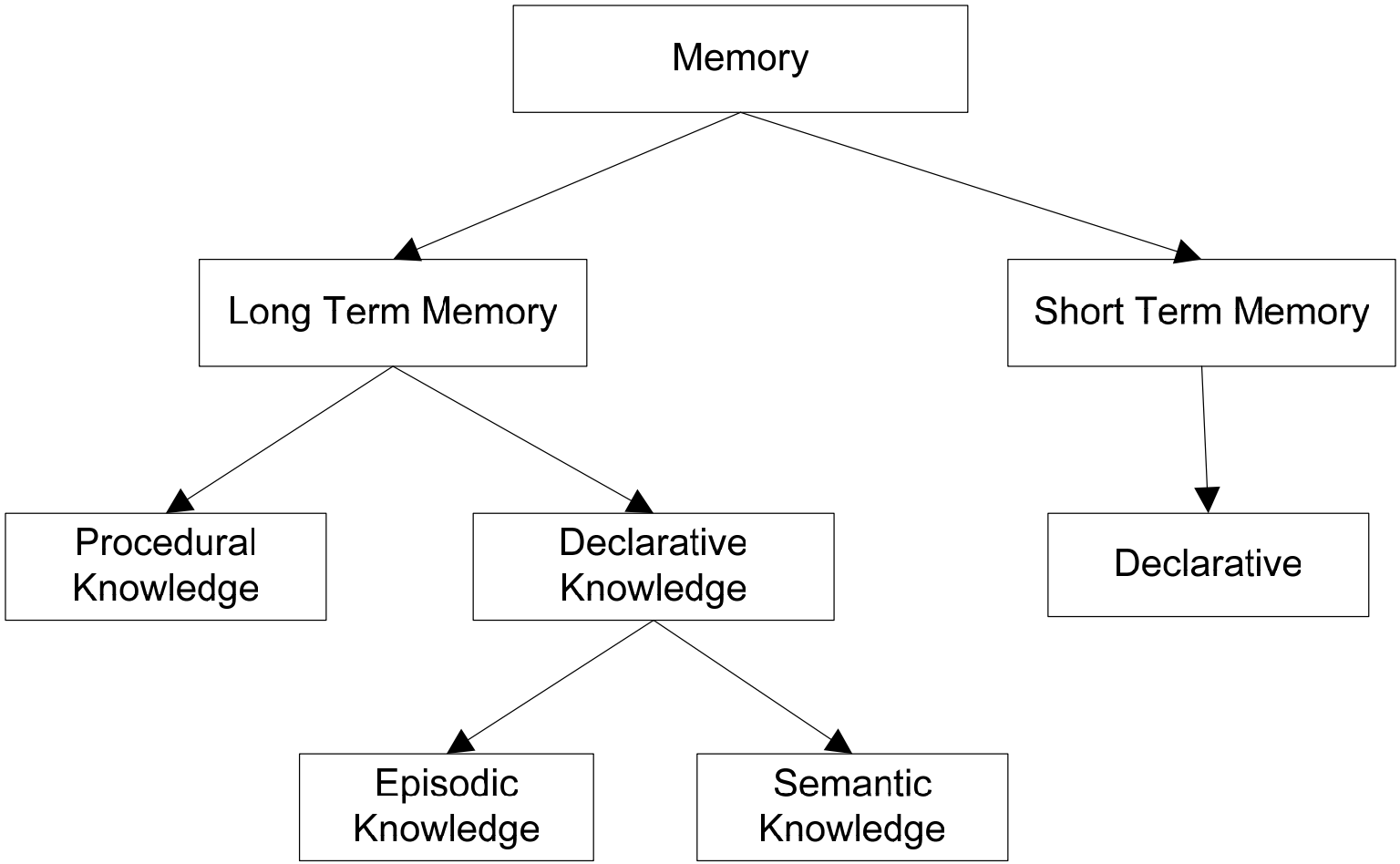
Soar Semantic Memory

Yongjia Wang
University of Michigan

Memory and Knowledge Representation

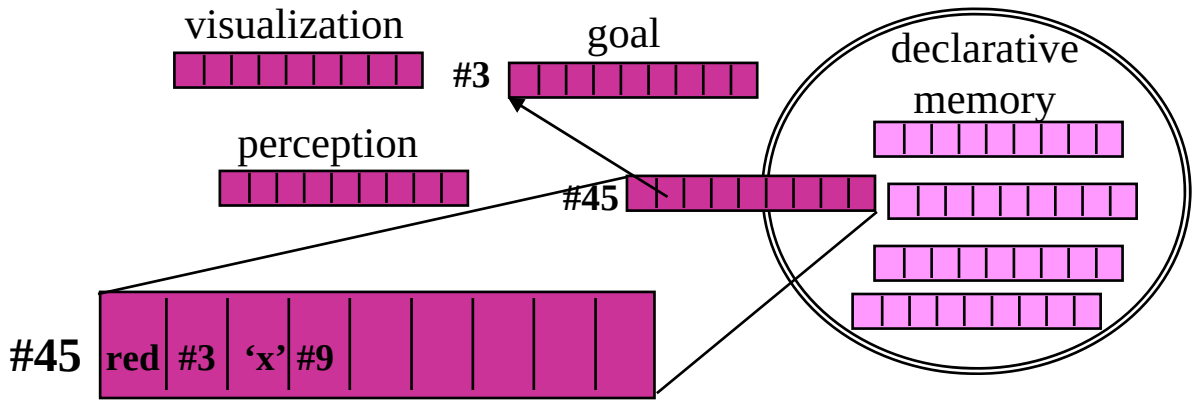


Soar Semantic Memory

- Current status of Soar
  - Long-term procedural memory
  - Only short-term declarative storage
- Data learning problem (*Young, R.M. 2004*)
- Integrate *architectural* declarative memories with Soar
  - Episodic memory (events)
  - Semantic memory (facts)

ACT-R

- The ‘chunk’ declarative data structure
- Buffers holding single chunk
