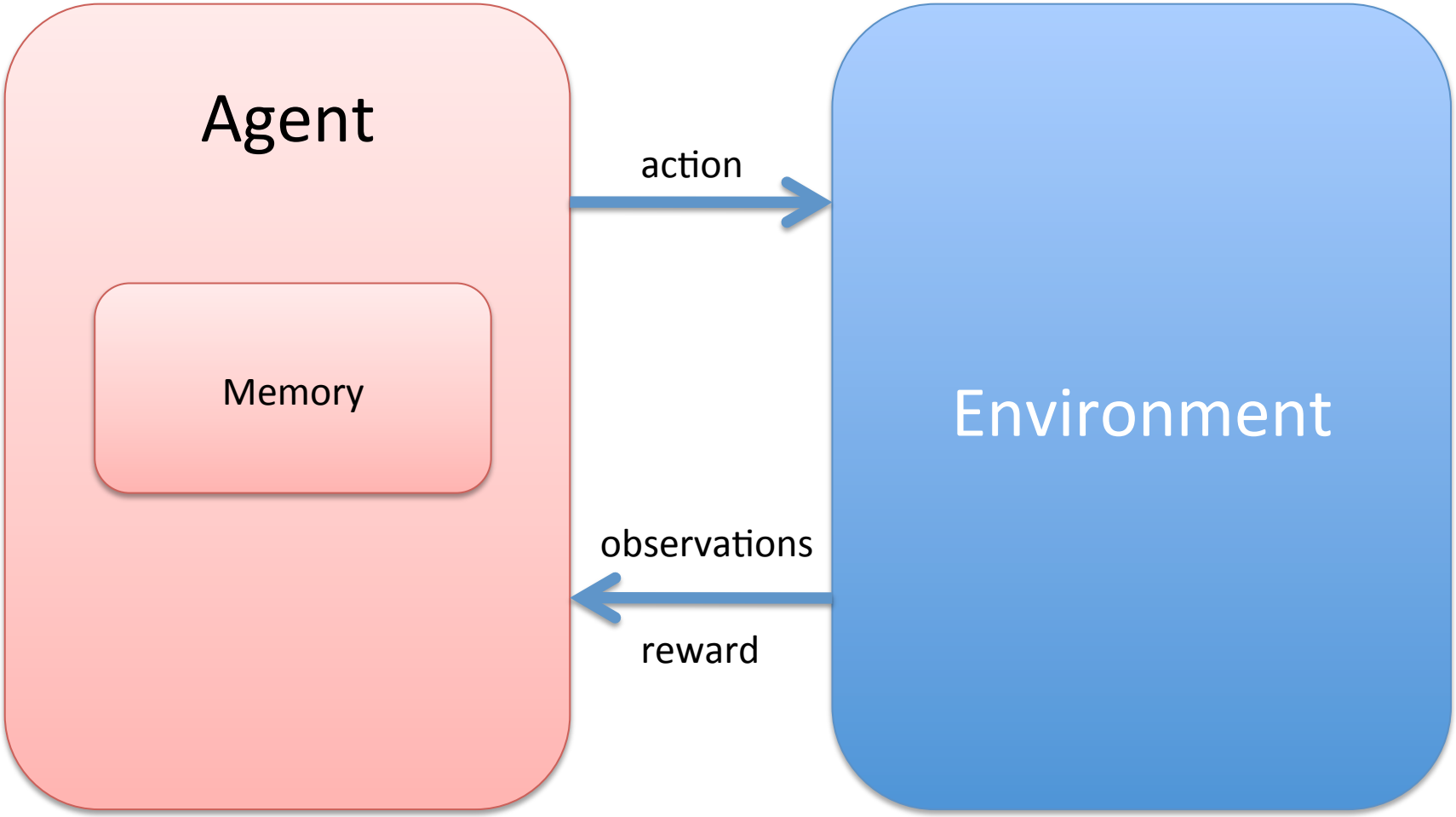
Learning To Use Memory

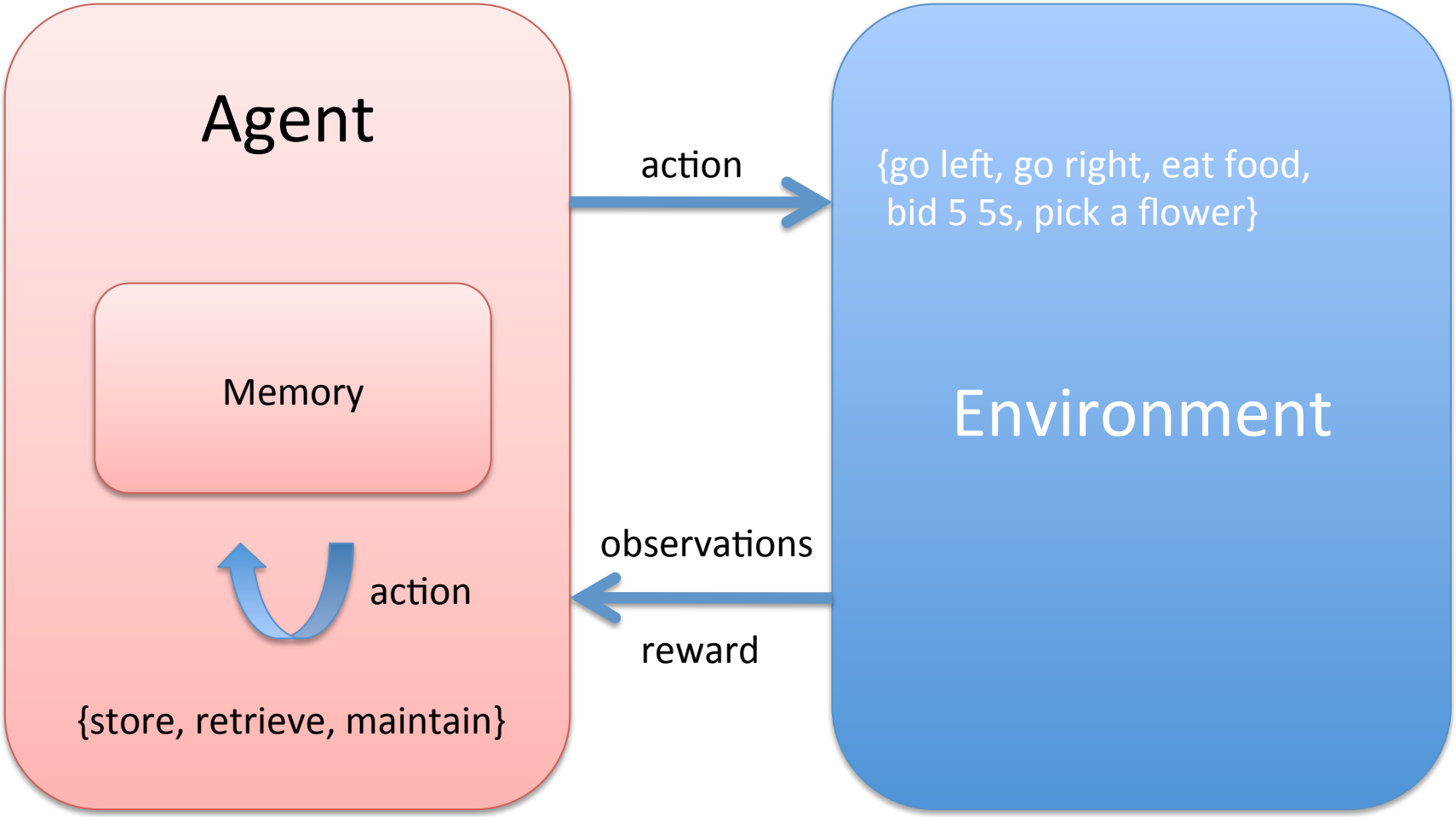
Nick Gorski & John Laird

Soar Workshop 2011

Memory & Learning



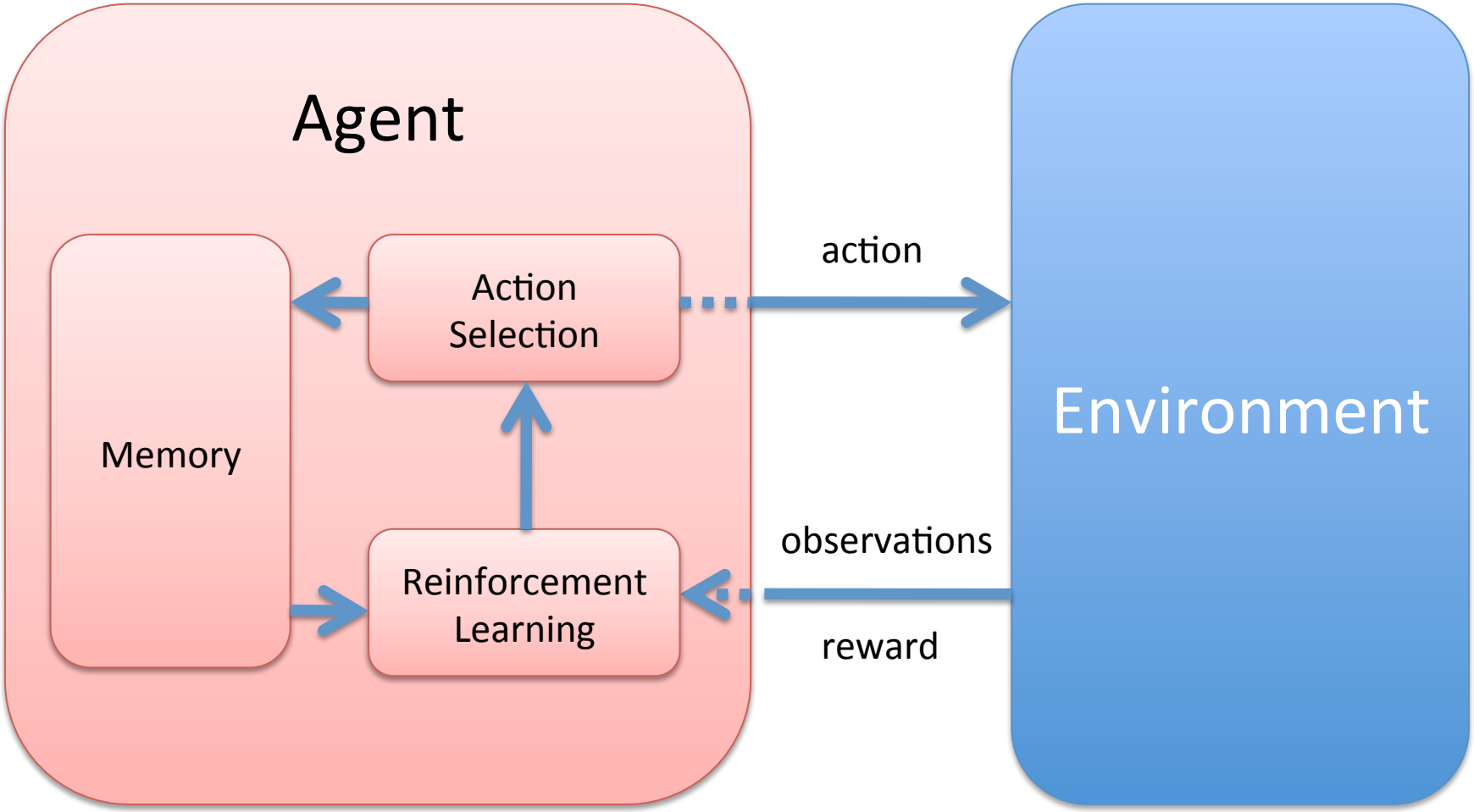
Actions, Internal & External



Internal Actions Over Memory

- Internal actions are deliberate or automatic
- Automatic actions are in the background
  - Architectural and always happen
