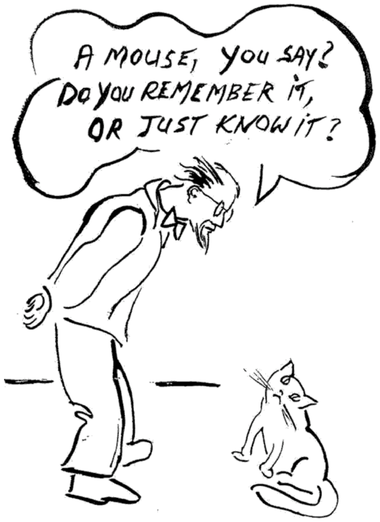
EPISODIC MEMORY:
A DBMS PERSPECTIVE

Nate Derbinsky

Outline

- Motivation
- Problem Characterization
- Soar-EpMem Implementation
- Results
- Future Work

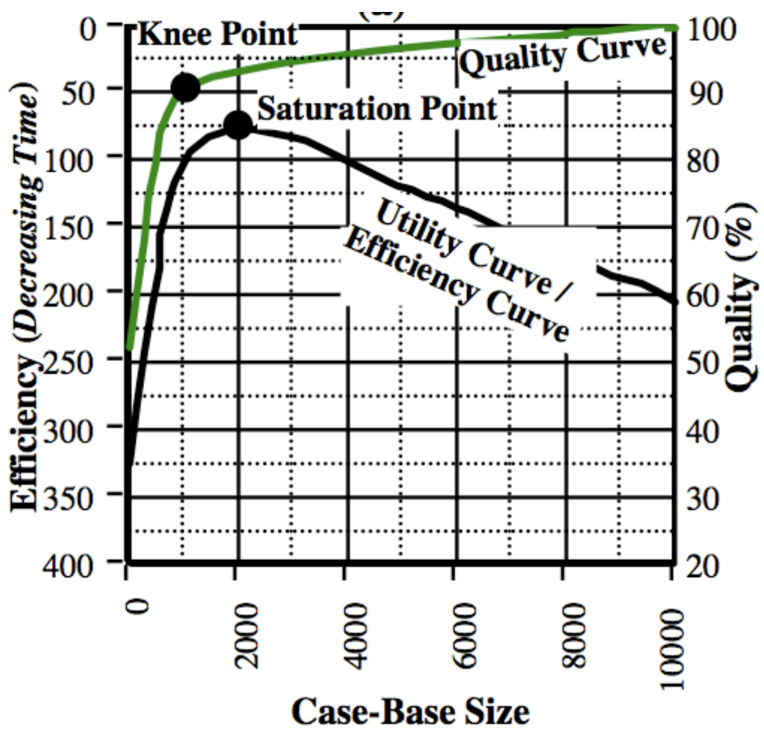
What is Episodic Memory?



- □ Long-term, contextualized store of specific events
  - □ *Tulving, E. (1983). Elements of Episodic Memory.*
- □ Comparable to CBR
- □ Affords agents cognitive capabilities
- □ Constrained: (encoding, storage, retrieval)
  - □ *Nuxoll, A. (2007). Enhancing Intelligent Agents with Episodic Memory.*

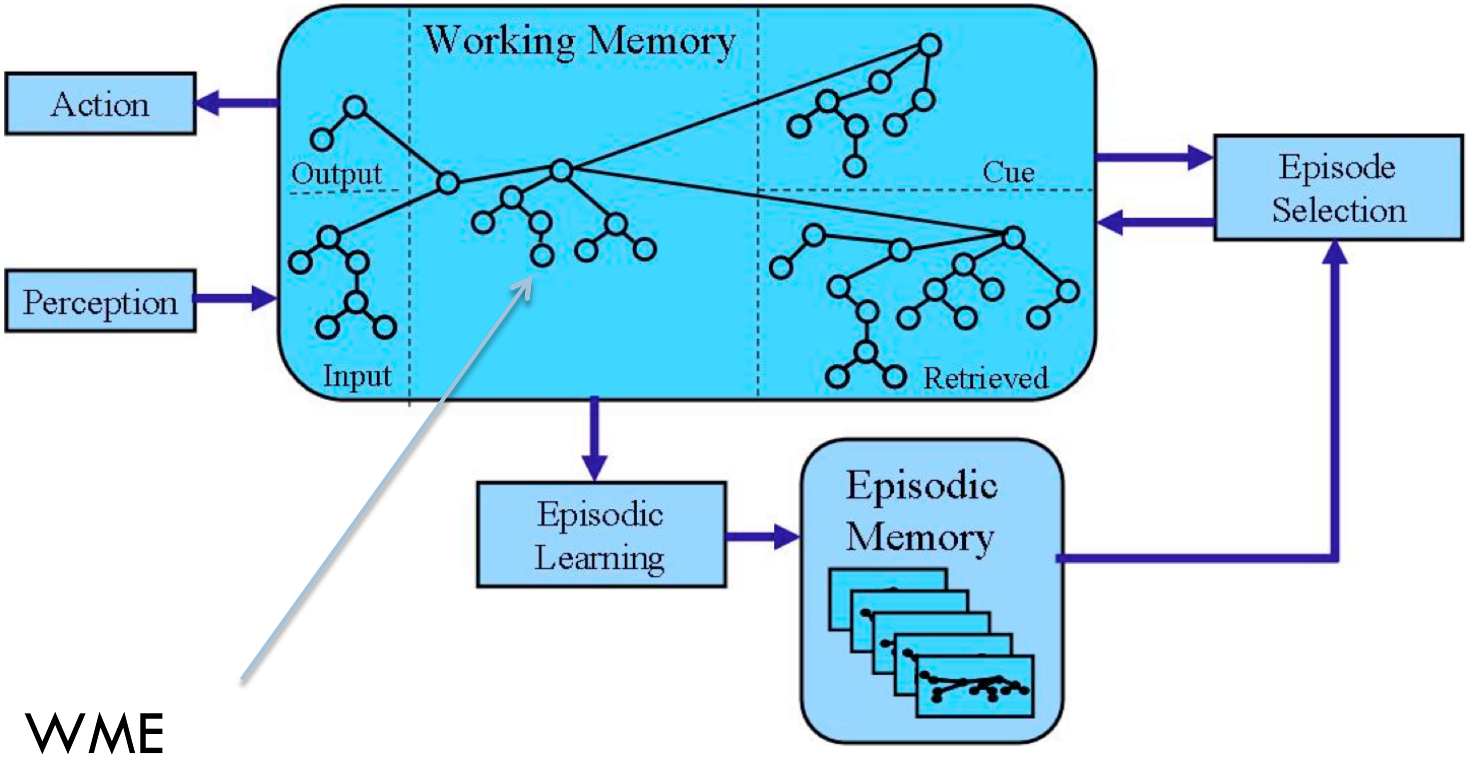
Utility of Episodic Memory

- Episodic memory can be crucial to an agent's efficacy
- The functional specification may lead to a monotonically increasing store



Smyth, B. and Cunningham, P. (1996).
The Utility Problem Analysed: A Cased-Based Reasoning Perspective.

Problem: EpMem Integration



WME

`(id ^attrib value)`

Nuxoll, A. and Laird, J. (2007). Extending Cognitive Architecture with Episodic Memory.

