Using Imagery to Simplify Perception

29th Soar Workshop

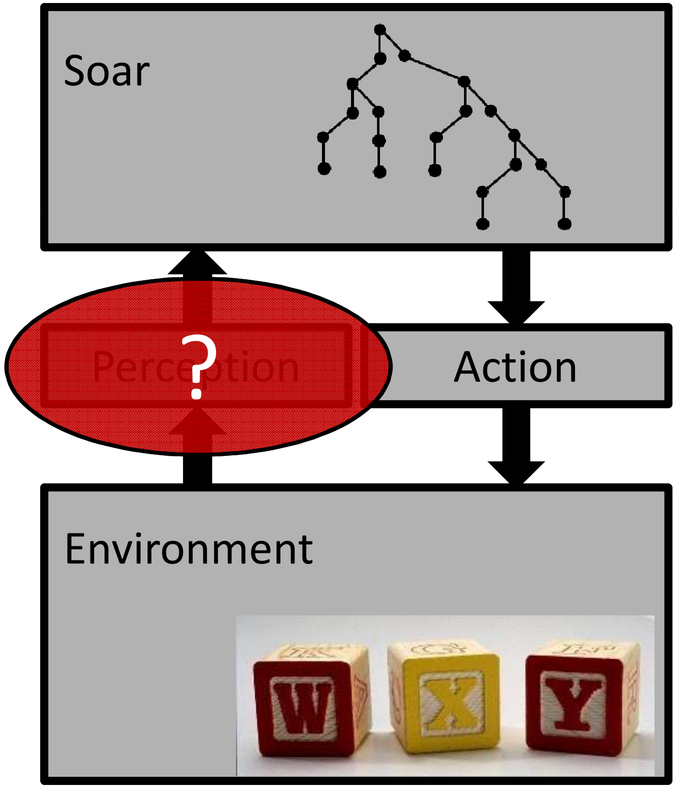
Samuel Wintermute
University of Michigan

SVS Project Background

- Soar Visual/Spatial (SVS) adds visual and spatial processing to Soar
- Previous workshops:
  - Architecture was presented
  - Pieces of it were examined
  - Agents were shown
- This year: studying *sufficiency* and *generality* in representing spatial problems
  - Giving Soar the tools it needs to solve lots of spatial problems

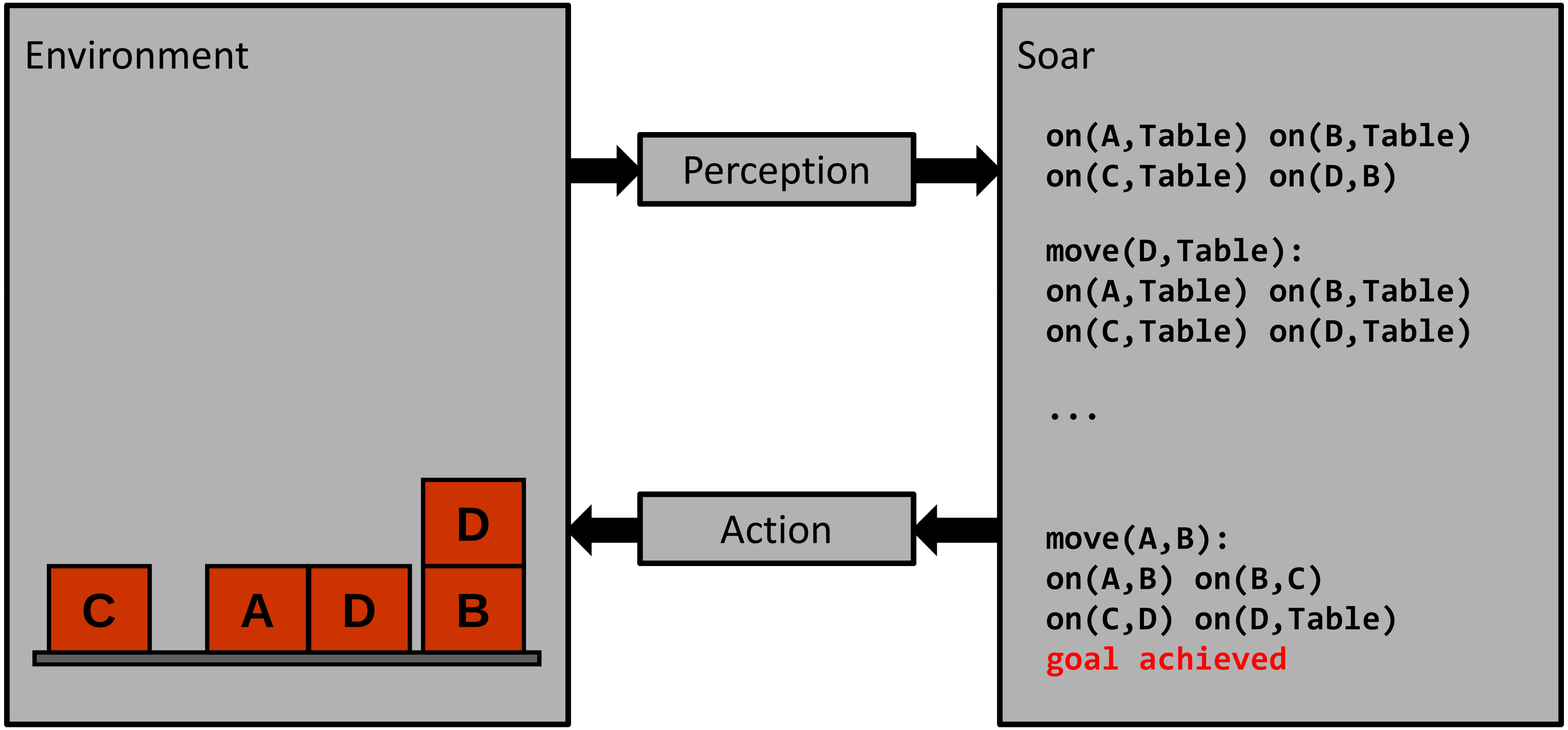
Getting Symbols from Sensors

- Goal: Allow Soar to solve arbitrary spatial problems
- Problem state information in Soar is (almost always) abstract
- Basic perceptions available to an embodied agent might have much more detail
- What perceptual information should be provided to Soar?
- This problem is about *what* is calculated by the perception system, not about *how* to calculate it



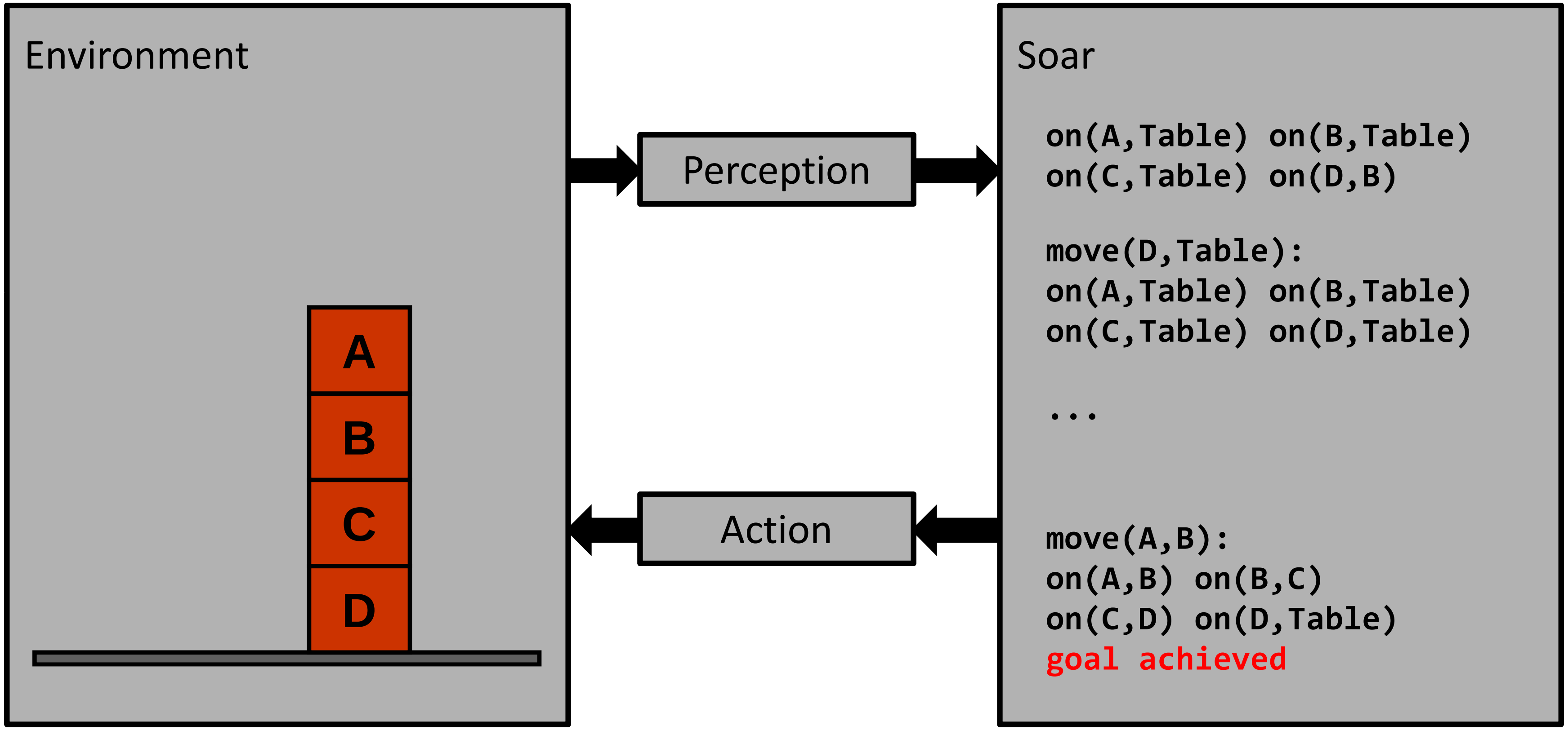
Motivating Example (1)