  - Ex: storage to episodic memory
- Deliberate actions are in the foreground
  - Procedural knowledge and cognitive
  - Ex: storage to working memory

Internal Reinforcement Learning



Assumptions

- Custom framework, not using Soar
- Simple memory models
- Simple tasks

Learning to Use Memory

- Research Question:
  - When can agents learn to use memory?
- Idea:
  - Investigate dynamics of memory and environment independently
- Need:
  - Simple, parameterized task

An Interactive TMaze

LEFT (observation)

{forward} (avail. actions)

An Interactive TMaze

DECIDE

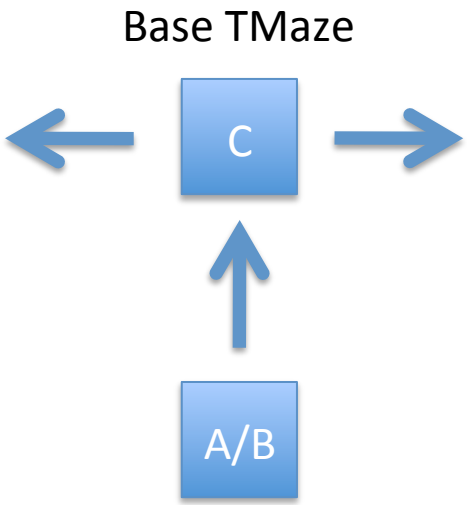
(observation)

{left, right}

(avail. actions)

ard)

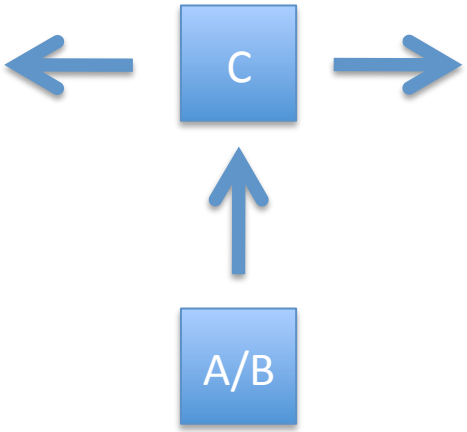
TMaze



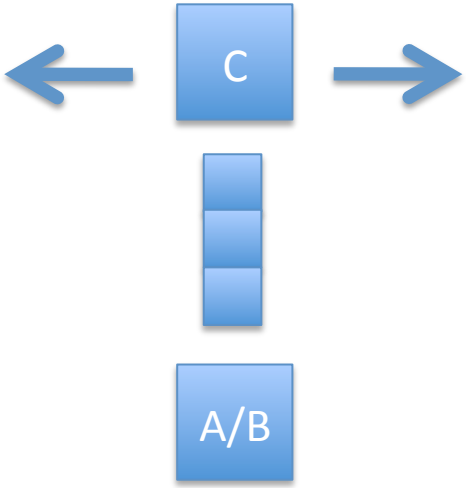
Question: how much knowledge is needed to perform this task?

