ACT-R model times.

	Scale (a)	Index ($v = b + c$ * ACT-R-time)	
		Intercept (b)	Slope (c)
Encoding	1.26	0.00	0.79
Planning	5.04	0.00	0.20
Solving	3.45	0.00	0.29
Responding	0.91	0.83	1.10

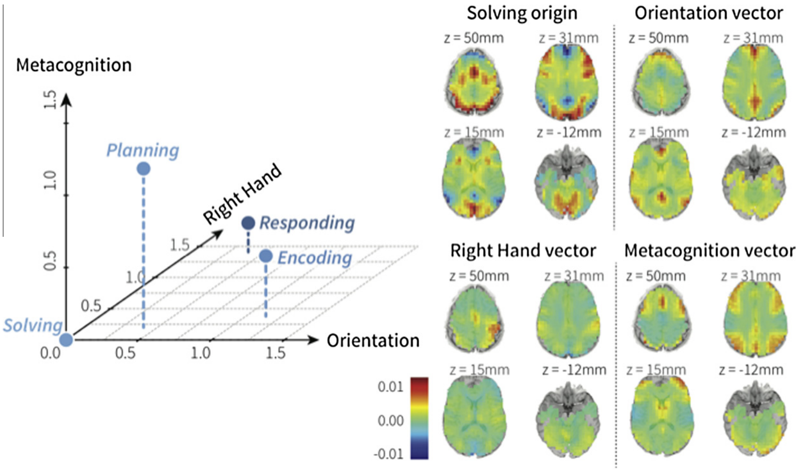

Fig. 17. Projection of the 4 brain signatures identified by the ACT-R HMM into 3 dimensions.

The superiority of the ACT-R HMM over the 128-condition HMM implies that the estimates of brain signatures given by the ACT-R HMM are more accurate.[13] For the remainder of the paper we will use the parameters from the ACT-R HMM as providing a refined definition of the states. While the 4 brain signatures are defined with respect to 20 PCA means and can be mapped back to 8365 activation values, since they are only 4 points they can be represented in a 3-dimensional subspace (just as 2 points can be represented along a 1-dimensional line connecting them in a higher-dimensional space). Fig. 17 provides a representation of the position of the states in that 3-dimensional space, choosing an origin and a set of orthogonal axes to help reveal the nature of these states. To map points in this subspace back to the original space one needs to know the where the origin in the subspace is in the original space. One also needs to know the orientations of the axis are in the original space. Therefore, Fig. 17 also shows the brain reconstructions of the origin and the vectors that define the 3 axes. The ACT-R HMM states can be reconstructed by weighting the vector images by the coordinates of the point and adding these to the origin image.

1. *Origin (solving state):* We set the origin to be the Solving state. This displays a pattern of parietal and prefrontal activation that we have come to recognize as quite typical of routine mathematical problem solving (e.g., Anderson, 2005; Rosenberg-Lee, Lovett, & Anderson, 2009). There is high activation in the region of PPC (mapped to ACT-R's Imaginal module) and the LIPFC (mapped to ACT-R's Retrieval module).
2. *Axis 1 (Orientation):* This vector adds a little activation in visual areas and more activation from default mode regions such a posterior cingulate and polar frontal (Buckner, Andrews-Hanna, & Schacter, 2008). We consider this pattern to reflect external orientation (Mayer, Dorflinger, Rao, & Seidenberg, 2004). The Encoding state is not on this axis, but is relatively close with coordinates of (1.14, 0.35, 0.39).
3. *Axis 2 (Right hand):* This vector adds activation in the vicinity of the left motor region. It seems clearly associated with actions of the right hand. While the Responding state is not on this axis it is close with coordinates of (0.35, 1.38, and 0.16).
4. *Axis 3 (Metacognition):* This vector adds activation in regions that look very much like what we described as the metacognitive network in our empirical reports (Anderson et al., 2011; Wintermute et al., 2012). Particularly noteworthy is activation in the regions of the RLPFC and angular gyrus, but there is also activation in other prefrontal and parietal regions. While the Planning state is not on this axis it is relatively close with coordinates of (0.39, 0.16, and 1.07).

[13] The brain signatures are not that different. The biggest difference is in the Planning state where the two sets of factor scores still correlate .958.

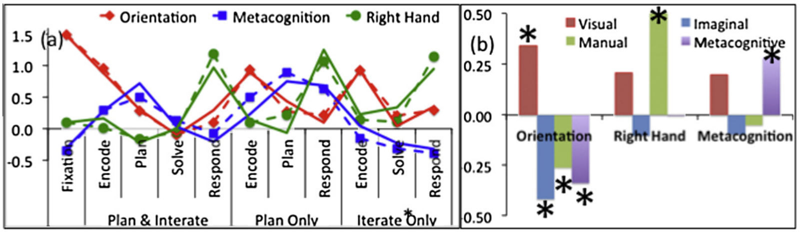

Fig. 18. (a) Average values for the fixation and problem states on the three dimensions in Fig. 17 Smooth lines are fitted results obtained in a regression with ACT-R modules. (b) Weights for ACT-R modules when regressed against the values in (a). Significant (i.e., with 95% confidence intervals that do not include 0) weights are starred.

Fig. 18a shows how the values for the dimensions in Fig. 17 varied across the fixation and problem types, using the same format Fig. 15. Predictably, there is relative little difference for states that are designated to be the same for different problem types. Interestingly, the fixation period is close to zero on the Right Hand and Metacognition vectors while being even greater on the Orientation vector than the Encode state. This reinforces our interpretation of the Orientation vector. While the ACT-R model does not directly make predictions about the values on these three dimensions, one can regress the vector values against module activity and generate predictions as weighted combinations of the modules. Fig. 18a also show the predictions that arise from regressing the Visual, Imaginal, Motor, and Metacognitive Modules (omitting the Retrieval module because of its high correlation with the Metacognitive module). Given that we are estimating 4 weights and fitting 11 points, it is perhaps not surprising that the ACT-R model gives such good predictions. What is more interesting are the regressor weights[14] shown in Fig. 18b. It confirms a strong connection between the Visual module and the Orientation vector, but the Orientation vector has negative weights for all the other modules, consistent the apparent default-mode pattern. The Right-Hand vector has a strong connection the Manual module. The Metacognitive vector weights most strongly on the Metacognitive Module.