Problem: Storage

- Characteristics
  - Maintain episode content/structure
  - Relatively frequent
  - Monotonically increasing store
- Regularities
  - Temporal persistence
  - Structural persistence

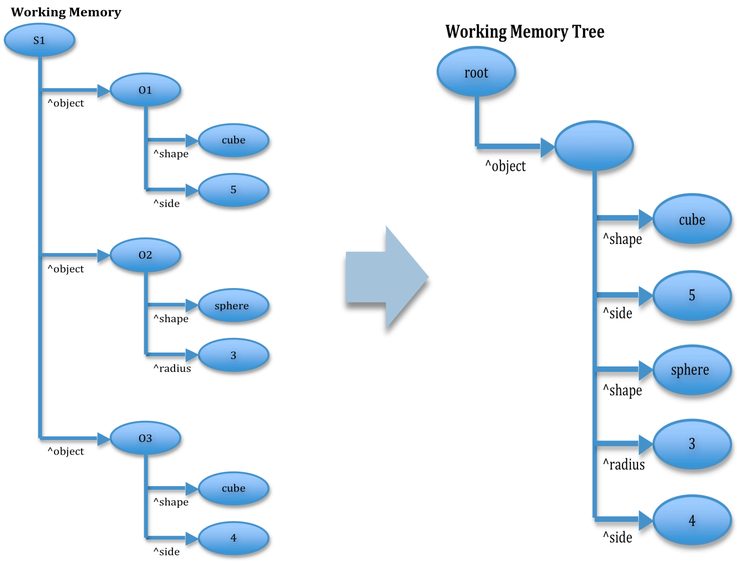
Problem: Query

- Characteristics
  - Cue: deliberate, declarative, structured
  - Retrieval
    - Nearest-Neighbor (NN)
      - cardinality + activation (feature weighting)
    - Biased by recency
- Regularities
  - Literature points to <u>structure</u>
    - Surface vs. Deep
    - Cardinality bias

Storage ("Insert")

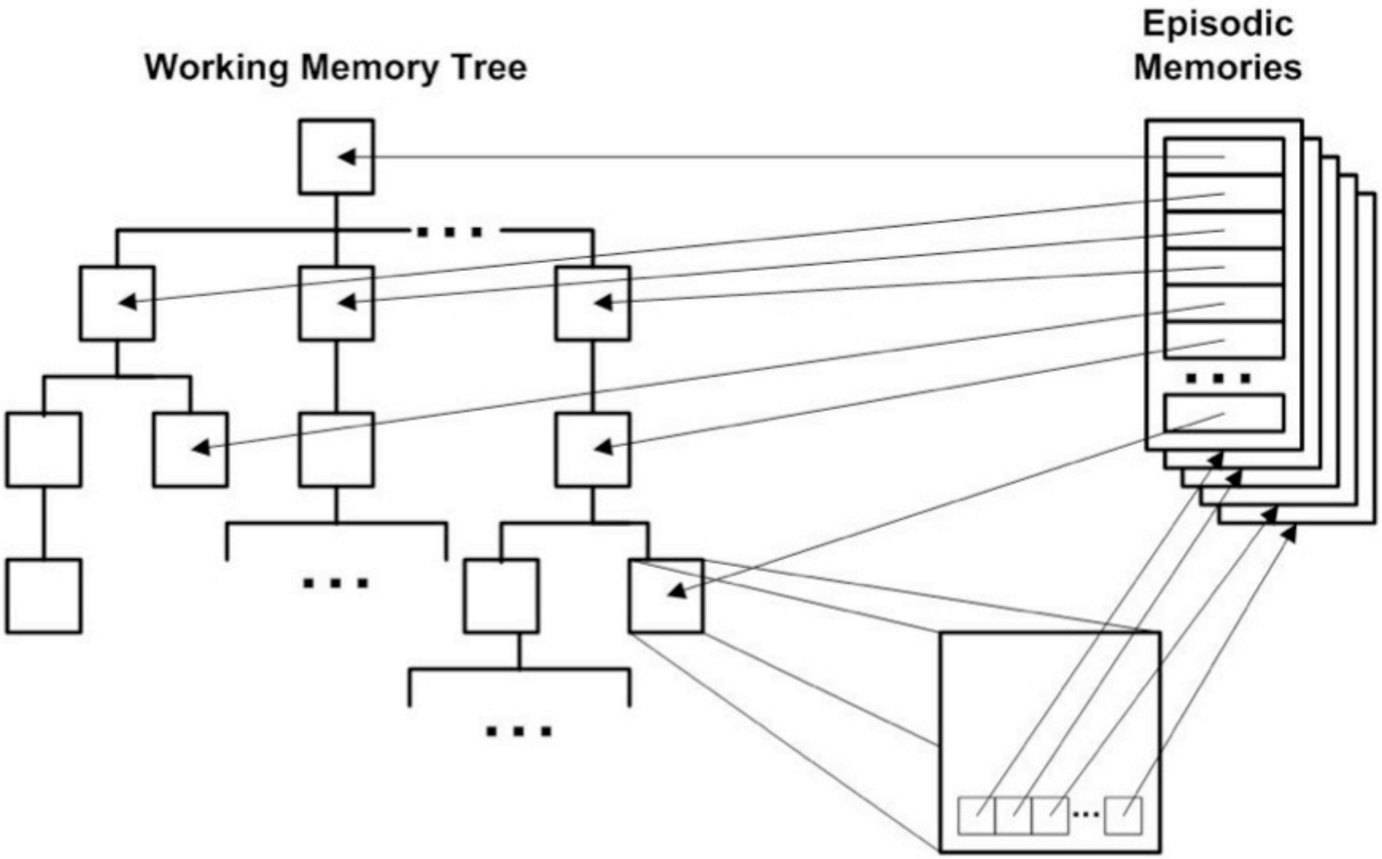
Working Memory Tree

- □ Maintains structural identity of all unique attribute-value pairs
- □ (id, parent, name, value)