Parameterized TMazes

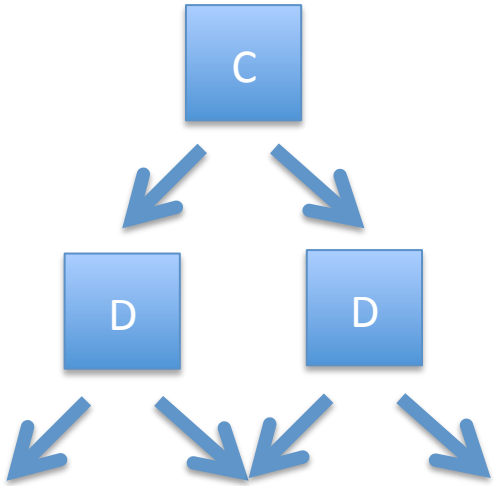
Base TMaze



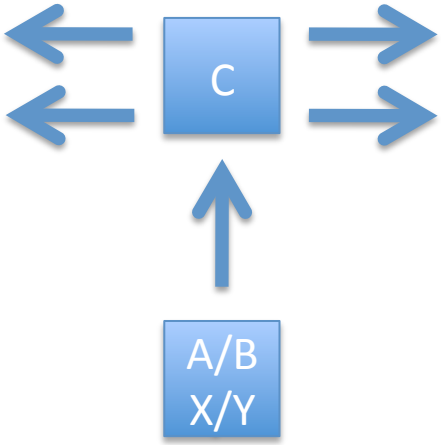
Temporal Delay



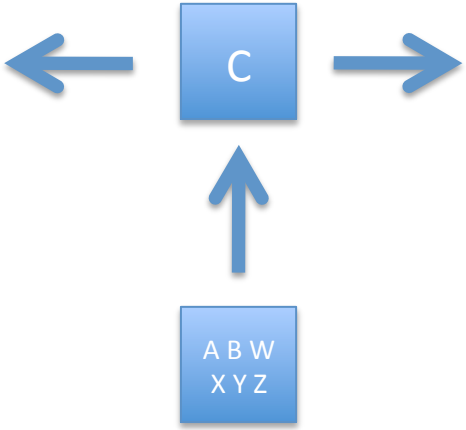
Dependent Actions



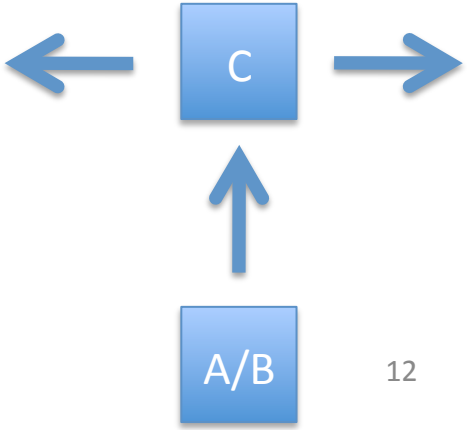
Concurrent Knowledge



Amt. of Knowledge

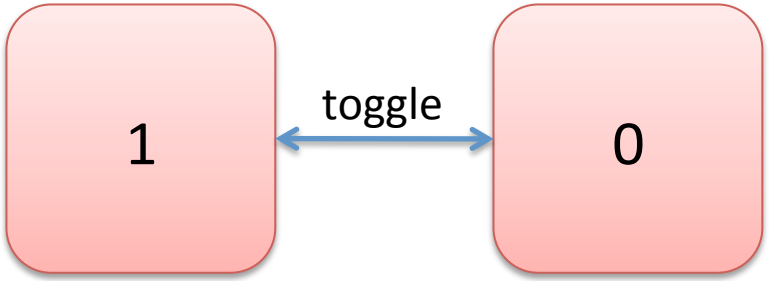


2nd Order Knowledge

