Efficiently Implementing Episodic Memory

Nate Derbinsky
University of Michigan

What is Episodic Memory?

- Long-term, contextualized store of specific events
  - Tulving, E.: Elements of Episodic Memory (1983)
- Functionally:
  - Architectural
  - Automatic
  - Autonoetic
  - Temporally indexed

The Promise of EpMem

- Virtual Sensing
- Action Modeling
- Retroactive Learning

…

Nuxoll, A.: Enhancing Intelligent Agents with Episodic Memory. (2007)

Efficient Implementation

Goals

- Develop a system that is practical for real-world tasks
- Establish baseline results for graph-based, task-independent EpMem implementations

Assumptions

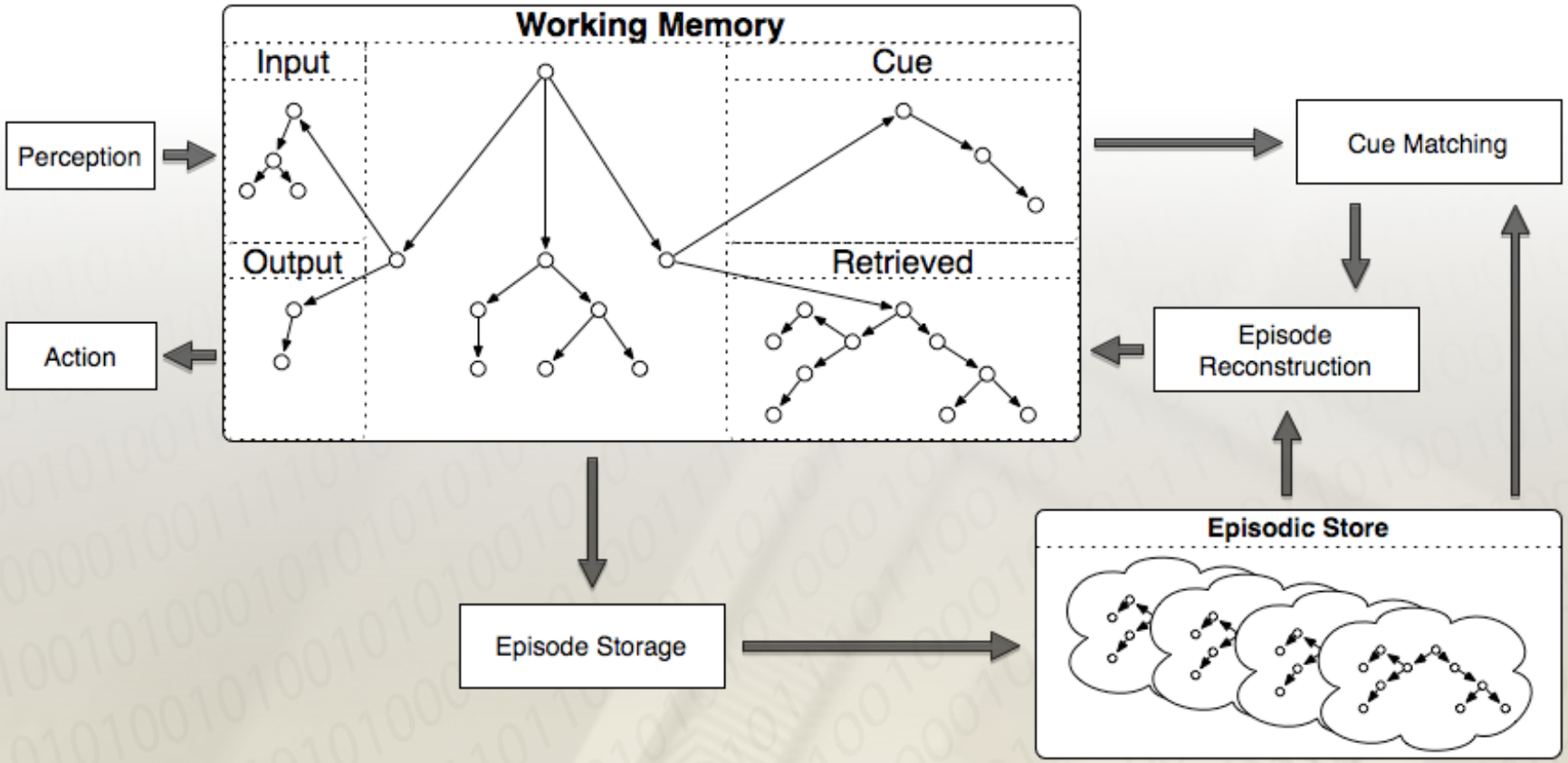
- Stored episodes do not change over time
- Qualitative Nearest Neighbor (NN) cue matching