- Long term production memory
- Long term declarative memory



Relevant Soar - ACT-R Differences

- Soar
  - Single generic Working Memory
  - WME structures represent individual attributes
  - Activations associated with individual attributes
  - Complex WM structures, parallel/serial rule firing
- ACT-R
  - Specialized buffers
  - Chunk is the atomic retrieval unit
  - Activations associated with chunks
  - Each buffer holds single chunk, serial rule firing

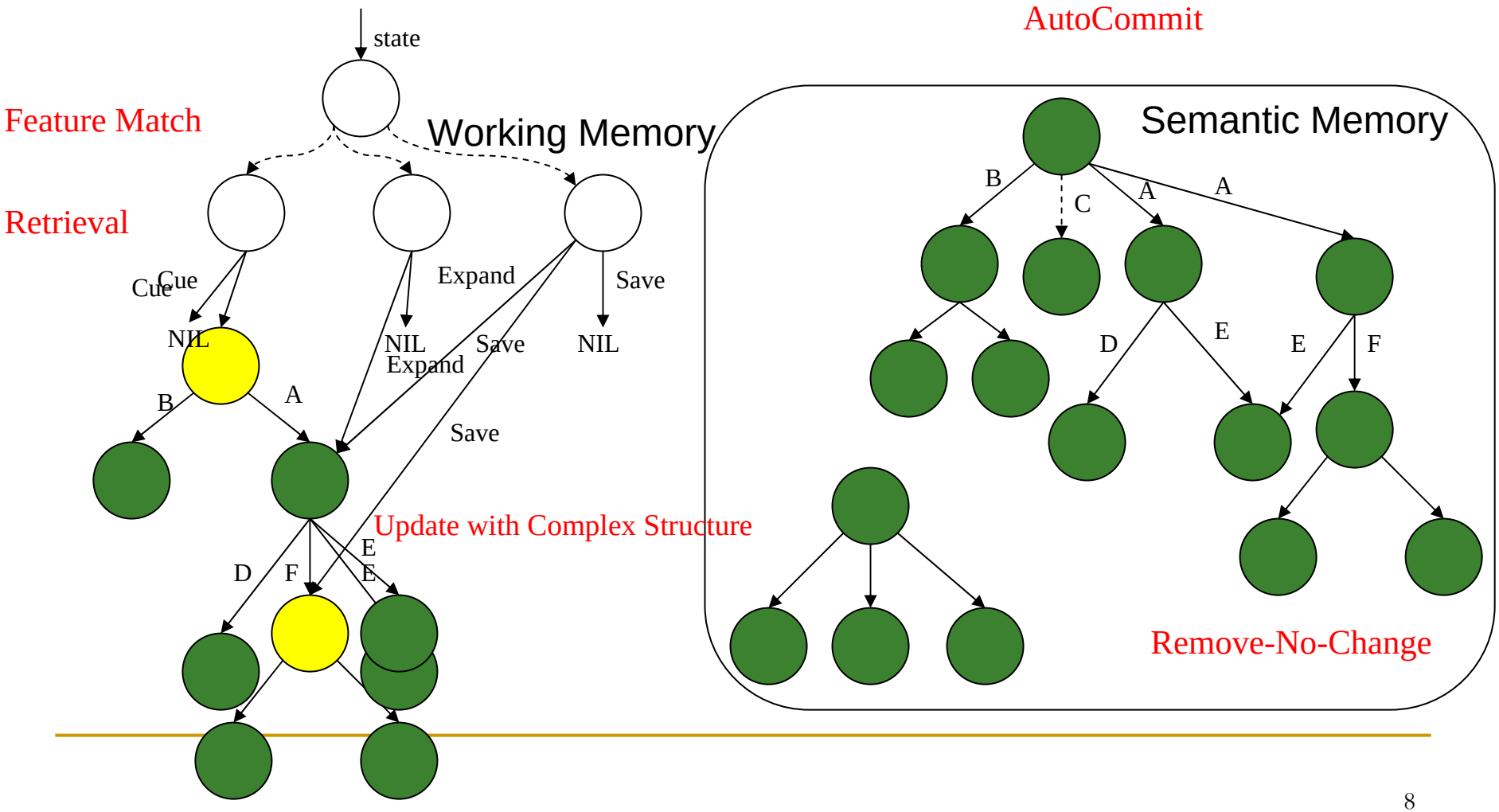
Proposed Changes for Soar (9)