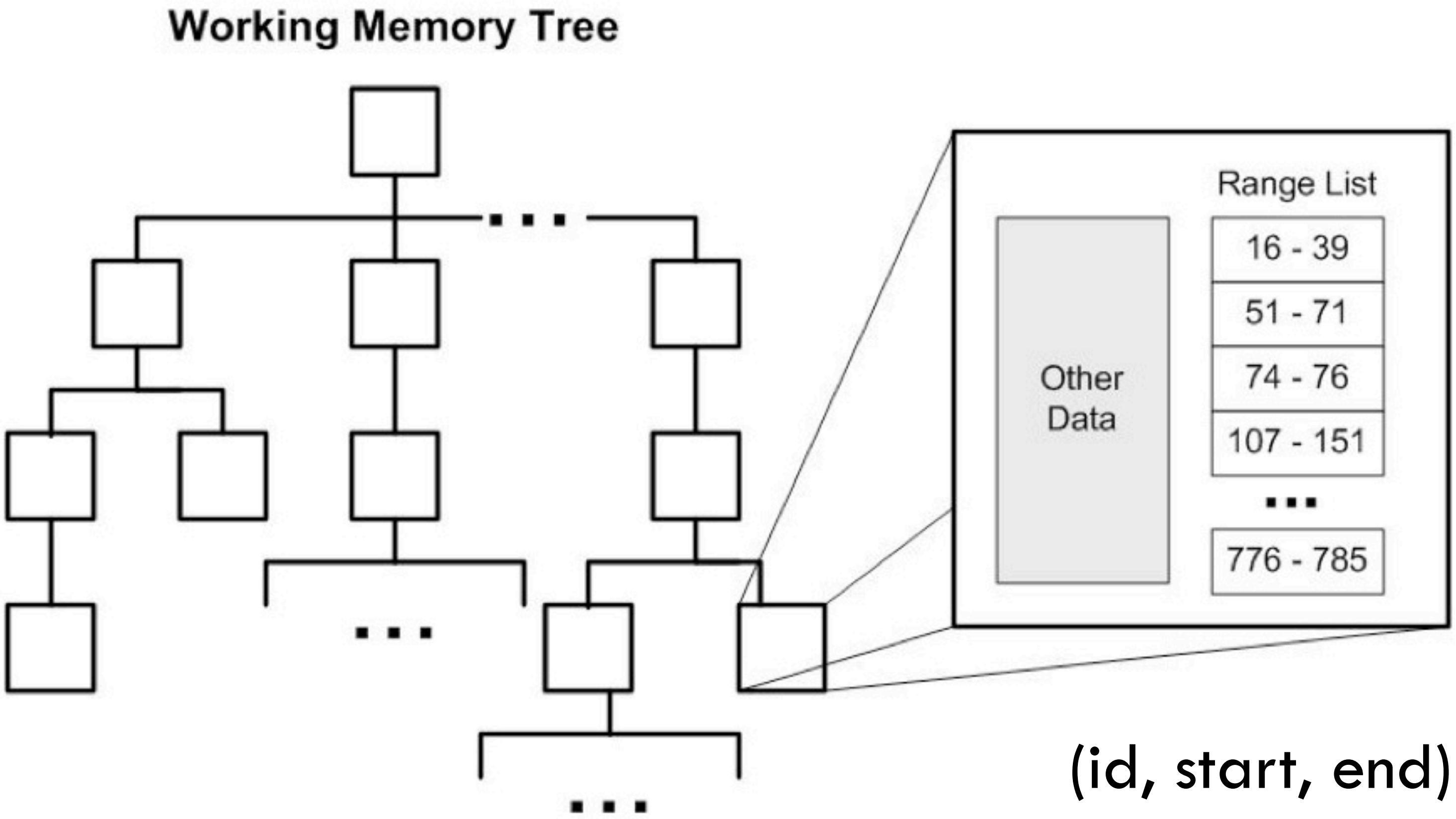


Storage. A scheme for associating nodes of the Working Memory Tree with a temporal id.

Instance Indexing



Range Indexing



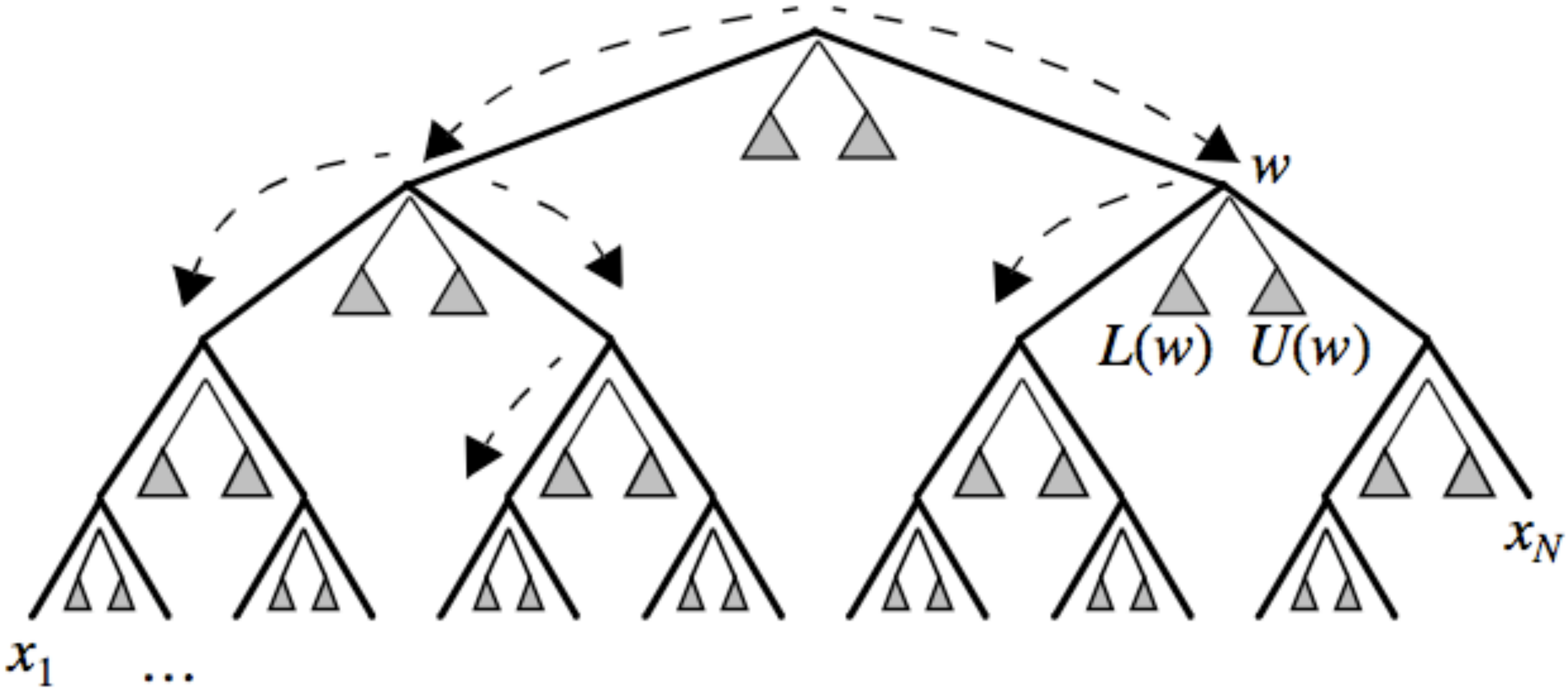
(id, start, end)

Reconstruction

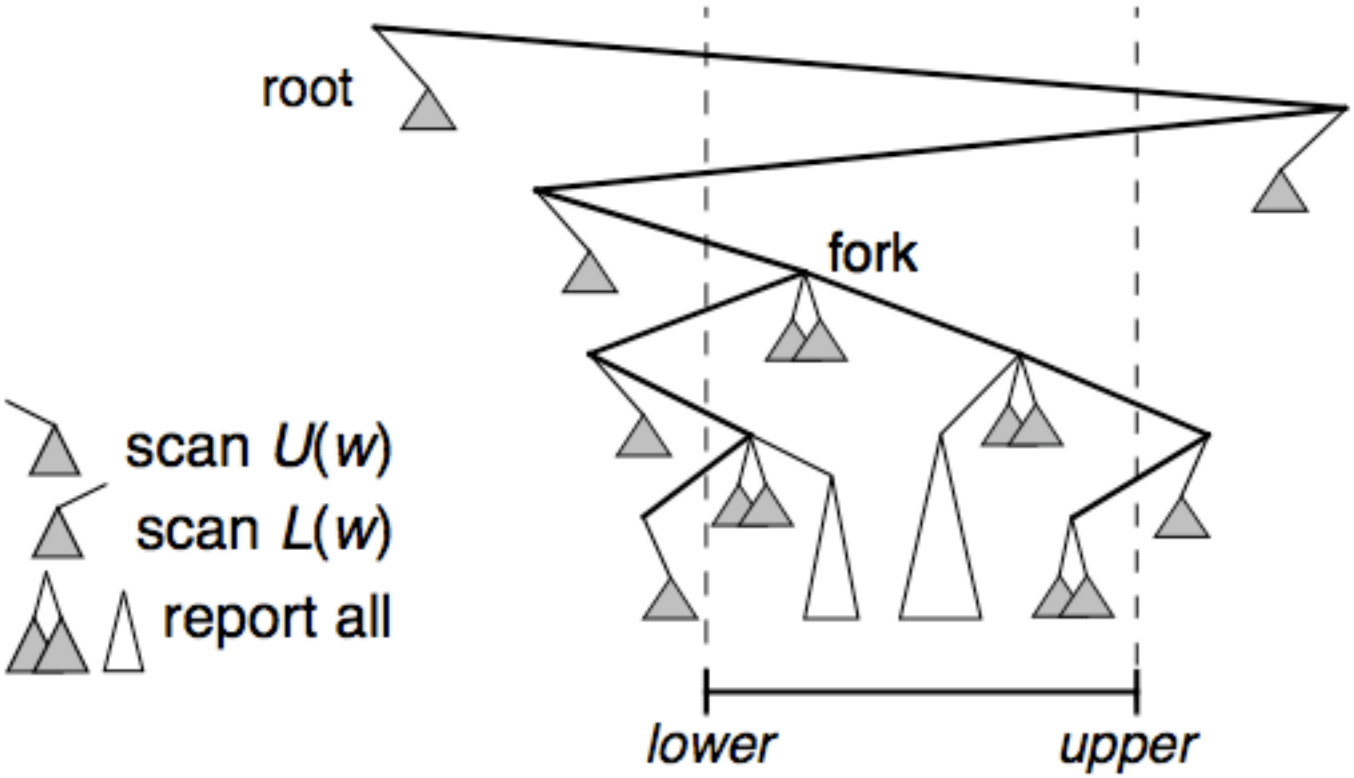
- Reconstructing an episode given a range representation is an interval intersection query
- Implemented Relational Interval Tree (RIT)
  - Adds computed attribute: “node”
  - Adds temporary, indexed relations to be populated at time of query