- Symbolic Planning in the Blocks World



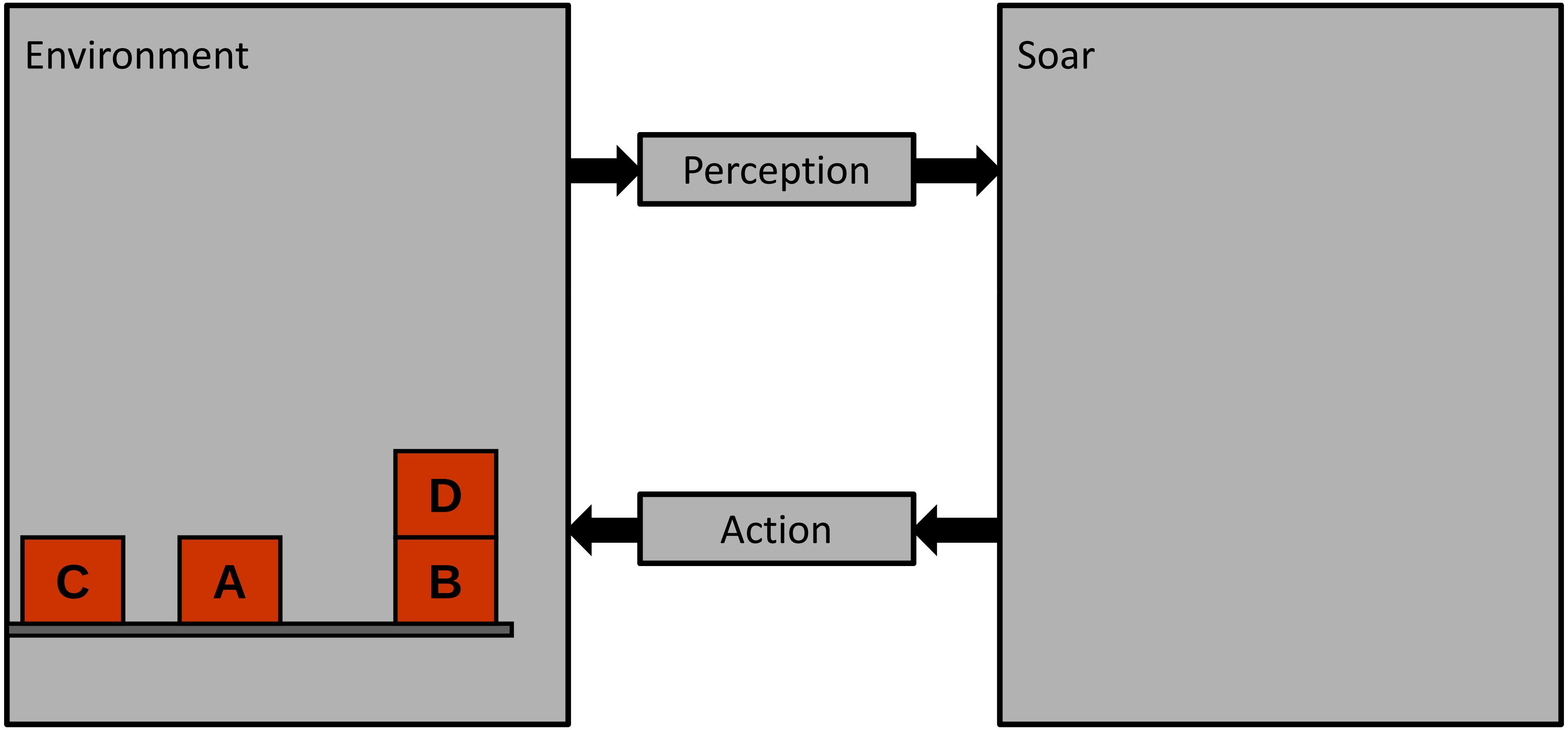
Motivating Example (1)

- Symbolic Planning in the Blocks World



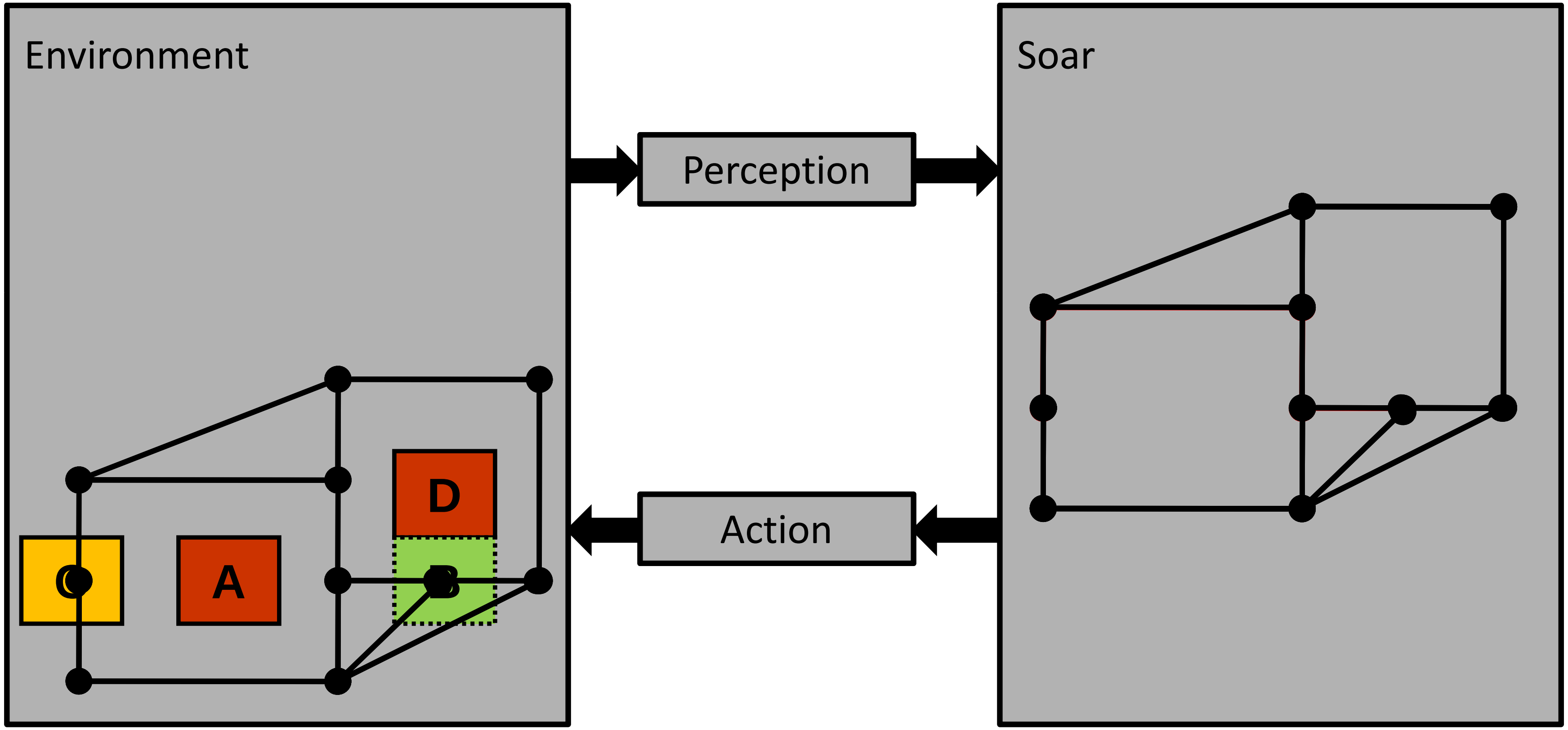
Motivating Example (2)

- Motion Planning



Motivating Example (2)

- Motion Planning (robot, obstacles, goal)