- Long-term declarative storage
  - Long-term identifiers
  - WMEs related by the same identifier equivalent to an ACT-R chunk, long-term identifier equivalent to chunk name
- Uni-valued attributes
  - Soar has been allowing for multi-valued attributes
    - Needed to support old values of attributes
  - ACT-R uses uni-value attribute scheme

Basic Semantic Memory Functionalities

- Encoding
  - What to save?
  - When to add new declarative chunk?
  - How to update knowledge?
- Retrieval
  - How the cue is placed and matched?
  - What are the different types of retrieval?
- Storage
  - What are the storage structures?
  - How are they maintained?

Soar 9 Semantic Memory Functionalities



Encoding

- What to save?
  - WMEs with same identifier as a declarative chunk
  - Individual WMEs as semantic links between chunks
- When to add new declarative chunk(s)?
  - Save all WMEs ever exist
  - Selectively save WMEs
    - Controlled by domain-independent criteria (activation)
    - Controlled by rules deliberately via Save Link

Encoding

- When and how to modify existing chunks?
- Considerations:
  - Updating knowledge doesn't necessarily mean modifying existing chunks
  - 2 options:
    - Overwrite old values by new values
    - Create a separate chunk with updated values
  - Both options seem to have problem

Encoding – Updating knowledge

Make knowledge level decision by rules

- Create a separate new chunk
- Modify a retrieved chunk
  - Add new attributes (WMEs)
  - Can NOT remove existing attributes
  - Change the values of existing attributes
    - Keep old values in long term memory
    - Only one value can be retrieved (uni-valued attribute)
    - Updated value tend to ‘mask’ older values

Retrieval

- How to place a cue
  - Cue Link:

    (state <s>) (<s> .. <cue>)

    -->

    <s>.memory.cue-link.cue <cue>
- What patterns does cue support
  - Instance: long-term identifiers as instance values
  - Variable: temporary identifier represents variable
  - Negated test: need new syntax

Retrieval

- Different types of retrieval
- ‘Global’ search
  - Soar: Cue Link
  - ACT-R: +retrieval> <cue>
- ‘Local’ information access
  - Soar: Expand Link
  - ACT-R: +retrieval> <chunk-name>

Storage

- Conceptually the same structures as WMEs
- Multi-value attributes in long term storage
- Activations maintained for individual attributes (WMEs)
- Long-term identifiers serve as chunk name

Nuggets and Coals

- Nuggets
  - Identified critical issues for semantic learning
  - Proposed a semantic memory design that unifies ideas from Soar and ACT-R
- Coals
  - Not implemented yet