Soar-EpMem Tutorial

Soar Workshop 29

Nate Derbinsky

University of Michigan

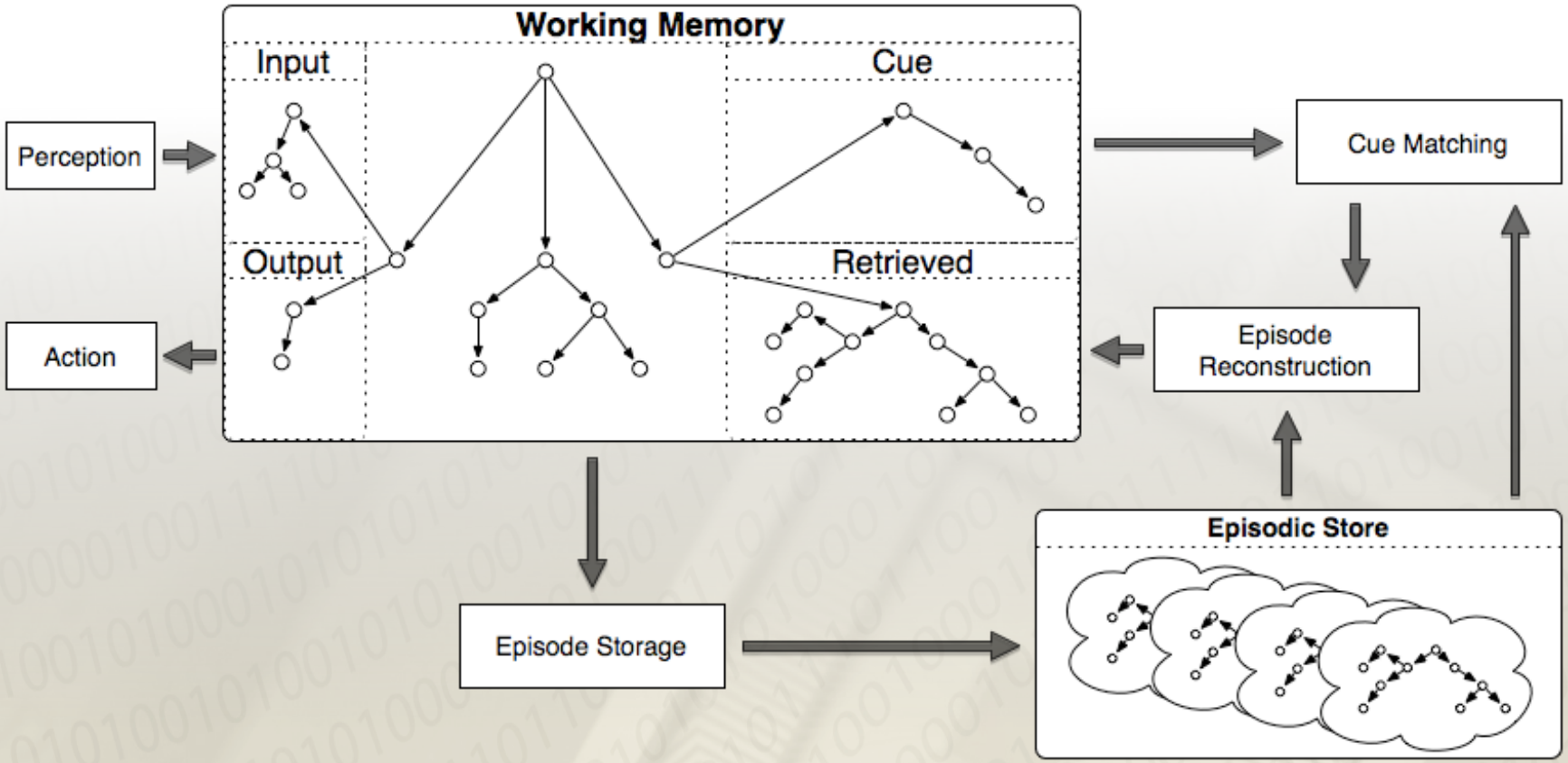
Agenda

- History
- Soar-EpMem as a learning mechanism
- Demo agent
- Episodic storage
- Episodic retrievals
- Architectural details