Performance Challenges

- Consider a year of episodic memories…
  - 16 hours/day -> 42M to 420M episodes
  - 100 – 1000 features/episode (10-100 bytes/feature)
    - **21GB to 21TB**
    - 2GHz CPU -> **10 seconds/scan**

Laird, J.E., Derbinsky, N.: A Year of Episodic Memory (2009)

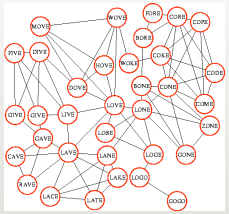
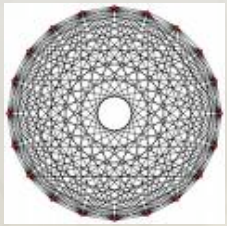
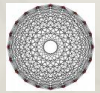
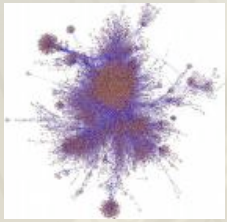
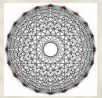
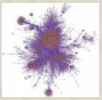
Integrating EpMem with Soar



Episodic Storage

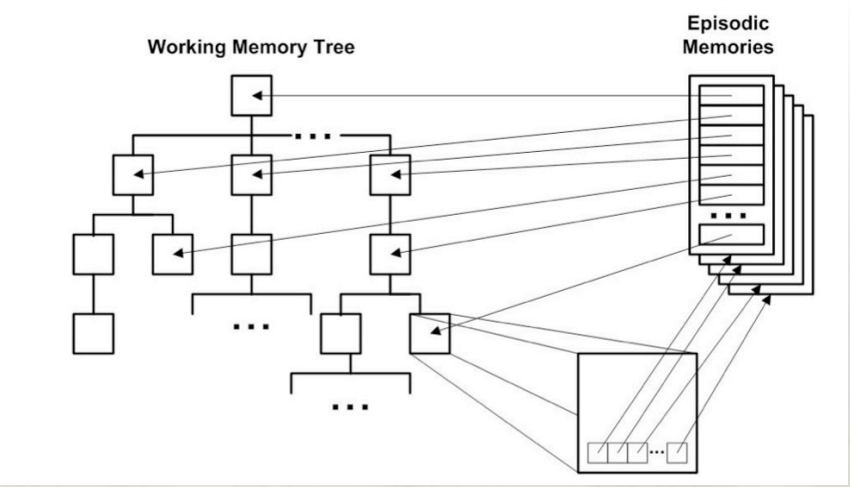
- Faithfully capture Soar's top state of working memory
- Incrementally update indexing structures to facilitate efficient cue matching
- Minimize
  - Memory (monotonically increasing store)
  - Time (relatively frequent operation)

