

Do's and Don't's of Episodic Memory

Justin Li

Computer Science and Engineering
University of Michigan
`justinnh@umich.edu`

2011-06-15

Goal of this Talk

1. Give high-level overview of algorithms behind Epmem
2. Suggest techniques for making faster Epmem queries
3. Present ideas for making Epmem more efficient

The Purpose and Interface of Episodic Memory

Purpose

- ▶ Record an agent's experience
- ▶ Capture knowledge not important *a priori*

The Purpose and Interface of Episodic Memory

Purpose

- ▶ Record an agent's experience
- ▶ Capture knowledge not important *a priori*

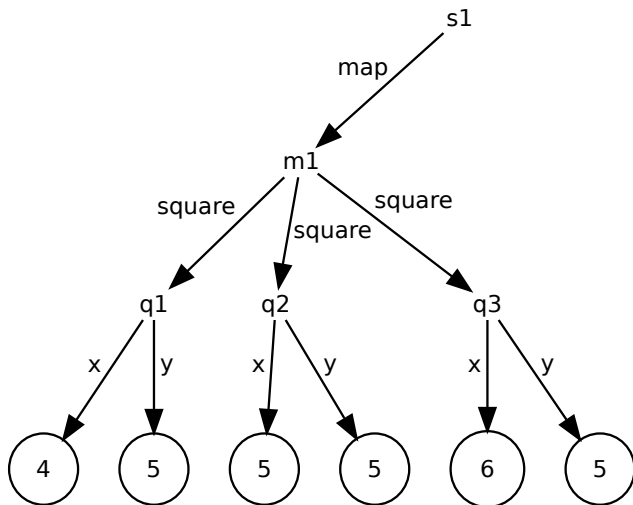
Interface

- ▶ Storage: Automatic
- ▶ Retrieval:
 - ▶ Of a particular episode
 - ▶ Of episodes similar to a *cue*
 - ▶ query
 - ▶ neg-query
 - ▶ before
 - ▶ after
 - ▶ prohibit
 - ▶ Of previous/next episodes
- ▶ Refer to Soar 9.3.1 manual for details

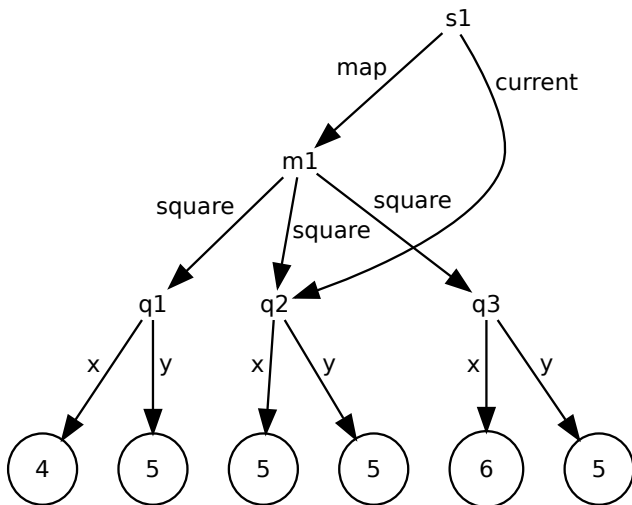
Data Structures and Algorithms by Example

- ▶ Taken from [Derbinsky and Laird, 2009]

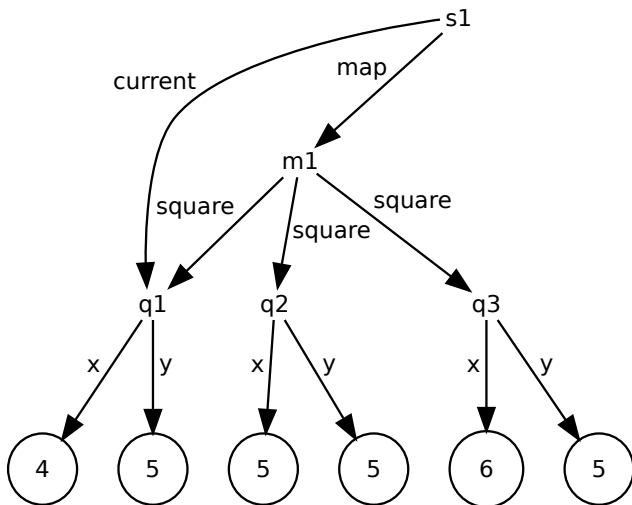
Data Structures and Algorithms by Example