Some History

- 2004
  - Initial implementation
  - *A Cognitive Model of Episodic Memory Integrated with a General Cognitive Architecture*
    - Andrew Nuxoll, John Laird (ICCM)
- 2007
  - *Extending Cognitive Architecture with Episodic Memory*
    - Andrew Nuxoll, John Laird (AAAI)
  - *Enhancing Intelligent Agents with Episodic Memory*
    - Andrew Nuxoll (Dissertation)
- 2008
  - Re-engineered Soar-EpMem, released as Soar 9.1.0-beta
  - *Efficiently Implementing Episodic Memory in Soar*
    - Nate Derbinsky (Prelim)
- 2009
  - Soar-EpMem refinements (9.1.1)
  - *Efficiently Implementing Episodic Memory*
    - Nate Derbinsky, John Laird (ICCBR)
  - *A Year of Episodic Memory*
    - John Laird, Nate Derbinsky (IJCAI Workshop)
  - *Learning to Use Episodic Memory*
    - Nick Gorski, John Laird (ICCM)

Soar-EpMem: Big Picture



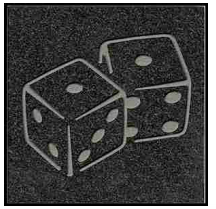
Soar-EpMem as a Learning Mechanism

- EpMem is a weak learning mechanism
  - Automatically stores and temporally indexes agent state
  - Provides the agent an *efficient* content-addressable interface to prior experience

- Supports cognitive capabilities to improve performance on future tasks
  - Virtual Sensing
  - Action Modeling
  - Retroactive Learning
  - …

Demo Task

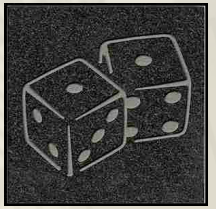
- Produce a random number in WM
  - Soar-EpMem automatically records this episode



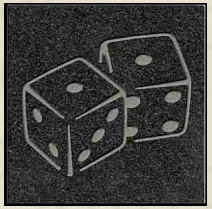
- Remove the number from WM
  - Write to the trace (for later verification)

- Query Soar-EpMem for an episode containing a random number

- Extract (and print) the number from the retrieved episode



Produce a Random Number



```
sp {propose*initialize-epmem-rand
   (state <s> ^superstate nil
              -^name)
-->
   (<s> ^operator <o> +)
   (<o> ^name initialize-epmem-rand)
}

sp {apply*initialize-epmem-rand
   (state <s> ^operator <op>)
   (<op> ^name initialize-epmem-rand)
-->
   (<s> ^name epmem-rand
        ^random.num (cmd rand))
}
```

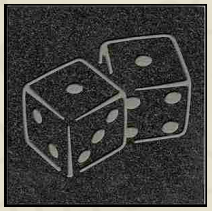
Remove the Number

```
sp {epmem-rand*propose*remember
   (state <s> ^name epmem-rand
              ^random)
-->
   (<s> ^operator <op> + =)
   (<op> ^name remember)
}

sp {apply*remember*cue
   (state <s> ^operator <op>
              ^random <rand>
              ^epmem.command <cmd>)
   (<op> ^name remember)
   (<rand> ^num <num>)
-->
   (write |Removing: | <num>)
   (<s> ^random <rand> -)
   (<cmd> ^query.random <remember-random>)
}
```