Problem Dependencies in Perception

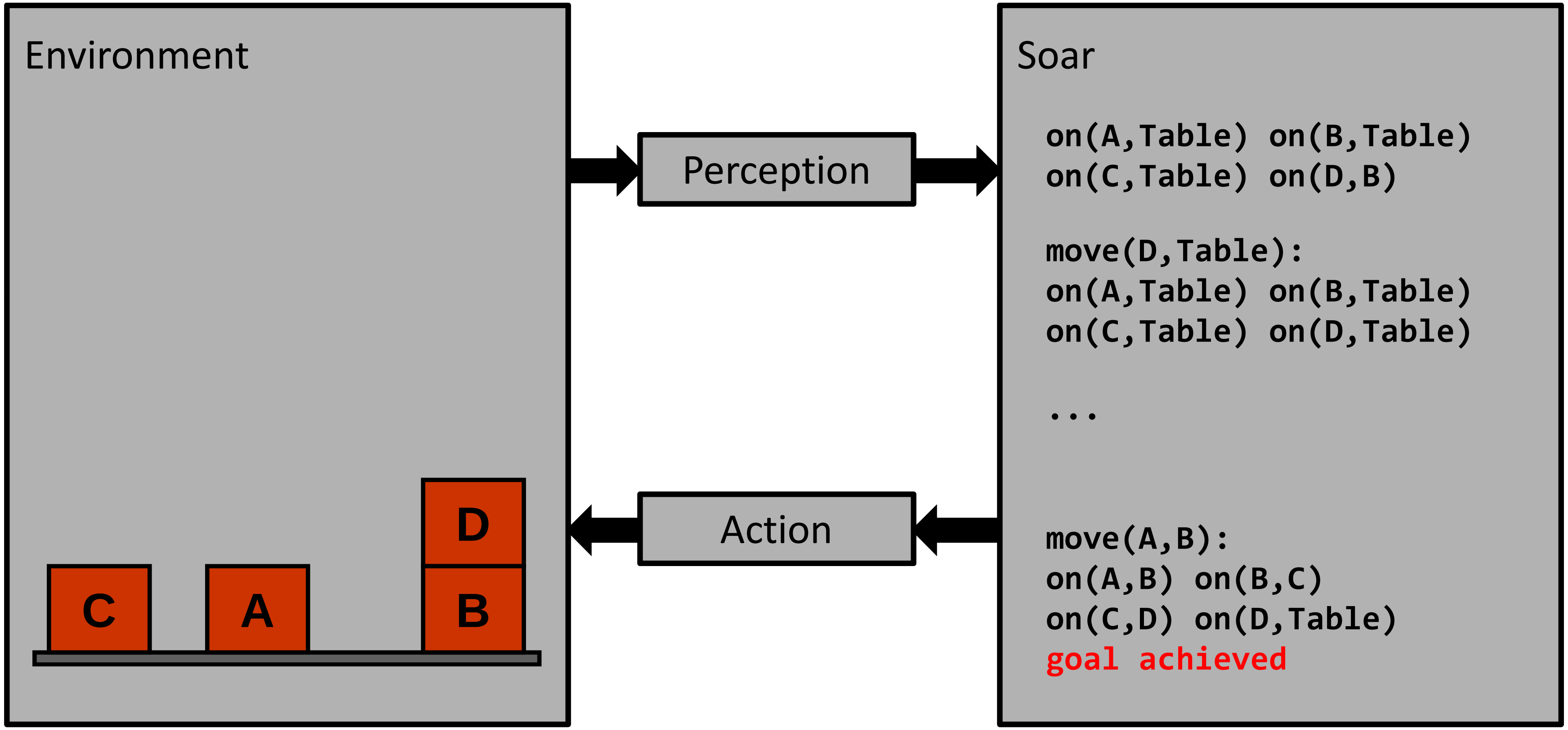
- Problems can be geometrically similar, but require very different symbolic abstractions
  - Any given agent only has one perception system
  - An agent may need to solve unforeseen problems

- Possible solution: space -> symbol transformation can be considered task knowledge.
  - Soar rules for blocks world would map a set of coordinates to “on” relationships
  - Soar rules for navigation would take a set of coordinates, calculate configuration space, and determine locations and adjacencies
  - This is certainly possible, but math-intensive and hard to learn

- Possible solution: abstractions can be symbolically composed from problem-independent primitives
- These domains are also very simple…

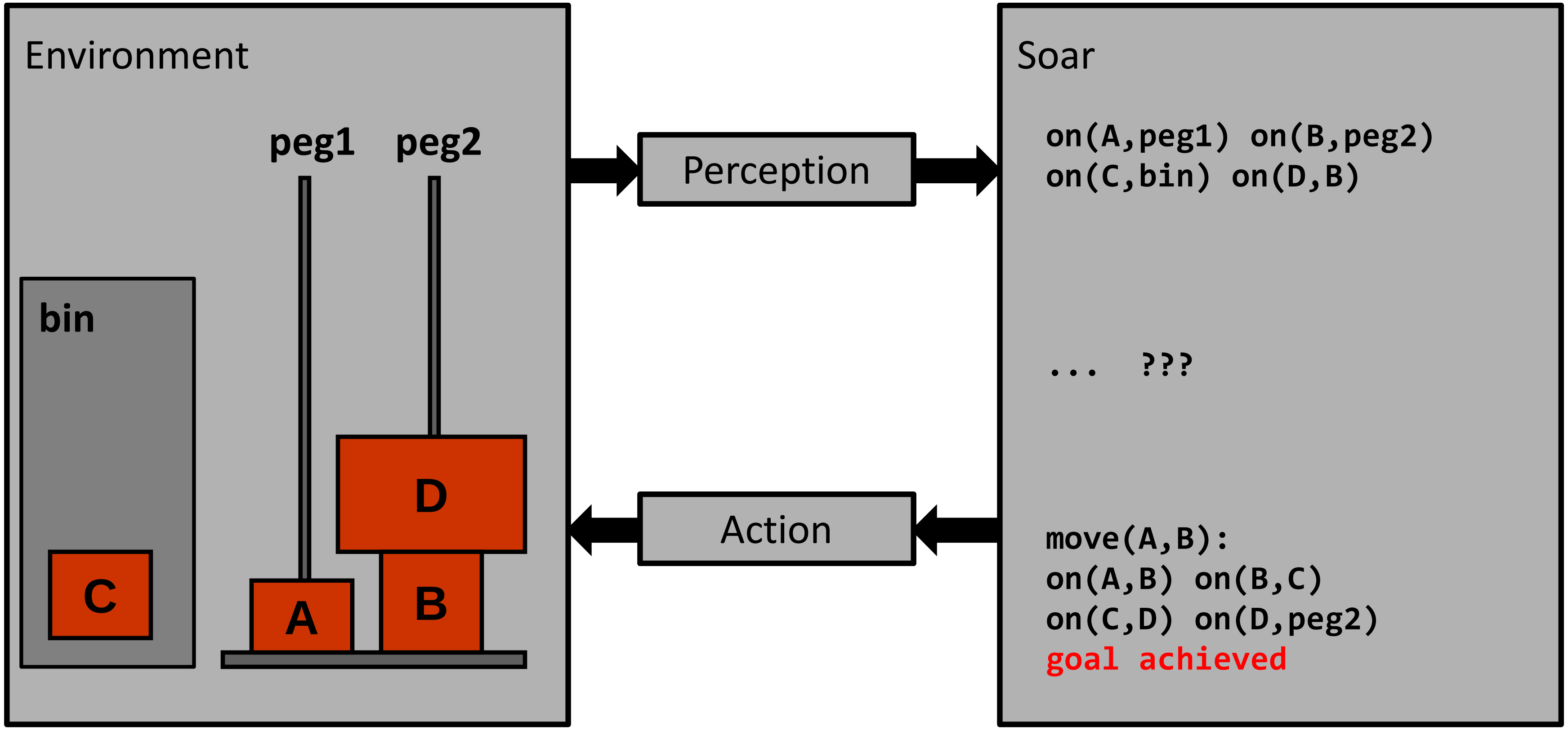
Motivating Example (3)

- Symbolic Planning in the Blocks World



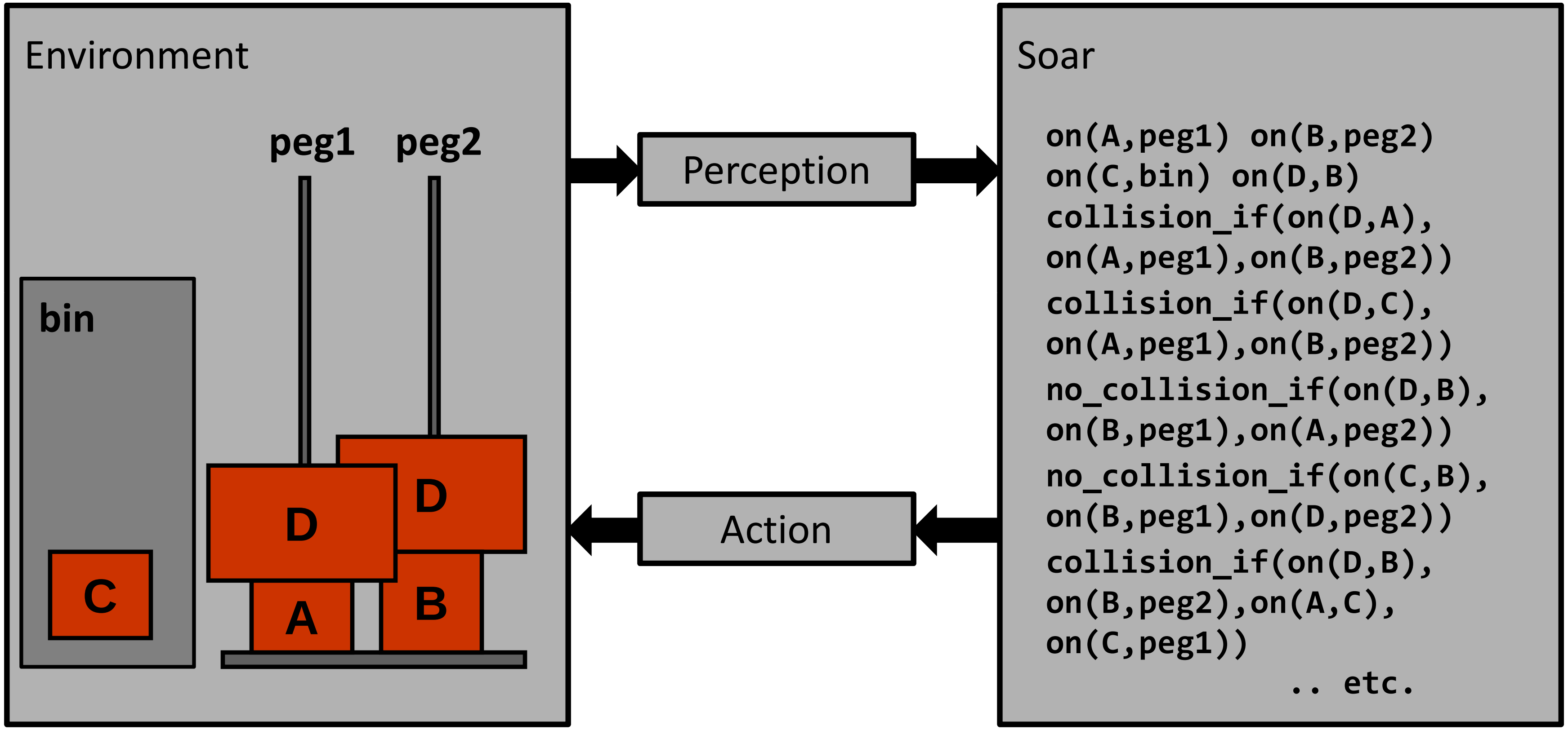
Motivating Example (3)