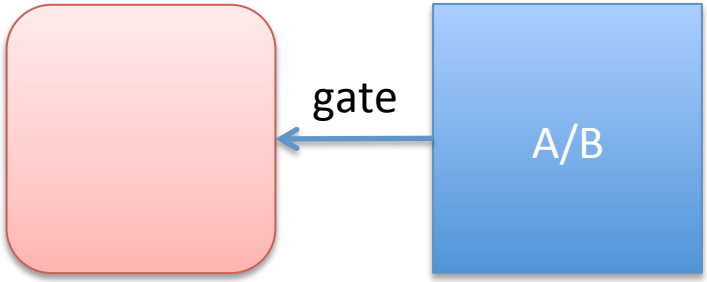


Two Working Memory Models

Bit memory



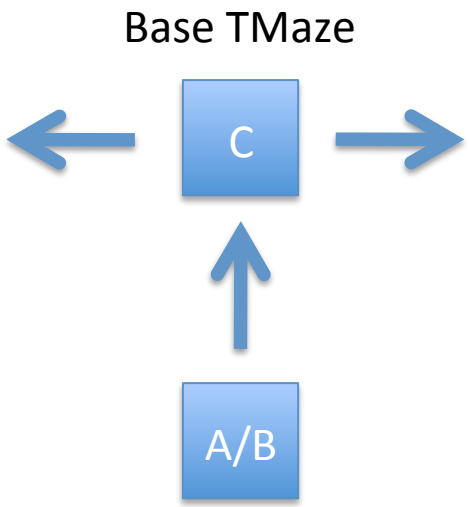
Gated WM

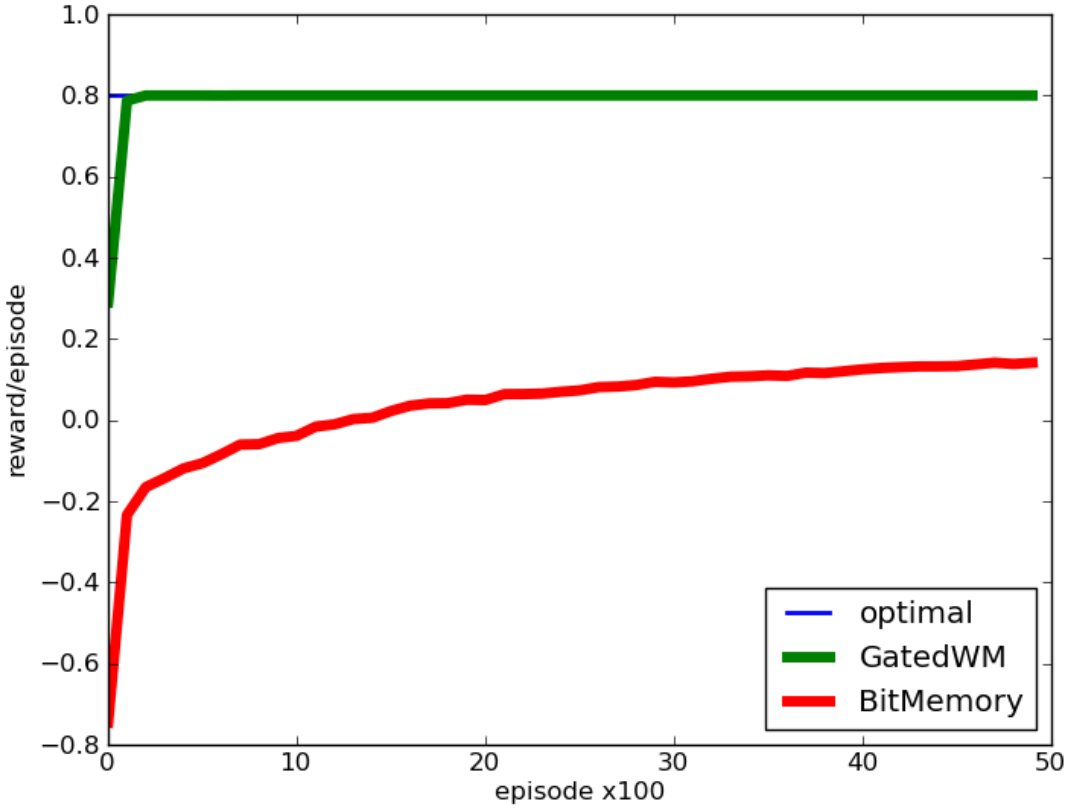


- Internal action toggles between memory states
- Less expressive
- Ungrounded knowledge

- Internal action stores current observation
- More expressive
- Grounded knowledge