Kriegel, H., Potke, M., Seidl, T. (2000).
Managing Intervals Efficiently in Object-Relational Databases.

Interval Tree Queries



Interval Tree Queries

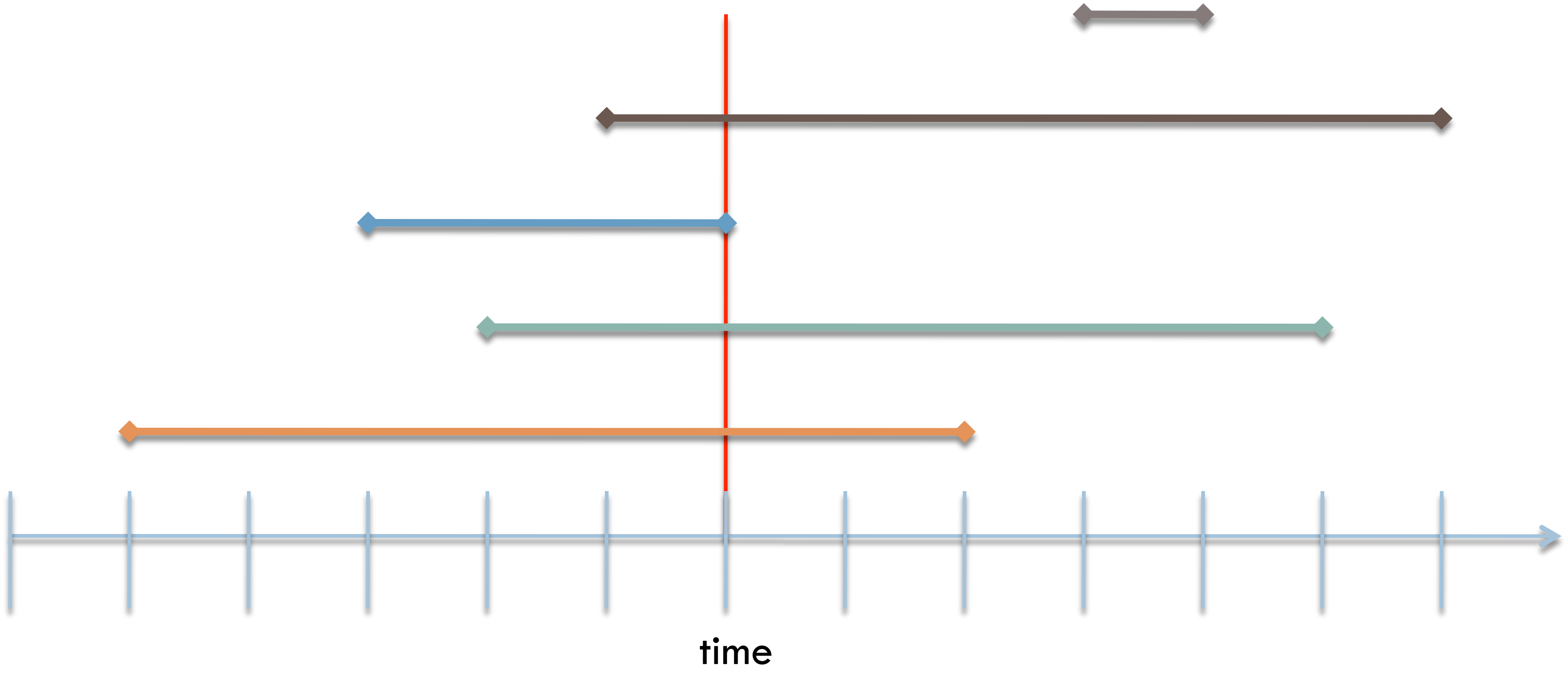


Supplementing RIT

- The RIT algorithm requires additional storage/ computation for stored intervals
- We can exploit alternative representations for two classes of ranges
  - “Now” ranges -> (id, start)
  - “Point” ranges -> (id, start)

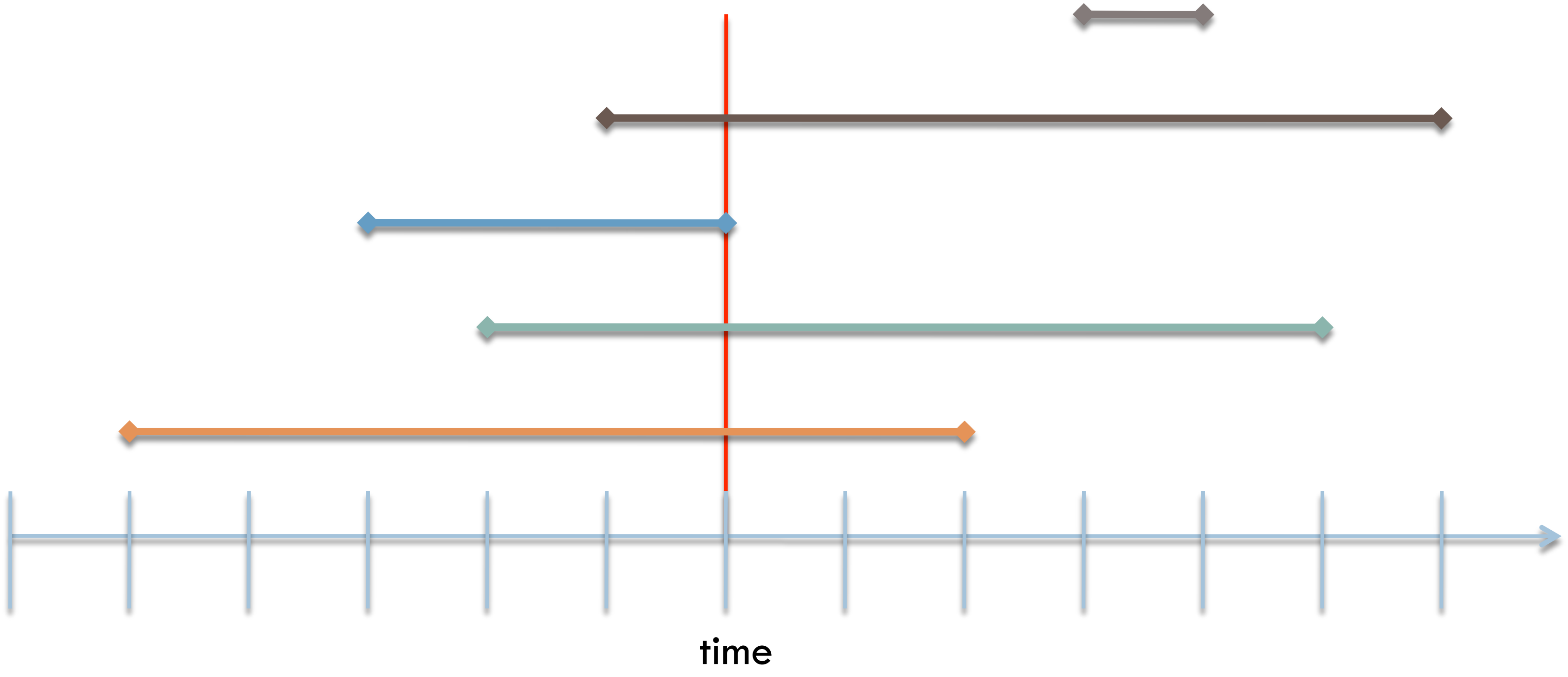
Query ("Select")