Episodic Storage: Naïve Implementation

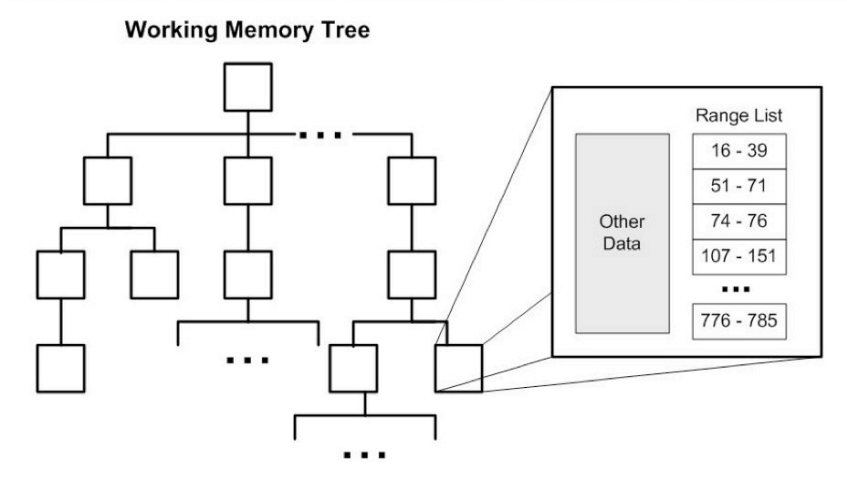
Time	Working Memory		Episodic Store		
1			1		
					
2			1	2	
					
3			1	2	3
					

Compression via Global Memory Structure

- Observation
  - Agents tend to re-use WM structures
- Result
  - Maintain a global record of unique structures
  - Define episodes as "bag of pointers"



Gains via Interval Representation



- Observation
  - An episode will differ from the previous (and next) only in a relatively small number of features
- Result
  - Define episodes implicitly as temporal changes

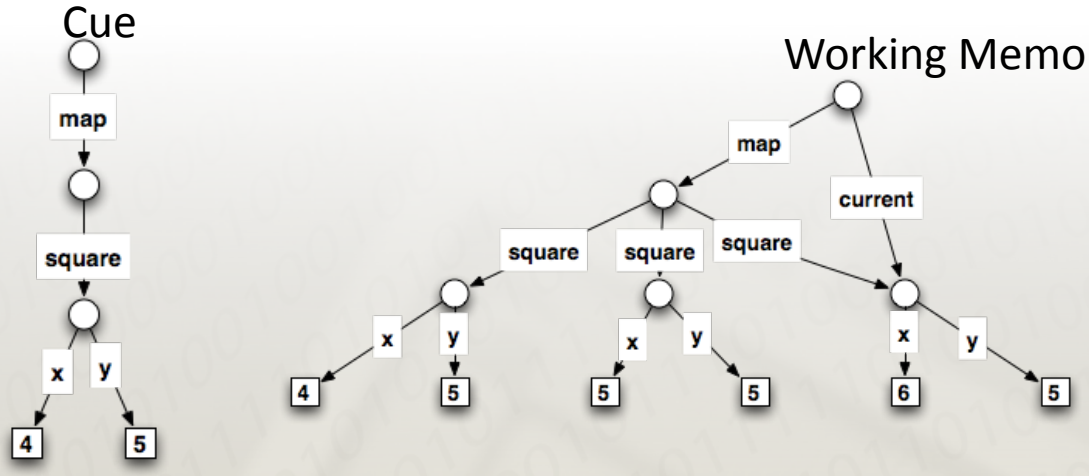
Episodic Storage Summary

- Maintain record of unique WM structures
- Maintain associated intervals on WME addition/removal
  - *Only process changes!*

Episodic storage performs in time/space linear in the *changes* in working memory.

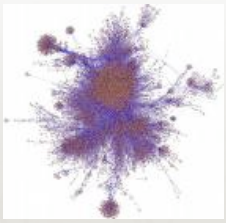
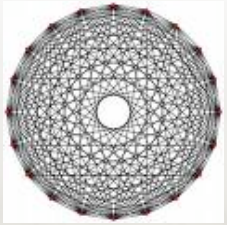
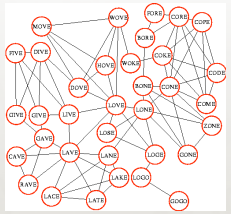
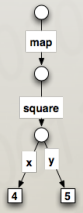
Cue Matching