8. Step 4. Interpreting the fMRI data

Now we turn to what this imaging analysis can say about two interesting aspects of the experiment that we have ignored to this point. First, 40 of the participants were adults at Carnegie Mellon and the other 35 were children between the ages of 12 and 14. What were the differences between these two populations? Second, we have only considered correct responses. What were the differences between the state characteristics of correct and incorrect answers? To address these questions, we fit the ACT-R HMM to all non-time-out trials, getting an estimate for the state durations of each trial.[15] Because the estimation is not informed by whether or not the participant is a child or whether or not the trial is correct, we are using the same instrument to estimate the state durations for all cases defined by these two dimensions of interest.

In interpreting the results it is good to keep in mind that there can be multiple sources of errors. For instance, consider 207\$4 = *X* (answer: 200 × 4 + 7 + 6 + 5 + 4 = 822). One participant produced 821 and another 622. The first seems produced by an error in addition and the second by an error multiplying the hundreds. However, another answer was 828, which seems the result of 207 × 4 (i.e. a wrong procedure), while another was 654, which seems impossible to diagnose. There were also answers of 5 and 7, which may reflect a participant giving up and just entering a number.

A complication is that different participants saw somewhat different subsets of problems (random problem selection) and, more seriously, answered different numbers of problems correctly.

[14] ACT-R module activity was *z*-scored to give each module equal variability and make the regression coefficients comparable.

[15] Because we were also fitting slower error times we did not use the parameter constraint $a * b = 1$ that was used when we fit the corrects only. Thus, all 3 parameters *a*, *b*, and *c* were estimated for each state. The intercept parameter *b* still estimates to be 0 for the all but the Responding State.

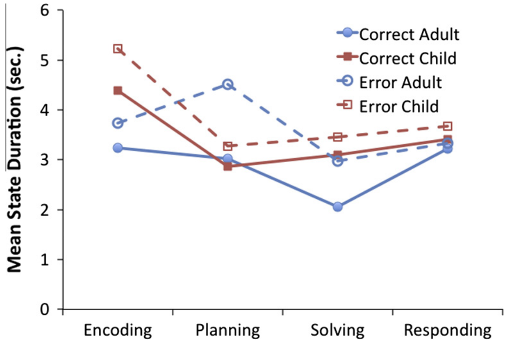

Fig. 19. Mean time per state, plotted separately for corrects and errors and for children and adults.

A participant who was doing well would have an over-representation of hard problems among the corrects compared to a participant who was doing less well. Therefore we regressed out the effects of problem, treating the state time T_{ij} for a participant i on problem j as

$$T_{ij} = M + S_i + P_j$$

where M is the grand mean, S_i is the effect due to the participant, and P_j is the effect due to the problem. The problem estimates P_j were constrained to average 0 over the 128 problems. The estimate of state time for participant i was calculated as $M + S_i$.[16] We used this procedure to estimate 4 state times for correct problems and separately to estimate 4 state times for incorrect problems, thus applying 8 independent estimation processes.

Fig. 19 shows the mean time per state for the four categories. We performed a 3-way analysis of variance on these data where the factors were population (adult or child), correctness, and state. All effects and interactions were highly significant ($p < .001$) except for the correctness-by-population interaction, which was only marginal ($F(1,73) = 2.90$; $p < .1$). The highly significant three-way interaction ($F(3,219) = 6.49$, $p < .001$) suggested it would be useful to analyze separately the effects in each state:

8.1. Encoding

The Encoding time is greater for errors ($F(1,73) = 36.21$, $p < .0001$) and longer for children ($F(1,73) = 29.07$, $p < .0001$) with no significant interaction ($F(1,73) = 2.48$). These effects on Encoding time are a striking contrast to the earlier results of no differences in Encoding times as a function of problem type (e.g., Fig. 8). In the ACT-R model these times include the classification of the problem and identifying a strategy for constructing a solution procedure. The effects in Fig. 19 for Encoding time probably arise from the strategy identification phase.

8.2. Planning

The Planning time is again greater for errors ($F(1,73) = 25.66$, $p < .0001$) but adults take longer than children ($F(1,73) = 5.13$, $p < .05$). This longer time for adults is only for incorrect problems, resulting in a quite significant correctness-by-population interaction ($F(1,73) = 8.24$, $p < .005$). Children had a higher probability of not producing an answer for a problem (7.2% timeouts vs. 3.5% – timeout trials were excluded in this analysis). One explanation is that, on difficult problems, children simply failed to

[16] While this method deals with potential selection artifacts, the results do not change much if we just calculate averages over all correct and incorrect problems for a participant without the correction.

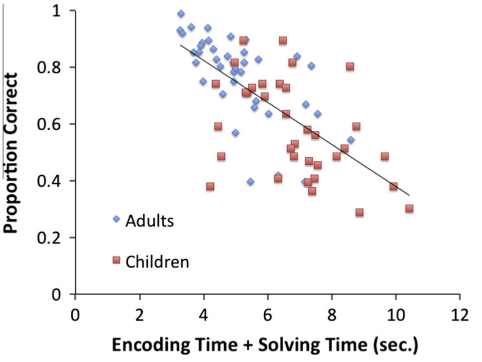

Fig. 20. Accuracy on children and adults as a function of the sum of the encoding and solving time on correct problems.

find a solution strategy while adults came up with strategies that resulted long planning times and incorrect procedures.

8.3. Solving

Solving time is greater for errors ($F(1,73) = 17.22$, $p < .0001$) and longer for children ($F(1,73) = 5.67$, $p < .05$) and the interaction was marginal ($F(1,73) = 3.36$, $p < .10$). It seems likely these effects are driven by the retrieval of arithmetic facts. There is a general speed up in arithmetic retrieval with age (Ashcraft, 1987) and a tendency to make more errors in retrieving arithmetic facts that are retrieved more slowly (Geary, 1994).