Query Soar-EpMem

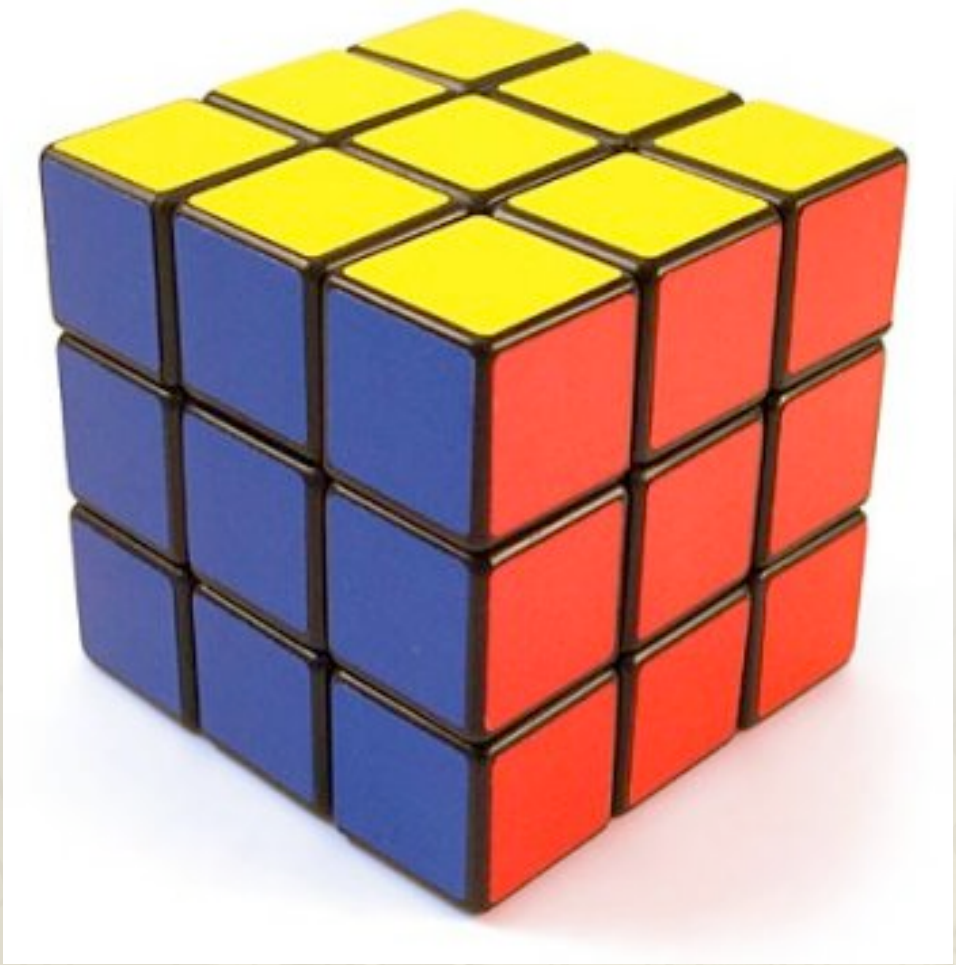
```
sp {epmem-rand*propose*remember
   (state <s> ^name epmem-rand
              ^random)
-->
   (<s> ^operator <op> + =)
   (<op> ^name remember)
}

sp {apply*remember*cue
   (state <s> ^operator <op>
              ^random <rand>
              ^epmem.command <cmd>)
   (<op> ^name remember)
   (<rand> ^num <num>)
-->
   (write |Removing: | <num>)
   (<s> ^random <rand> -)
   (<cmd> ^query.random <remember-random>)
}
```

Report Result



```
sp {done
   (state <s> ^epmem.result.retrieved.random.num <num>)
-->
   (write |Remebered number: | <num>)
   (halt)
}
```

Two-Minute Break



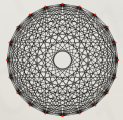
Episodic Storage

- **Who** stores episodes?

- **Why** are episodes stored?

- **When** are episodes stored?

- **What** is stored?



- **Where** are episodes stored?

Who: Architectural Storage

- Soar-EpMem records new episodes automatically
  - Does **not** require deliberate action/consideration by the agent

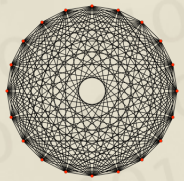
Why: Storage Triggers

- Automatic storage can be **triggered** by
  - **Output** (default)
    - A new WME, rooted at the *output-link* identifier, is added to WM
  - **Decision Cycle**

When: EpMem Phase

- All Soar-EpMem processing (storage and retrievals) takes place at the end of a phase
  - **Output**
    - Default
  - **Selection**

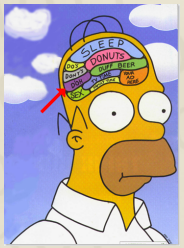
What: Episode Contents



- By default, Soar-EpMem stores the entire contents of the top state of Soar's Working Memory
  - Supports multi-valued attributes and shared identifiers

- For performance reasons, whole branches of WM can be **excluded** from storage by attribute

Where: Episodic Store



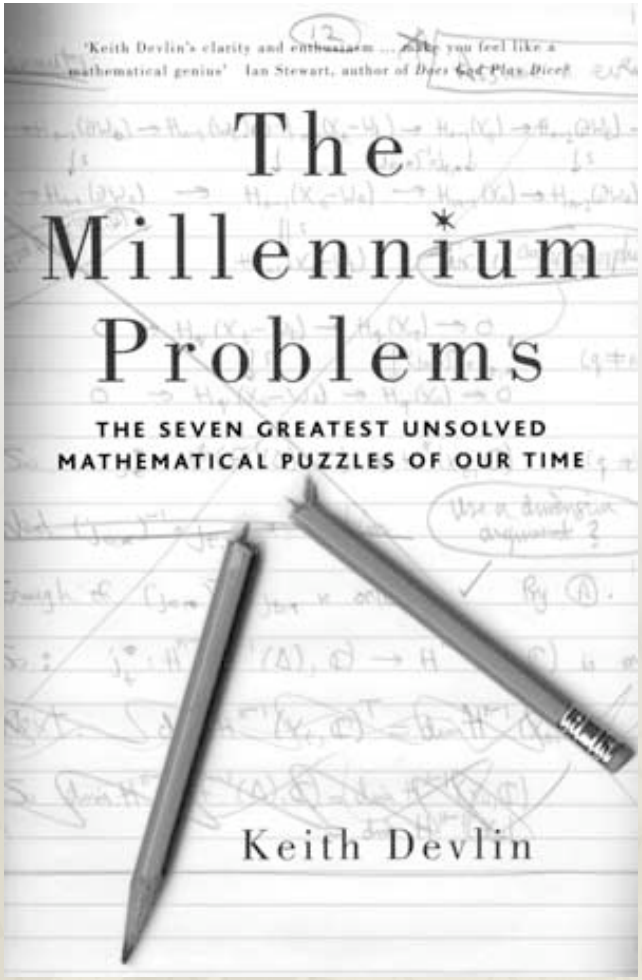
- Soar-EpMem uses the SQLite3 library to manage the episodic store