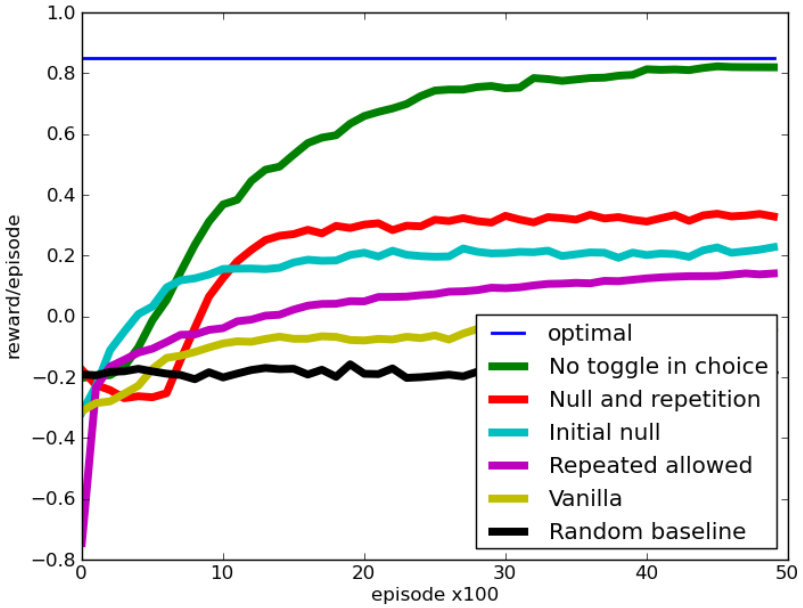
TMaze


Base TMaze
C
A/B

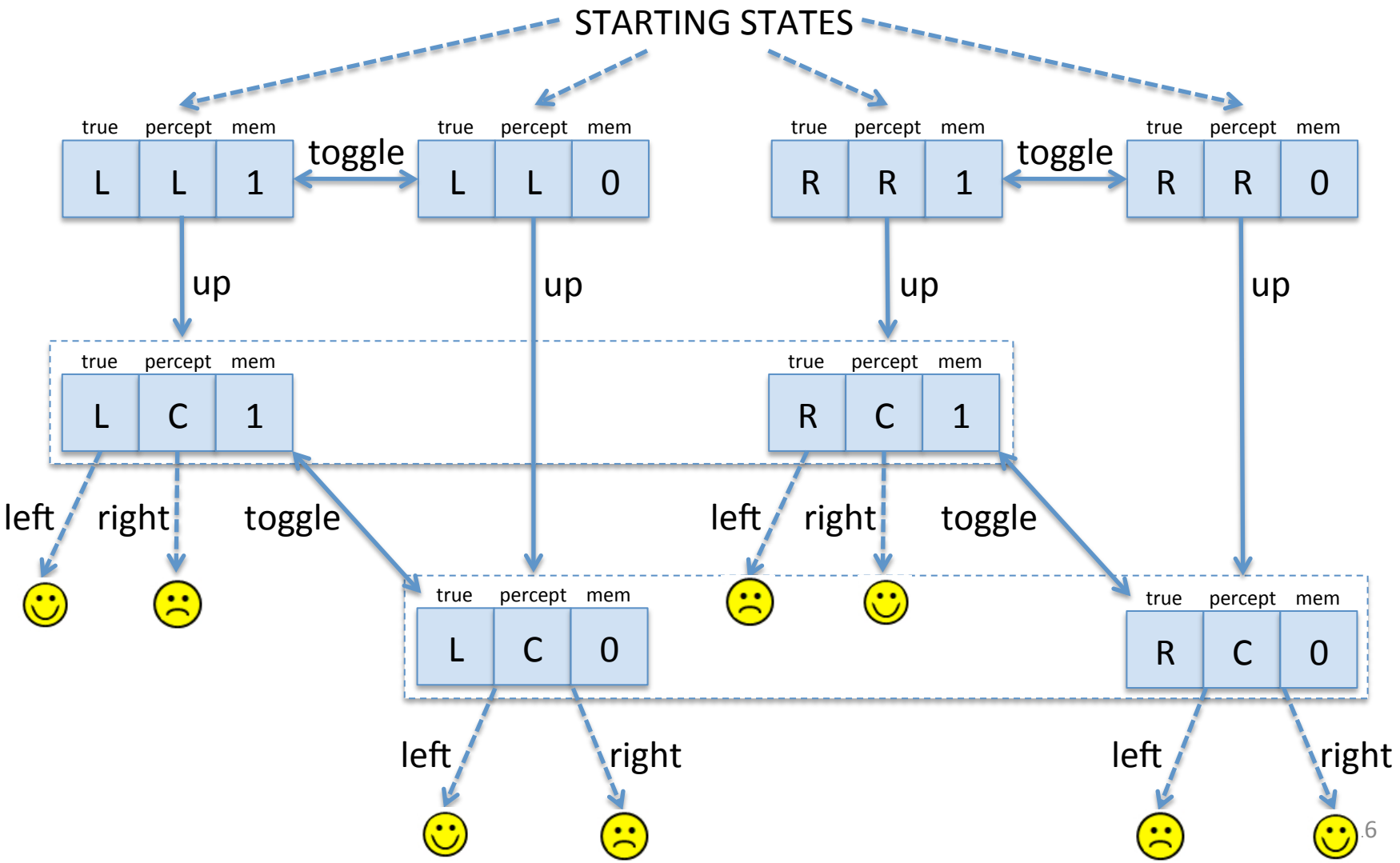

1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
reward/episode
0
10
20
30
40
50
episode x100
optimal
GatedWM
BitMemory