8.4. Responding

Errors are slightly longer ($F(1,73) = 5.14$, $p < .05$) but there is neither a significant population effect ($F(1,73) = 1.72$) nor an interaction ($F(1,73) = 1.04$). The reason errors are slightly longer is probably because they averaged slightly more characters (1.92 vs. 1.72). The lack of difference between the two groups may just reflect the fact that both underwent the same keypad proficiency training before going into the scanner.

Overall adults are more accurate than children – 77% correct versus 58% correct. We correlated average state times for correct problems with accuracy across the participants. There are quite significant correlations with both Encoding time ($r = -.56$; $t(73) = 5.03$, $p < .0001$) and Solving time ($r = -.53$; $t(73) = 3.60$, $p < .0001$). There are also correlations with Planning time ($r = -.30$, $p < .05$) and Responding time ($r = -29$; $p < .05$) but they are substantially weaker. In a stepwise regression, both Encoding and Solving time make significant ($p < .0001$) contributions after the other is entered. Having entered these two, the other two state times have no significant contribution to add. Covarying out these 2 time variables, the accuracy difference between the two populations becomes 72% versus 64%, which is reduced but still significant ($t(71) = 2.09$, $p < .05$). Fig. 20 shows how well the simple sum of Encoding and Solving times can predict accuracy. The correlation with this sum is $-.60$. The graph reveals that the relationship holds within and between the two populations.

The relationship between speed and accuracy can be related to accessing relevant declarative knowledge in somewhat different ways for Encoding and Solving times. In the case of Solving time, we think the effects can be explained in terms of retrieval of arithmetic facts. For instance, Lebiere (1999) presented his ACT-R model with 20 years of simulated practice on arithmetic facts, accumulating strength with practice. When facts were weak they were not only recalled slowly but also incorrect facts can intrude.

In the case of Encoding time, we think long latencies reflect difficulty in retrieving a strategy for planning a solution procedure. To retrieve an appropriate strategy the student must have some rele-

Table 4
128-Condition HMM planning and solving times (seconds) classified according to whether the ACT-R model predicted the times would be above median or not.

	Model's planning state	
	Short	Long
(a) Participant's estimated planning time		
Model's solving stage		
Short	2.93	3.95
Long	2.73	4.70
(b) Participant's estimated solving time		
Model's solving stage		
Short	2.20	2.97
Long	4.77	5.15

vant knowledge. Consider the protocol of the college student in the introduction who was solving $X\$4 = X$:

"different... Oh, that's interesting...x, 3x + 6"

This student was probably solving this as a successive integer problem (which is a common high school problem) and reasoning (with a sign error) as follows:

$$X + (X - 1) + (X - 2) + (X - 3) = X$$

$$4X - 6 = X$$

$$3X - 6 = 0$$

This is the reasoning pattern in the ACT-R procedure for this problem (see Appendix B). Reconceptualizing this problem as a successive integer problem is easily accessible if one has experience with such problems. Many of the participants (particularly the children) would have had little or no such experience. Their long Encoding times reflect difficulty in retrieving such a basis. If they failed to retrieve a basis, they may give up and perhaps resort to a strategy of just guessing an answer, resulting in a shorter Planning time. This may produce the crossover (Fig. 19) in the incorrect times for adults and children going from Encoding to Planning.

9. Conclusions

Our past experimental studies (e.g. Anderson et al., 2011; Wintermute et al., 2012) had shown that a wide network of regions becomes active when participants are challenged to extend their knowledge to solve a problem. The current research has shown that this pattern is not constant throughout problem solving but is concentrated around the period of time when participants retrieve a solution strategy and plan their procedure based on that. The ACT-R model has codified one sense of "metacognition" from the many things the term can refer to in the study of cognition. The Metacognitive module creates, modifies, and rehearses declarative representations of cognitive procedures.

The work focused on the RLPFC as representative of this metacognitive activity. There are a number of other regions we could have chosen. For instance, the angular gyrus shows similar patterns of activity for pyramid problems. However, it is less reliably activated in other knowledge-extension tasks that we have studied (Lee, Fincham, Betts, & Anderson, 2014) and seems to be associated with a rather wide range of disparate functions (Seghier, 2013). In contrast, the RLPFC seems always activated when participants must extend their knowledge. As reviewed earlier, considerable other research relates the RLPFC to reflective functions. While the functionalities of all ACT-R modules are probably achieved by a distributed set of regions, one can perform stronger tests of the theory if there is a predefined region whose activity is expected to reflect the activity of each module.

In the theory-free exploration of the state space, the best performing HMM was one that treated each of the 128 problems separately. However, the HMM based on the ACT-R model revealed that this apparent complexity was largely driven by three properties of these problems – the number of changed or new steps in planning, the number of iterations of repetitive adding, and the number of characters that had to be entered to produce the answer (Fig. 17). While these factors are specific to our class of problems, they illustrate the principle that the apparent diversity of complex problem solving becomes systematic when analyzed with a cognitive model that addresses in detail how these problems are solved (e.g., Lee & Anderson, 2001).

Critical to the goals of this paper, the individual problems varied in their predicted Planning times and Solving times. We classified the predicted Planning and Solving times as above or below median for the 80 problems for which the ACT-R model predicted both a non-zero Planning time and a non-zero Solving time. These 80 problems divided nearly evenly into the four possible categories: 21 problems had both short predicted Planning and Solving times, 19 had short Planning times and long Solving times, 21 had both long Planning and Solving times, and 19 had long Planning and short Solving times. Table 4 shows the estimated times for the states from the 128-condition HMM (same HMM as in Fig. 16. Since we are using the 128-condition HMM, the estimation is not informed by the model. We did ANOVAs of the estimated Planning and Solving times, using the individual problems to get an effect of variability. With respect to Planning time, the large effect of predicted Planning difficulty is highly significant ($F(1,76) = 44.93$, $p < .0001$) and the effect of predicted Solving difficulty is not ($F(1,76) = 0.71$) nor is the interaction ($F(1,76) = 2.12$). Conversely, with respect to Solving time, the large effect of predicted Solving difficulty is highly significant ($F(1,76) = 35.74$, $p < .0001$) and the effect of Planning difficulty is not ($F(1,76) = 2.12$) nor is the interaction ($F(1,76) = 0.24$).