- A cue is an acyclic graph, partially specifying a subset of an episode



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

Cue Matching: Naïve Implementation

Time	N	N-1	N-2
Episode			
			
Score	§	Ω 	ß

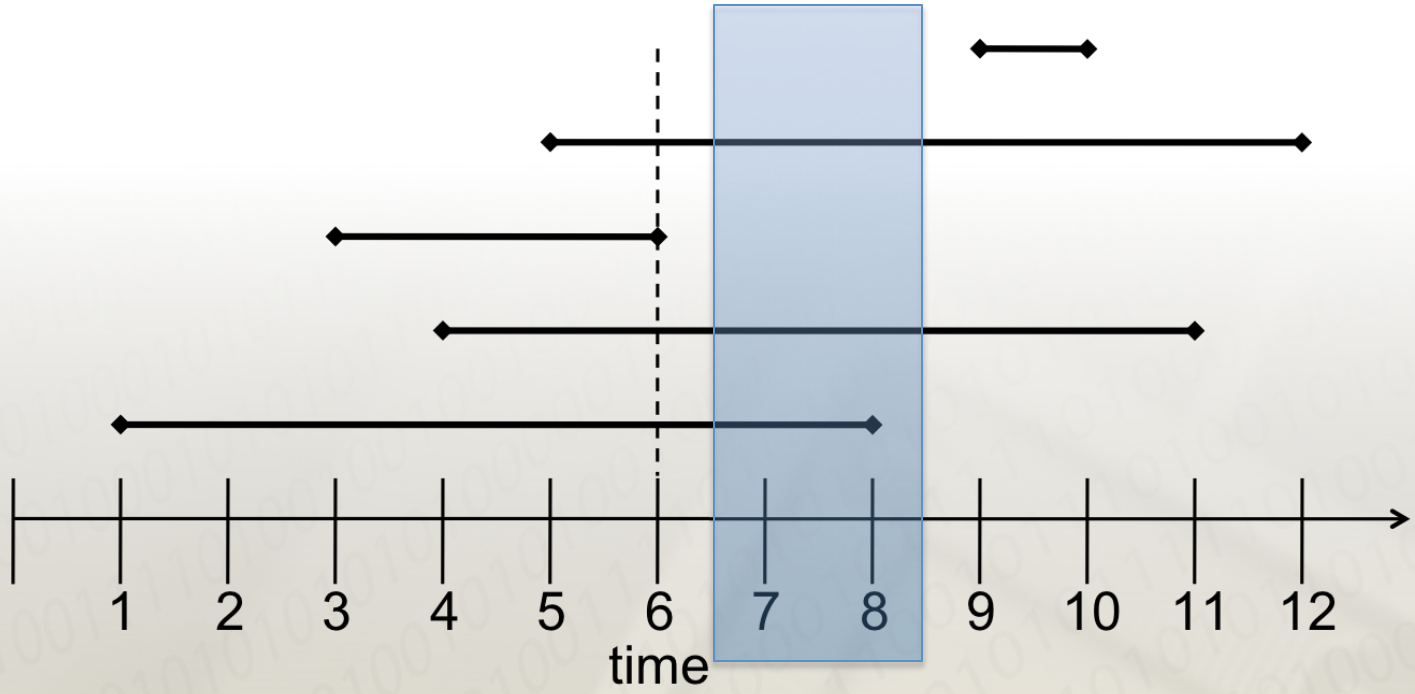
Minimizing Combinatorics via Two-Stage Matching