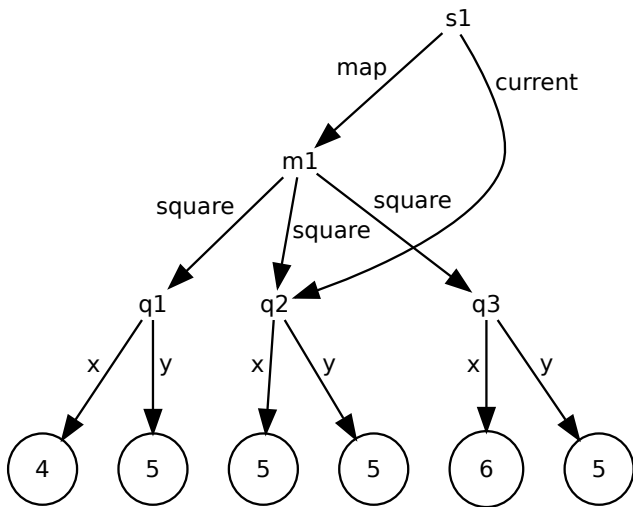
Data Structures and Algorithms by Example



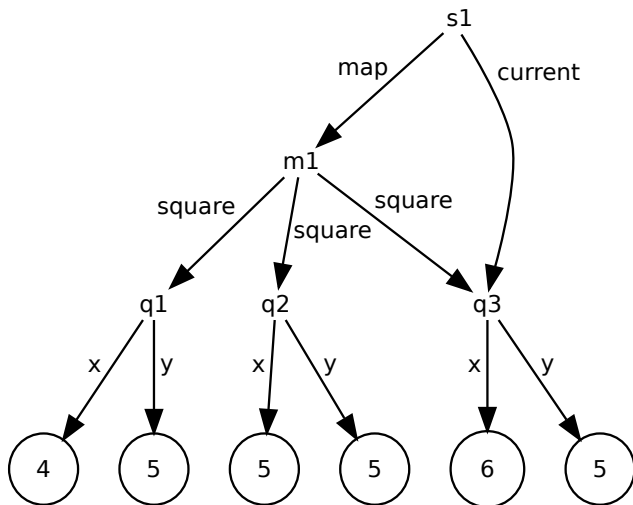
Data Structures and Algorithms by Example



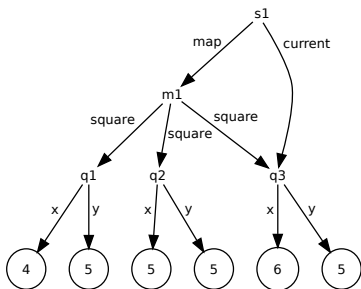
Data Structures and Algorithms by Example



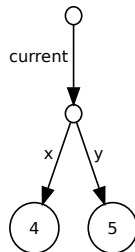
Data Structures and Algorithms by Example