Bit Memory & TMaze

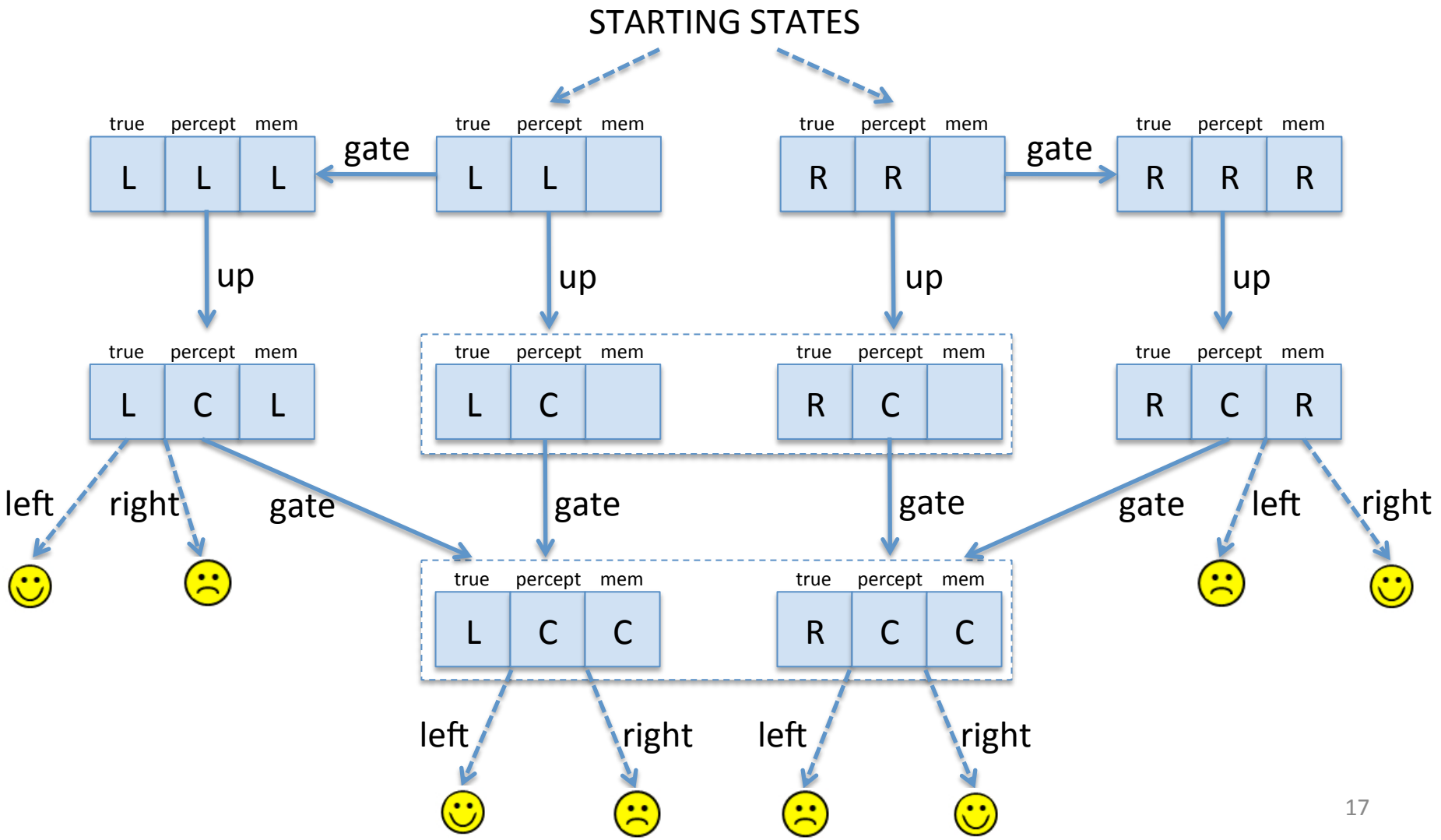
- Methodology:
  - Modify memory to attribute blame
- Interfering behavior in choice location
- Doesn't manifest with GWM