Through the combination of the HMM–MVPA and ACT-R modeling we were able to analyze the source of individual differences in problem solving. The same model fit both adults and children equally well.[17] Overall, the children took longer but these time differences were concentrated in the Encoding and Solving phases of the tasks. This pattern suggested that the differences in success depend on identifying a solution strategy and successfully performing the calculations. These inferences could not be achieved without combining the HMM–MVPA and modeling approaches.

To repeat the NCTM goal from our introduction (Romberg, 1992), students should be able to "generate new procedures and extend or modify familiar ones". The conclusion of this paper is that this ability depends on having access to a declarative representation of one's procedures and the resources (modeled in ACT-R's Metacognitive Module) to reflect on these procedures and change them. The activity of this module seems concentrated in the planning phase of problem solving. A striking result of this research is that the children showed no deficit in planning relative to adults. While the planning process varied widely among problems, it did not seem to vary between the two populations when the problem was correctly solved (Fig. 19). If the children could retrieve a strategy, they seemed perfectly capable of planning a procedure based on that strategy.

To further illuminate population similarities and differences, an analysis of activity in the ACT-R predefined regions was conducted, averaging for the mean activity in each state for the two populations (see Fig. 1). The analysis was restricted to the 80 problem types that the ACT-R model assumes have all 4 states. In the RLPFC, associated with ACT-R's Metacognitive module, there is a highly significant effect of state ($F(3,219) = 43.04$, $p < .001$) but no difference between adults and children $F(1,73) = .02$) nor is there a significant interaction ($F(3,219) = 1.60$). The absolute difference in the size of the RLPFC response was small (mean of 0.38% for adults and 0.37% for children). In contrast, two other predefined regions (Fig. 1) showed quite significant differences between the populations. Adults show higher activation in the fusiform (0.35% versus 0.14%, $F(1,73) = 12.02$, $p < .001$) which corresponds to developmental trends in the fusiform (Peelen, Glaser, Vuilleumier, & Eliez, 2009). Adults also show greater activation in the PPC (0.71% vs. 0.56%, $F(1,73) = 8.18$, $p < .01$), which again corresponds to developmental trends in this region (Rivera, Reiss, Eckert, & Menon, 2005).

In conclusion with respect to developmental effects, we see no difference between our young adolescents and adults in their ability to construct a solution procedure if they know a strategy for

[17] However, children younger than our population (ages 12–14) might have produced different results.

constructing such a procedure. The differences between the populations were in their Encoding and Solving processes. Children's slower Encoding times reflected the fact that they often did not have as ready access to a strategy for modifying their procedure. Their slower Solving times reflected less practice with basic arithmetic facts. Fig. 20 shows that, controlling for the speed of Encoding and Solving, there is relatively little difference between the populations and that the two populations overlap.

Acknowledgments

This work was supported by the National Science Foundation Grant DRL-1007945, ONR Grant N000140910098, and a James S. McDonnell Scholar Award. We would like to thank Jelmer Borst and Aryn Pyke for their comments on the paper. We would also like to thank Jelmer Borst for his help in constructing Fig. 17.

Appendix A. The 5-state solution

The analysis in Fig. 5 found best solutions for the 128-condition 5-state solution, but we chose to focus on the 128-condition 4-state solution. Table A1 shows the correlations of the factor means for these two solutions. States 1–4 of the 4-state solution correspond to States 1, 2, 3, and 5 of the 5-state solution. State 4 of the 5-state solution appears to be a combination of states 3 and 4 of the 4-state solution. Also the factor scores of State 4 of the 5-state solution can be fit quite well as a combination of the adjacent States 3 and 5 ($r = .91$) – much better than any other factor score can be fit as a combination of its adjacent states.

State 4 in the 5-state solution has a curious temporal distribution. As can be seen in Fig. A1 85% of the cases are at 2 s and 15% are at 0 s. Fig. A1 involves a binning of durations to the nearest half second. As might be expected, the actual times in States 1, 2, 3, and 5 average about .12 s from the nearest half second. In contrast, the durations of State 4 average less than .01 s from exactly 0 or 2 or, very rarely, 4 s. Looking at the state occupancies it is apparent that State 4 is being assigned either zero or one whole scan. The estimation process is capturing the fact that most problems involve a scan during

Table A1
Correlations between the brain signatures of the 4-state and 5-state solutions with 128 conditions.

	5 State solution				
	State 1	State 2	State 3	State 4	State 5
4 State solution					
State 1	0.998	0.079	−0.434	−0.392	−0.259
State 2	−0.028	0.996	0.227	0.120	−0.306
State 3	−0.481	0.112	0.984	0.460	−0.063
State 4	−0.296	−0.231	−0.129	0.842	0.990

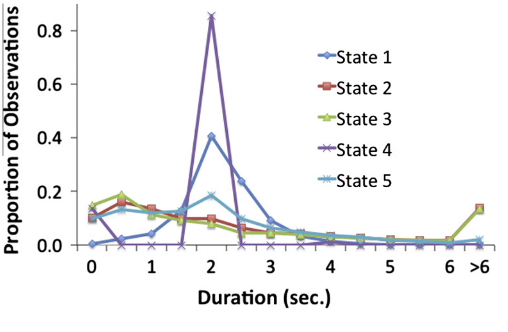

Fig. A1. Proportion of trials involving states of various durations rounded to the nearest half second.