Data Structures and Algorithms by Example

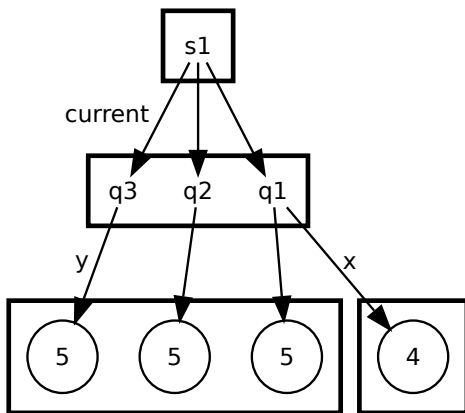


Episode



Cue

Data Structures and Algorithms by Example



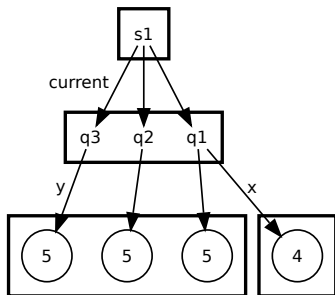
Disjunctive Normal Form (DNF) Graph

Data Structures and Algorithms by Example

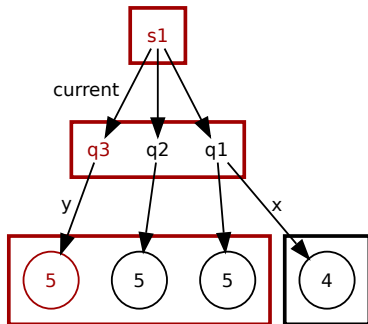


Working Memory Graph

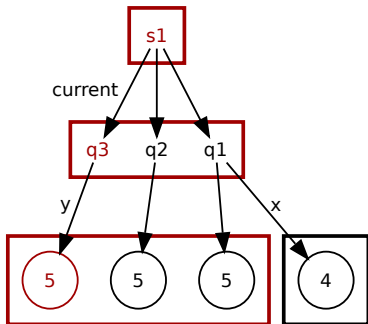
Data Structures and Algorithms by Example



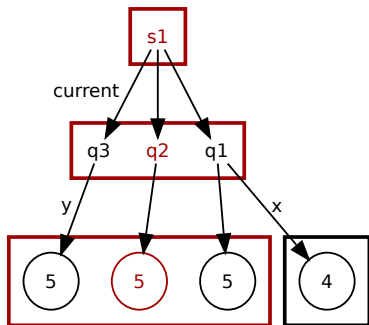
Data Structures and Algorithms by Example



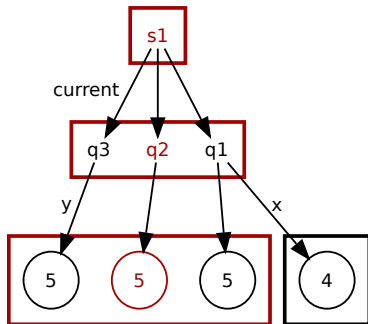
Data Structures and Algorithms by Example



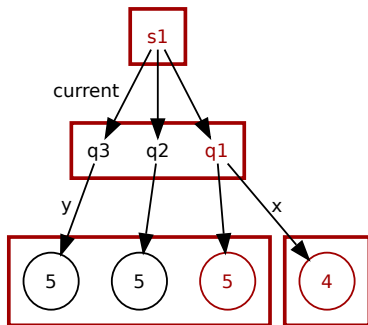
Data Structures and Algorithms by Example



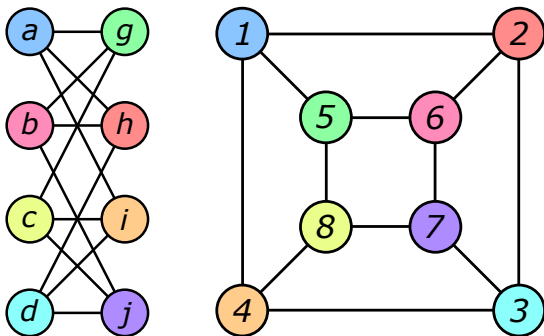
Data Structures and Algorithms by Example



Data Structures and Algorithms by Example



Data Structures and Algorithms by Example



Graph Match

Complexity of Episodic Memory Search

- ▶ Graph match is **NP-Complete**
- ▶ Reduction from the induced subgraph isomorphism problem
 - ▶ Root has edges to all nodes
 - ▶ All edges have the same label
 - ▶ All edges have back-edges