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

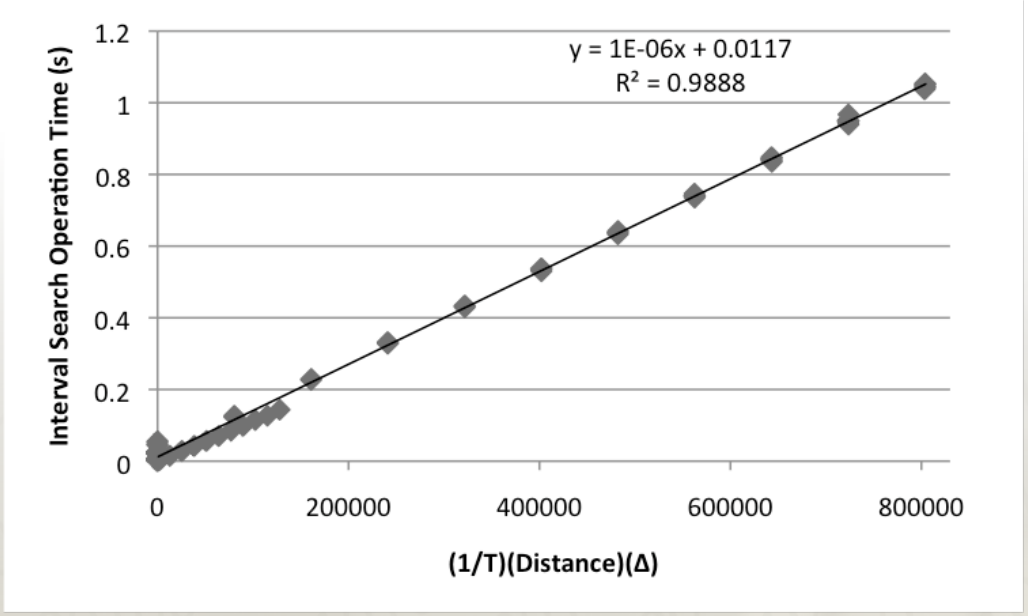
End search on perfect match or no more episodes.

Minimize Episode Evaluation via Interval Endpoint Search



Episode match score changes only at interval endpoints!

Interval Search Model



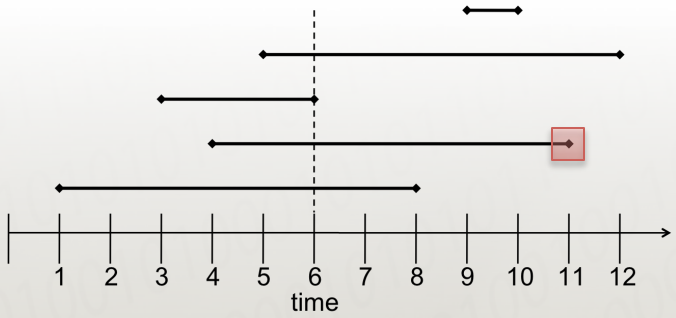
T = Total episodes

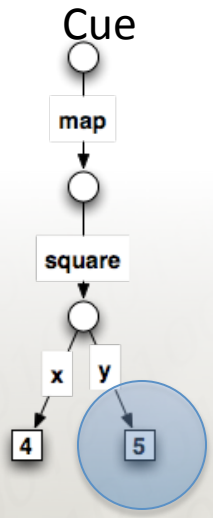
Distance = Temporal distance to best match

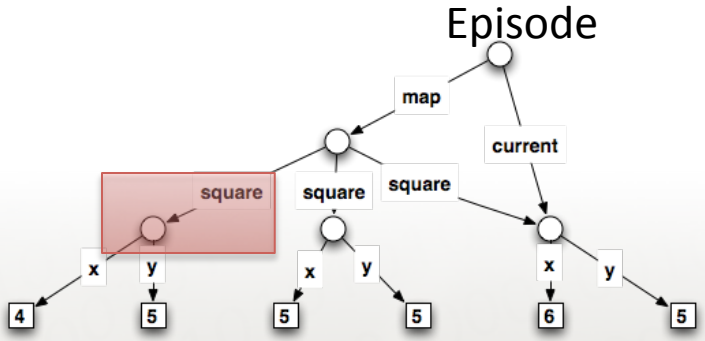
Δ = Cue intervals (~ WM changes)

Interval search is dependent upon the number of candidate episodes evaluated.