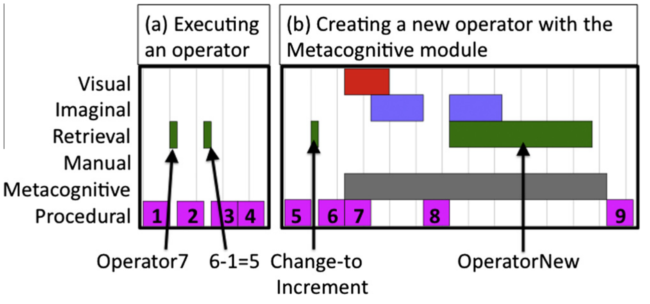

Fig. A2. Swimlane representations of snippet of the model activity when executing a step (a) and when creating a new operator (b). The identities of the declarative retrievals are indicated and the production firings numbered for reference in the text. Each gray bar marks off 50 ms.

which the solution process is completing and the response process in starting. Variance can be maximized by creating a transitional state for this mixture and having low variability in the expected duration of this state. In fact, for the 128-condition model the estimation process estimates nearly zero means for 19 of the 128 problems,[18] 2 scan means (4 s) for 1 problem, and 1 scan means (2 s) for the remaining 108. Because there is no variability in state duration, it does not contribute to an understanding of how state duration varies with problem characteristics. This is one example of the fact that an analysis that maximizes explained variation does not necessarily maximize explanation.

There are two possible interpretations of what might have created this state. The first is that it is an artifact of the fact that the points at which states change do not correspond to the 2-s scan boundaries. The second is that there may be a period of time when motor planning and problem solving are intermixed, perhaps when participants are confirming the answer they calculated. It is not possible to distinguish these two explanations given the temporal sloppiness of fMRI.

Appendix B. Module interaction during operator execution and operator creation

The actual model is available at http://act-r.psy.cmu.edu/?post_type=publications&p=16145. The purpose of this Appendix is to illustrate the interactions among the modules that occur when executing a declarative operator and when creating a new operator. Fig. A2 presents expansions of two small portions of swimlane representations like those in Figs. 11–14. Production firing in the Procedural buffer has been added to Fig. A2 to provide a more complete picture of the module interactions.

B.1. Executing a declarative operator (Fig. A2a)

According to the ACT-R theory of declarative procedure following, learners start out with declarative representations of procedures called operators, which consist of preconditions, primitive actions that ACT-R can perform, arguments for these actions, and post-actions. For instance, the step of decrementing the term in Fig. 10 is represented:

Operator7:
precondition iterate
action decrement
arg1 term
post-action term-updated

[18] These 19 problems have significantly shorter answers (1.26 characters versus 1.68 characters ($t(126) = 2.77$, $p < .01$) and so have less of a response period to divide up.

The operator states that when a decision has been made to iterate again (*precondition iterate*) then decrement then term. Production 1 in Fig. A2 requests retrieval of the operator. In response to the retrieval of this step the following production makes a request to retrieve the digit that is one less than *term*:

Production 2:
If the operator requires decrementing a variable
  And num1 is the current value of the variable
Then retrieve the number one less than num1
  And note retrieving a fact

Fig. A2a illustrates 6–1 = 5 being retrieved in response to such a request. Then the following production harvests the result of the retrieval and makes this the new value of *term*.

Production 3:
If retrieving a fact to change a variable
  and num2 is the value retrieved
Then set the variable to num2
  And note ready to do the next step

Production 4 in Fig. A2a is the same as Production 1 and requests retrieval of the next operator to interpret.

B.2. Creating a new operator (Fig. A2b)

One function of the Metacognitive module is to create operators to replace an operator like the Operator7 above. For instance, in Fig. 12, the step of decrementing is changed to incrementing. The implementation of the change begins by a production noting that the operator to apply at the iterate state needs to be changed:

Production5:
If changing the step at a state
Then try to retrieve a specification for a change at this state

The following chunk is retrieved as the specification of the change:

Negative-Height-Change3:
    State iterate
    new-action increment
    new-arg1 term
    post-action decrement-done

which states to change decrementing of the term to incrementing and then continue on as if it had decremented. When this is retrieved yet another production sends this information to the Metacognitive module to construct a new step:

Production6:
If a state change has been retrieved
Then move the specification to the metacognitive module

As Fig. A2b illustrates, the Metacognitive module is then engaged for 500 ms. In this time it constructs the new a new operator:

OperatorNew:
precondition iterate
action increment
arg1 term
post-action term-updated

The Metacognitive module then rehearses the chunk by repeatedly placing it in the Retrieval module (in Fig. A2b this is shown by the extended period of this chunk in the retrieval module). Productions 7 and 8, which fire during this time, are principally responsible for updating the Imaginal buffer. Since this is the last of the 3 changes for negative height, the last production in Fig. A2b, Production9, starts the actual execution of the procedure.

Appendix C. ACT-R details

The full ACT-R model and instruction for running it are available at http://act-r.psy.cmu.edu/?post_type=publications&p=16145. Here we will discuss the parameters that control the behavior of the model and the exception procedures not described in the body of the paper.

C.1. Parameterization of model

While the parameters for the ACT-R model were not set by a careful optimization process, they were set with the mean problem times in mind. Below we review parameters that were set:

C.1.1. Retrieval module

Retrieval times are exponential functions of activation, A:

$$t = \mathrm{F}e^{-A}$$

The latency scale factor F was set to 0.1 s. Different types of facts were given different levels of activation:

1. The declarative representation of steps of the pyramid procedure supporting information (such as the use of a dollar sign) had activation 2, reflecting their high level of practice.
2. New steps rehearsed by the Metacognitive module were temporarily given an activation level of 4 so that they would be retrieved instead of the regular steps but they then decayed to a sufficiently negative value that they would not be retrieved.
3. Addition facts involving the operands 1–9 where given an activation that reflected the size of the arguments: 3.5 − log(operand1) − log (operand2) – such that *1 + 1 = 2* had an activation level of 3.5 while *9 + 9 = 18* had an activation level of −0.9.
4. Other odd facts (e.g., square root of 25, 33 divided by 5 rounded to the nearest integer) were set to have activations of −1.5. These might better have been modeled with some computational procedure.

