ACT-R: modeling the brain

A Cognitive Architecture
“If the brain were so simple we could understand it, we would be so simple we couldn’t.”

Lyall Watson
Overview

- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary

\[ A_i = B_i + \sum W_i S_i + \xi \]
ACT-R:

- stands for Adaptive Control of Thought - Rational
- developed by Anderson at CMU in 1993
- it’s a theory about how the mind works, and also a runnable program, written in lisp
- can be used for general problem solving
- but most useful for modeling aspects of human behaviour
ACT-R

- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary
ACT-R

the architecture

\[ A_i = B_i + \sum W_i S_i + \xi \]

- act-r is a hybrid cognitive architecture
- symbolic level
  - production system
  - facts, rules, goals
- subsymbolic level
  - learning without making it explicit with production rules
  - determines which facts are retrieved, which rules are used, how long the retrieval takes

“best of both worlds”
ACT-R

the architecture  \[ A_i = B_i + \sum W_i S_i + \xi \]

- various parts ("modules") of the brain is involved in cognition
- input from our eyes, movement of our arms, remembering facts stored in our long term memory etc.
- not all of this information available at all time
- buffers corresponding to the modules reflect this, for instance we don’t know all our knowledge at once, or attend to all objects in our visual field at once
The architecture of ACT-R

- Intentional Module
  - Goal Buffer
- Declarative Module
  - Retrieval Buffer
- Visual Module
  - Visual Buffer
- Manual Module
  - Manual Buffer
- External World

Mathematical equation:

$$A_i = B_i + \sum W_i S_i + \xi$$
ACT-R

\[ A'_i = B_i + \sum W_i S_i + \xi \]

- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary
the building blocks

chunks

- declarative memory contains facts, the smallest units of knowledge, in ACT-R this is called a chunk

- a chunk type is defined with the command
  (chunk-type chunk-name slot1 slot2 ...)

- an instance of a chunk is added to declarative memory with the add-dm command
ACT-R

the building blocks chunks

- (chunk-type bird feet flies)
- (ostrich isa bird feet 2 flies no)

the first one is the template or class, and the second is an instance, a single fact, or in act-r terminology: a chunk
ACT-R

the building blocks
production rules

- the central production system contains rules, which makes up the procedural knowledge
- these rules are called productions
- matched against the chunks available in the buffers
- but only one production get to fire in each cycle

Feb. 8th 07  T. Liadal  12/43
ACT-R

the building blocks

production - syntax

- a production consists of a LHS and a RHS
- the LHS must access the goal buffer
- a production is written as
  \[(p \text{ LHS} = \implies \text{ RHS})\]
- LHS and RHS access buffers and matches chunks with conditions
the building blocks production execution

- the production system has a cycle that executes the matching, selection and execution every simulated time unit, typically 50 ms
- a production is thought of as a simple step of cognition
- it’s how act-r models the smallest units of cognitive processing
the building blocks
buffers

- goal, retrieval, visual, manual most common
- buffers correspond to modules, and more modules can be added
- is accessed through \(-\text{buffer-name}\) (modifies), \(+\text{buffer-name}\) (request and indirect purge) or \(-\text{buffer-name}\) (purge)
- the only way to access chunks; retrieval from buffers to the central production system
the building blocks buffers

- Communication with the “outside world” (simulated or real through experiments) happen in the perceptual-motor modules.

- Current object of attention from these modules are also made available in buffers.
ACT-R

processing

- the productions all exist in the central matching-selection-execution system
- the different buffers can contain information
- sequentiality from
  - only one chunk per buffer at a certain time
  - only one production gets to fire per production cycle
variables

- variables are written like \( =\text{var-name} \)
- \( =\text{goal} \) and \( =\text{retrieval} \) are also variables, bound to whatever is in the goal and retrieval buffers
- the first time a variable is used in a production, it gets assigned (bound to) the values of the specified slot from the chunk in the buffer
- if the slot does not have a value, then the pattern does not match
ACT-R

- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary
**ACT-R**

**example: count**

\[ A_i' = B_i + \sum W_i S_i + \xi \]

(\text{chunk-type count-order first second})

(\text{chunk-type count-from start end count})

(\text{add-dm})

(b ISA count-order first 1 second 2)

...

(f ISA count-order first 5 second 6)

(first-goal ISA count-from start 2 end 4)

)

(p start

  =goal>

    ISA count-from

    start =num1

    count nil

  ==>

  =goal>

    count =num1

  +retrieval>

    ISA count-order

    first =num1

  ...
**ACT-R**

**Example: count**

\[ A'_i = B_i + \sum W_i S_i + \xi \]

(P increment

=goal>

ISA count-from

count =num1

- end =num1

=retrieval>

ISA count-order

first =num1

second =num2

==>

=goal>

count =num2

+retrieval>

ISA count-order

first =num2

!output! (=num1)

)

(P stop

=goal>

ISA count-from

count =num

end =num

==>

-goal>

!output! (=num)

)

Feb. 8th 07

T. Liadal
\[ A'_i = B_i + \sum W_i S_i + \xi \]
ACT-R

\[ A'_i = B_i + \sum W_i S_i + \xi \]

- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary
ACT-R

**subsymbolic features**

- learning of probability of goal achievement
- learning of relevance of knowledge at hand
- conflict resolution: which chunks to retrieve, which production to fire
- possible to implement the subsymbolic functionality using production rules, but gets cumbersome and expensive

\[ A_i = B_i + \sum W_i S_i + \xi \]
**ACT-R**

**subsymbolic features**

- parameters determine if they are used, and how, but ACT-R is supposed to be a parameter-free model, not fitting the model to the data.