Efficient Surface Evaluation via Incremental DNF Satisfaction



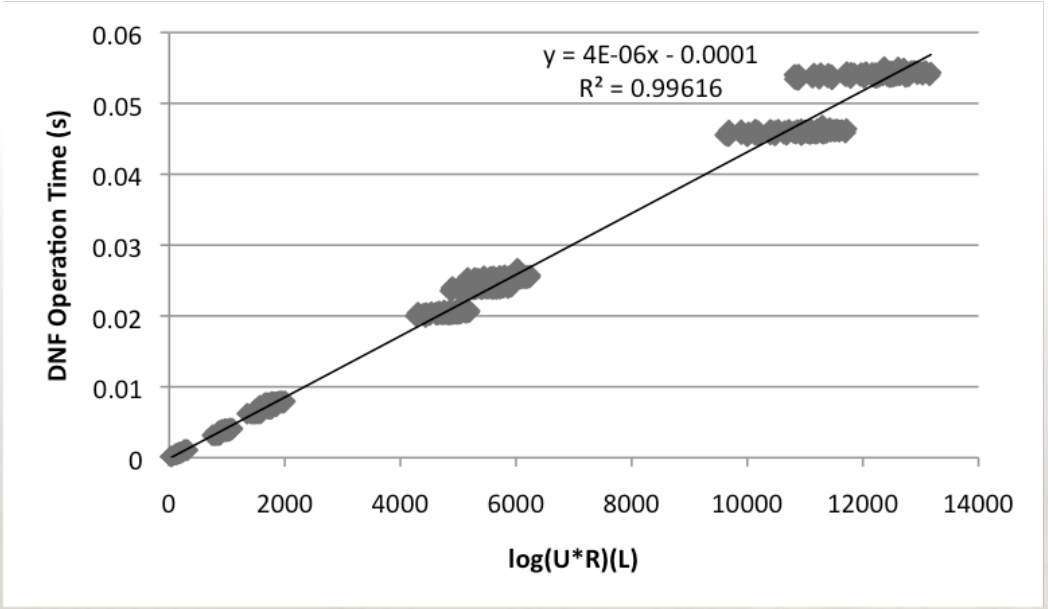




- sat(y=5) := (root AND map[1] AND square[1] AND y=5[1]) OR
  (root AND map[1] AND square[2] AND y=5[2]) OR
  (root AND map[1] AND square[3] AND y=5[3])

- Surface matching can be expressed as evaluating the satisfaction of a set of disjunctive normal form (DNF) Boolean equations
  - **Each interval endpoint inverts the value of a single variable**

DNF Model



U = Unique nodes

R = Stored intervals
(~ changes)

L = Cue node literals

DNF performance is dependent upon the changes in working memory.

Cue Matching Summary

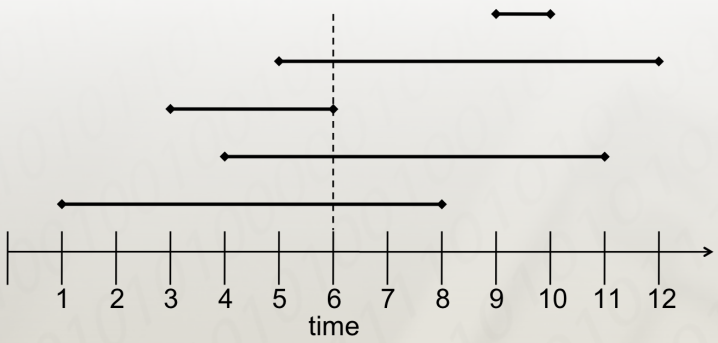
- Minimize candidates by only considering episodes with at least one cue node
- Minimize combinatorics via two-stage matching policy
  - Exponential growth in the worst case
- Minimize episode evaluation via interval endpoint search
  - Linear growth in the worst case
- Minimize surface evaluation cost by only processing cue node changes

Episode Reconstruction

- The process of faithfully reproducing all episode content and structure within the agent's working memory
  - Collect contributing episode elements
  - Add elements to working memory