- The store can be maintained in **memory** (default) or as a **file** on disk
  - If stored on disk, can be opened/queried using any SQLite3 program

A Peek at the Episodic Store

- Load Demo
  - Debugger, source
- Modify Parameter
  - `epmem --set path /path/to/database`
- Run, Quit
- Open Sqliteman
  - File -> Open (`/path/to/database`)

Two-Minute Break



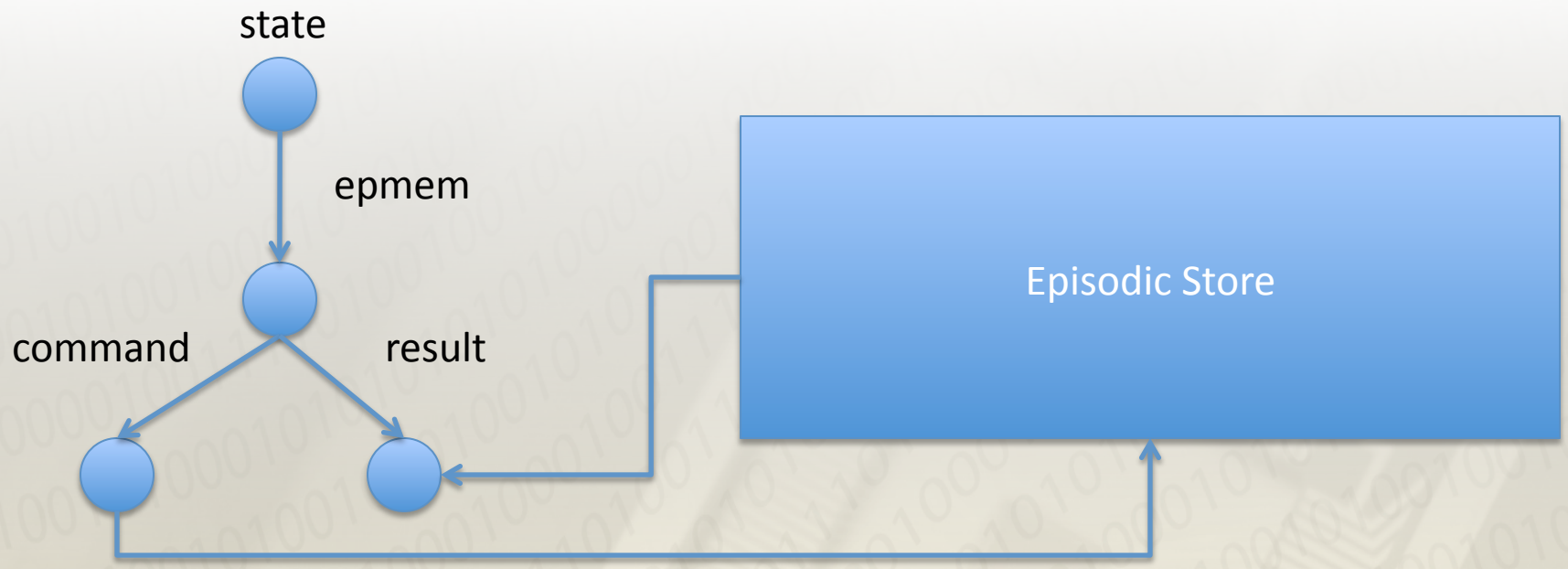
Episodic Retrievals

- Agent API

- Cue-matching details
  - Cue semantics
  - Retrieval meta-data
  - Performance considerations