Overall, this amounts to 5 retrieval parameters – the latency scale 0.1, the default activation level of 2, the activation level of 4 set by Metacognitive module through rehearsal, the 3.5 as the highest activation value for addition facts, and the −1.5 as the activation level for odd facts.

C.1.2. Metacognitive module

The Metacognitive model spent 0.5 s to transform and rehearse each procedural step.

C.1.3. Imaginal module

Each update to ACT-R's Imaginal buffer (its working memory for partial results) took .1 s.

C.1.4. Manual module

Participants response rate was slower in the scanner, probably because they could not see the keypad and where on their back while using it. To model the slower response times we added 50% to the default motor times in ACT-R.

In addition to these processing times, we needed to calculate predicted BOLD functions. The BOLD response associated with any module is calculated by convolving a hemodynamic response function with a boxcar function that is 1 whenever the module is engaged and 0 otherwise. The hemodynamic response function is the SPM function: gamma(6,1) – gamma(16,1)/6 (Friston et al., 2011).

C.2. Other exception procedures

Fig. A3 presents the other 10 procedures implemented in the ACT-R model for different Exception problems. Here we discuss each briefly. The first 4 are implemented as modifications to the standard iterative procedure and the last 6 are new procedures that bypass the iterative procedure:

(a) *Unknown height:* This is the simplest of all the edits – it just requires changing the test for termination (to be that the sum equal the value) and changing the response (to be the count).
(b) *Fractional height:* The procedure extends the original procedure to add in a fraction of the next term. For example, $6\$3\frac{1}{3} = 6 + 5 + 4 + \frac{1}{3}(3) = 16$.
(c) *Negative base:* To avoid doing signed arithmetic, the procedure switches the sign on the base before iterative addition and then switches it back afterwards. The other changes involve incrementing the term rather than decrementing it.
(d) *Unknown base:* The procedure divides the value by the height and adds 1 to provide a possible value for the base. Then the iterative procedure is applied. If it produces an answer smaller than the sum, the base is incremented again and given as the answer.
(e) *Mirror:* To compute the height at which the positive and negative values will cancel, the procedure doubles the base and adds 1 (e.g. 200 * 2 + 1 = 401). If this breakpoint equals the height, it returns 0. If it is one less it returns the base. If it is 1 more it returns –(base + 1).

There are three possible types of double-X problems and a different procedure is required for each:

(f) *V$X=X:* Because the base is known the procedure runs something similar to the standard iterative procedure.
(g) *X$V=X:* The procedure implements the same procedure as attempted by the student whose protocol was discussed in the main paper. Basically, it represents $X\$V = X$ as $X + (X - 1) + \ldots + (X - (V - 1)) = X$, which leads $aX - b = X$ which leads to $X = b/(a - 1)$. For instance, for $X\$4 = X$ this leads to $X = 6/(4\text{–}1)$. The first 6 boxes in the figure calculate the sum of the first $(V - 1)$ integers, which is b. The next two boxes calculate the $a - 1$. Then the division is performed, the base and value are assigned, and the answer returned.
(h) *X$X=V.* The procedure guesses a height 1 that is more than the square root of the base. It then uses the same procedure as in (i) below for unknown-base problems.

To model the students who guessed the base for unknown-base problems and the height for unknown-height problems, we implemented procedures that bypass iterative addition. The model uses an equal combination of the iterative procedures in parts (a) and (d) and the guessing procedures in parts (i) and (j). The procedures in parts (i) and (j) rely on the insight that b$h is close to but smaller than $b * h$, (where the approximation is proportionately better for larger bases and smaller heights).

(i) *Alternative procedure for unknown base:* For Unknown base, the base is guessed to be 1 bigger than the integer result of dividing the value by the height. If the height is 4 or 5 this is incremented once more.