- Symbolic Planning in the Pegged Blocks World



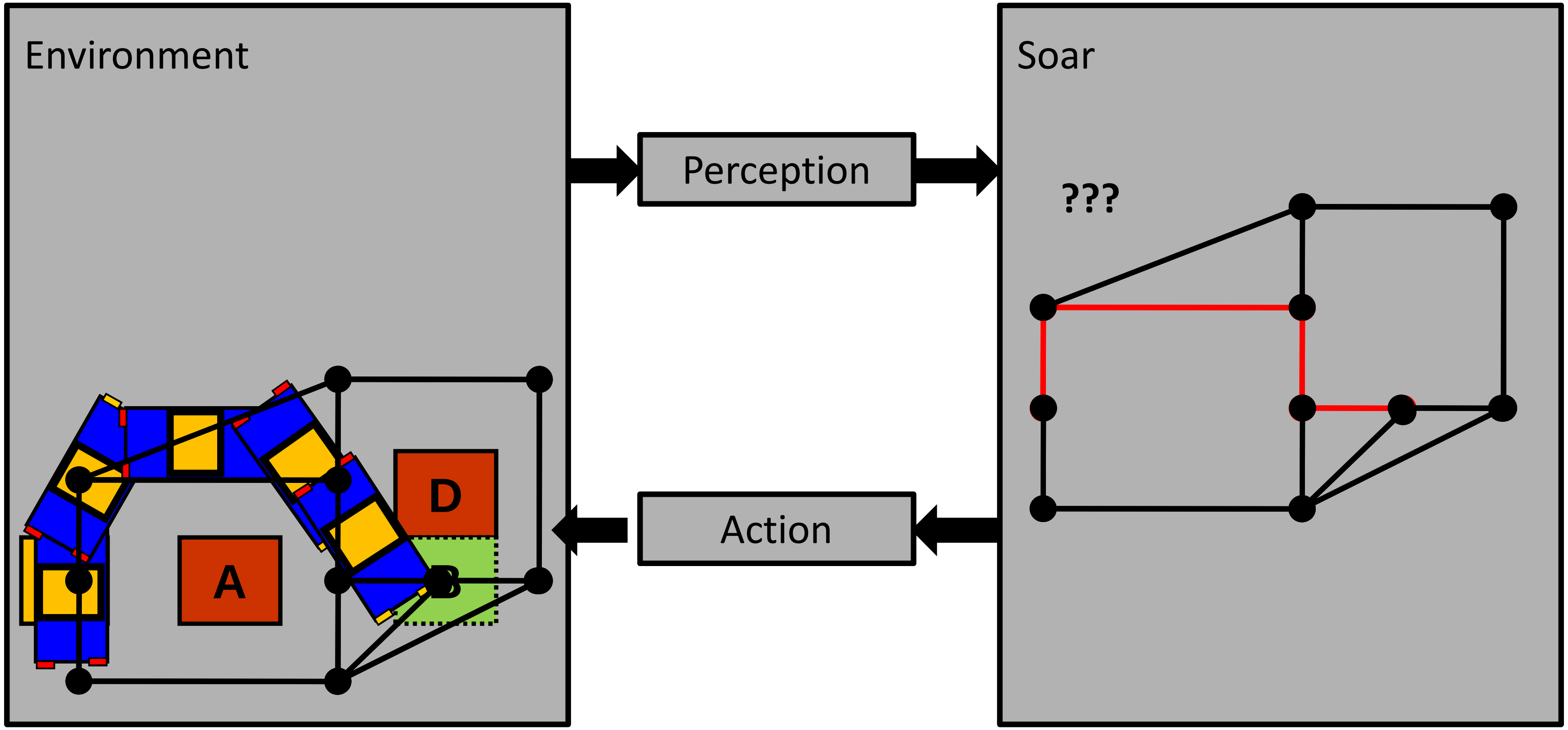
Motivating Example (3)

- Symbolic Planning in the Pegged Blocks World



Motivating Example (4)

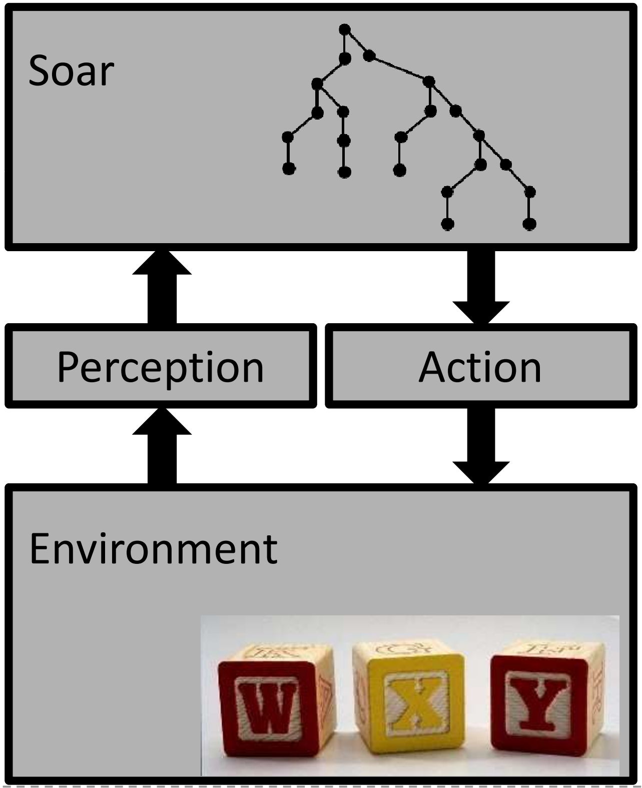
- Non-Holonomic Motion Planning (robot, obstacles, goal)



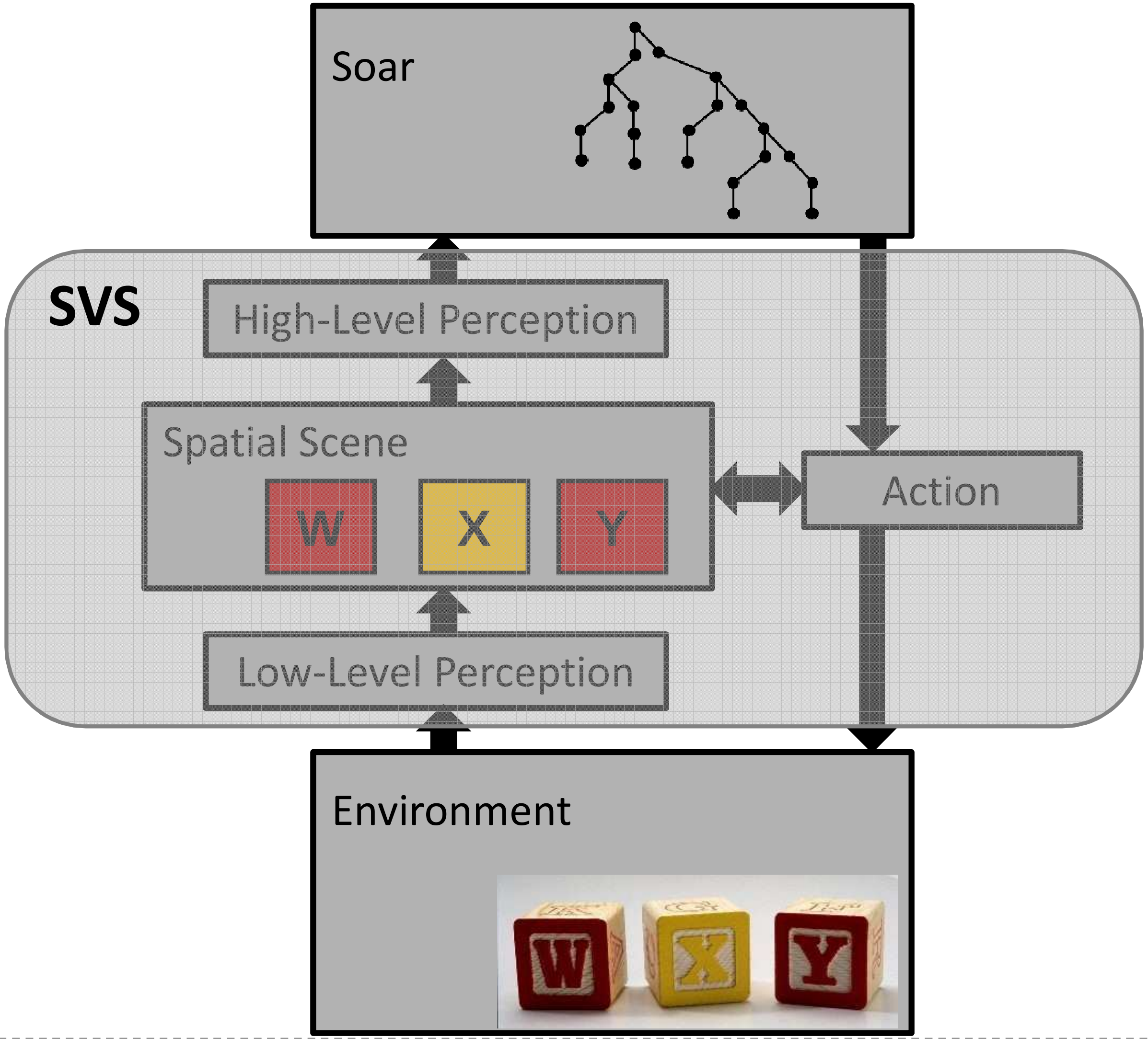
Difficulties in perception

- Different problem domains require very different qualitative symbolic relationships
  - But any given agent only has one perception system
- For some problems, computing *any* useful qualitative symbolic representation is difficult (or impossible)
  - And if it is possible, the calculated relationships will be extremely problem-specific
- Solution: reconsider one-way perception