- act-r often criticized for involving too much parameter tweaking.

- the goal is a adaptive architecture with fixed parameters that are able to predict data from psychological experiments.
ACT-R

**Chunk selection**

activation \( (A) \)

\[ A'_i = B_i + \sum W_i S_i + \xi \]

**Activation** = recency of use + usefulness + noise

- A chunk is selected on basis of its recency of use and usefulness in similar situations
- The noise component adds some randomness to the selection process
production selection
utility

\[ U = P^*G - C + \text{noise} \]

- A production is selected on basis of its likelihood \( P \) to lead to the goal minus the cost \( C \) of execution

- \( P = \frac{\text{Successes}}{\text{Successes} + \text{Failures}} \)

- The goal value \( G \) is a global parameter, cost and probability of success is learned
ACT-R

how it fits together

<table>
<thead>
<tr>
<th>Symbolic</th>
<th>Declarative</th>
<th>Procedural</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrieval of chunks</td>
<td>activation</td>
<td>application of production rules</td>
</tr>
</tbody>
</table>

$A_i = B_i + \sum W_i S_i + \xi$

Feb. 8th 07  
T. Liadal  
28/43
ACT-R

Overview

- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary
ACT-R

well this looks like a standard production system, what makes act-r so special?

\[ A_i = B_i + \sum W_i S_i + \xi \]
ACT-R

unified theory of cognition

- Proposed by Allen Newell in 1987
- Newell argued for the need of a set of general assumptions for cognitive models that account for all of cognition
- Presents a unified way of viewing and modeling all cognitive processes
- act-r proposed as a candidate for UTC
The brain:

- the retrieval from the declarative memory and the processing of chunks in the production system are roughly equivalent to the flow from the cortex to the basal ganglia and back.

- the basal ganglia are thought to be where we acquire new knowledge.
ACT-R

It's thought to be a mapping to how the brain actually works...

\[ A_i = B_i + \sum W_i S_i + \xi \]

**Intentional Module**

**Declarative Module**

**Goal Buffer**

**Retrieval Buffer**

**Visual Buffer**

**Manual Buffer**

**Visual Module**

**Manual Module**

**External World**

Basal ganglia

VL cortex

DL cortex

Parietal

Occipital

Motor

Motor/cerebellum

33/43
but...

- Of course it’s a simplification.
- Direct links between brain regions exist.
- Brain functions not fully understood.
- Not a general agreement in the cognitive psychology community that the human cognitive processes can be modeled as a production system.
ACT-R

- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary
ACT-R

learning
new chunks

- chunks can be acquired from outside world by using for instance the perceptual-motor modules

production compilation

- if two productions are fired in sequence, we can combine them into a new production
- if the new production is better, it will eventually replace the old ones

$A_i = B_i + \sum W_i S_i + \xi$
ACT-R

- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary
ACT-R

USES

- Human-Computer Interaction to produce user models that can assess different computer interfaces
- Education (cognitive tutoring system) to “guess” the difficulties that students may have and provide focused help
- CMU’s cognitive math tutor a commercial success
- Neuropsychology, to interpret brain scan data
act-r and emotion

- Emotion plays a role in problem solving.
- Optimism makes us pursue a certain strategy, pessimism gets us out of local minima.
- act-r can be used to model how emotions affect our problem solving strategies (Belavkin, 2002).
- For instance, we can model an optimist as going from a low goal value and much noise, to a high goal value and less noise, optimism makes us focus on current solving strategy.
ACT-R

- about act-r
- the architecture
- the building blocks
  - chunks
  - production rules
  - buffers
- example: count
- subsymbolic features
  - chunk selection
  - production selection
- learning
- UTC
- uses
- challenges
- summary
ACT-R

challenges

- hard to write plausible models
- one has to be both skilled in writing computer programs and in modeling cognitive processes
- thus confined to a small set of academics
- knowledge of lisp almost required
- documentation is not very good, and the different versions of act-r are not backwards compatible
ACT-R

summary

- act-r introduces a way to think about the brain and the parts of it involved in learning and planning
- it is proposed as a unified theory of cognition
- act-r assumes that the cognitive processes in the brain can be modeled as a production system
- hard to write good models
- not a complete solution to cognitive modeling, but hopes to bring some insight
ACT-R

$A_i = B_i + \sum W_i S_i + \xi$

Questions?