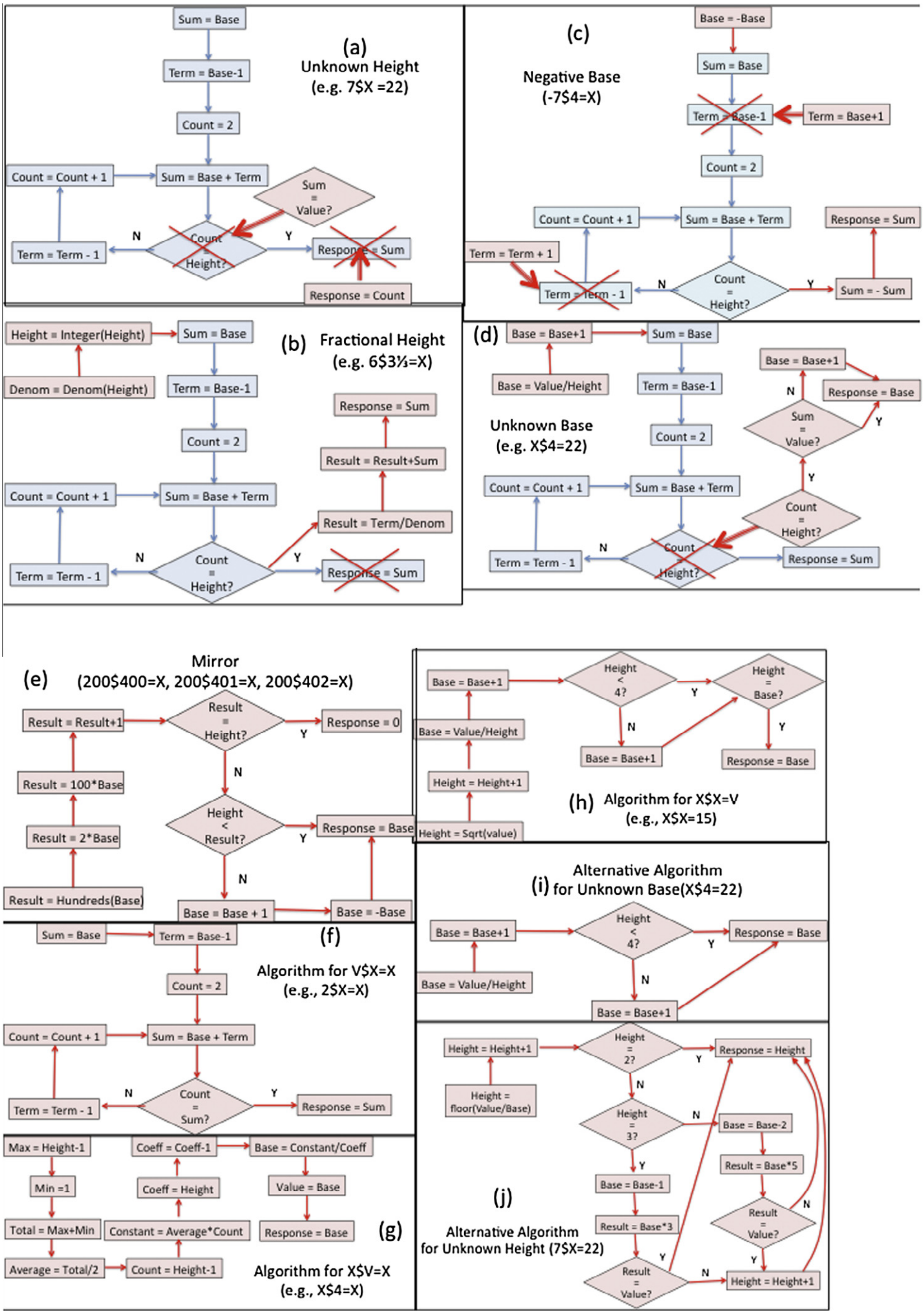

Fig. A3. (a–d) Modifications to iterative procedure for different problem types. (f–j) Alternative procedures for other problems types.

(j) *Alternative procedure for unknown height:* The height is similarly guessed to be 1 more than the integer part of dividing value by height. If the result is 2, the procedure stops and returns 2 as the answer. If the result is 3, the actual height might be either 3 or 4. If it is 3, the value should be 3 times the base minus 1. If the result is 4, the height might either be 4 or 5. If it is 5 the value should be the height times the bases minus 2.

References

Altmann, E. M., & Gray, W. D. (2008). An integrated model of cognitive control in task switching. *Psychological Review, 115*(3), 602.

Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science, 29*, 313–342.

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

Anderson, J. R., & Fincham, J. M. (2014). Discovering the sequential structure of thought. *Cognitive science, 38*(2), 322–352.

Anderson, J. R., Betts, S., Ferris, J. L., & Fincham, J. M. (2011). Cognitive and metacognitive activity in mathematical problem solving: Prefrontal and parietal patterns. *Cognitive, Affective, and Behavioral Neuroscience, 11*, 52–67.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036.

Anderson, J. R., Carter, C. S., Fincham, J. M., Ravizza, S. M., & Rosenberg-Lee, M. (2008). Using fMRI to test models of complex cognition. *Cognitive Science, 32*, 1323–1348.

Ashcraft, M. H. (1987). Children's knowledge of simple arithmetic: A developmental model and simulation. In J. Bisanz, C. J. Brainerd, & R. Kail (Eds.), *Formal methods in developmental psychology: Progress in cognitive development research* (pp. 302–338). New York: Springer.

Badre, D., & Wagner, A. D. (2007). Left ventrolateral prefrontal cortex and the cognitive control of memory. *Neuropsychologia, 45*, 2883–2901.

Bassok, M. (1990). Transfer of domain-specific problem solving procedures. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*, 153–166.

Benoit, R. G., Gilbert, S. J., Frith, C. D., & Burgess, P. W. (2012). Rostral prefrontal cortex and the focus of attention in prospective memory. *Cerebral Cortex, 22*, 1876–1886.

Berger, J. O., Ghosh, J. K., & Mukhopadhyay, N. (2003). Approximations and consistency of Bayes factors as model dimension grows. *Journal of Statistical Planning and Inference, 112*, 241–258.

Bovair, S., Kieras, D. E., & Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. *Human-Computer Interaction, 5*(1), 1–48.

Bransford, J. D., & Schwartz, D. L. (1999). Rethinking transfer: A simple proposal with multiple implications. *Review of Research in Education, 24*, 61–100.

Buckner, R. L., Andrews-Hanna, J. R., & Schacter, D. L. (2008). The brain's default network: Anatomy, function, and relevance to disease. *Annals of the New York Academy of Sciences, 1124*, 1–38.

Bunge, S. A., Helskog, E. H., & Wendelken, C. (2009). Left, but not right, rostrolateral prefrontal cortex meets a stringent test of the relational integration hypothesis. *Neuroimage, 46*, 338–342.

Bunge, S. A., Wendelken, C., Badre, D., & Wagner, A. D. (2005). Analogical reasoning and prefrontal cortex: Evidence for separable retrieval and integration mechanisms. *Cerebral Cortex, 15*, 239–249.

Carraher, D., & Schliemann, A. (2002). The transfer dilemma. *The Journal of the Learning Sciences, 11*(1), 1–24.

Clark, D., & Wagner, A. D. (2003). Assembling and encoding word representations: fMRI subsequent memory effects implicate a role for phonological control. *Neuropsychologia, 41*, 304–317.

Cohen, N. J., Poldrack, R. A., & Eichenbaum, H. (1997). Memory for items and memory for relations in the procedural/declarative memory framework. *Memory, 5*(1–2), 131–178.

Cox, R. W. (1996). AFNI: Software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical Research, An International Journal, 29*, 162–173.

Cox, R. W., & Hyde, J. S. (1997). Software tools for analysis and visualization of fMRI data. *NMR in Biomedicine, 10*, 171–178.

Davachi, L., Maril, A., & Wagner, A. D. (2001). When keeping in mind supports later bringing to mind: Neural markers of phonological rehearsal predict subsequent remembering. *Journal of Cognitive Neuroscience, 13*, 1059–1070.