Imagery

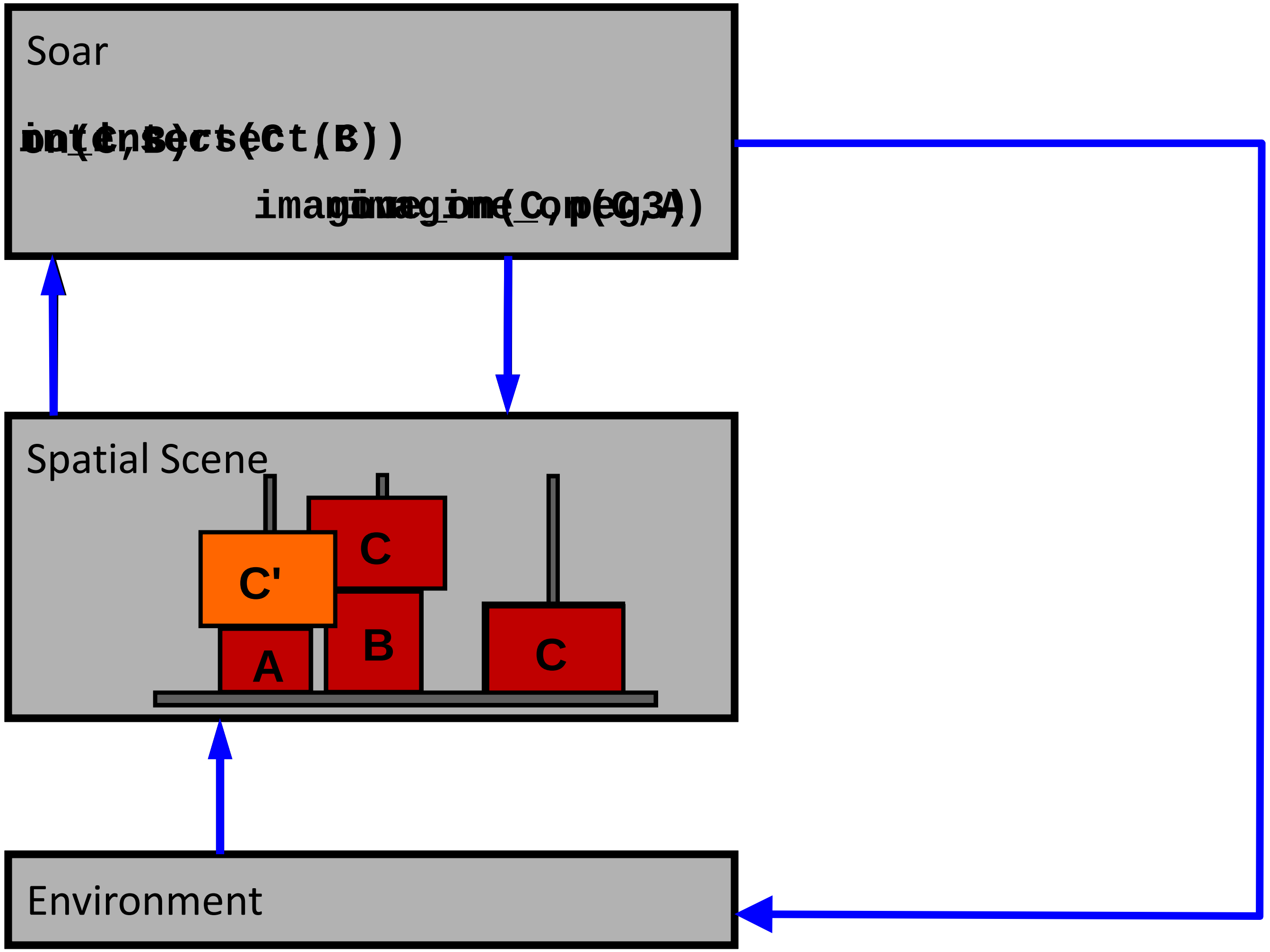


Imagery

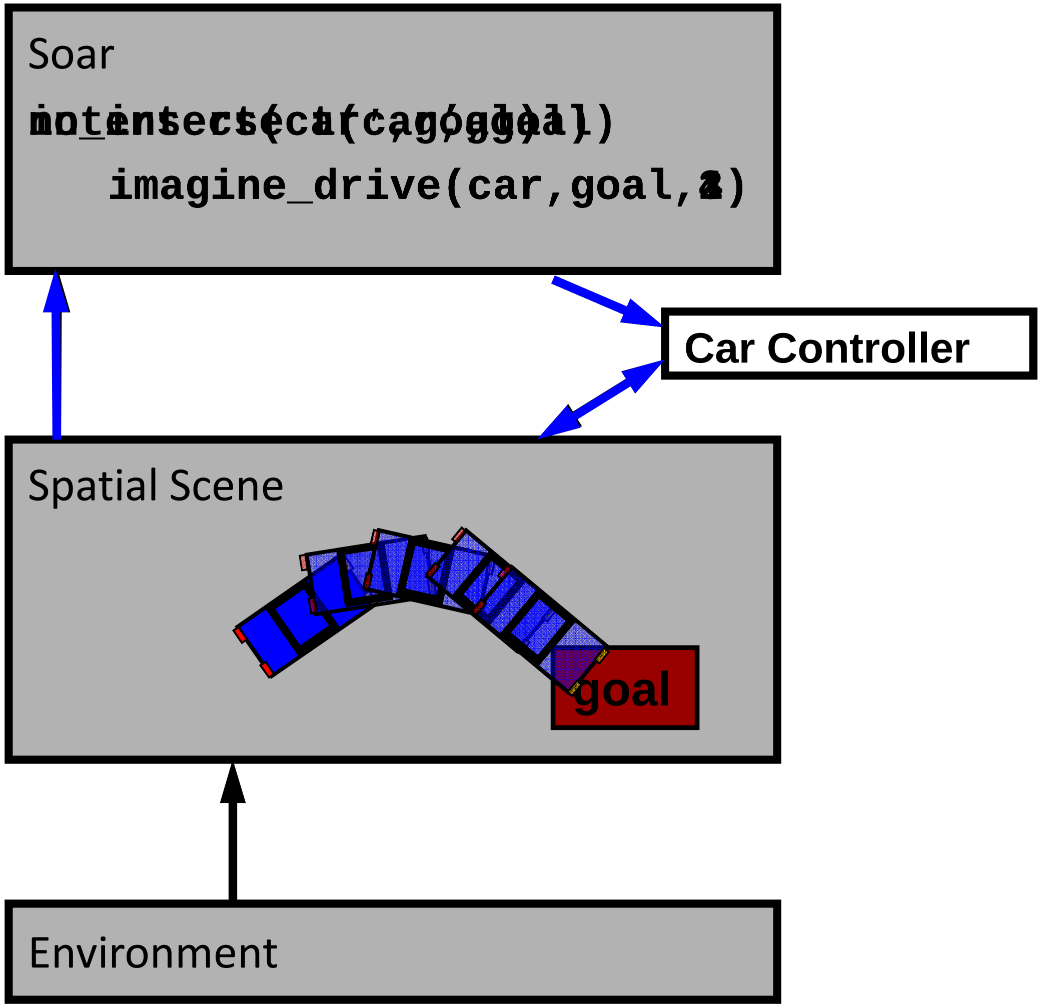


Soar
SVS
High-Level Perception
Spatial Scene
W
X
Y
Action
Low-Level Perception
Environment

Imagery in the Pegged Blocks World



Imagery for Non-Holonomic Motion

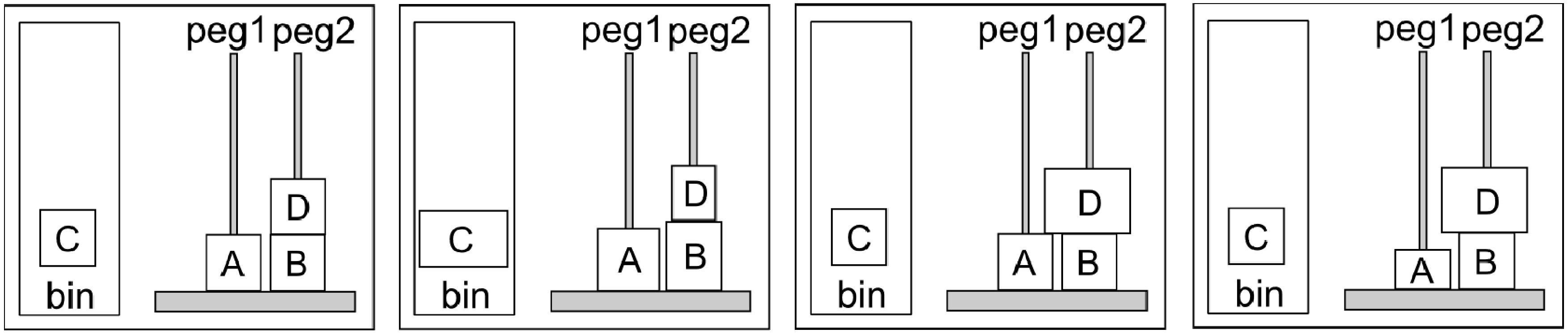


Soar / Imagery Interface

- High-level perceptions to Soar:
  - Object identities
  - Object topology (intersecting or not)
  - Object distances
  - Object directions (left-of, right-of, etc.)
- Imagery actions:
  - Qualitative predicate projection (e.g., imagine A on B)
  - Motion simulation
  - Memory retrieval

Pegged Blocks World Agent

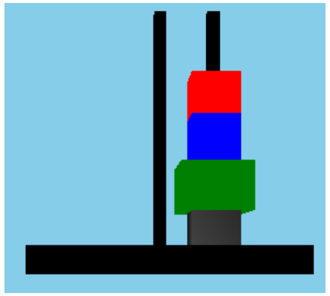
- All instances considered have the same abstract initial state and goal
- Any instance may be in one of four cases where the optimal plan differs

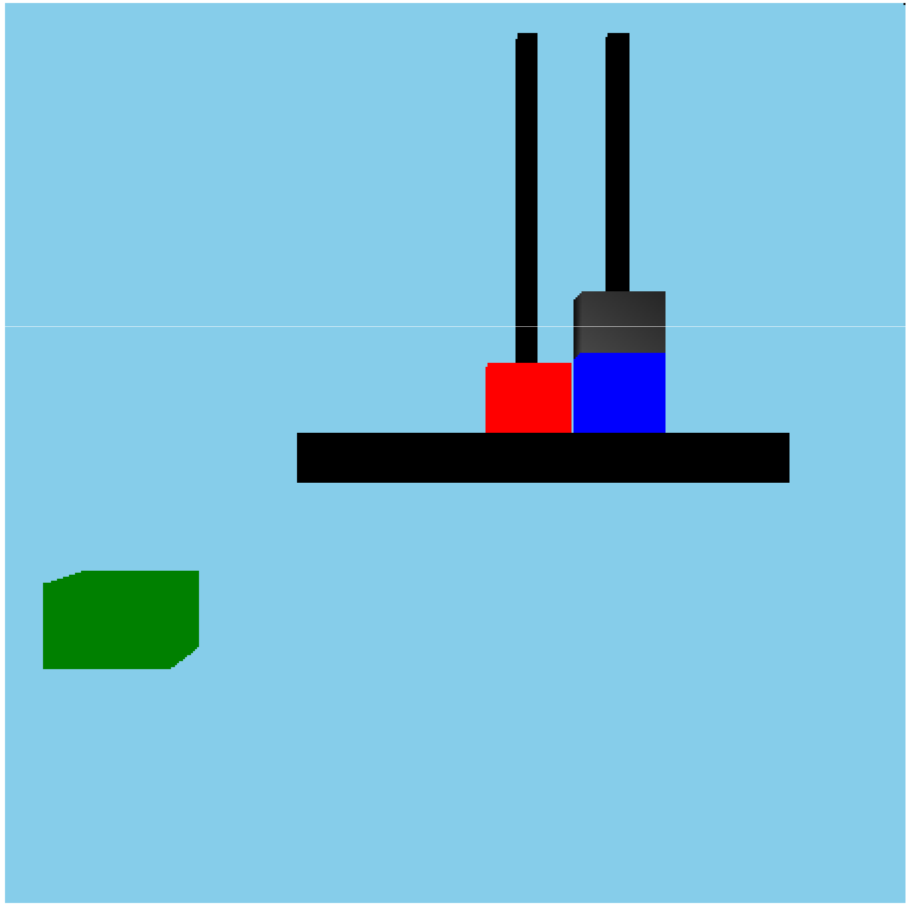


- Soar with SVS can encode a plan to get (almost) optimal behavior in all four cases
  - This would be extremely difficult without imagery

Pegged Blocks World Agent

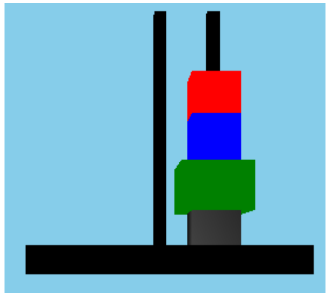
- goal :



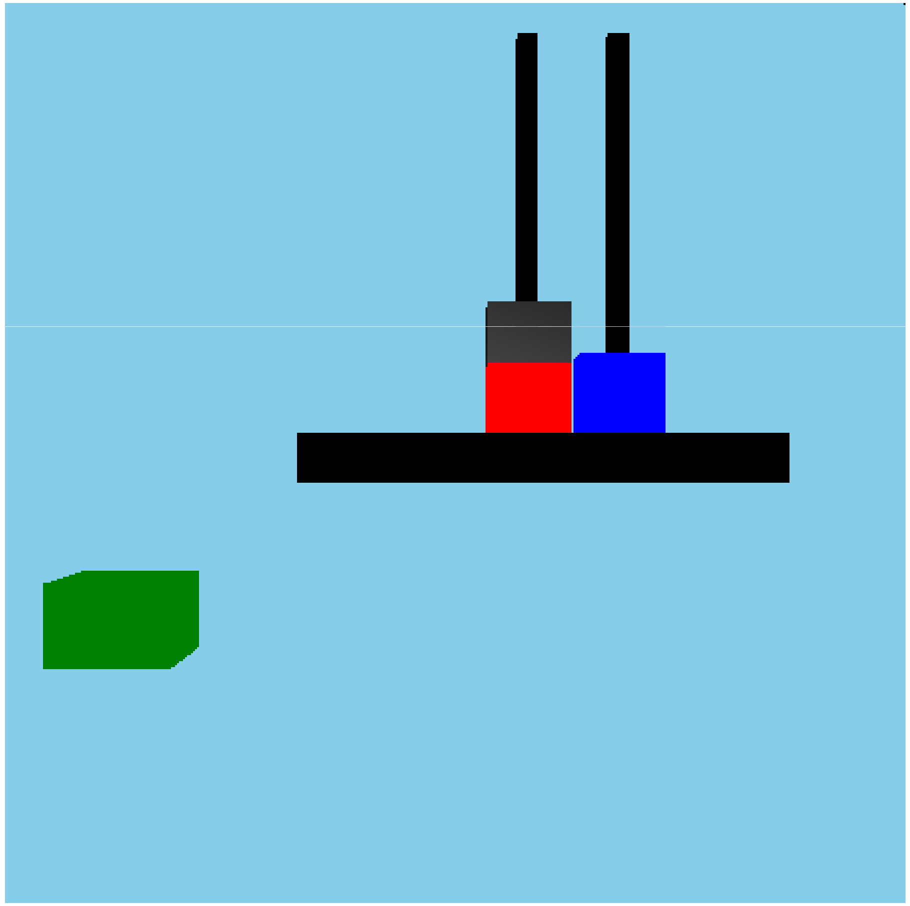


Pegged Blocks World Agent

- goal :

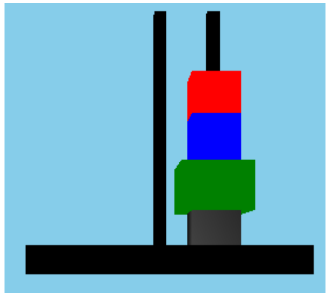


- move black above red, if it fits

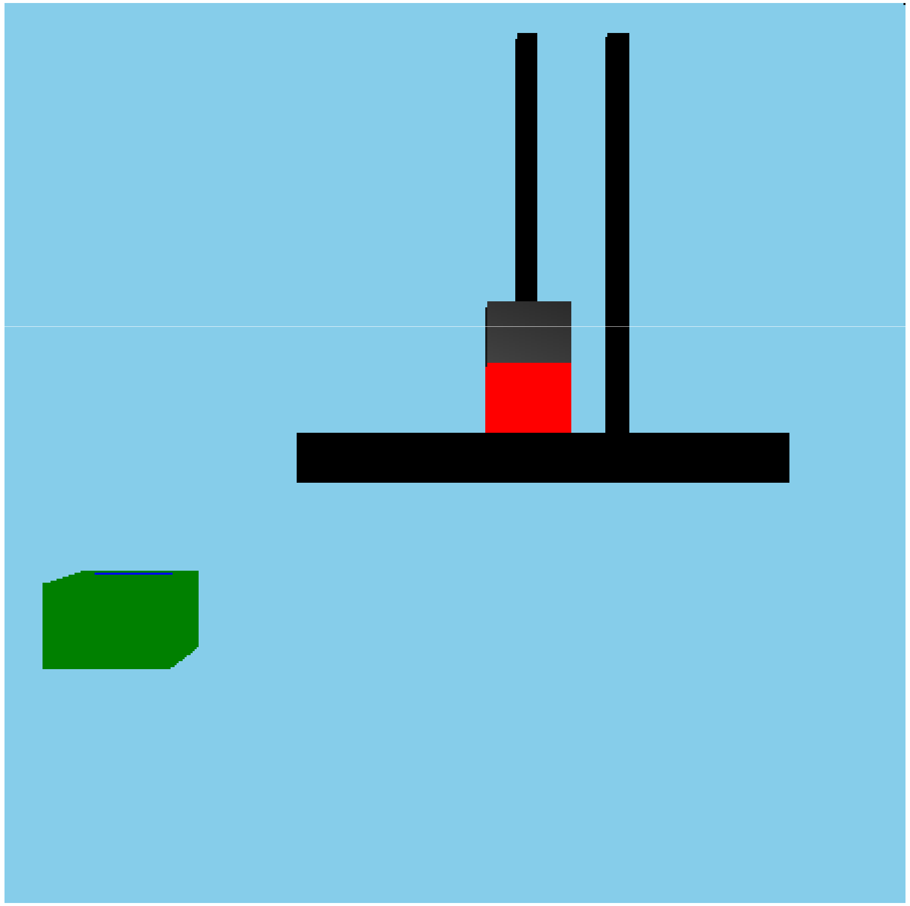


Pegged Blocks World Agent

- goal :