- Brute force tactic (discrete time)
  - Instantiate ranges (w.r.t. time), find best "sum"
  - O(n), n = # episodes

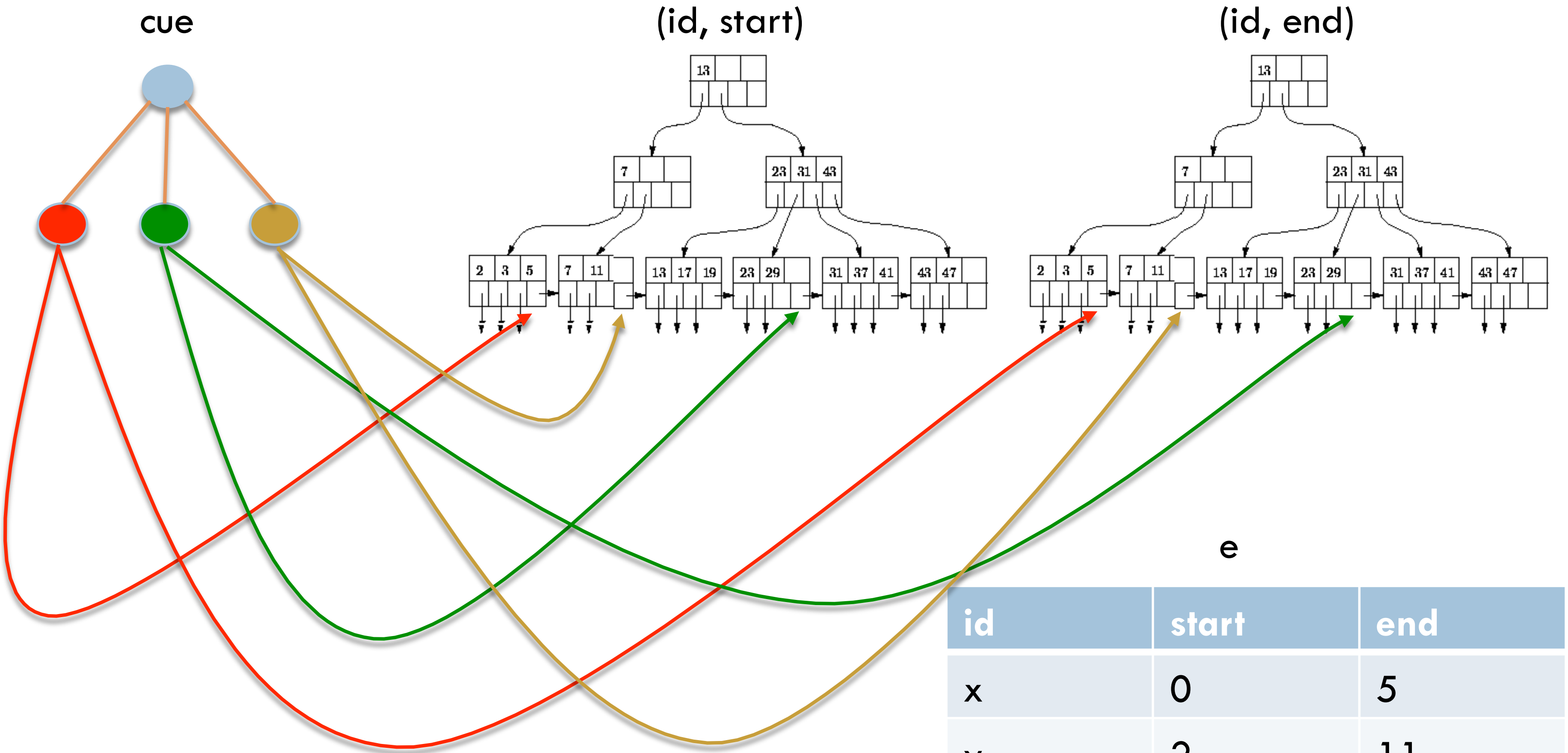


Query ("Select")

- Smarter algorithm
  - Walk range endpoints, keep track of current/best "sum"
  - O(m), m = # pertinent ranges