Dehaene, S., Piazza, M., Pinel, P., & Cohen, L. (2003). Three parietal circuits for number processing. *Cognitive Neuropsychology, 20*(3–6), 487–506.

Detterman, D. L. (1993). The case for the prosecution: Transfer as epiphenomenon. In R. J. Sternberg & D. K. Detterman (Eds.), *Transfer on trial: Intelligence, cognition, and instruction*. Norwood, NJ: Ablex.

Fleming, S. M., Huijgen, J., & Dolan, R. J. (2012). Prefrontal contributions to metacognition in perceptual decision making. *Journal of Neuroscience, 32*, 6117–6125.

Friston, K. J., Ashburner, J. T., Kiebel, S. J., Nichols, T. E., & Penny, W. D. (Eds.). (2011). *Statistical parametric mapping: The analysis of functional brain images: The analysis of functional brain images*. Academic Press.

Geary, D. C. (1994). *Children's mathematical development: Research and practical applications*. American Psychological Association.

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology, 12*, 306–355.

Glover, G. H. (1999). Deconvolution of impulse response in event-related BOLD fMRI. *Neuroimage, 9*, 416–429.

Gonzalez, R. E. W., & Woods, R. E. (2002). *Digital image processing*. New York: Prentice Hall.

Grill-Spector, K., Knouf, N., & Kanwisher, N. (2004). The fusiform face area subserves face perception, not generic within-category identification. *Nature Neuroscience, 7*, 555–562.

Jones, R. H. (2011). Bayesian information criterion for longitudinal and clustered data. *Statistics in Medicine, 30*, 3050–3056.

http://dx.doi.org/10.1002/sim.4323 (Epub 2011 Jul 29).
Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association, 90*, 773–795.
Lebiere, C. (1999). The dynamics of cognition: An ACT-R model of cognitive arithmetic. *Kognitionswissenschaft, 8*(1), 5–19.
Lee, F. J., & Anderson, J. R. (2001). Does learning of a complex task have to be complex? A study in learning decomposition. *Cognitive Psychology, 42*, 267–316.
Lee, H. S., Fincham, J. M., Betts, S., & Anderson, J. R. (2014). An fMRI investigation of instructional guidance in mathematical problem solving. *Trends in Neuroscience and Education*.
Mayer, A. R., Dorflinger, J. M., Rao, S. M., & Seidenberg, M. (2004). Neural networks underlying endogenous and exogenous visual–spatial orienting. *Neuroimage, 23*(2), 534–541.
McCandliss, B. D., Cohen, L., & Dehaene, S. (2003). The visual word form area: Expertise for reading in the fusiform gyrus. *Trends in Cognitive Sciences, 7*, 293–299.
Peelen, M. V., Glaser, B., Vuilleumier, P., & Eliez, S. (2009). Differential development of selectivity for faces and bodies in the fusiform gyrus. *Developmental Science, 12*(6), F16–F25.
Reichle, E. D., Carpenter, P. A., & Just, M. A. (2000). The neural basis of strategy and skill in sentence–picture verification. *Cognitive Psychology, 40*, 261–295.
Reynolds, J. R., McDermott, K. B., & Braver, T. S. (2006). A direct comparison of anterior prefrontal cortex involvement in episodic retrieval and integration. *Cerebral Cortex, 16*, 519–528.
Rivera, S. M., Reiss, A. L., Eckert, M. A., & Menon, V. (2005). Developmental changes in mental arithmetic: Evidence for increased functional specialization in the left inferior parietal cortex. *Cerebral Cortex, 15*, 1779–1790.
Romberg, T. (Ed.). (1992). *Mathematics assessment and evaluation: Imperatives for mathematics educators*. SUNY Press.
Rosenberg-Lee, M., Lovett, M., & Anderson, J. R. (2009). Neural correlates of arithmetic calculation strategies. *Cognitive, Affective, and Behavioral Neuroscience, 9*, 270–285.
Rugg, M. D., Henson, R. N., & Robb, W. G. (2003). Neural correlates of retrieval processing in the prefrontal cortex during recognition and exclusion tasks. *Neuropsychologia, 41*, 40–52.
Salvucci, D. D. (2013). Integration and reuse in cognitive skill acquisition. *Cognitive Science, 37*, 829–860.
Seghier, M. L. (2013). The angular gyrus multiple functions and multiple subdivisions. *The Neuroscientist, 19*(1), 43–61.
Singley, K., & Anderson, J. R. (1989). *The transfer of cognitive skills*. Cambridge, MA: Harvard University Press.
Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review, 120*(3), 439–471.
Taatgen, N. A., Huss, D., Dickison, D., & Anderson, J. R. (2008). The acquisition of robust and flexible cognitive skills. *Journal of Experimental Psychology: General, 137*(3), 548.
Thompson-Schill, S. L., D'Esposito, M., & Kan, I. P. (1999). Effects of repetition and competition on activity in left prefrontal cortex during word generation. *Neuron, 23*, 513–522.
Volle, E., Gilbert, S. J., Benoit, R. G., & Burgess, P. W. (2010). Specialization of the rostral prefrontal cortex for distinct analogy processes. *Cerebral Cortex, 20*, 2647–2659.
Wendelken, C., Nakhabenko, D., Donohue, S. E., Carter, C. S., & Bunge, S. A. (2008). "Brain is to thought as stomach is to??": Investigating the role of rostrolateral prefrontal cortex in relational reasoning. *Journal of Cognitive Neuroscience, 20*, 682–693.
Wertheimer, M. (1959). *Productive thinking*. New York: Harper and Row.
Wheeler, M. E., & Buckner, R. L. (2004). Functional-anatomic correlates of remembering and knowing. *Neuroimage, 21*, 1337–1349.
Willingham, D. B., Nissen, M. J., & Bullemer, P. (1989). On the development of procedural knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*(6), 1047.
Wintermute, S., Betts, S., Ferris, J. L., Fincham, J. M., & Anderson, J. R. (2012). Brain networks supporting execution of mathematical skills versus acquisition of new mathematical competence. *PLoS One, 7*, e50154 (pmcid:3519473).