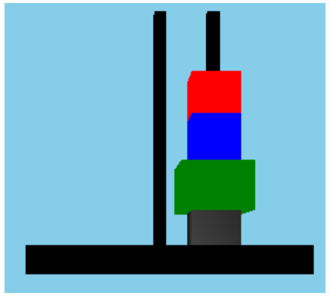


- move blue to bin

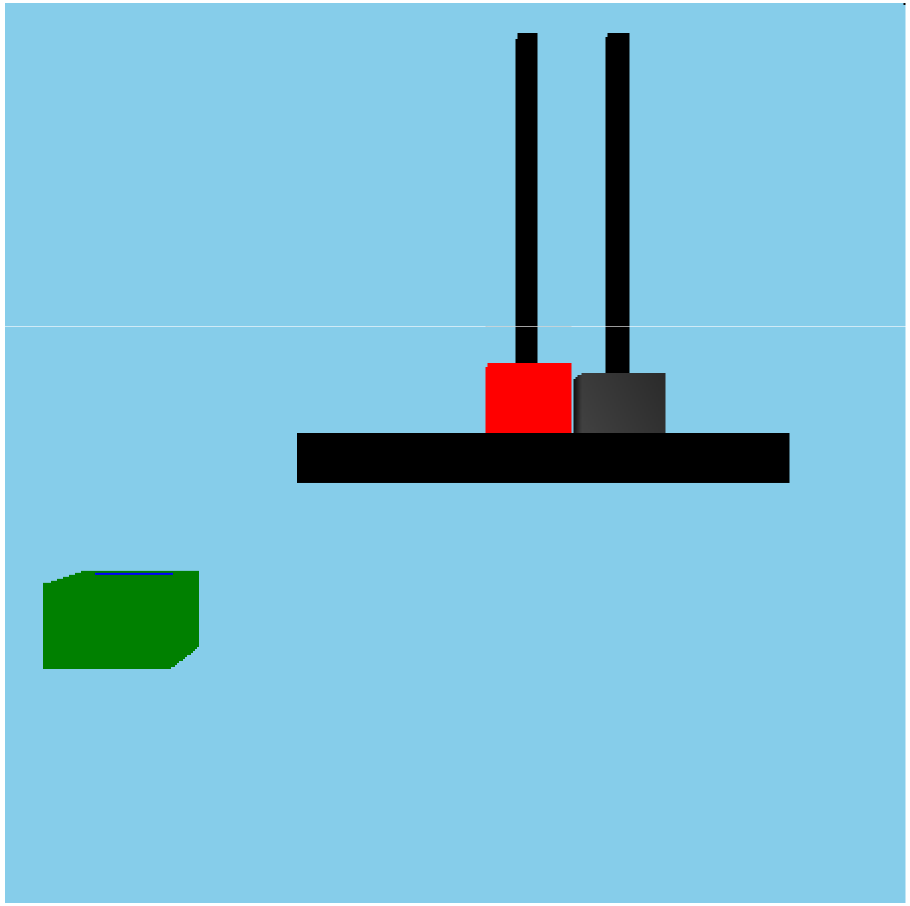


Pegged Blocks World Agent

- goal :

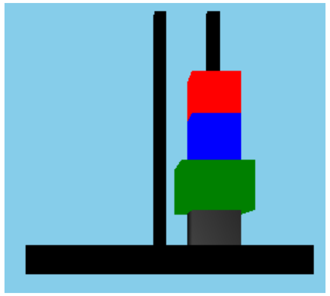


- move black next to red, if it fits

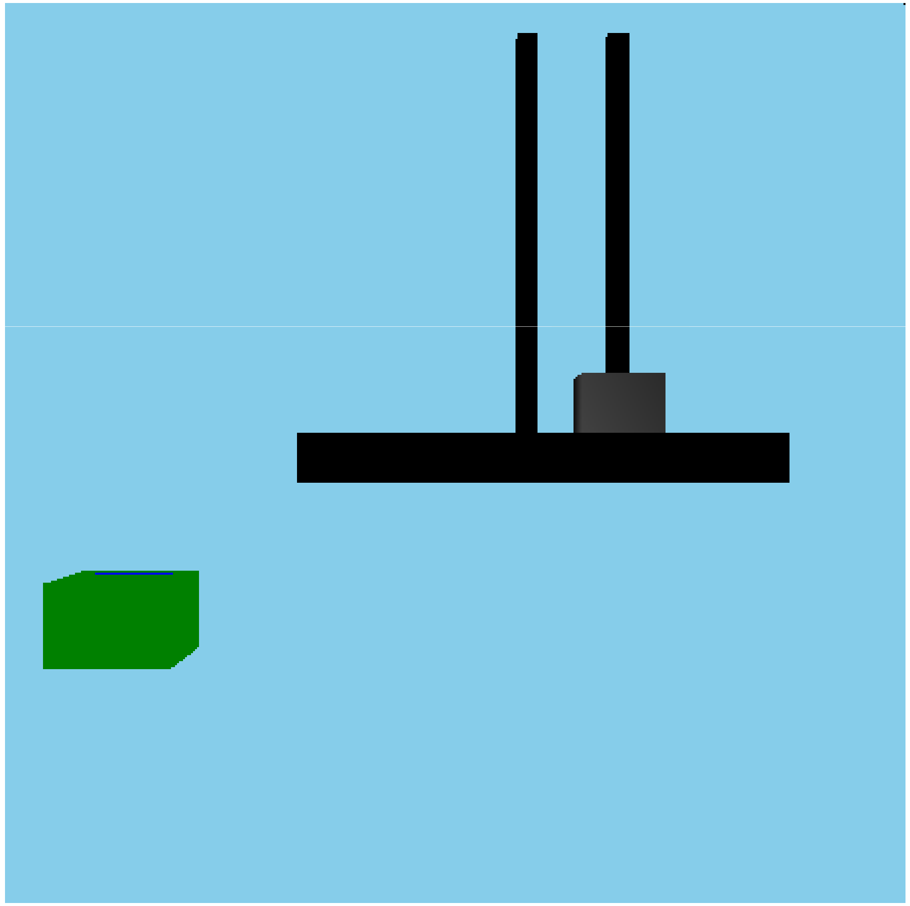


Pegged Blocks World Agent

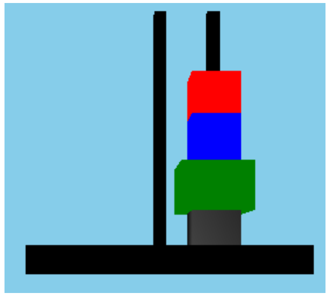
- goal :



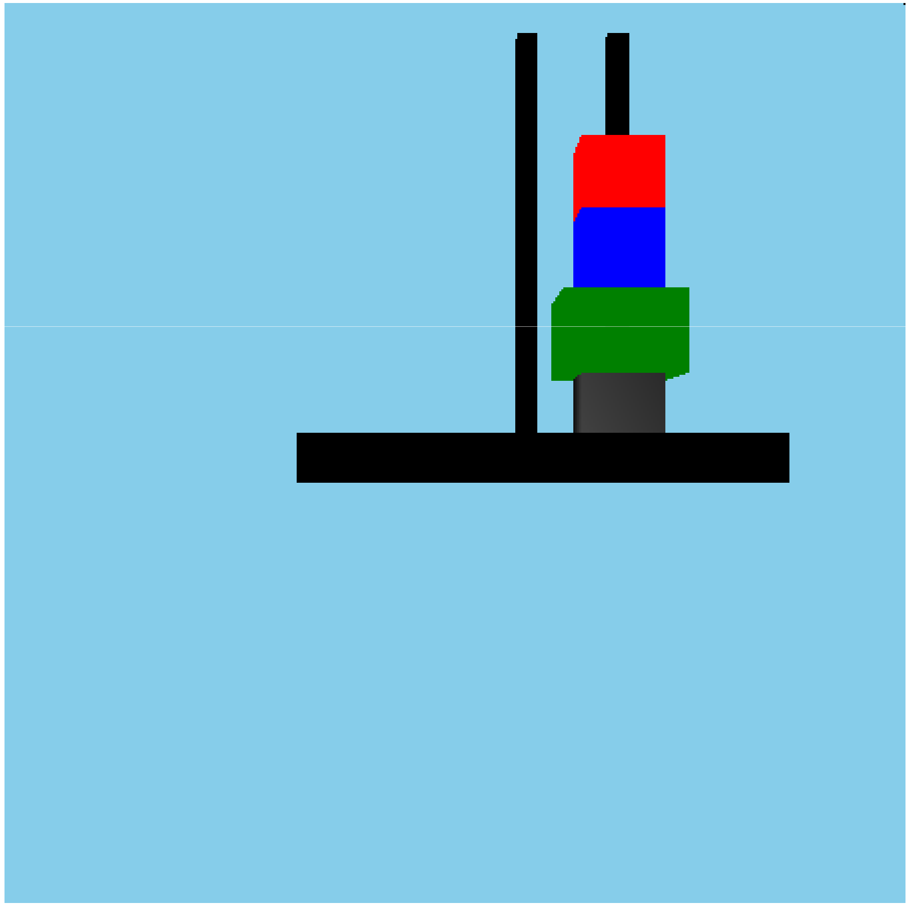
- move green above black, if it fits
- else, move red to bin