State Diagram: Bit Memory TMaze

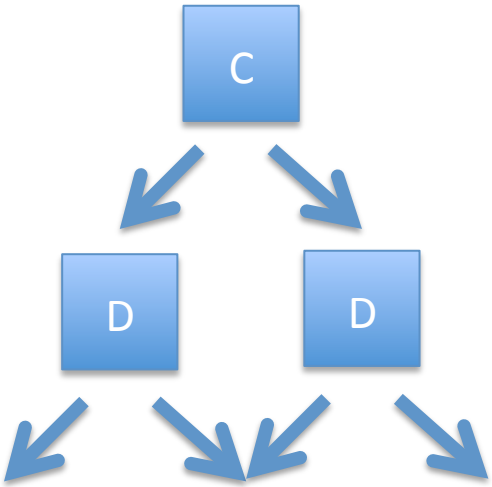


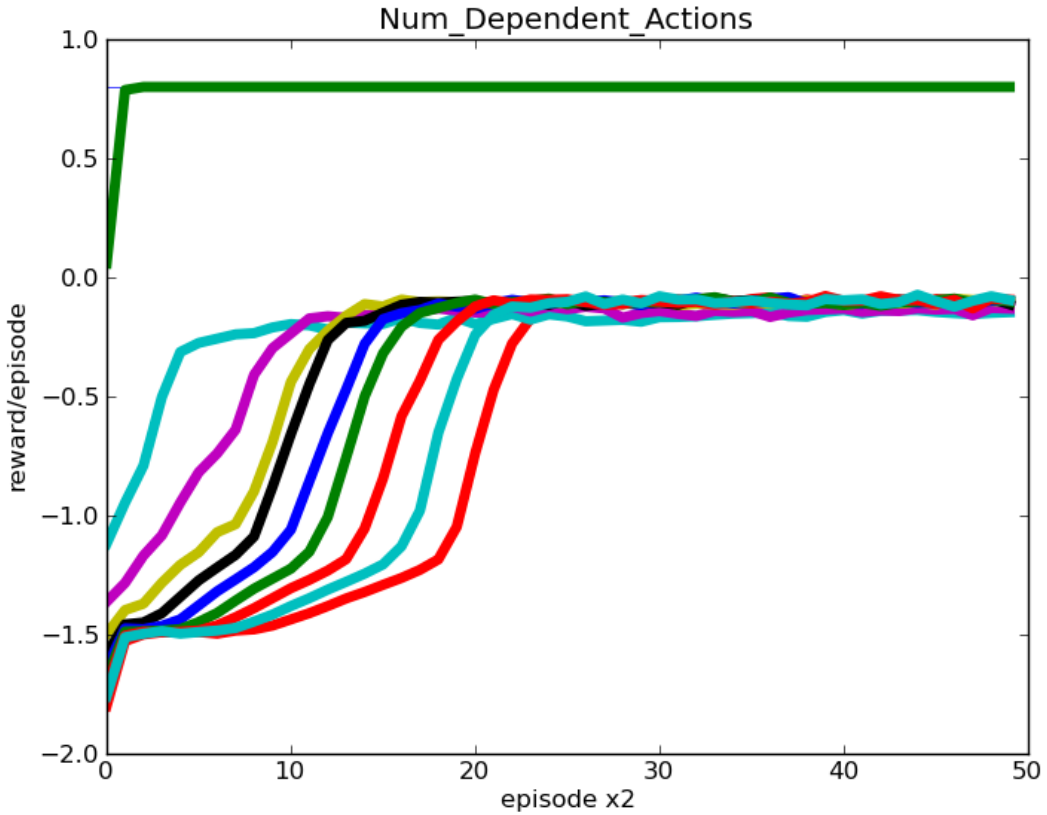
State Diagram: GWM TMaze



Number of Dependent Actions

Dependent Actions





What We've Learned

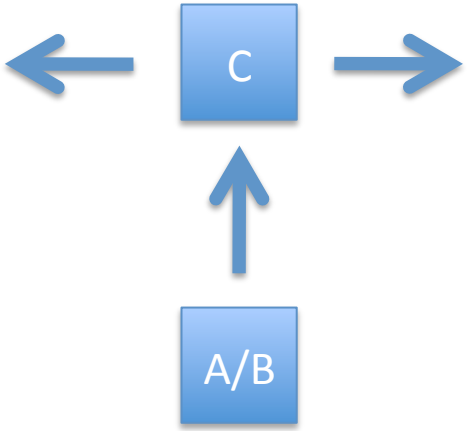
- Our machine learning intuition is often wrong (and yours probably is, too!)
- Chicken & Egg Problem
- State ambiguity is very problematic to learning

Chicken & Egg Problem

- Prospective uses of memory are hard
- Case study: bit memory & TMazes

Bit memory

Base TMaze



Chicken & Egg Problem:

- Must learn an association between 1 & 0 and A & B
- Must learn an association between 1 & 0 and left & right
- To be effective, can't self-interfere with memory in C!

- Endemic across memory models

Implications for Soar

- Soar natively supports learning internal acts.
- Next step: learning to use Soar's memories
- Learning alongside hand-coded procedural knowledge is potentially strong approach
- Soar got the WM model right
- RL will never be a magic bullet

Nuggets & Coal

- Nearly finished!
- Better understanding of RL + memory, and thus Soar 9
- Parameterized, empirical evaluations of RL gaining traction
- Optimality not only metric of performance

- Not quite finished!
- Qualitative results, but no closed form results yet
- No recent results for long term memories
- Not immediately applicable to Soar