

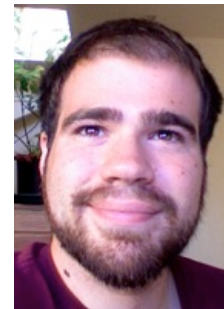
# Angelic Hierarchical Planning



Bhaskara Marthi  
Willow Garage



Stuart Russell  
UC Berkeley



Jason Wolfe  
UC Berkeley

# Levels of Decision-Making

- Task: which object to move where?
- Grasp planner: where to grasp object?
- Inverse kinematics: arm/whole-body configuration
- Motion planner: path through cspace
- Trajectory following: joint torques/velocities

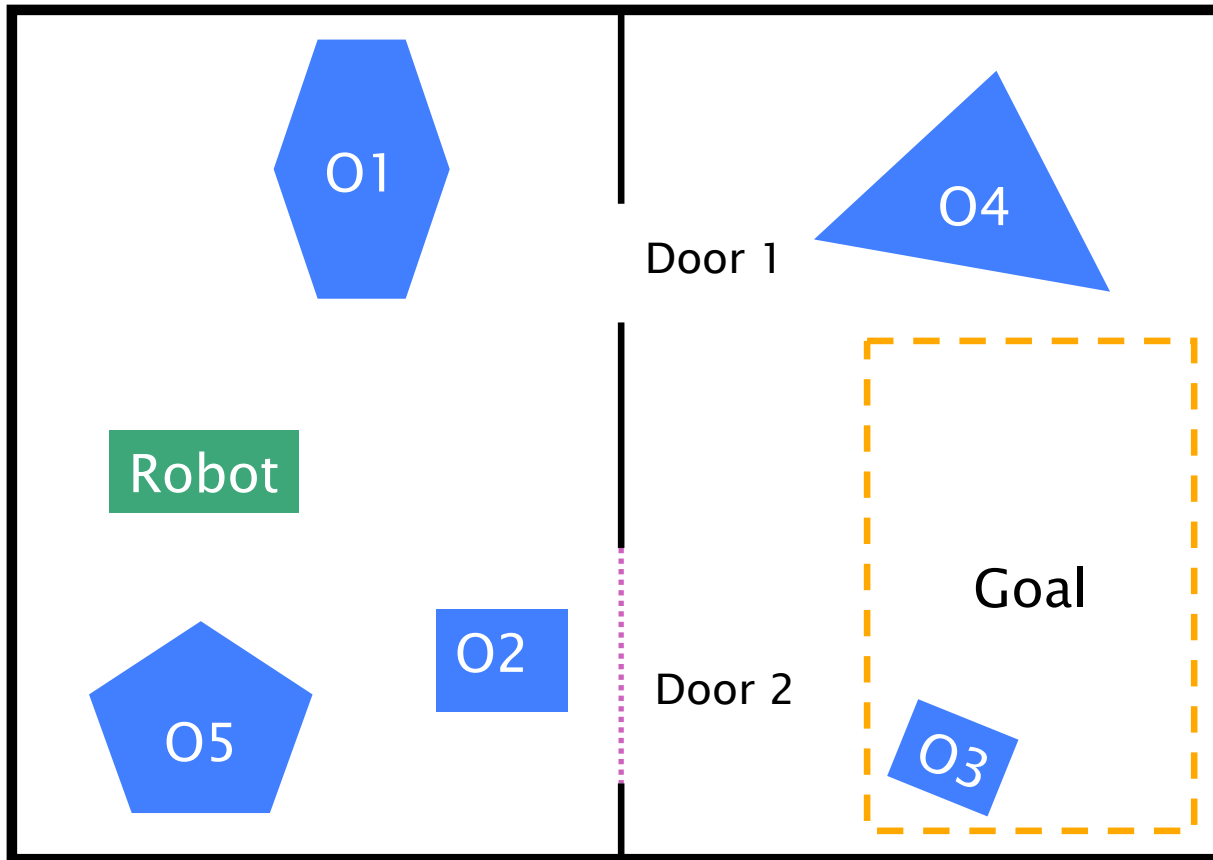
# How to connect levels?

- Standard model: top-down
  - Decide which object to move next
  - Choose grasp point
  - Find IK solution
  - Etc
- Protocol
  - High level gives low level a planning instance
  - Low level returns a single solution or fails
- Works well when levels mostly independent<sub>s</sub>

# This talk

- Angelic semantics: levels communicate via (bounds on) **reachable sets**
- Resulting planners
  - Do most work at high level => efficient
  - Yet provably sound and complete