Agent Interaction

- Agents interact with an **epmem** structure on each state



Command Processing

- Only one type of command can be issued in a single decision cycle
  - Multiple states may issue commands simultaneously
- After a command has been processed, Soar-EpMem will ignore it until some aspect of the **command** structure changes (via addition/removal of WMEs)
  - When a command modification is detected, all WMEs rooted at the **result** identifier are removed from WM

Retrieval Commands

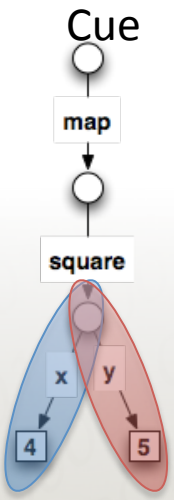
- Retrieve an episode known by temporal id
  - `state.epmem.command.retrieve temporal-id`
- Request a cue-based retrieval
  - `state.epmem.command.query <cue>`
- Retrieve the episode immediately preceding/following the last retrieved episode
  - `state.epmem.command.next <n>`
  - `state.epmem.command.previous <p>`

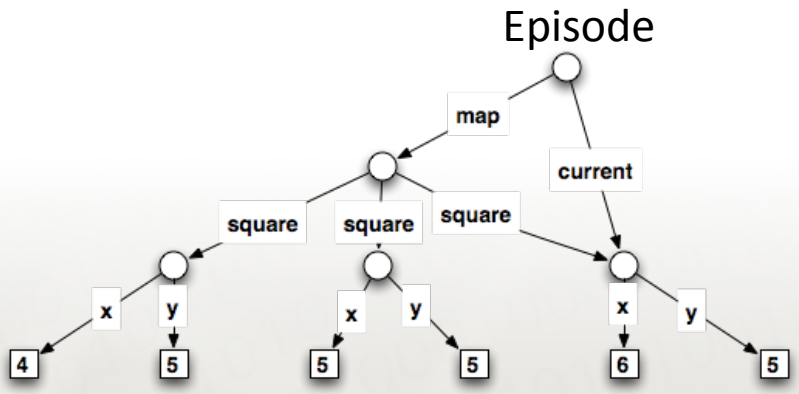
Cue-Based Retrievals

- A cue is an acyclic graph, composed of WMEs that partially describe a **top state** of WM in the retrieved episode

- Cue matching returns the most recent episode containing the greatest number of cue leaf elements

Cue Example





sat(x=4) := (root AND map[1] AND square[1] AND x=4[1])

sat(y=5) := (root AND map[1] AND square[1] AND y=5[1]) OR
(root AND map[1] AND square[2] AND y=5[2]) OR
(root AND map[1] AND square[3] AND y=5[3])

Cue-Based Retrieval Modifiers

- Require that the retrieved episode come relatively **before** a supplied temporal id
  - `state.epmem.command.before temporal-id`
- Require that the retrieved episode come relatively **after** a supplied temporal id
  - `state.epmem.command.after temporal-id`
- Individually **prohibit** one or more episodes from being retrieved
  - `state.epmem.command.prohibit temporal-id`

Retrieval Meta-Data

- **present-id**
- **status**

- Retrieval
  - **retrieved**
  - **memory-id**

- Match
  - **cue-size**
  - **match-score**
  - **normalized-match-score**
  - **match-cardinality**
  - **graph-match**
    - **mapping**

2-Stage Matching Algorithm

1. Evaluate *candidate* episodes based upon relatively inexpensive surface match

2. Perform combinatorial structural match (graph-match via CSP backtracking) ONLY on candidate episodes with a perfect surface score

End search on perfect match or no more episodes.

Performance Considerations (What Makes Cue Matching Crawl)

- Cues forcing linearly increasing episodic store scan
  - "Find me an episode when the current time was both odd and even."
- Cues forcing graph-match backtracking
  - Lots of multi-valued attributes

Two-Minute Break

Soar-EpMem Parameters

- Get a parameter
  - **`epmem [-g|--get] <name>`**
- Set a parameter
  - **`epmem [-s|--set] <name> <value>`**
- Get all values
  - **`epmem`**

Useful Tidbits

- Statistics
- Timers
- Watch
- *kb Demo*

Additional Resources

- Soar-EpMem Manual
  - In the “Documentation” directory
- Soar-EpMem Demo Agent
  - In the “SoarLibrary/Demos” directory
  - *kb*: knowledge-base sequence of unit tests, implements all commands in the agent API
- SQLite: http://sqlite.org
  - Sqliteman: http://sqliteman.com/
- Derbinsky, N., Laird, J.E.: *Efficiently Implementing Episodic Memory*. To Appear: Proceedings of the 8th International Conference on Case-Based Reasoning (2009)