Pegged Blocks World Agent

- goal : 

- build goal stack from bin



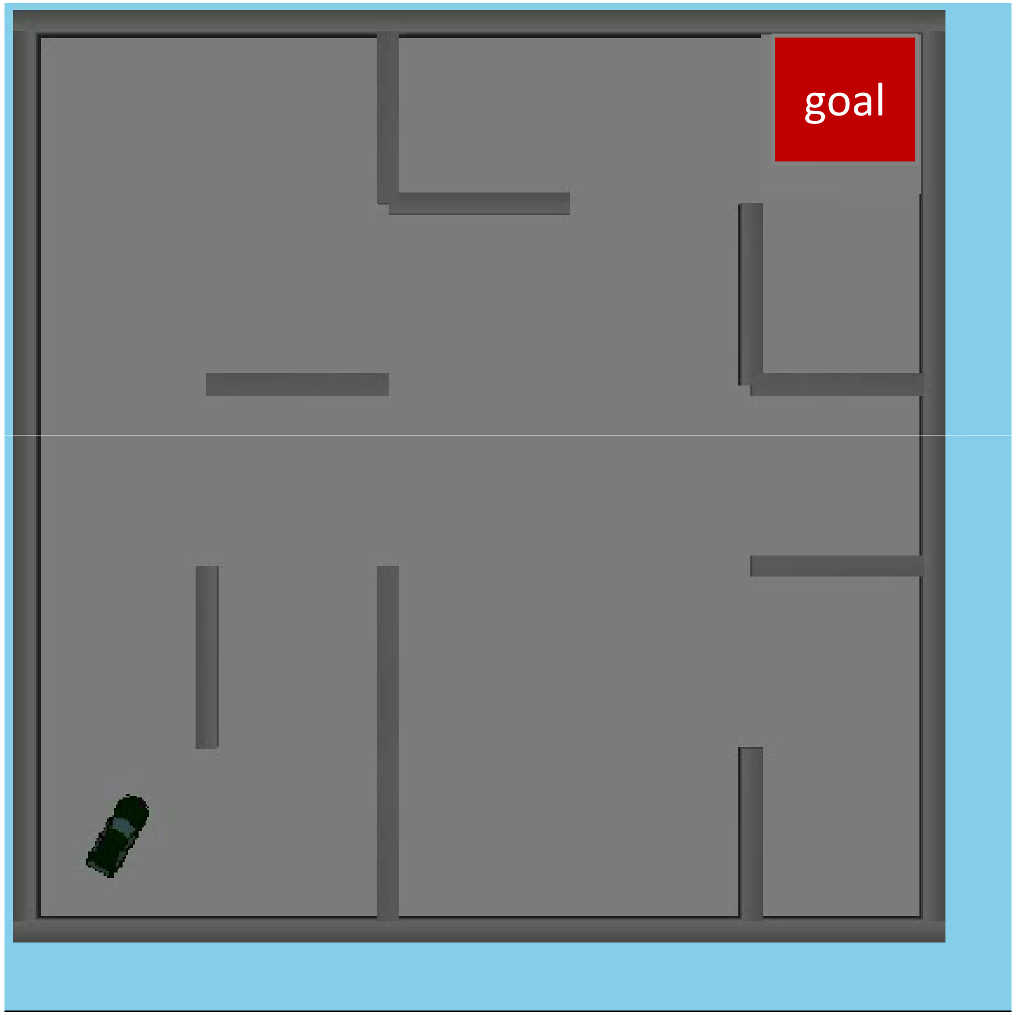
Motion Planning Agent

- Common robotics problem: must determine a sequence of actions to move from place to place in a fully-observed world
- Abstraction can be very difficult
- Result: sampling-based motion planning
  - RRT Algorithm: through simulation, build a tree of reachable configurations until the goal is reached

- This algorithm has been instantiated in Soar/SVS.

RRT Motion Planning

- Controller steers car toward a goal, biasing steering away from obstacles
- Soar keeps track of tree of possible configurations, and chooses which to expand next



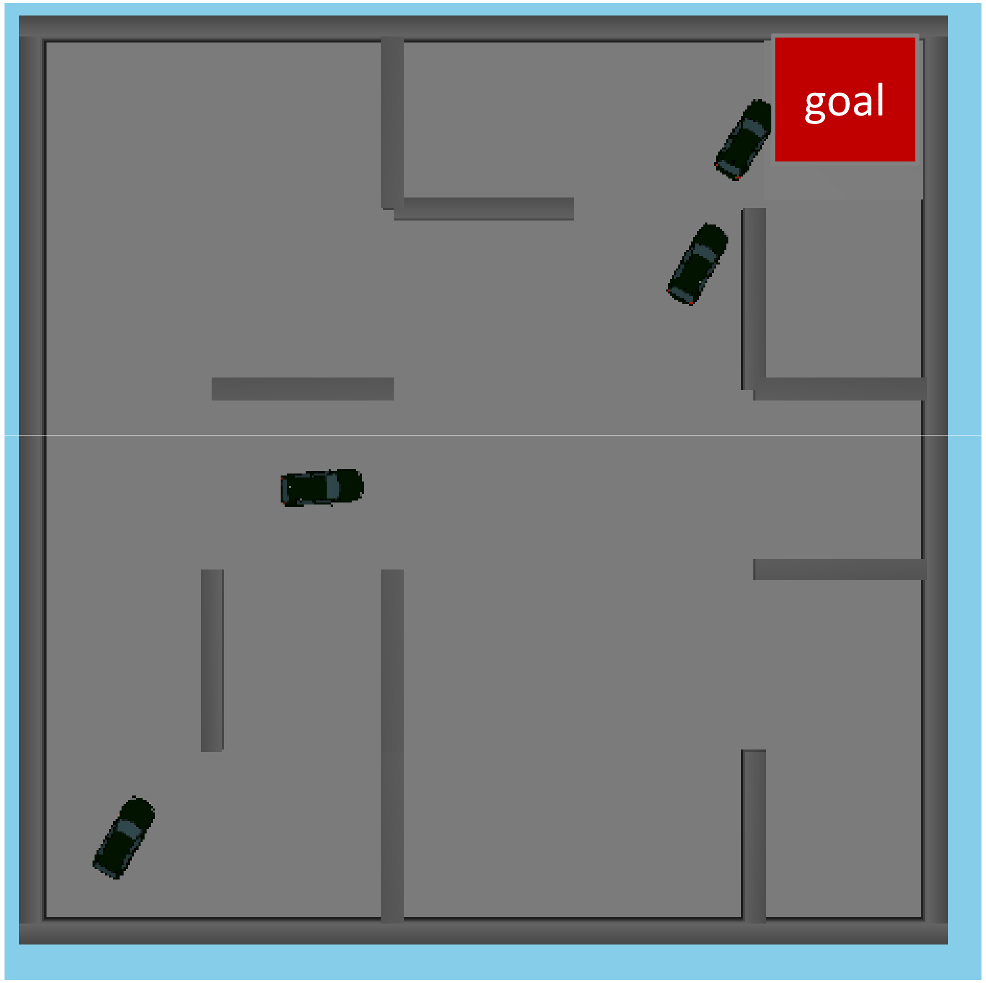
RRT Motion Planning

- Controller steers car toward a goal, biasing steering away from obstacles
- Soar keeps track of tree of possible configurations, and chooses which to expand next

RRT Motion Planning

- Controller steers car toward a goal, biasing steering away from obstacles

- Soar keeps track of tree of possible configurations, and chooses which to expand next



High-Level Perceptions Used

- Pegged Blocks World:
  - **`intersecting(X,Y); above(X,Y); distance(X,Y)`**
    - Symbolically composed to on(X,Y) and collision(X,Y) predicates
- RRT Motion Planning:
  - **`intersecting(X,Y); in-front-of(X,Y); distance(X,Y)`**
    - Used to determine if car has hit an obstacle or reached the goal, and choose which node to expand
- Low-level perception and object recognition are (hypothetically) the same in both cases.

Imagery Actions Used

- Pegged Blocks World:
  - predicate projection: imagine a copy of a block on top of an existing block, centered relative to it
- RRT Motion Planning:
  - predicate projection: imagine a random point within the floor
  - motion simulation: imagine a copy of the car in the future, given its motion model

Conclusion

- Nuggets:
  - Imagery allows spatial problems to be accurately solved even when no good abstraction is available
  - Imagery reduces complicated perceptual operations to simple operations performed over time
    - This results in a small set of perceptual primitives
    - This allows the same perceptual system to be used in many problem domains
- Coal:
  - Low-level perception is still hard
  - Generality is a hard claim to evaluate
  - Still no software release

References

- SVS
  - S. Wintermute, *An Overview of Spatial Processing in Soar/SVS*, Technical Report, University of Michigan Center for Cognitive Architecture, 2009.
- Pegged blocks world agents
  - S. Wintermute, "Representing Problems (and Plans) Using Imagery," *Submitted to the AAAI Fall Symposium on Multi-representational Architectures*, 2009.
  - S. Wintermute and J.E. Laird, "Imagery as Compensation for an Imperfect Abstract Problem Representation," *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, 2009.
- RRT algorithm
  - S.M. LaValle and J.J. Kuffner Jr, "Randomized Kinodynamic Planning," *The International Journal of Robotics Research*, vol. 20, 2001, p. 378.
- RRT agent
  - S. Wintermute, "Integrating Reasoning and Action through Simulation," *Proceedings of the Second Conference on Artificial General Intelligence*, 2009.