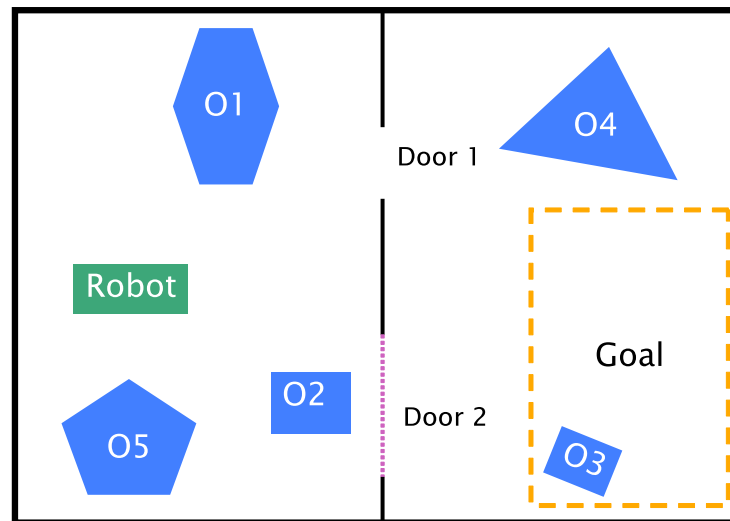
# Running example



**Goal:** Move O1 into the orange box

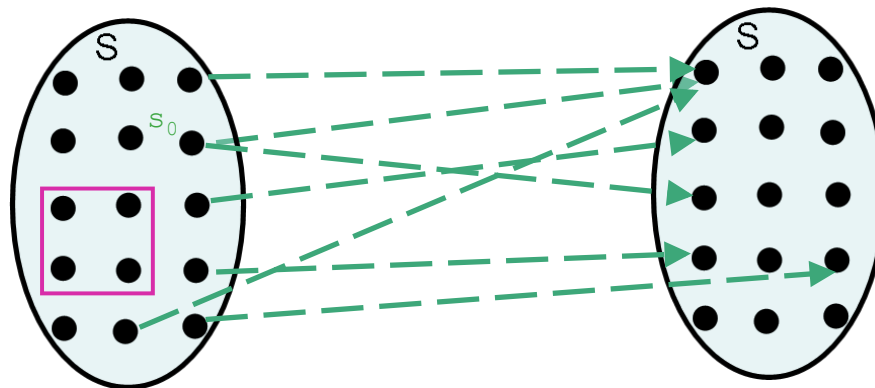
# What we would like

- Combine different types of reasoning
- Think hierarchically
  - Reject most bad plans at high level
  - Commit to good plans at high level if possible
  - Descend to lower levels when necessary



# Planning problems

- A (deterministic) **planning problem** consists of:
  - State space  $S$
  - Goal set  $T \subseteq S$
  - Initial state  $s_0 \in S$
  - Primitive action set  $A$
  - Transition function  $f : S \times A \rightarrow S$



Transition function for action  $a_2$

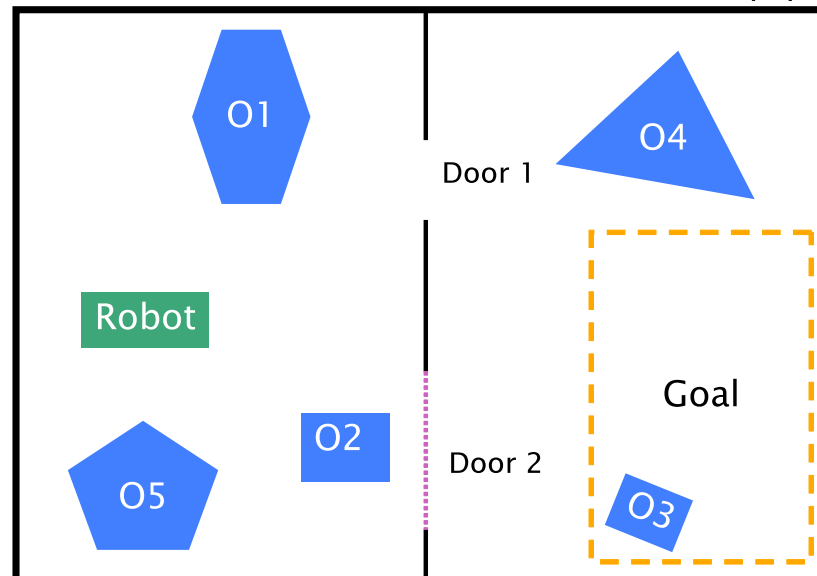
# Planning Problems

- State

- Robot configuration
- Confs of unheld objects
- Held object (if any)
- Relative conf of held object
- Door statuses

- Actions

- Move ( $\sigma$ ):  $\sigma$  directly reachable
- Grasp (o)
- Ungrasp (o)
- Open (d)
- Shut (d)



# Hierarchies

- A **hierarchy** for a planning problem consists of:
  - Set of **high-level actions**  $H$
  - Set of **top-level actions**  $Top$
  - For each high level action, set of **refinements**
  - Each refinement is a sequence of other actions

# Hierarchy

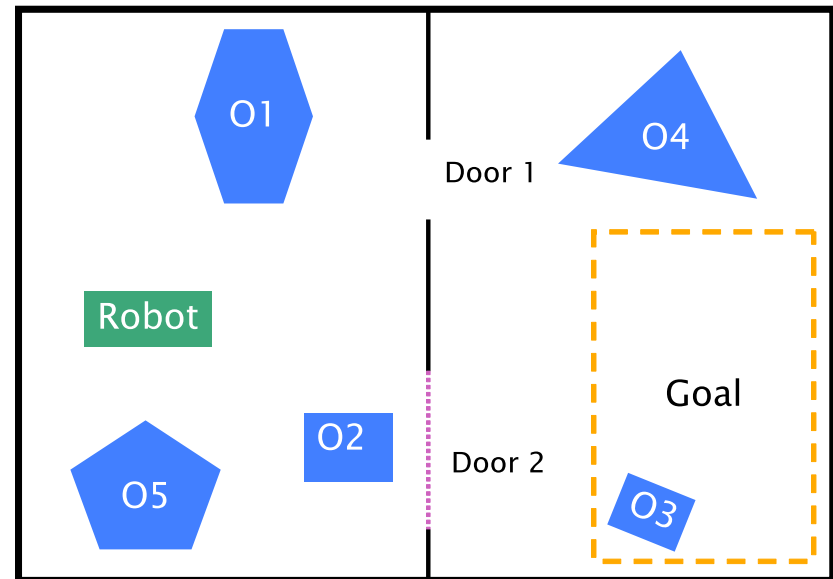
Top Level

Transfer (O1, G)

Transfer (O2, H)

...