Mapping Range Search -> RDBMS

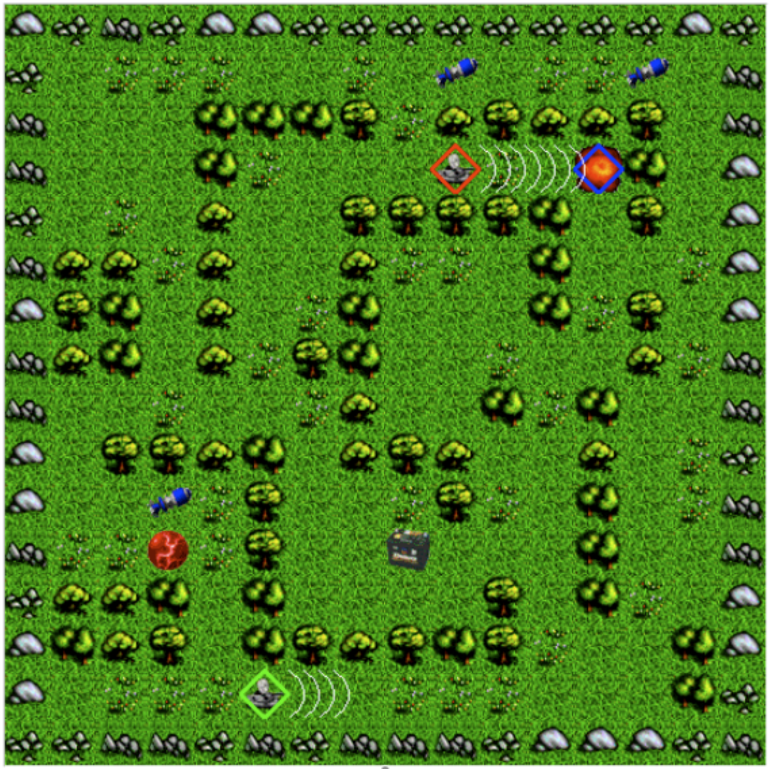


SELECT start/end FROM e WHERE e.id=X
ORDER BY start/end DESC

e

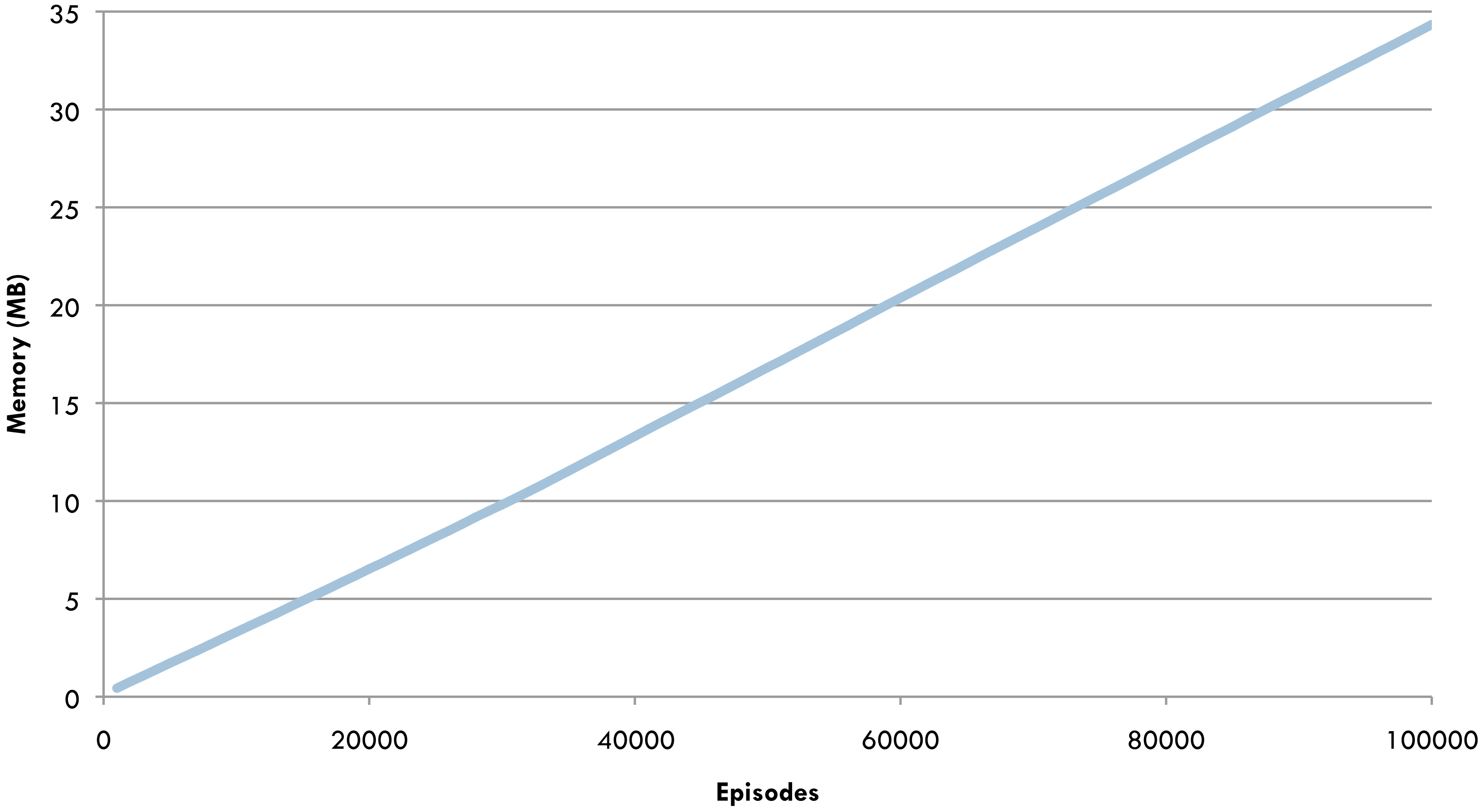
id	start	end
x	0	5
y	2	11
…	…	…

Experimentation

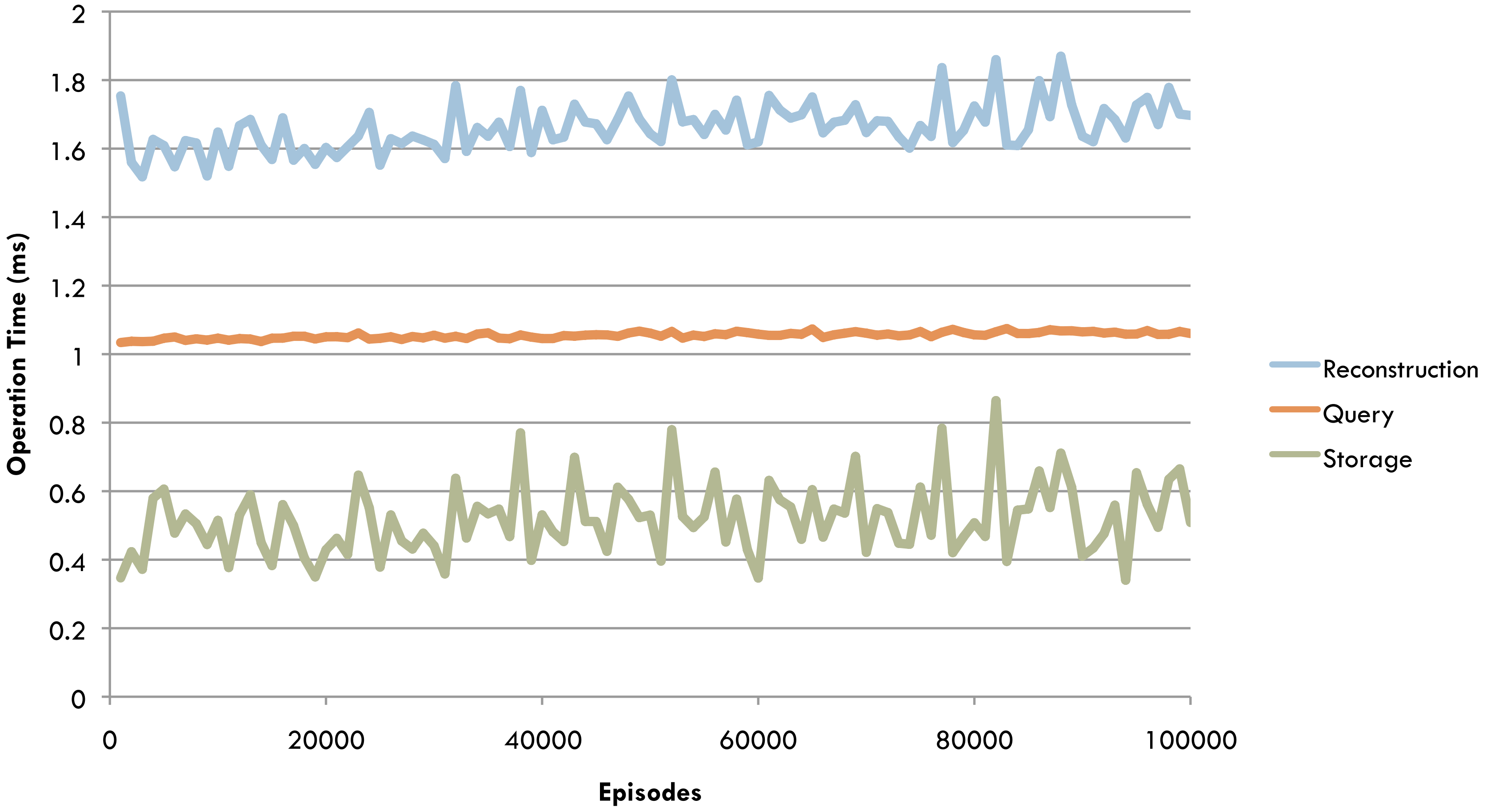


- Domain
  - Single agent, fixed number of decisions
  - Typically 100 WMEs
  - 70-90% change
  - 300 unique
- System
  - Soar 9.1.0-beta
  - SQLite3

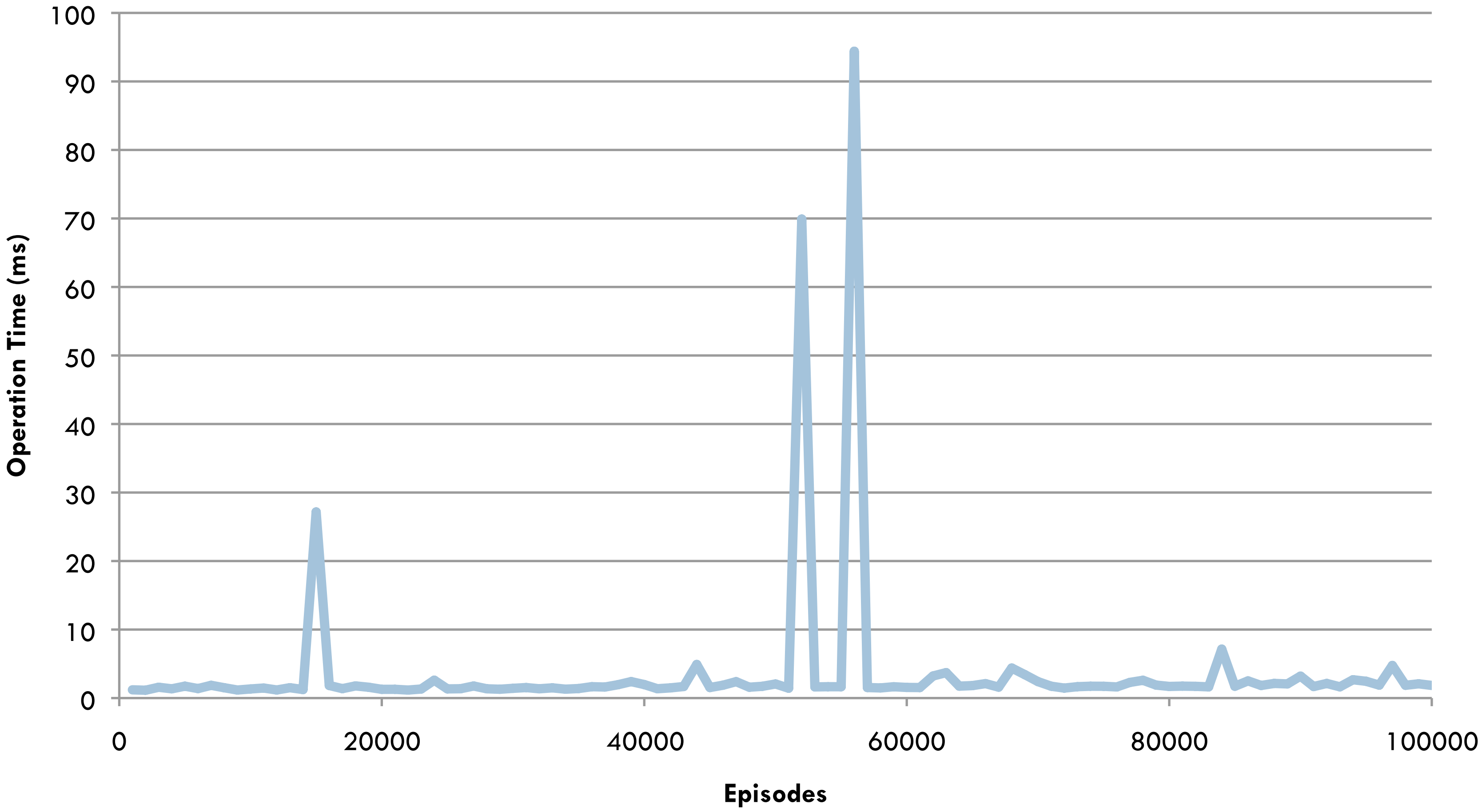
Results: Memory Consumption



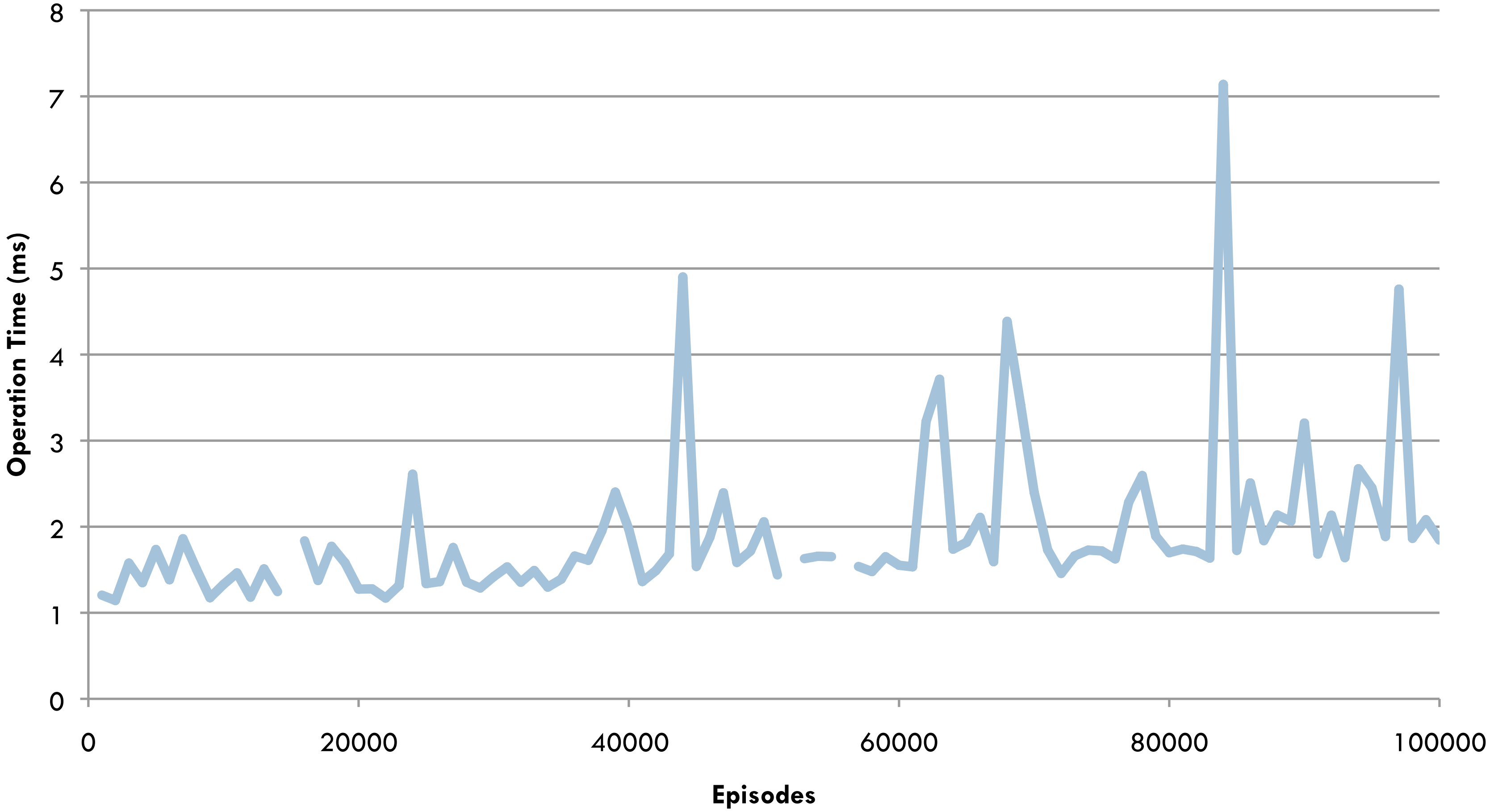
Results: (X, Y, Direction), Most Recent



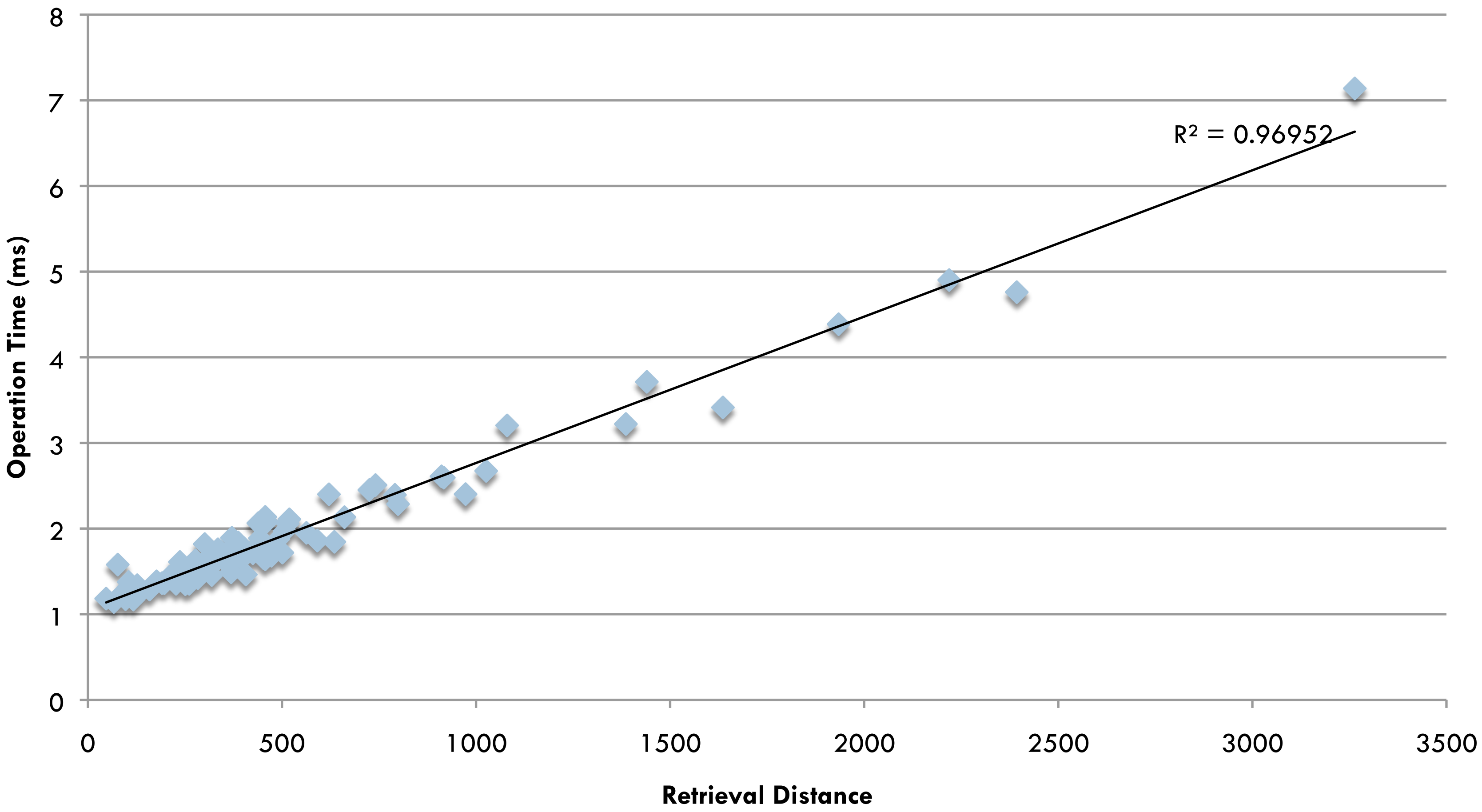
Results: (X, Y, Direction)



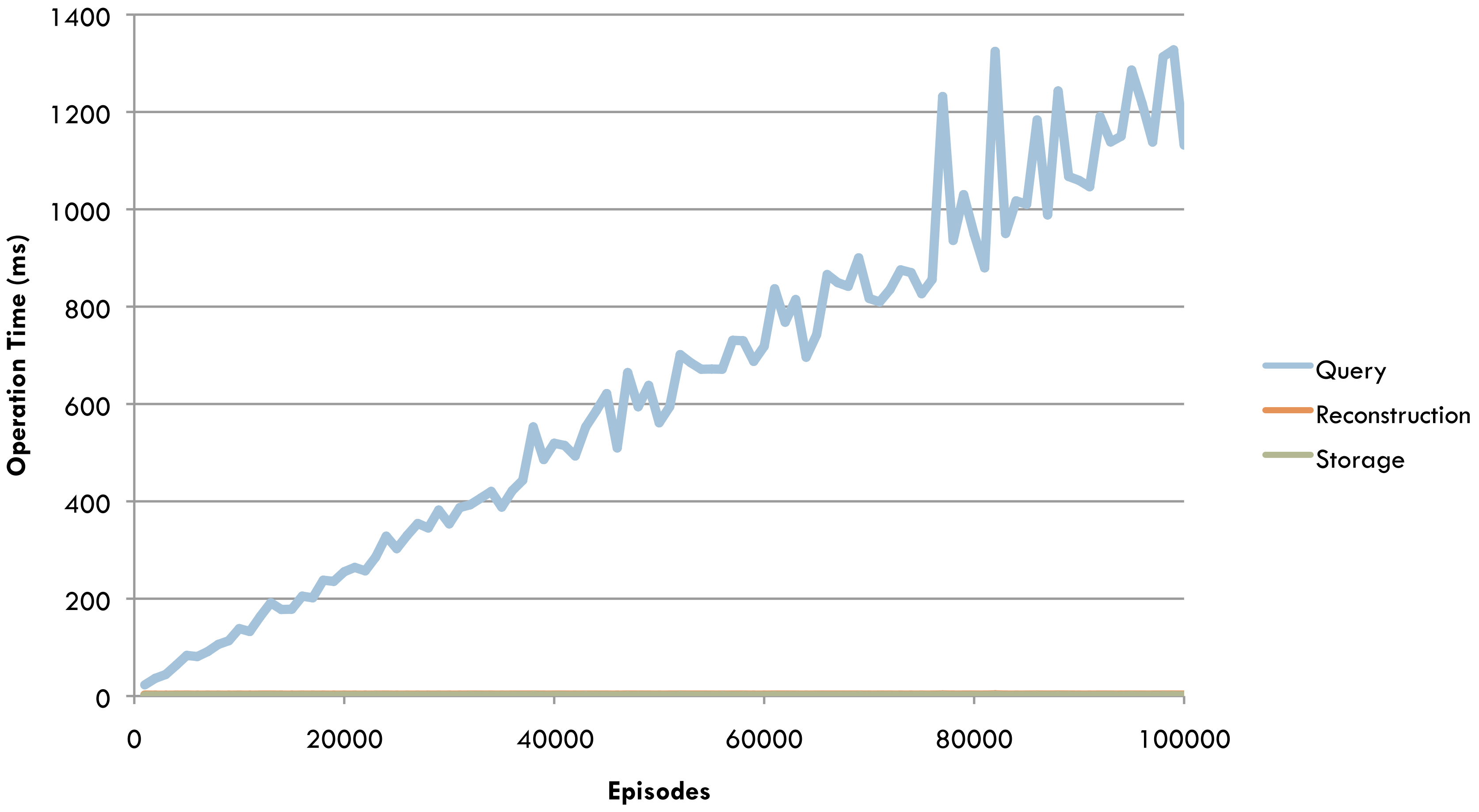
Results: (X, Y, Direction)



Results: (X, Y, Direction)



Results: Input-Link



Future Work

- Proper query test-bed => regularities => heuristics (ala "quality")
  - OLAP
  - 2-Stage Query
- Query optimization: OR's vs. AND's + statistics