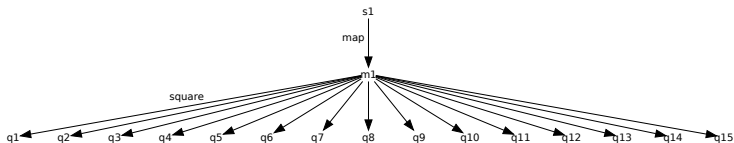
- ▶ Worse-case complexity:

$$O(\|cue\| * \|avg.IDs\| + \|memory\| * k^{\|cue\| + \|episode\|})$$

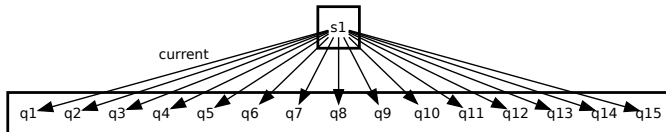
Complexity of Episodic Memory Search

- ▶ Focus on improving best and average case complexity
 1. Delaying work until it's necessary
 2. Reducing the number of intervals to iterate through
 3. Avoiding graph match if possible

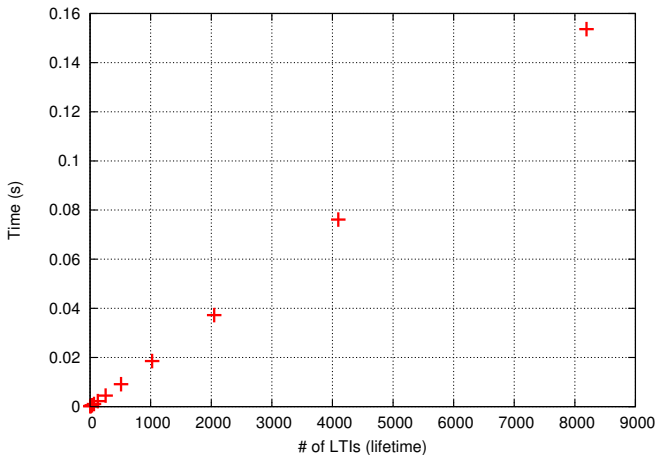
#1: Don't search for LTIs using short-term IDs



#1: Don't search for LTIs using short-term IDs



#1: Don't search for LTIs using short-term IDs



#1: Don't search for LTIs using short-term IDs

► Solutions:

Agent Augment episode with constant WME if important *a priori*

Architecture Incrementally grow the DNF graph, only including WMEs when necessary

► Complexity Reduction:

$$O(\|cue\| * \|memory\|) \Rightarrow O(\|cue\|)$$

#2: Use small, selective cues

#2: Use small, selective cues

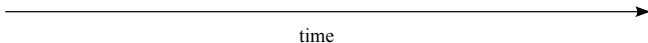
(s1 ^count 2)



time

#2: Use small, selective cues

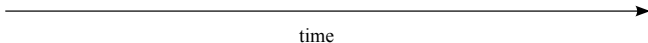
(s1 ^count 2)



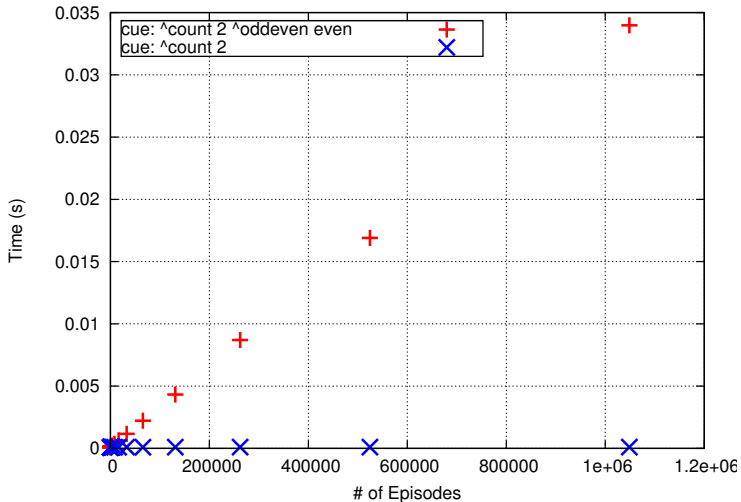
(s1 ^count 2)