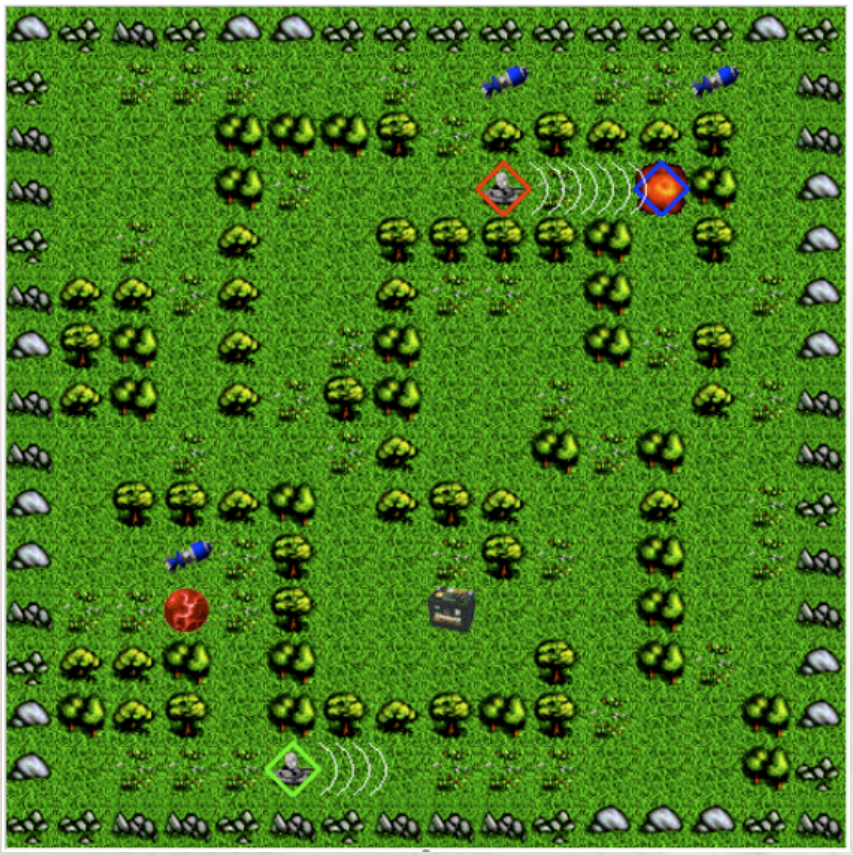
Logarithmic Interval Query via Relational Interval Tree

- Collecting episode elements in an Interval representation is tantamount to an interval intersection query:
  - Collect all elements that started before and ended after time *t*



- By implementing an interval tree, intersection queries are answered in time logarithmic with respect to the changes in working memory

Empirical Domain



- TankSoar *Mapping-Bot*
  - 2500 features
  - 70-90% of perceptual WMEs change each episode
- 2.8GHz, 4GB RAM
- SQLite3

Empirical Results

1 million episodes (~1 episode/decision), 10 trials

Storage	Cue Matching*	Reconstruction**	Total
2.68ms 625-1620MB (0.64-1.66KB/ep)	57.6ms	22.65ms	**82.93ms**

* 15 cues

** 50 random times

Future Work

- Better Evaluation
  - Characterize architecture performance with respect to properties of the environment, agent, cues, and task
  - Longer and multi-task runs
- Bound Cue Matching
  - Fast familiarity
  - Heuristic graph-match
- Algorithmic Variants
  - Selection bias: activation, arousal via appraisals, etc.
  - Characterize task vs. architecture performance

Evaluation

Nuggets

- Extended and improved Soar-EpMem implementation (9.1.1)
- 1M episode initial empirical study with predictive performance models

Coal

- Limited evaluation

Further Reading

- Derbinsky, N., Laird, J.E.: *Efficiently Implementing Episodic Memory*. To Appear: Proceedings of the 8th International Conference on Case-Based Reasoning (2009)

- Laird, J.E., Derbinsky, N.: *A Year of Episodic Memory*. To Appear: Workshop on Grand Challenges for Reasoning from Experiences, 21st International Joint Conference on Artificial Inteligence (2009)