G, H subsets of map

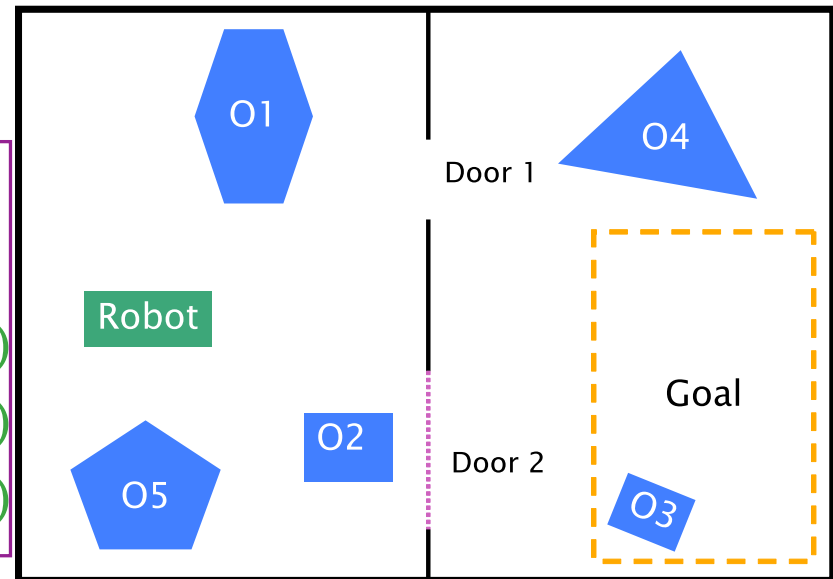


# Hierarchy

Transfer (O4, G)

## Refinements

GraspSide (O4,1), TransferTo(O4, G)  
GraspSide (O4,2), TransferTo(O4, G)  
GraspSide (O4,3), TransferTo(O4, G)



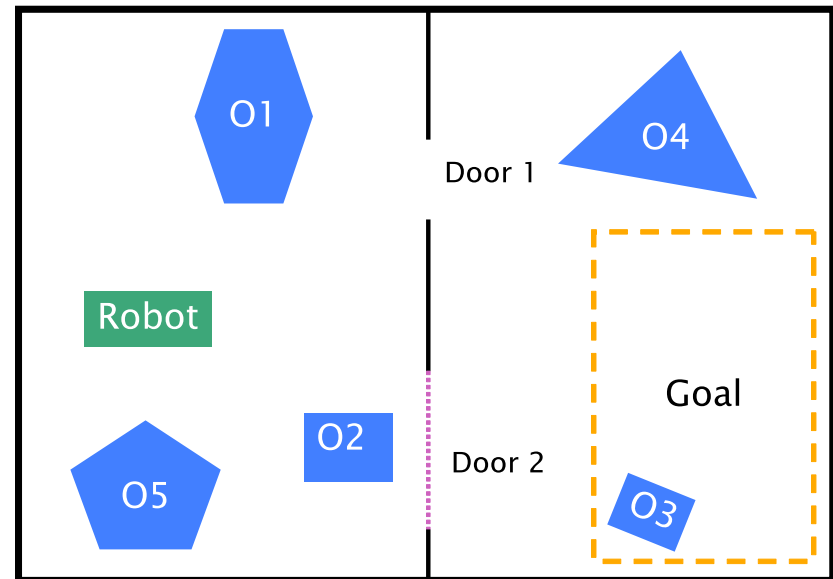
# Hierarchy

GraspSide (O4, 2)

## Refinements

Navigate (S), Grasp (O4)

S: configs from which side 2 of O4 can be grasped



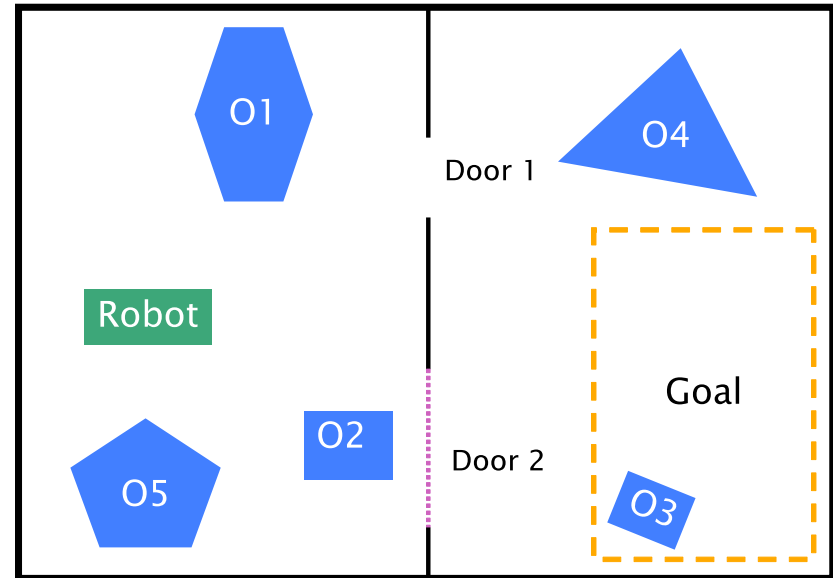
# Hierarchy

Navigate (S)

## Refinements

NavToRoom(R2, Door1), Nav(S)

NavToRoom(R2, Door2), Nav(S)

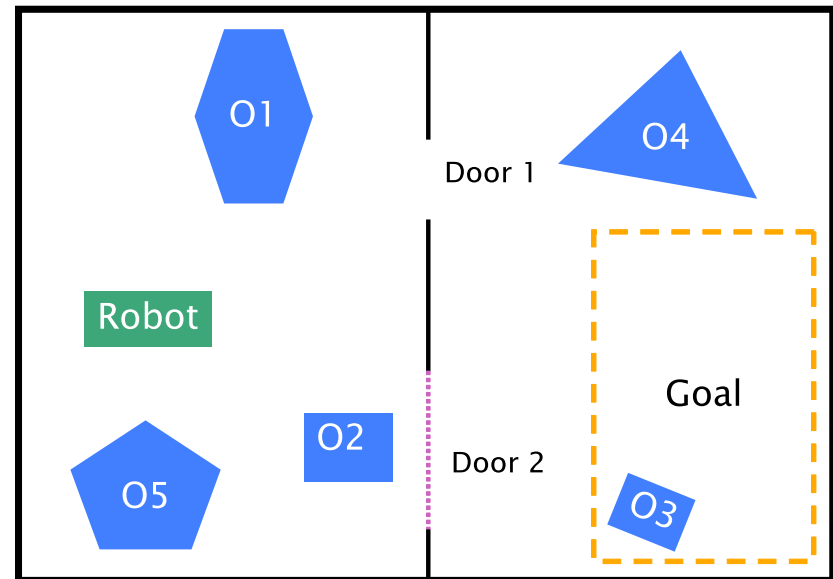


# Hierarchy

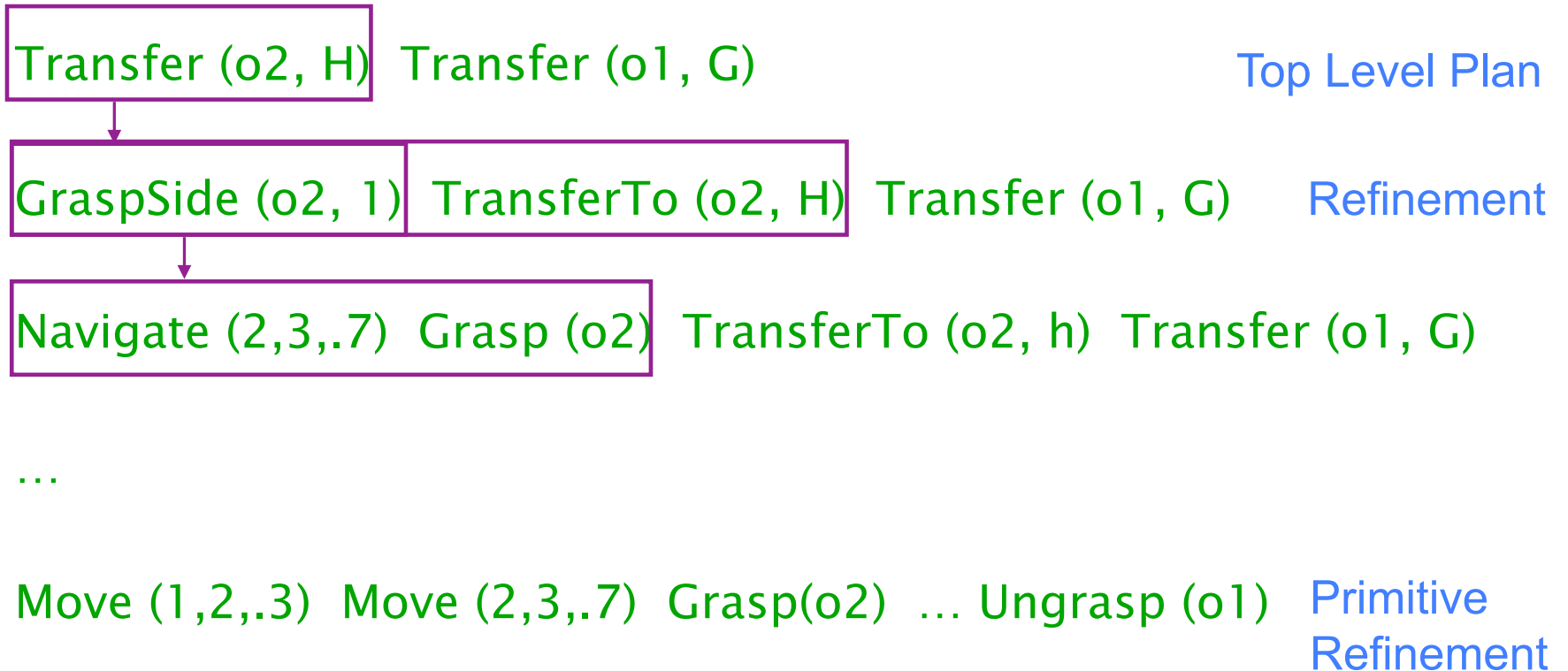
Nav (S)

## Refinements

Collision-free kinematic motion plans  
ending in S



# Refinements



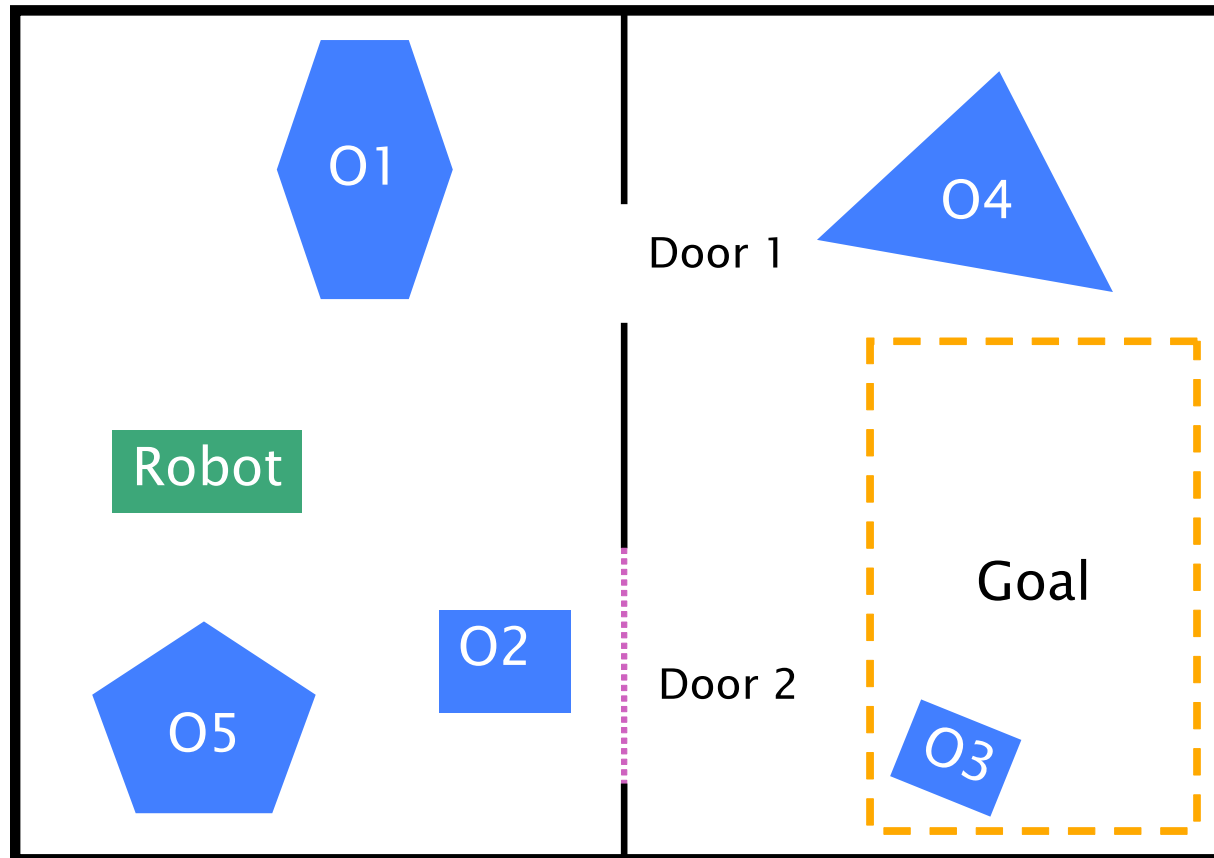
# Hierarchical planning

- Given
  - A planning problem
  - A hierarchy
- Look for a plan that:
  - reaches the goal
  - is a primitive refinement of a top-level plan

# Using hierarchies to plan

- # high level plans  $\ll$  # primitive plans
- So we win if we can
  - Reject most bad plans at the high level
  - Often commit to good plans at high level

# Using hierarchies to plan



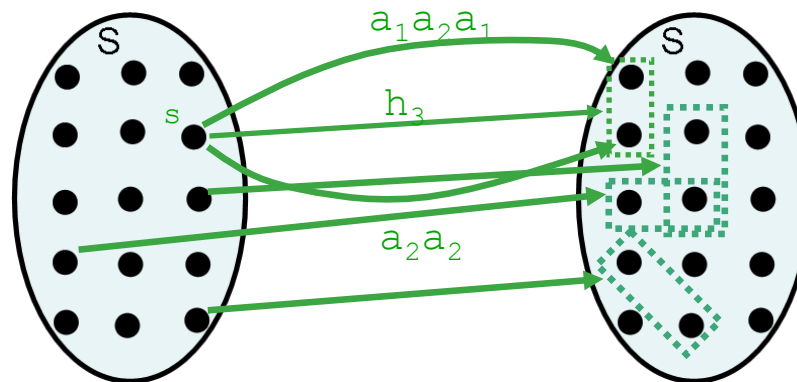
# Using hierarchies to plan

- # high level plans  $\ll$  # primitive plans
- So we win if we can
  - Reject most bad plans at the high level
  - Often commit to good plans at high level
- To do this we need **transition models** for HLAs

# Our approach: Reachable sets

- Reachable set of HLA  $h$  from state  $s$ :  
 $\{f_{a'}(s) \mid a' \in \text{PrimRef}(h)\}$
- Angelic description of  $h$  -  $F_h : S \rightarrow 2^S$   
 $F_h(s) = \text{Reachable set of } h \text{ from } s$

HLA:  $h_3$   
Primitive refinements:  
 $[a_1 a_2 a_1]$ ,  $[a_2 a_2]$



# Angelic nondeterminism

- HLA descriptions map states to **sets** of states
- May seem related to nondeterministic planning
- Key difference: how actual successor state chosen
  - Nondeterministic planning: **adversary** chooses
  - Stochastic planning: **nature** chooses (by flipping a coin)
  - HLA descriptions: **planner** will eventually choose

# Angelic descriptions...

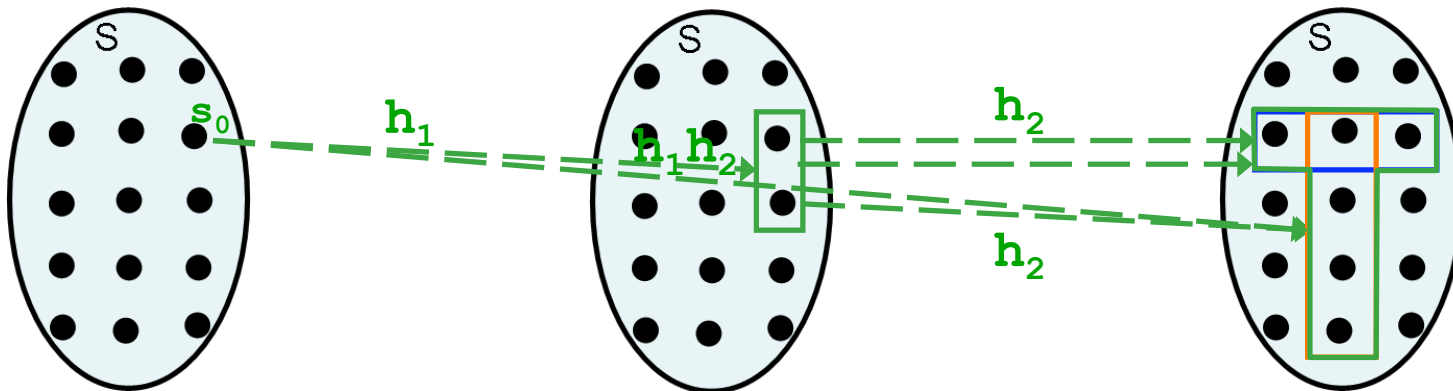
1. Are true
2. Are composable
3. Support high level reasoning

# Truth

- Angelic HLA descriptions
  - Logically follow from primitive descriptions
  - Are empirically verifiable
- Simplifies design/analysis of algorithms that
  - Derive HLA descriptions from primitive
  - Learn descriptions by observation

# Composability

- Can extend exact descriptions to:
  - Sets of states
  - Sequences of actions

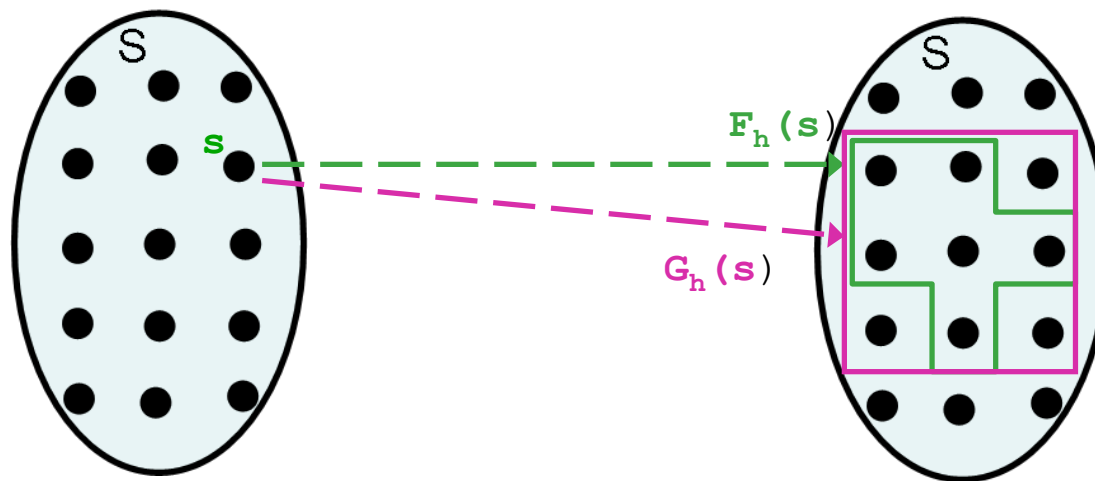


# High level reasoning

- Given high level sequence  $p$ , the following are equivalent
  - $F_p(s_0)$  intersects the goal set
  - $p$  has a primitive refinement that succeeds
- Allows rejecting/committing to plans at high level

# Optimistic descriptions

- Exact description  $F$  could be complicated
- NP-Hard to represent compactly
- Optimistic description: **superset** of reachable set



# Optimistic descriptions

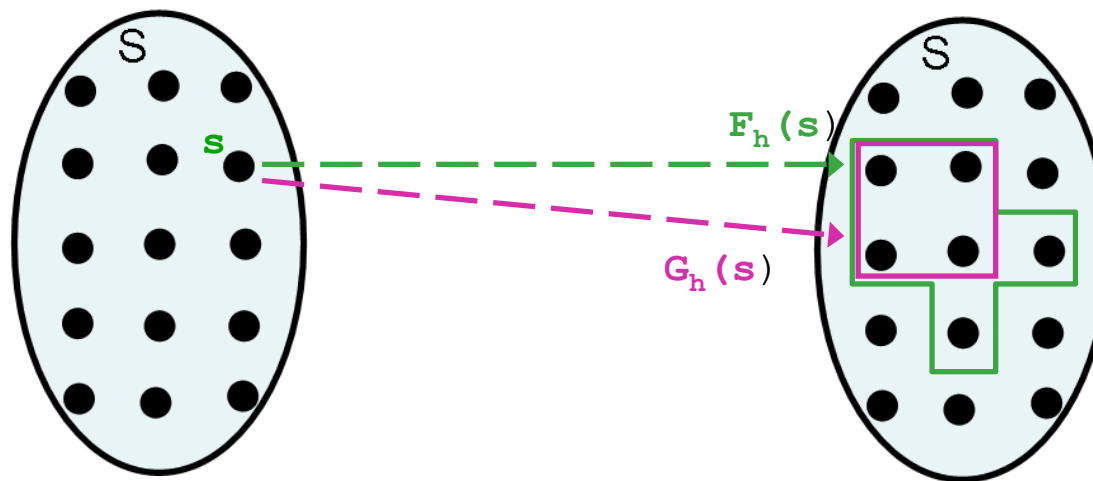
- **Transfer ( $o$ ,  $G$ )**
  - Object  $o$  must end up inside  $G$
  - Robot must end up within distance  $d$  of  $G$
- **NavToRoom ( $r$ )**
  - Use polygonal decomposition using outer bound of robot + grasped object

# Properties of optimistic descriptions

- **Theorem**: if the optimistic reachable set of plan  $p$  does not intersect the goal, then no refinement of  $p$  solves the problem
- Allows pruning away plans at high-level

# Pessimistic descriptions

- Dual to optimistic descriptions
- Pessimistic description: subset of reachable set



# Pessimistic descriptions

- Nav ( $G$ )
- If world has no bottlenecks
  - Can reach any configuration in  $G$
- Else if exists a path in roadmap to set  $G'$  in  $G$ 
  - $G'$  achievable
- Else
  - Empty set

## Properties of pessimistic descriptions:

- **Theorem:** if the pessimistic reachable set of plan  $p$  intersects the goal, then some primitive refinement of  $p$  solves the problem
- Allows committing to plans at high-level

# Numerical costs

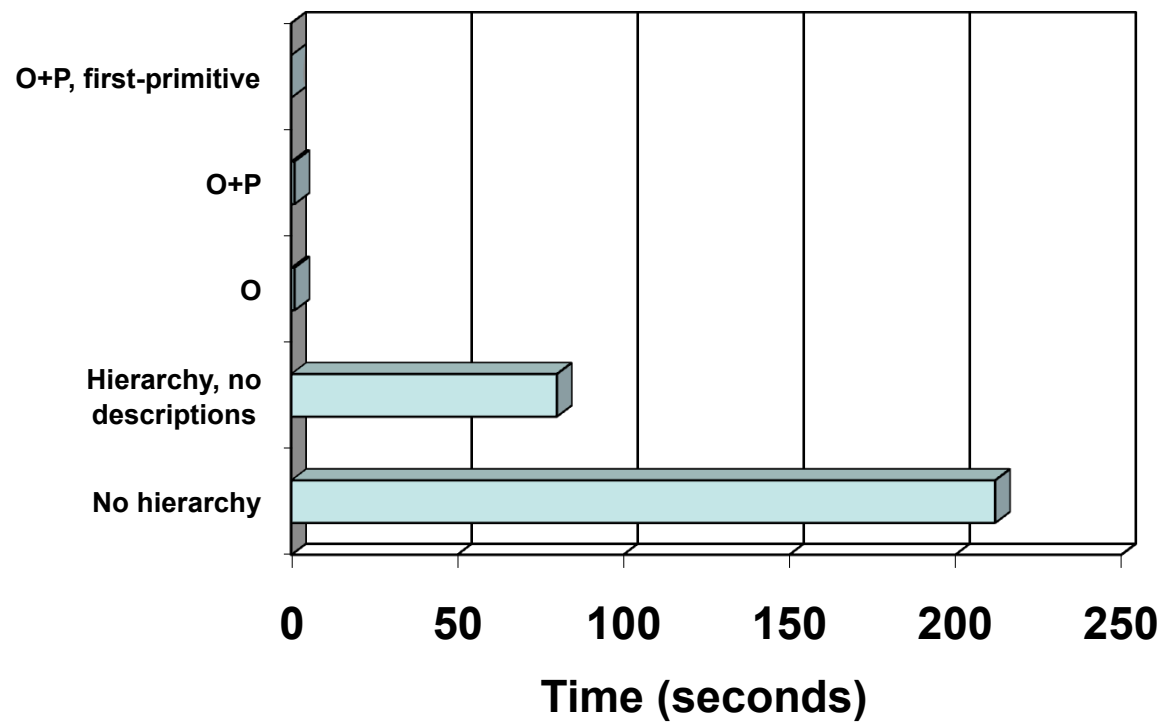
- So far, we've restricted to goal-based problems
- What if actions have numerical costs/rewards?
- Generalize descriptions, so HLA has
  - Pessimistic cost – upper bound on achievable cost
  - Optimistic cost – lower bound on achievable cost
- Example:  $\text{Nav}(G)$ 
  - Optimistic cost: at least Euclidean distance to  $G$
  - Pessimistic cost: distance in roadmap

# Planning algorithms

- Repeat
  - Get next plan
  - If fails optimistically, reject
  - If succeeds pessimistically, commit
  - Else, add refinements to set
- Examples
  - AHA\*: find hierarchically optimal plan
  - SHP: find any plan below a cost threshold
  - AHRTA\*: plan that begins with primitive action

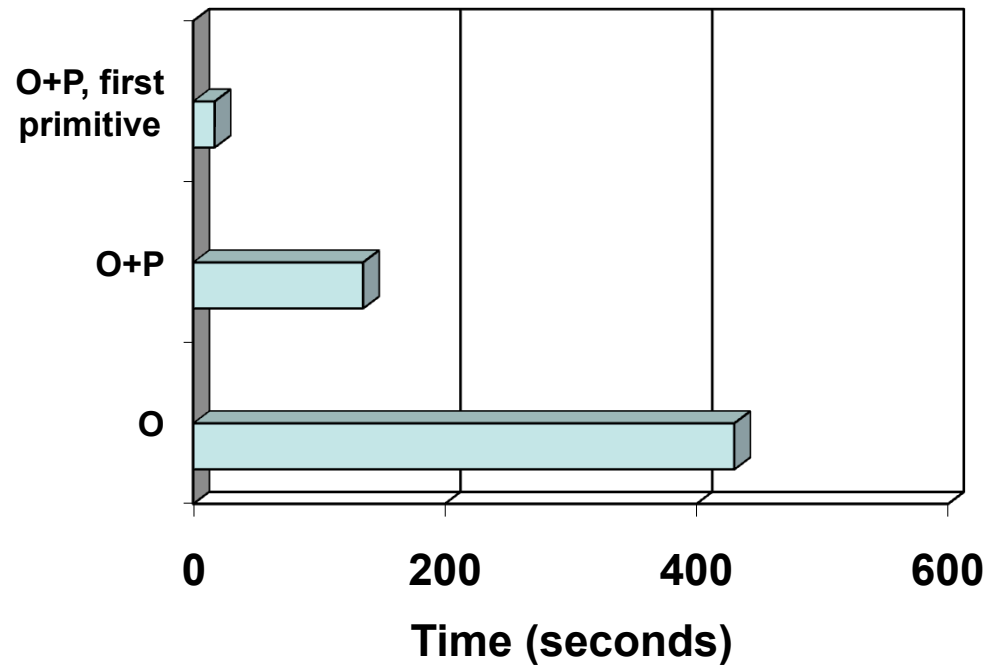
# Experiments

- Discretized manipulation problem: easy instance



# Experiments

- Harder instance
- Nonhierarchical methods did not terminate within 10,000 seconds



# Online navigation planning

The screenshot displays a software interface for online navigation planning. The central area is a 2D occupancy grid map showing a complex environment with walls and obstacles. A blue line represents a planned path, starting from a red square at the bottom center and moving towards the top right. A yellow dashed line indicates a previous or alternative path. The interface includes several panels:

- Displays Panel (Left):** Lists three PolyLine objects:
  - 10. PolyLine6 (PolyLine):** Enabled, Color (25,255,0), Render Operation Points, Topic /angelic/localmap.
  - 11. PolyLine7 (PolyLine):** Enabled, Color (25,255,0), Render Operation Lines, Topic /angelic/connect.
  - 12. PolyLine8 (PolyLine):** Enabled, Color (25,255,0), Render Operation Lines, Topic /angelic/traverse...
- Views Panel (Right):** Shows the current view as "Top-down Orthographic" and includes "Save Current", "Load", and "Delete" buttons.
- Selection Panel (Right):** A panel for selecting objects in the scene.
- Time Panel (Bottom):** Displays performance metrics:
  - Wall Time: 1246067687.722636
  - Wall Elapsed: 12780.658137
  - ROS Time: 12782.600000
  - ROS Elapsed: 12780.700000

# Online navigation planning

The screenshot displays a software interface for online navigation planning. The central area is a 2D occupancy grid map showing a robot's current position (a small yellow circle) and a planned path (a blue line). The path starts from the robot and moves through a series of corridors and rooms, ending at a goal point. The map is rendered in grayscale, with obstacles represented by black pixels and free space by white pixels.

On the left side, there is a 'Displays' panel with a list of objects and their properties:

- 10. PolyLine6 (PolyLine)**
  - Enabled:
  - Override Color:
  - Color: (25,255,0)
  - Loop:
  - Render Operation: Points
  - Z Position: 0
  - Alpha: 1
  - Topic: /angelic/localmap
- 11. PolyLine7 (PolyLine)**
  - Enabled:
  - Override Color:
  - Color: (25,255,0)
  - Loop:
  - Render Operation: Lines
  - Z Position: 0
  - Alpha: 1
  - Topic: /angelic/connect
- 12. PolyLine8 (PolyLine)**
  - Enabled:
  - Override Color:
  - Color: (25,255,0)
  - Loop:
  - Render Operation: Lines
  - Z Position: 0
  - Alpha: 1
  - Topic: /angelic/traverse...

Below the 'Displays' panel are buttons for 'Add', 'Remove', and two directional arrows (down and up).

On the right side, there is a 'Views' panel with a dropdown menu set to 'Top-down Orthographic' and buttons for 'Save Current', 'Load', and 'Delete'. Below it is a 'Selection' panel, which is currently empty.

At the bottom of the interface, there is a 'Time' panel with four input fields and a 'Reset' button:

- Wall Time: 1246068951.766710
- Wall Elapsed: 14044.702210
- ROS Time: 14046.600000
- ROS Elapsed: 14044.700000

# Conclusions

- Angelic hierarchical planning
  - Reason about reachable sets of HLAs
  - Reject/commit to plans at high level when possible
  - Descend to lower levels when necessary
- Future work
  - Extending planners to return reachable sets
  - Learning hierarchies and descriptions
  - Probabilistic dynamics
  - Partially observable state

Thank you!