(s1 ^oddeven even)



#2: Use small, selective cues



#2: Use small, selective cues

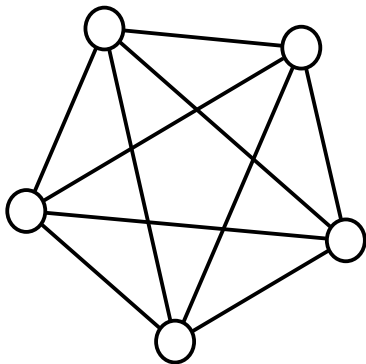
- ▶ Solutions:

Agent If two WMEs always appear together, use the more selective one to query episodic memory

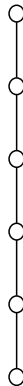
- ▶ Complexity Reduction:

$$O(\|cue\| * \|memory\|) \Rightarrow O(\|cue\|)$$

#3: Avoid multi-valued attributes

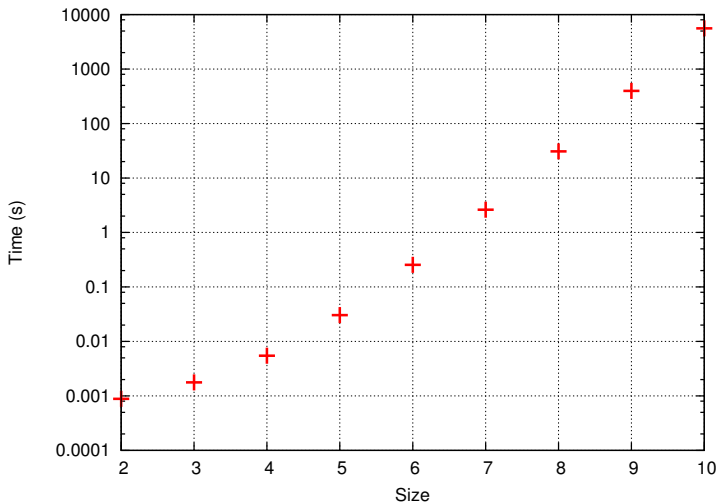


Episode:



Cue:

#3: Avoid multi-valued attributes



#3: Avoid multi-valued attributes

► Solutions:

Agent Keep the number of multi-valued cues small

Agent Use elaborations to make WMEs specific

Architecture Create stricter filters for performing graph match

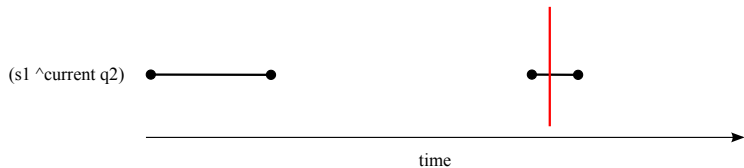
► Complexity Reduction:

$$O(\|memory\| * k^{\|cue\| + \|episode\|}) \Rightarrow O(\|memory\|)$$

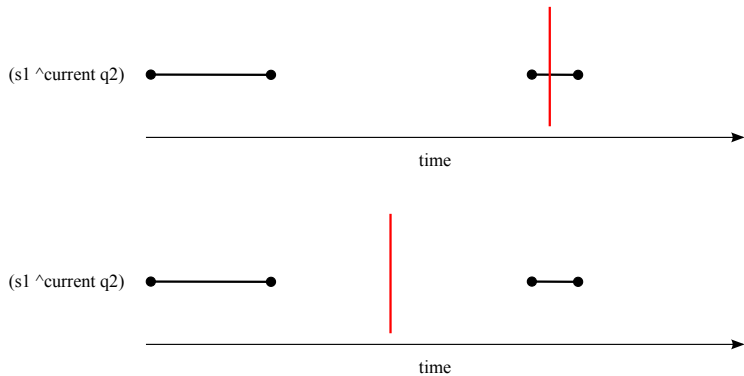
#4: before's offer little speed gain



#4: before's offer little speed gain



#4: before's offer little speed gain



#4: before's offer little speed gain

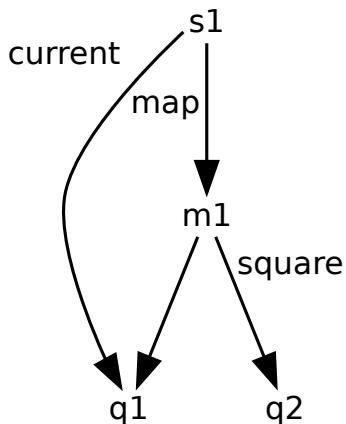
- ▶ Solutions:

- Architecture** Use database information to determine existence of WMEs

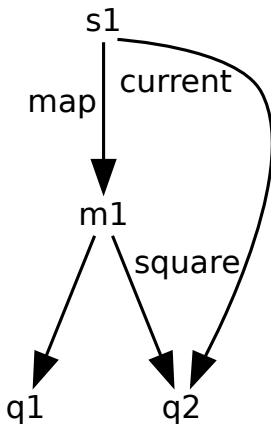
- ▶ Complexity Reduction:

$$O(\|memory\|) \Rightarrow O(1)$$

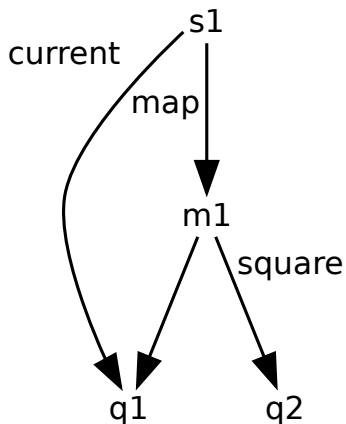
#5: Ignore working memory graph orphans



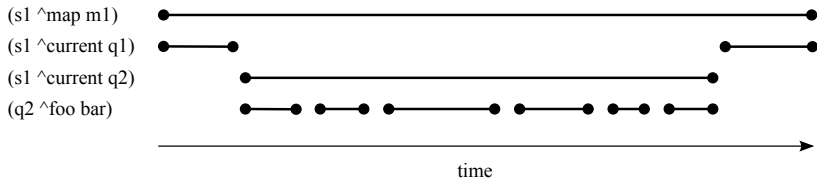
#5: Ignore working memory graph orphans



#5: Ignore working memory graph orphans



#5: Ignore working memory graph orphans



#5: Ignore working memory graph orphans

► Solutions:

Architecture Reduce work done for orphans

Architecture Jump ahead to parent's existence

► Complexity Reduction:

$$O(\|cue\| * \|memory\|) \Rightarrow O(1)$$

Evaluation

Coal

- ▶ Worst case search will always be exponential (unless $P = NP$)
- ▶ Current implementation requires time linear in episodes even for best case

Nuggets

- ▶ It's possible to restrict the exponential worst case with better cues
- ▶ On-going architectural work to improve efficiency of episodic memory

Thank You

Further Reading

- ▶ Derbinsky, N. and Laird, J. E. (2009). Efficiently Implementing Episodic Memory. In Proceedings of the 8th International Conference on Case-Based Reasoning (ICCBR).
- ▶ Nuxoll, A. M. and Laird, J. E. (2007). Extending Cognitive Architecture with Episodic Memory. In Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI).