# Learning Hierarchical Tasks from Situated Interactive Instruction

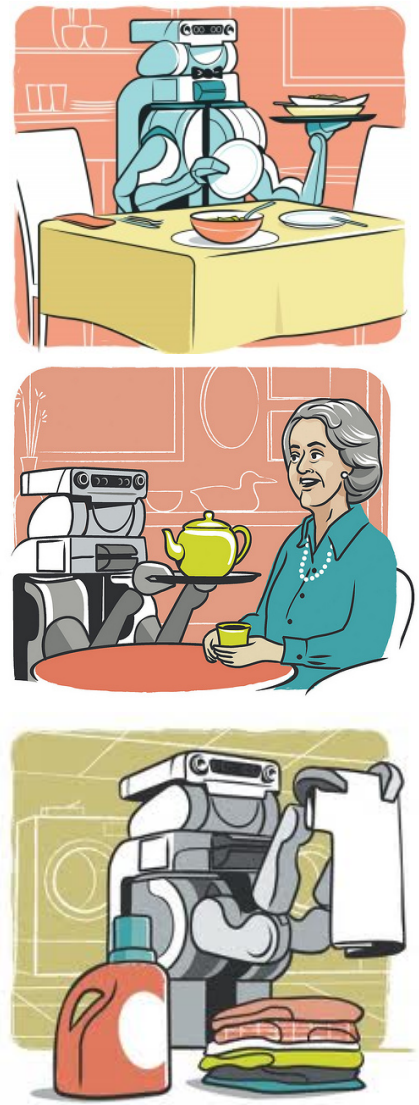
**Shiwali Mohan**

Computer Science and Engineering
University of Michigan

December $6^{th}$, 2013

# Generally Intelligent Agents

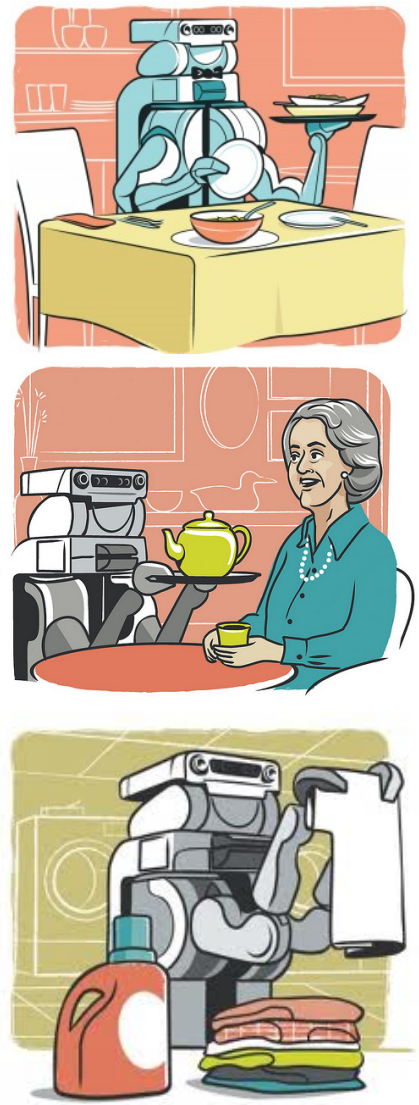
- Perform complex tasks
  - hierarchical
  - goal-oriented



[www.willowgarage.com]

# Generally Intelligent Agents

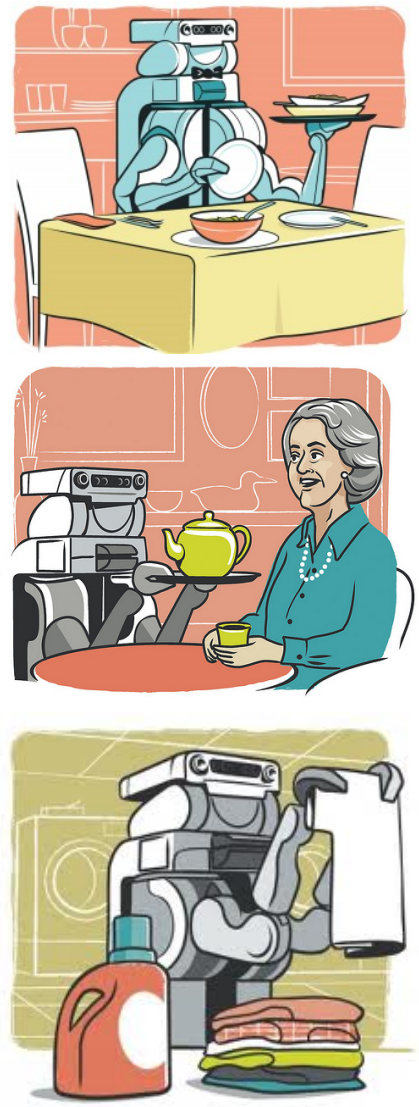
- Perform complex tasks
  - hierarchical
  - goal-oriented
- Dynamically learn task definitions
  - diversity in tasks
  - difference in environments
  - difference in preferences



[www.willowgarage.com]

# Generally Intelligent Agents

- Perform complex tasks
  - hierarchical
  - goal-oriented
- Dynamically learn task definitions
  - diversity in tasks
  - difference in environments
  - difference in preferences
- Users should be able to *program* agents
  - *embodied taskability*
  - natural interactions

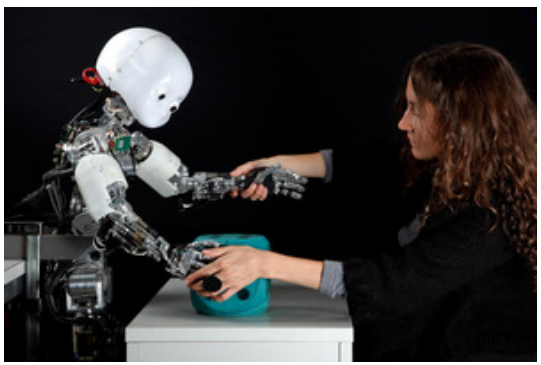


[www.willowgarage.com]

# Learning from Human-Agent Interaction

# Learning from Human-Agent Interaction

*Learning from Demonstration*
(LfD)

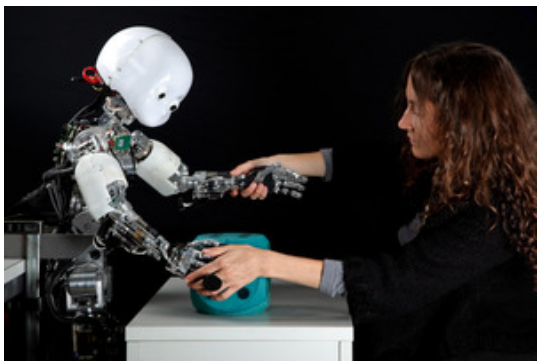


[B. Argall, Northwestern U]

kinesthetic training
teleoperation

# Learning from Human-Agent Interaction

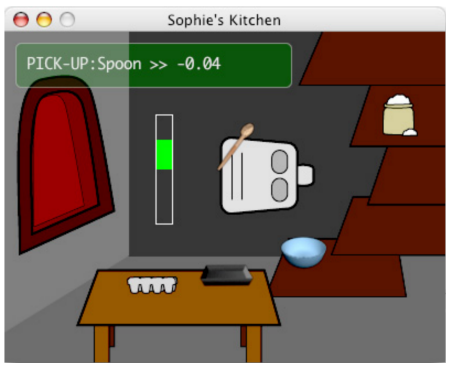
*Learning from Demonstration* (LfD)



[B. Argall, Northwestern U]

kinesthetic training
teleoperation

*Interactive Reinforcement Learning* (iRL)

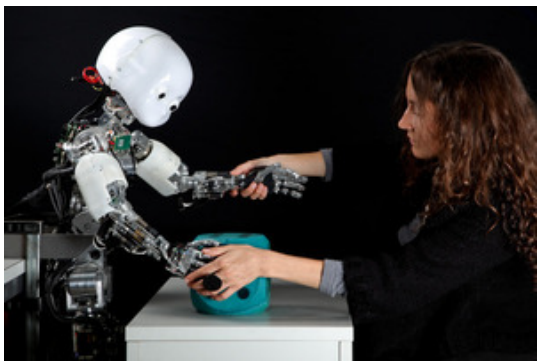



[A. Thomaz, Georgia Tech]

reward
feature selection

# Learning from Human-Agent Interaction

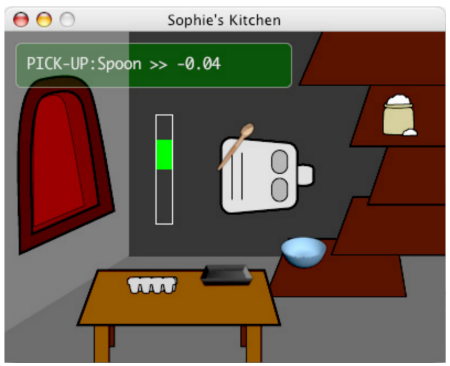
*Learning from Demonstration* (LfD)



[B. Argall, Northwestern U]

kinesthetic training
teleoperation

*Interactive Reinforcement Learning* (iRL)

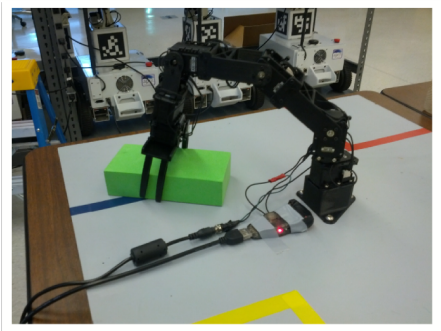



[A. Thomaz, Georgia Tech]

reward
feature selection

*Situated Interactive Instruction* (SII)

*"Store the green rectangle."*



natural language
conceptual reference

# ROSIE Demo


[www.blogging4jobs.com]

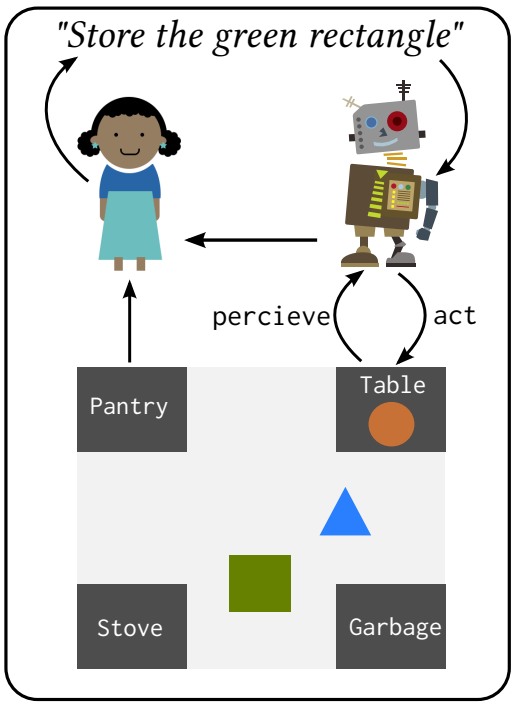
a cognitive robotic framework @ UofM
based on Soar Cognitive Architecture (Laird, 2012)

# Learning with Situated Interactive Instruction

Mohan, Kirk, and Laird. *A Computational Model of Situated Task Learning with Interactive Instruction.* International Conference on Cognitive Modeling. 2013.

# Learning with Situated Interactive Instruction

## Integrative Interaction



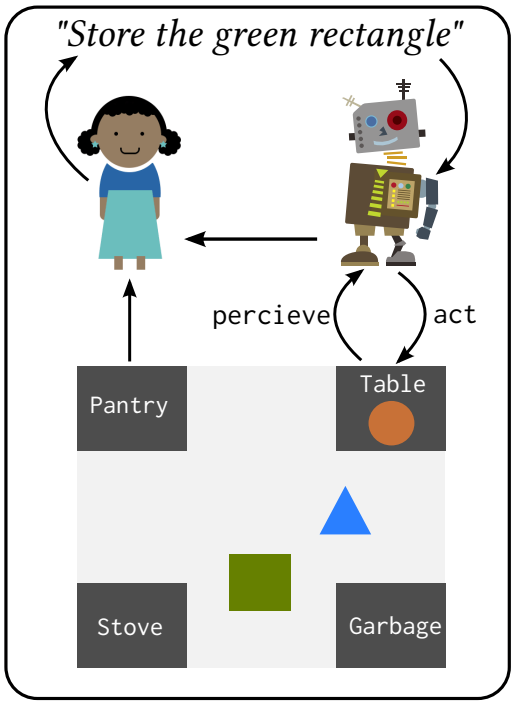

Theory of Discourse
Grosz and Sidner (1986)

---

Mohan, Kirk, and Laird. *A Computational Model of Situated Task Learning with Interactive Instruction.* International Conference on Cognitive Modeling. 2013.

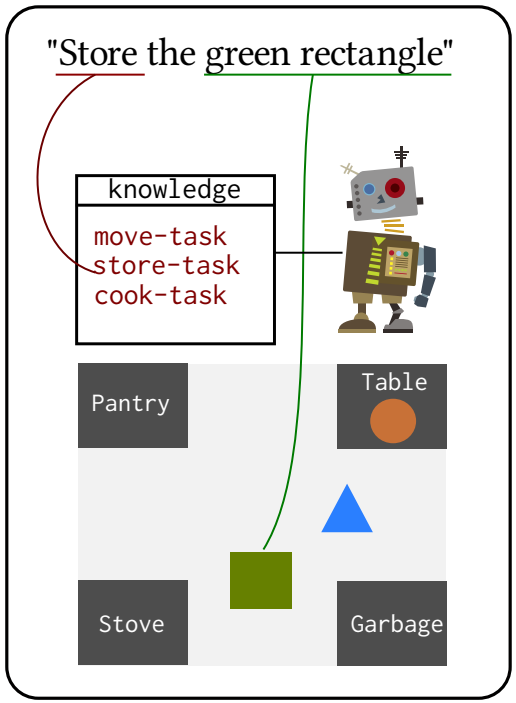
# Learning with Situated Interactive Instruction

## Integrative Interaction




Theory of Discourse
Grosz and Sidner (1986)

## Situated Comprehension



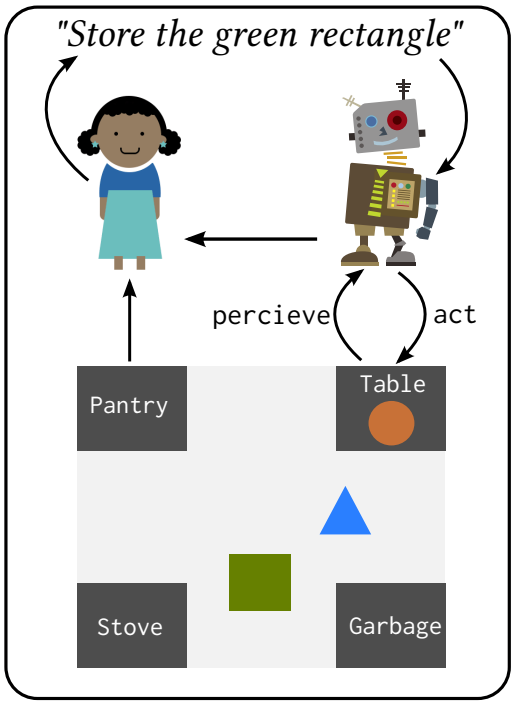

The Indexical Model
of Situated Comprehension

---

Mohan, Kirk, and Laird. *A Computational Model of Situated Task Learning with Interactive Instruction.* International Conference on Cognitive Modeling. 2013.

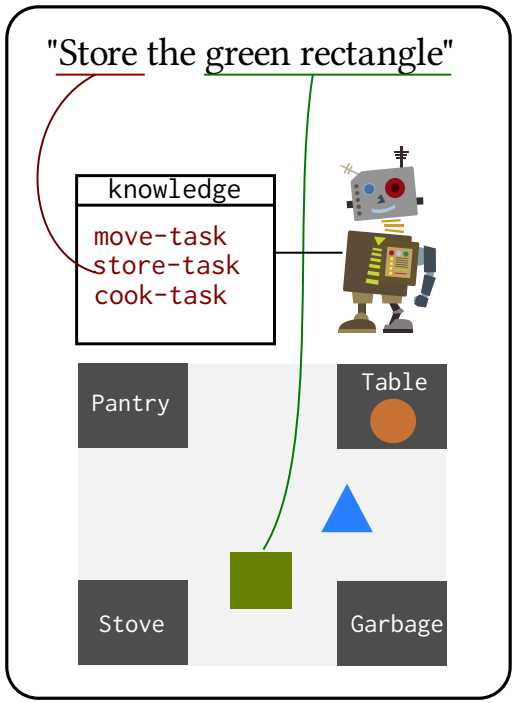
# Learning with Situated Interactive Instruction

## Integrative Interaction



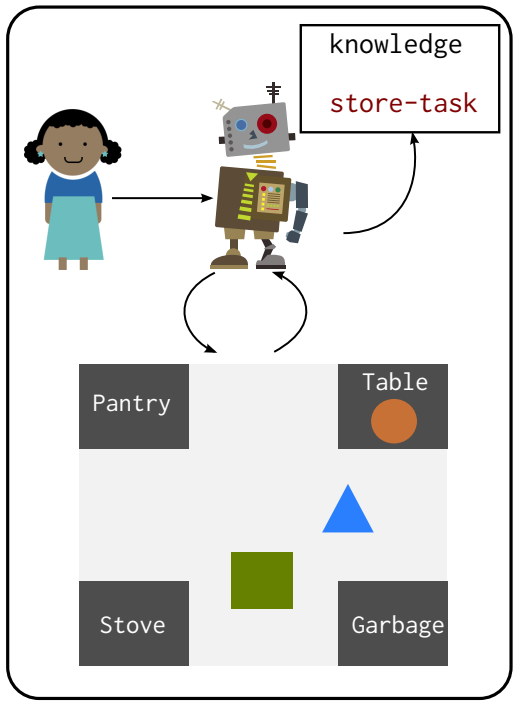

Theory of Discourse
Grosz and Sidner (1986)

## Situated Comprehension




The Indexical Model
of Situated Comprehension

## Task Learning




Representation + iEBL
Empirical Evaluation

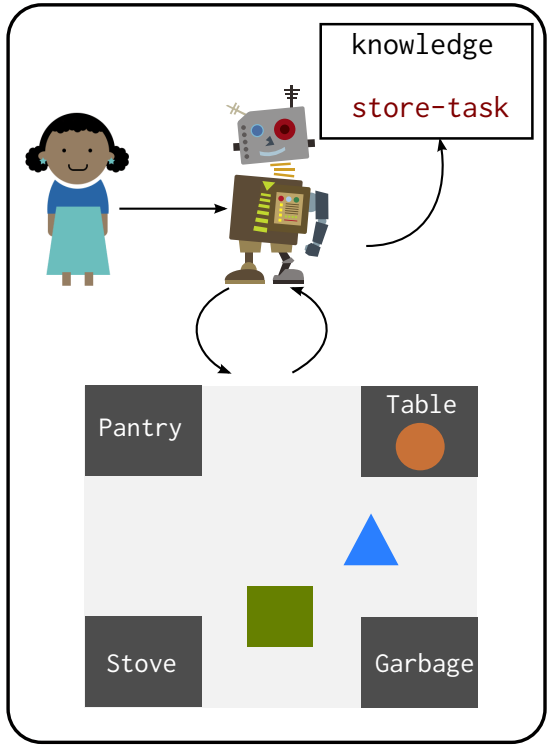
---

Mohan, Kirk, and Laird. *A Computational Model of Situated Task Learning with Interactive Instruction.* International Conference on Cognitive Modeling. 2013.

# Hierarchical Task Learning

## Outline

1. Problem Description
2. Approach
   - Representation
   - iEBL
3. Evaluation
   - Variability
   - Generalization
   - Transfer

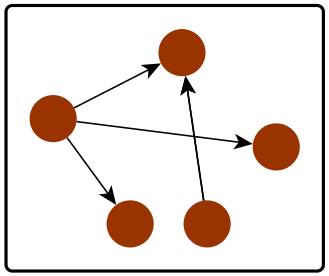



---

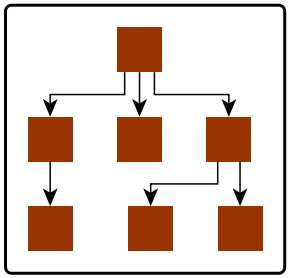
Mohan, Mininger, Kirk, & Laird. *Acquiring Grounded Representations of Words with Situated Interactive Instruction.* Advances in Cognitive Systems 2. 2012.
Mohan, Kirk, & Laird. *A Computational Model of Situated Task Learning with Interactive Instruction.* International Conference of Cognitive Modeling. 2013.
Mohan & Laird. *Learning Hierarchical Tasks with Situated Interactive Instruction.* submitted to Autonomous Agents and Multi-Agent Systems 2013.

# Task Learning

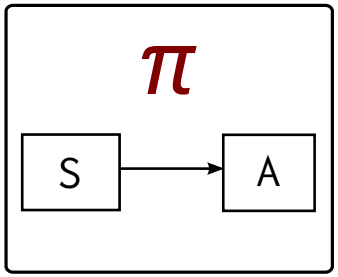
- Task knowledge



relational structure



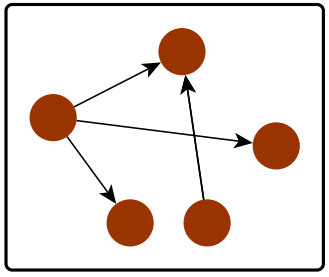
hierarchical decomposition



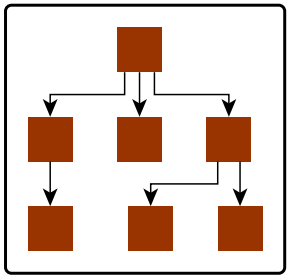

policy

# Task Learning

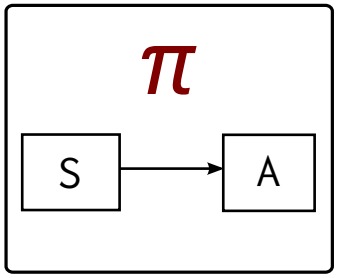
- Task knowledge



relational structure

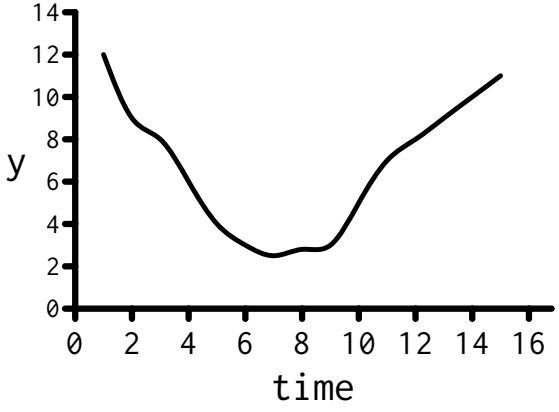


hierarchical decomposition

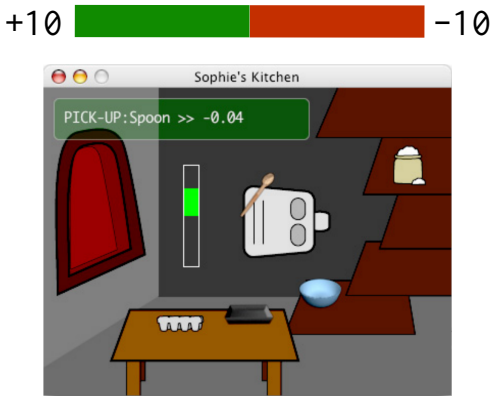



policy

- Kinesthetic Training

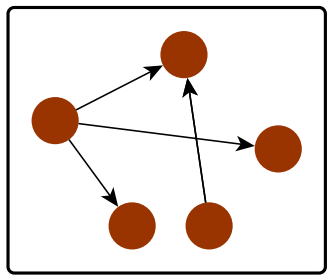



- Rewards

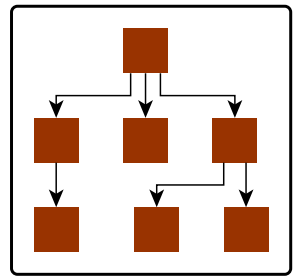


# Task Learning

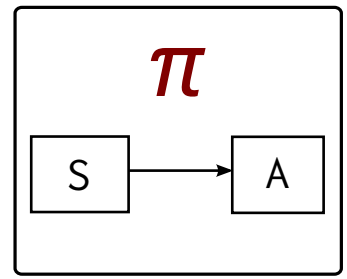
- Task knowledge



relational structure



hierarchical decomposition




policy

- Situated Interactive Instruction (SII)
  - Explicit human supervision
    - identify relevant features
    - define goals
    - decompose tasks
  - Agent-driven: learning when needed
  - HRI motivation
    - fast learning
    - transfer

# Task Knowledge

Given a set of primitive actions and their models
Acquire the following for `store`

**What?**

- parameters
  *Store the green cylinder.*
  `store(O2, pantry, in(O2,pantry))`
- subtasks
  `store`: `open`, `move [pick-up, put-down]`, `close`
- goal
  `in(O2,pantry)` $\wedge$ `closed(pantry)`

---

**How?**

- policy
  if `[state,task]` then `execute([subtask])`
- model
  if `[state,task]` then `[next-state]`

---

**When?**

- availability
  if `[state]` then `available(store)`
- termination
  if `[state]` then `terminate(store)`

# interactive Explanation-based Learning

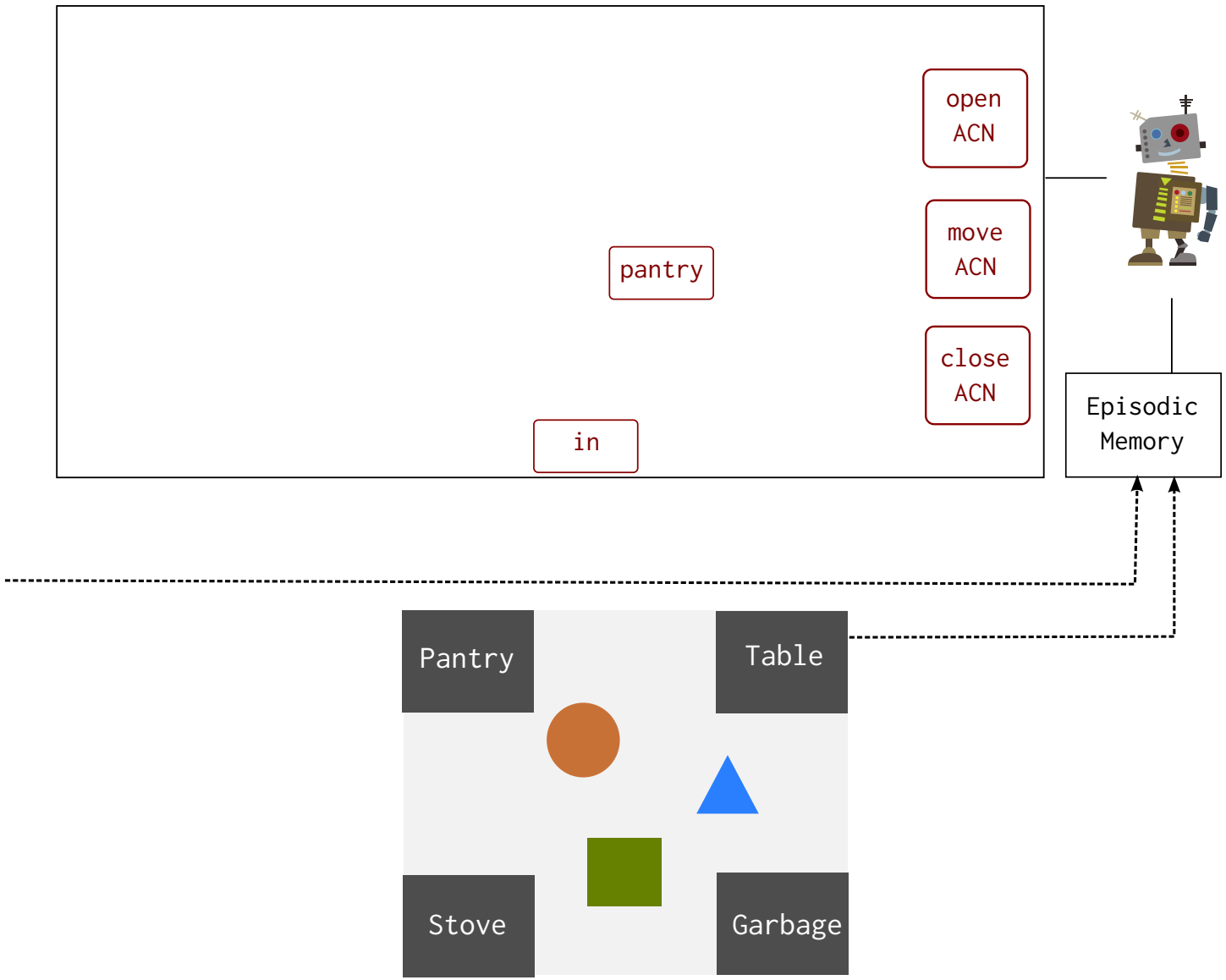
## EBL methods: Mitchell (1986), DeJong and Mooney (1986), Rosenbloom (1986)

Specific to general learning

1. (interactive) Acquire a specific example of how to execute a task.
2. (EBL) Generalize the specific experience

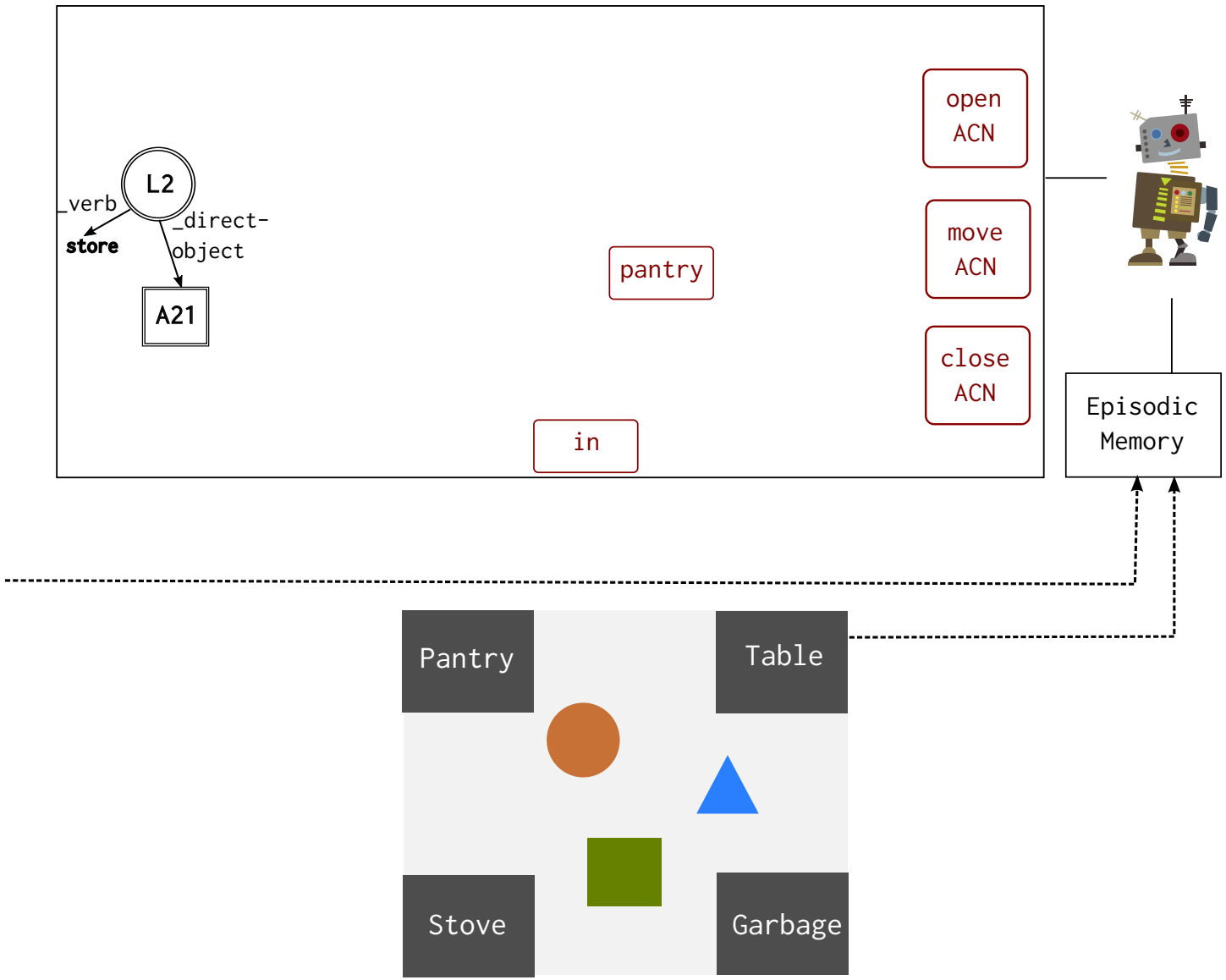
# 1. Interactive Example Execution

Interaction trace
Instructor: Store the green rectangle.



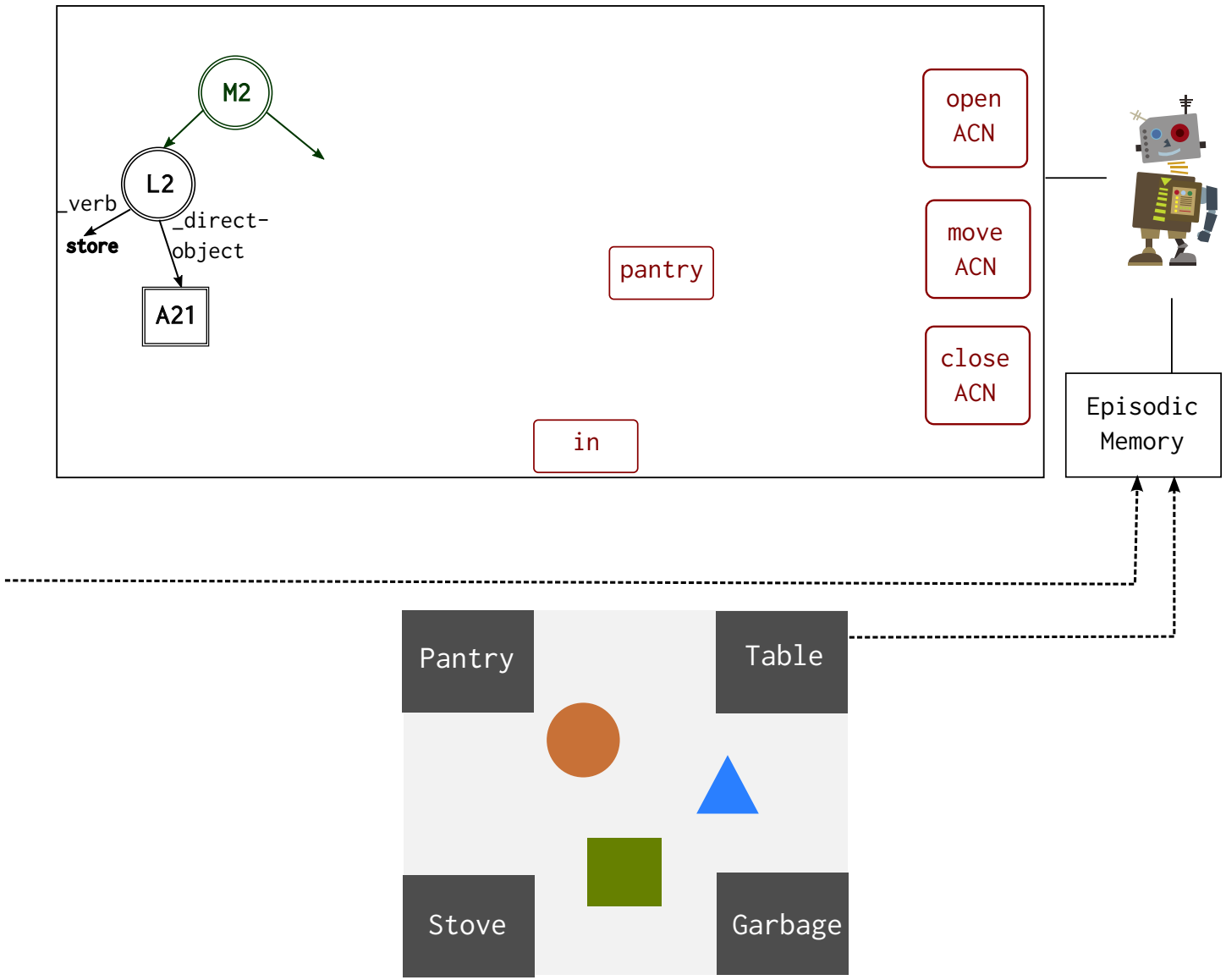
# 1. Interactive Example Execution

Interaction trace
Instructor: Store the green rectangle.



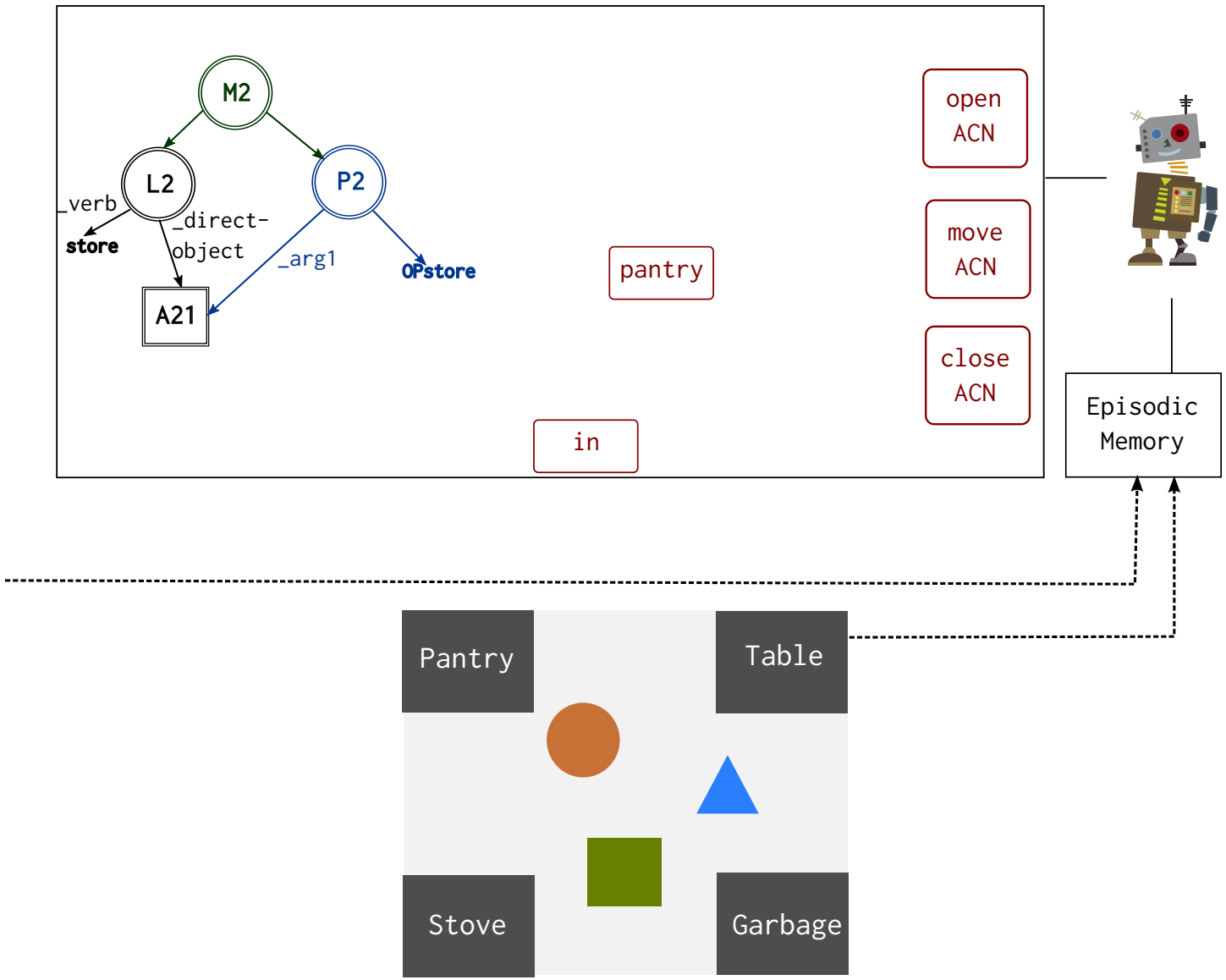
# 1. Interactive Example Execution

Interaction trace
Instructor: Store the green rectangle.



# 1. Interactive Example Execution

Interaction trace
Instructor: Store the green rectangle.

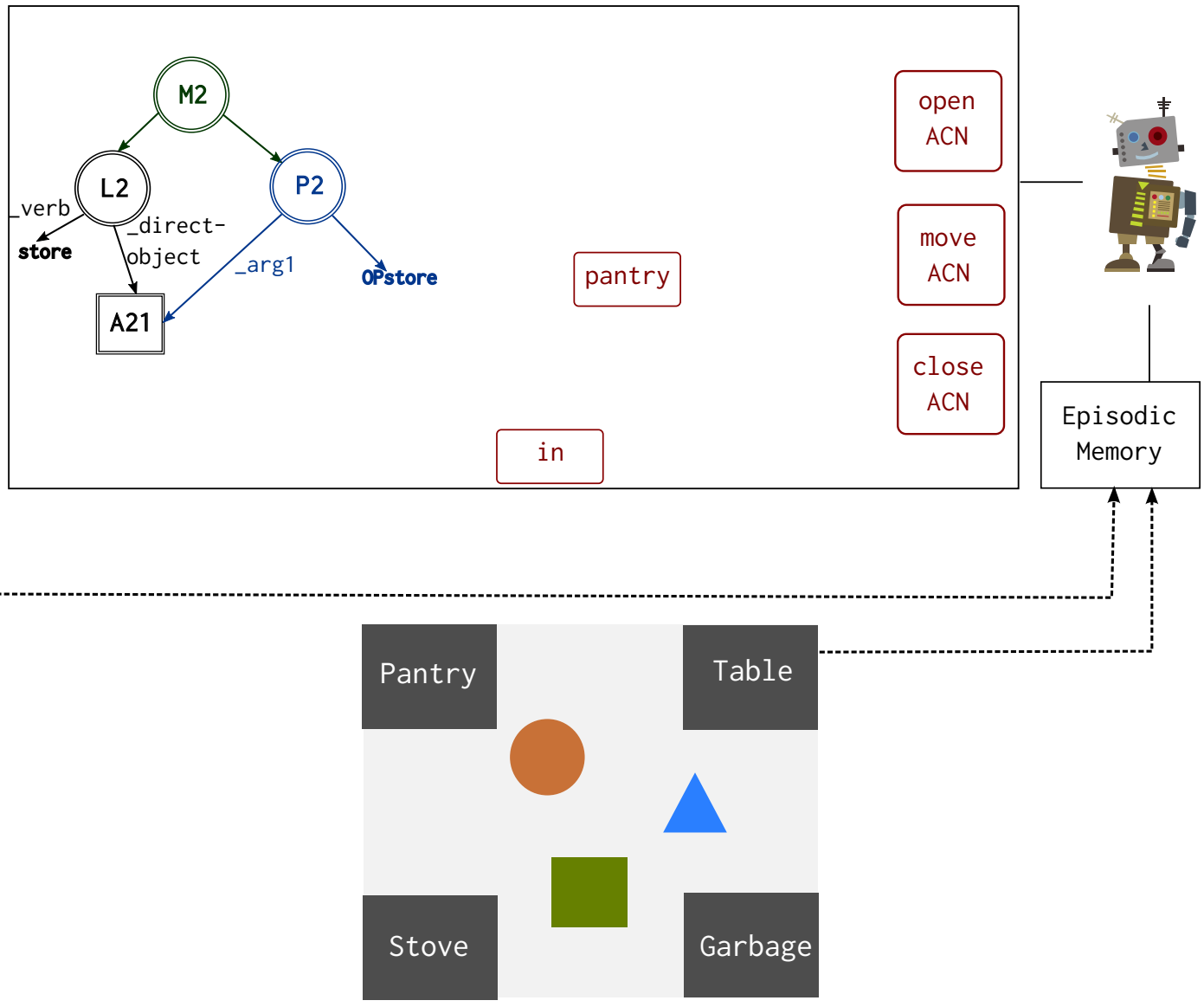


# 1. Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?



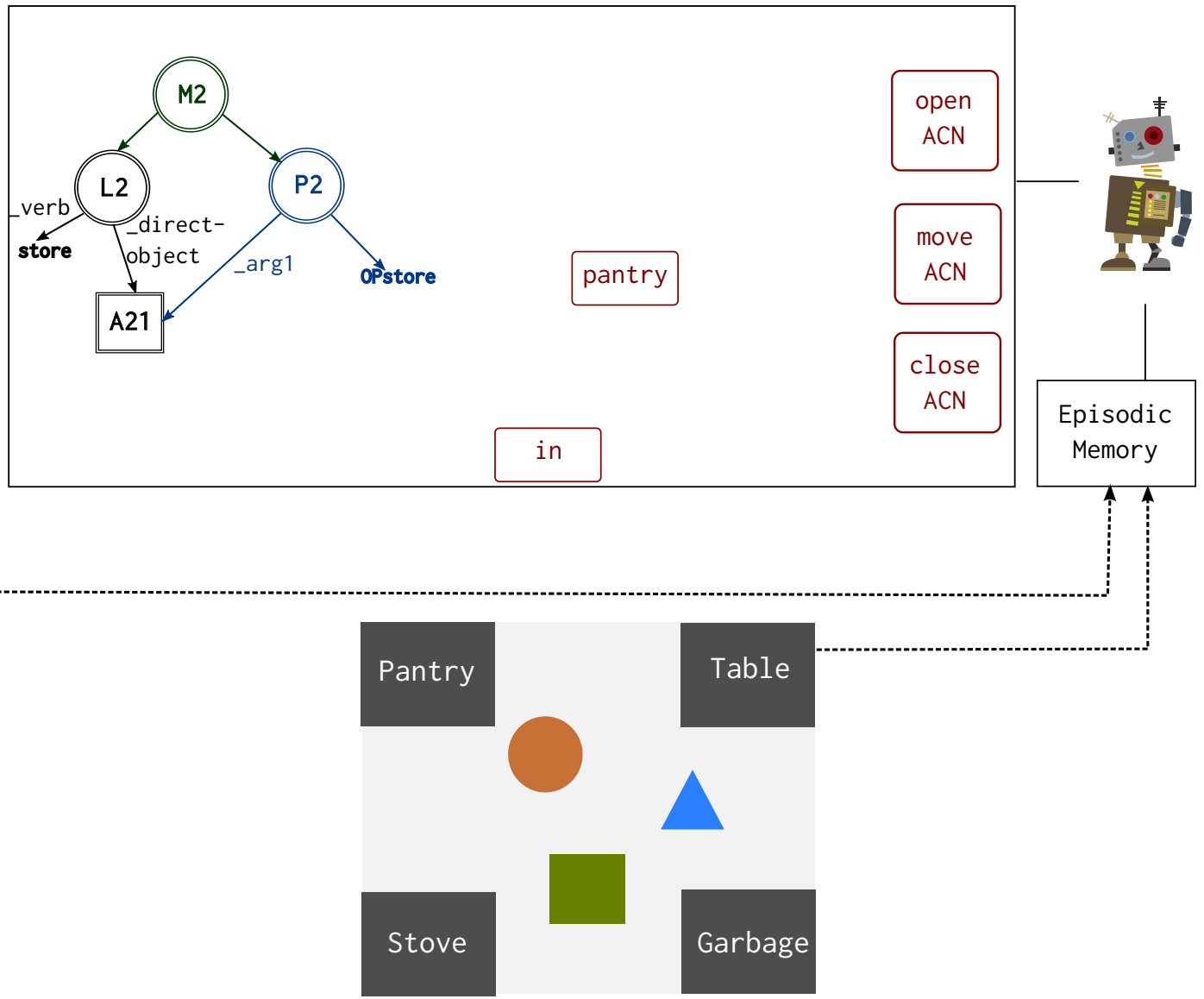
# 1. Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.



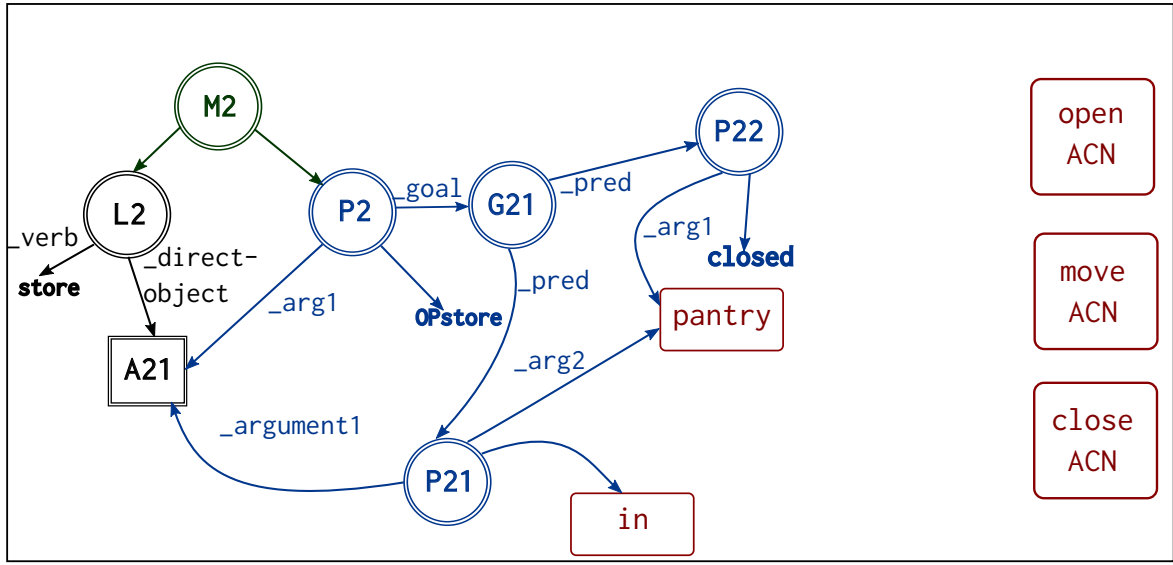
# 1. Interactive Example Execution

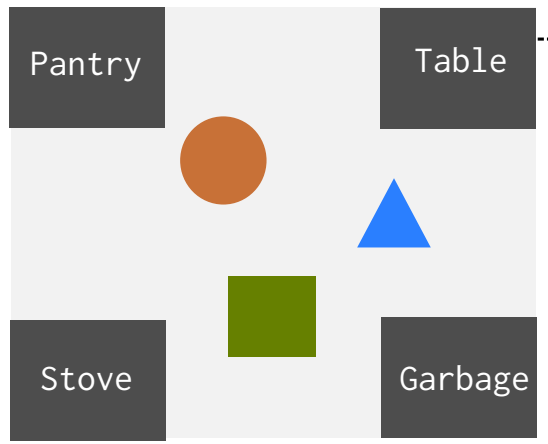
Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.



# 1. Interactive Example Execution

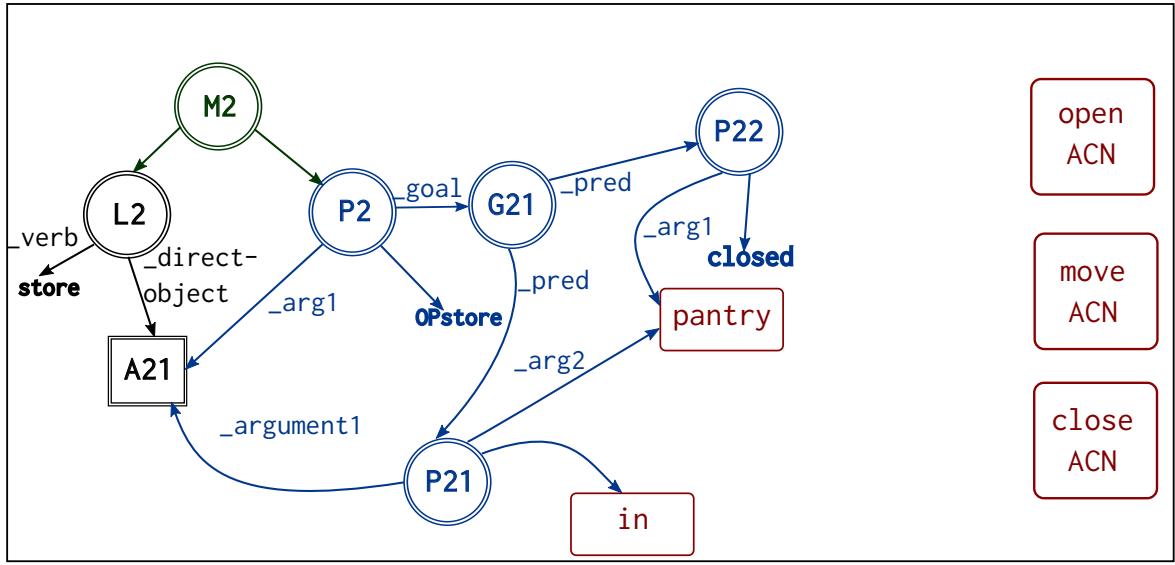
Interaction trace

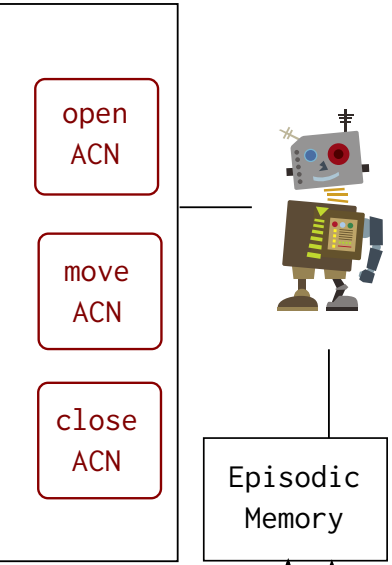
Instructor: Store the green rectangle.

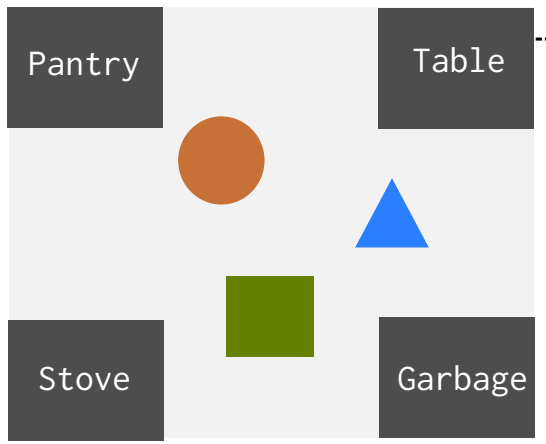
Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?



# 1. Interactive Example Execution

Interaction trace

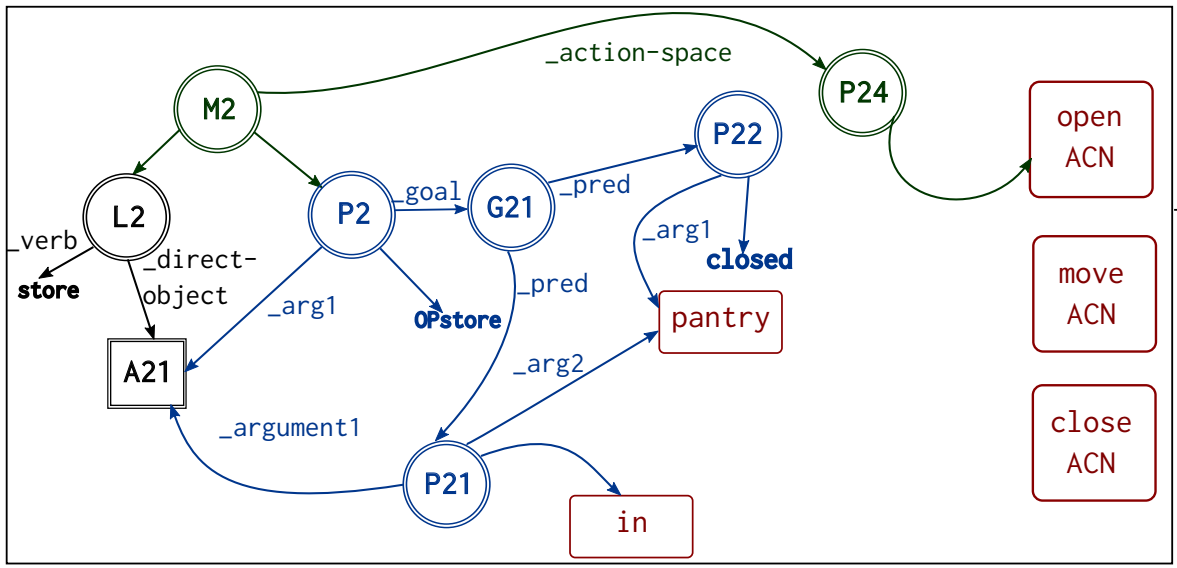
Instructor: Store the green rectangle.

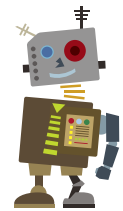
Agent: What is the goal of the action?

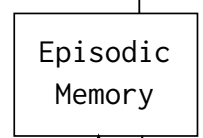
Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

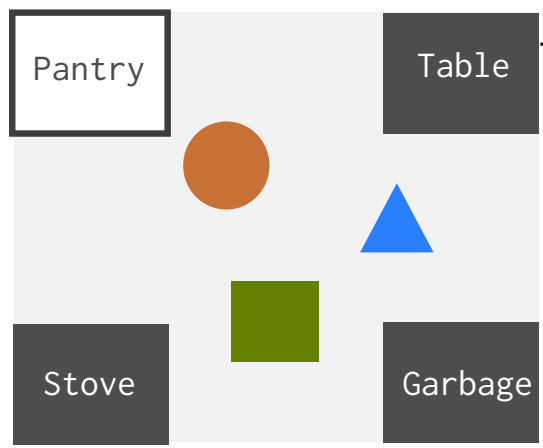
Agent: Which action should I take?

Instructor: Open the pantry.





# 1. Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

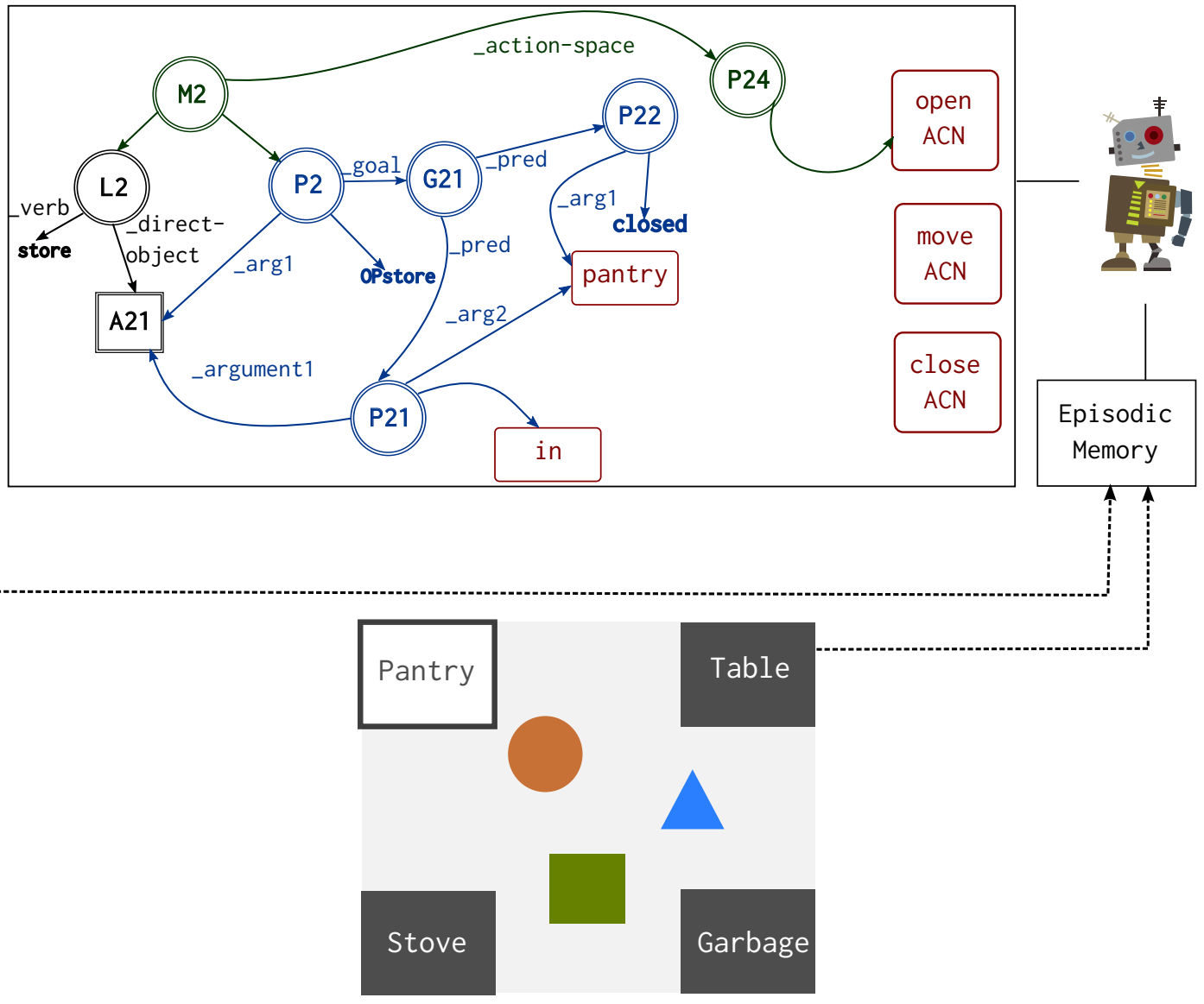
Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?

Instructor: Open the pantry.

Agent: Which action should I take?



# 1. Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

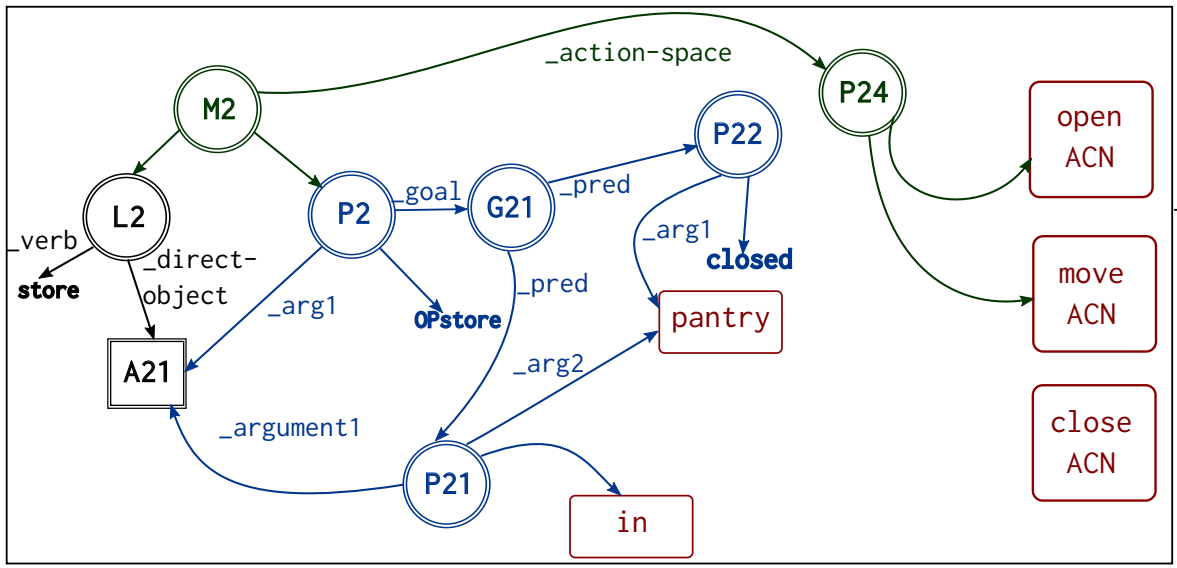
Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

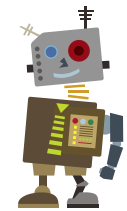
Agent: Which action should I take?

Instructor: Open the pantry.

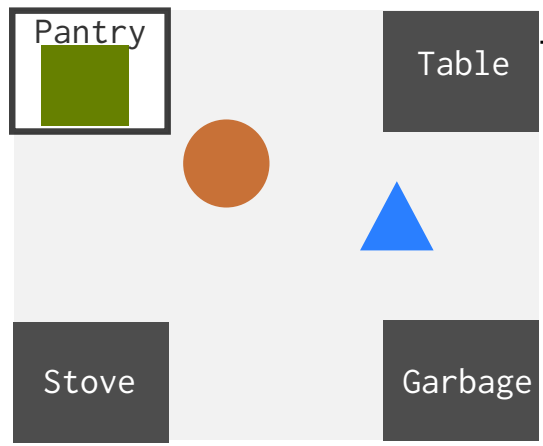
Agent: Which action should I take?

Instructor: Move the object to the pantry.



Episodic Memory



# 1. Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

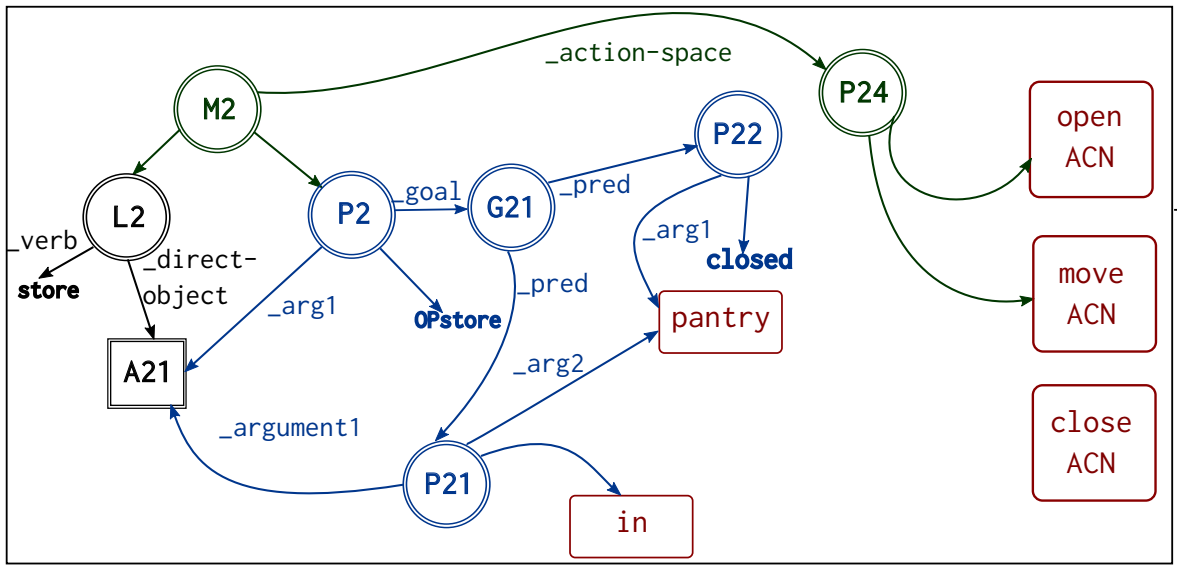
Agent: Which action should I take?

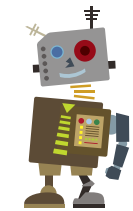
Instructor: Open the pantry.

Agent: Which action should I take?

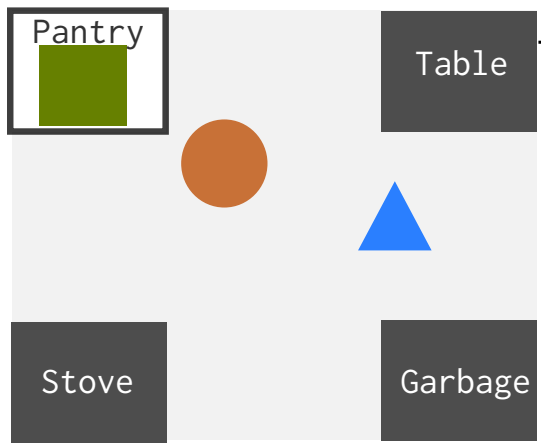
Instructor: Move the object to the pantry.

Agent: Which action should I take?



Episodic Memory



# 1. Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?

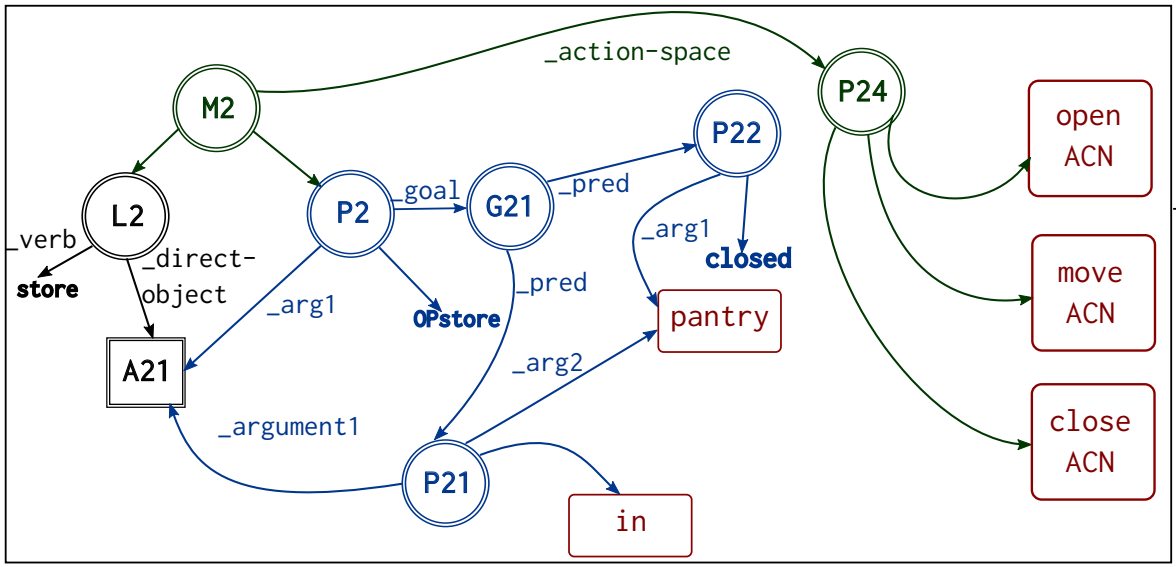
Instructor: Open the pantry.

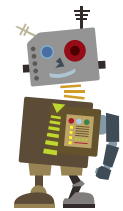
Agent: Which action should I take?

Instructor: Move the object to the pantry.

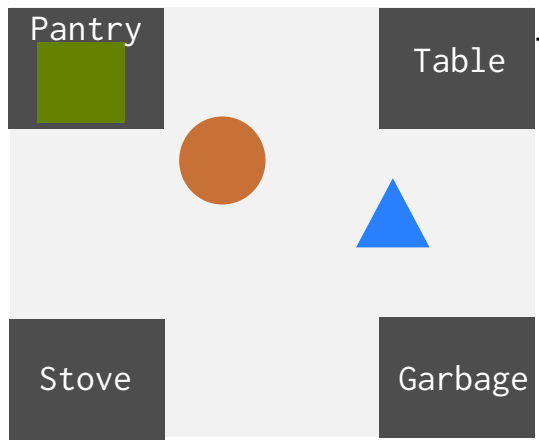
Agent: Which action should I take?

Instructor: Close the pantry.



Episodic Memory



# 1. Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?

Instructor: Open the pantry.

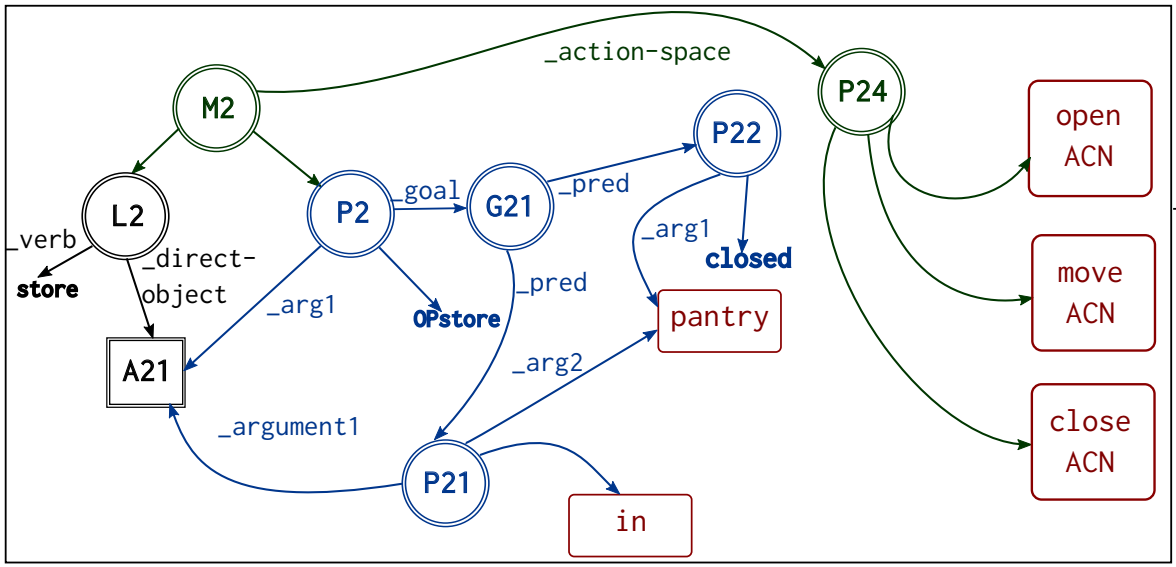
Agent: Which action should I take?

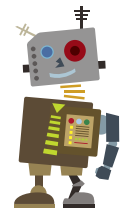
Instructor: Move the object to the pantry.

Agent: Which action should I take?

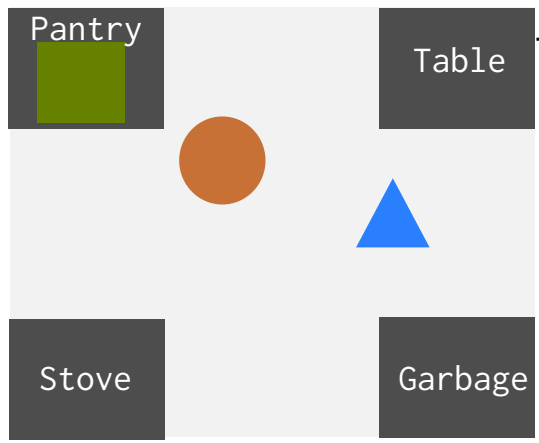
Instructor: Close the pantry.

Agent: Which action should I take?



Episodic Memory



# 1. Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry is closed.

Agent: Which action should I take?

Instructor: Open the pantry.

Agent: Which action should I take?

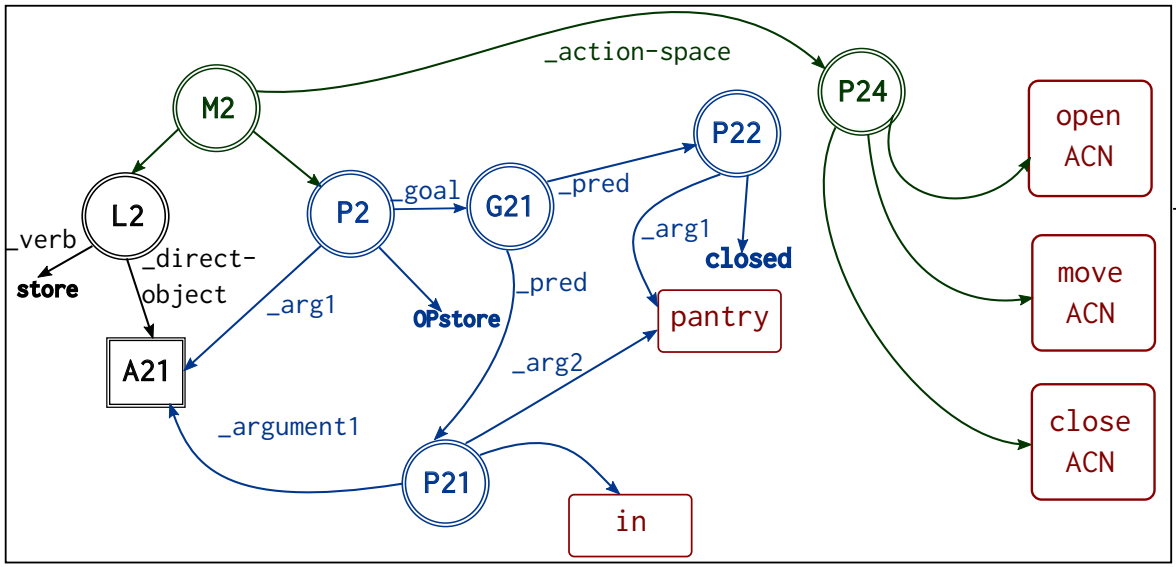
Instructor: Move the object to the pantry.

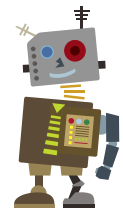
Agent: Which action should I take?

Instructor: Close the pantry.

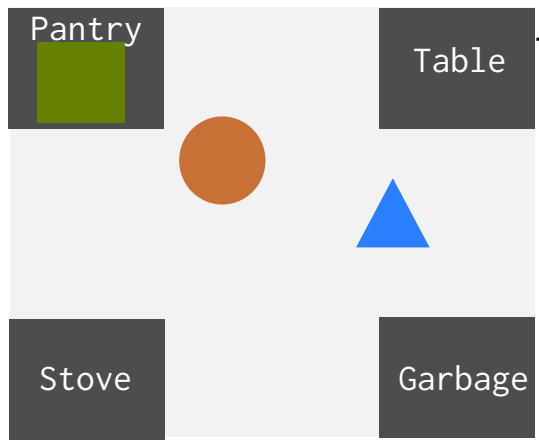
Agent: Which action should I take?

Instructor: You are done.



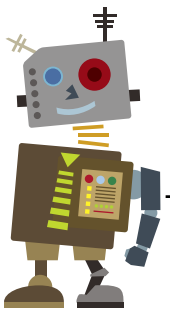
Episodic Memory



# 2. Retrospective Explanation

Concept Network

Episodic Memory

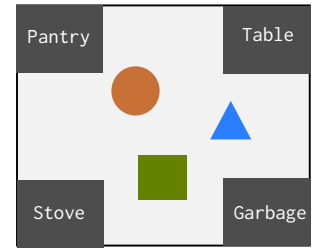


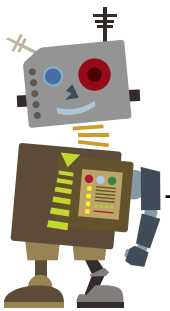
# 2. Retrospective Explanation

Concept Network

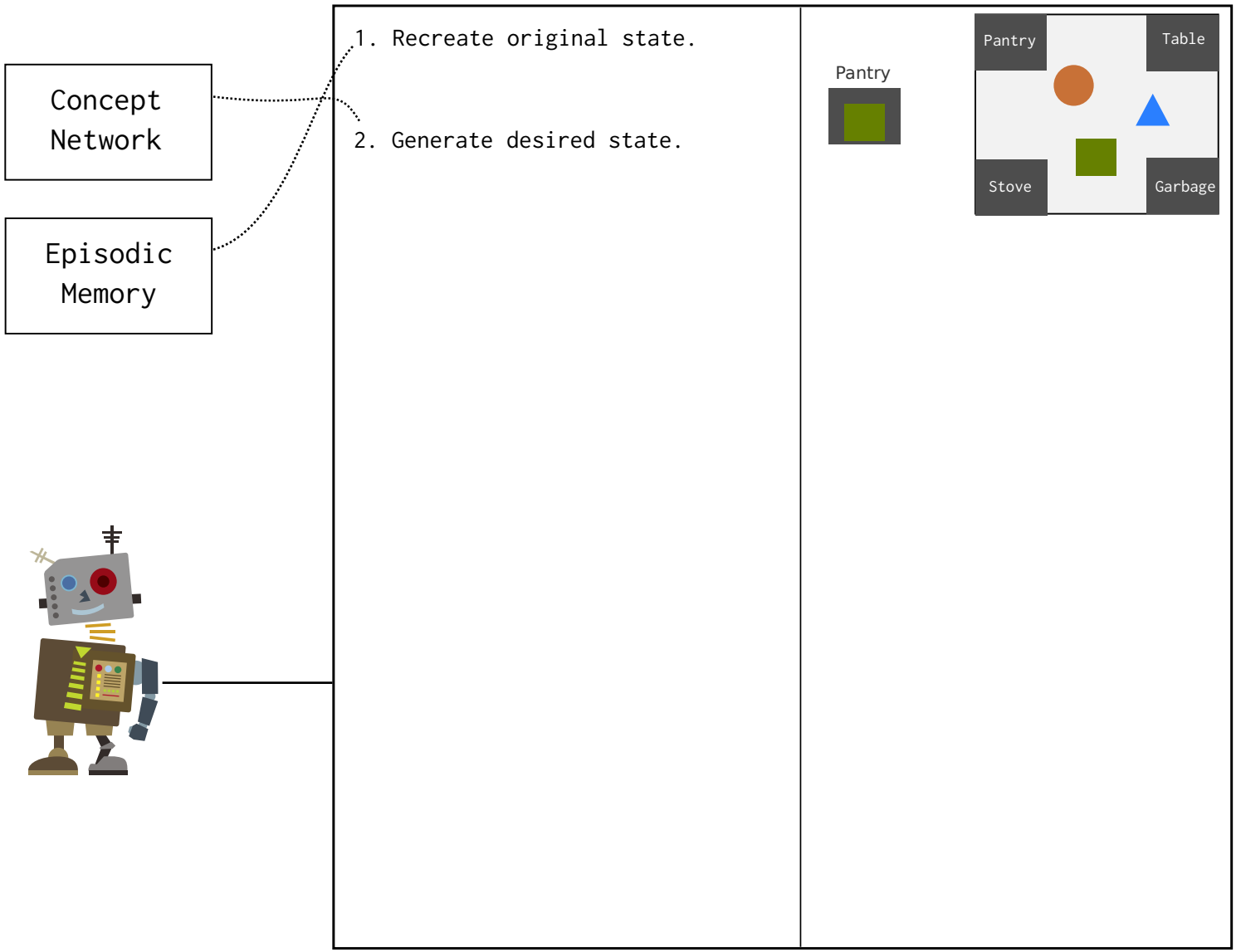
Episodic Memory

1. Recreate original state.

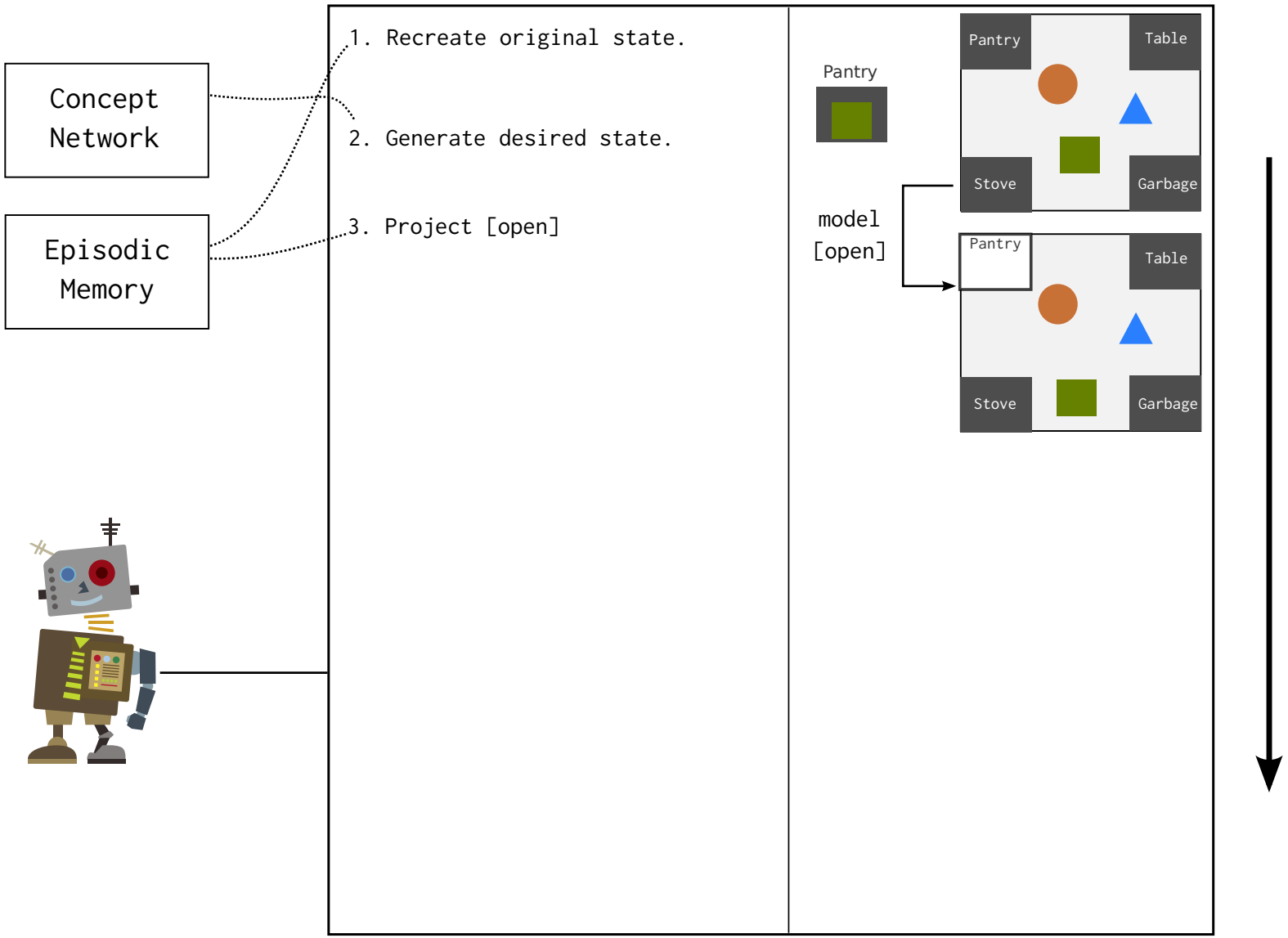


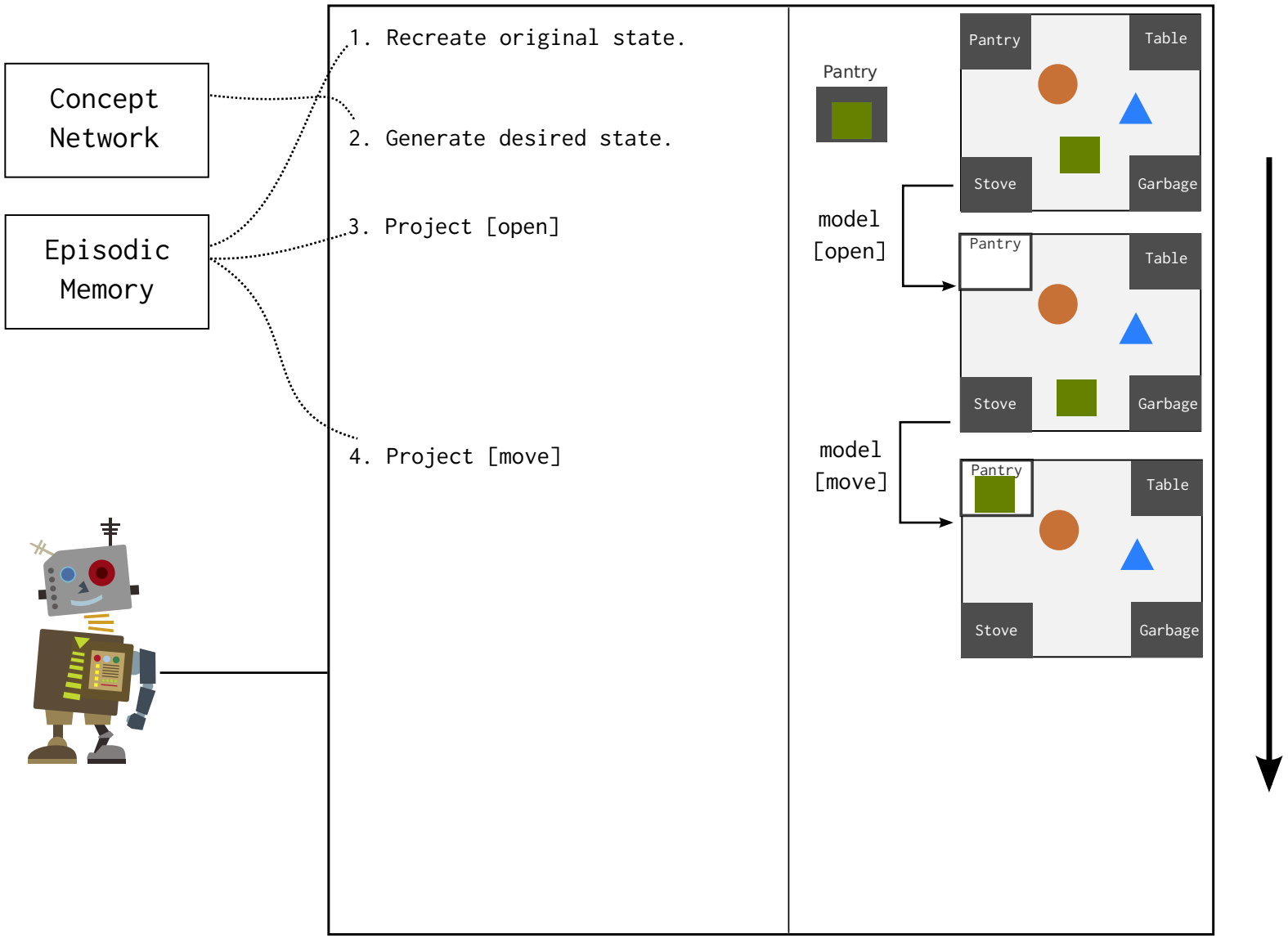
# 2. Retrospective Explanation



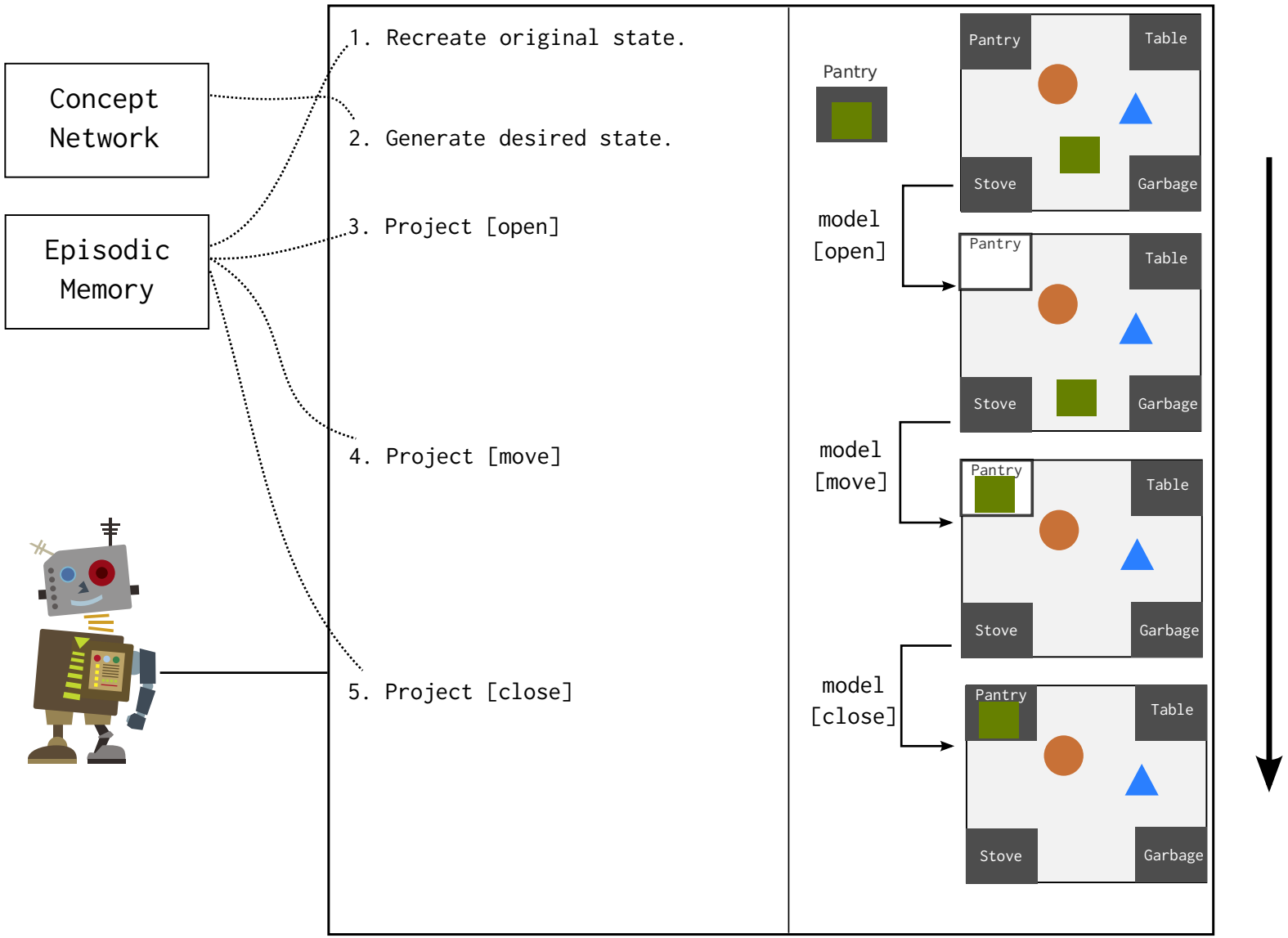
# 2. Retrospective Explanation



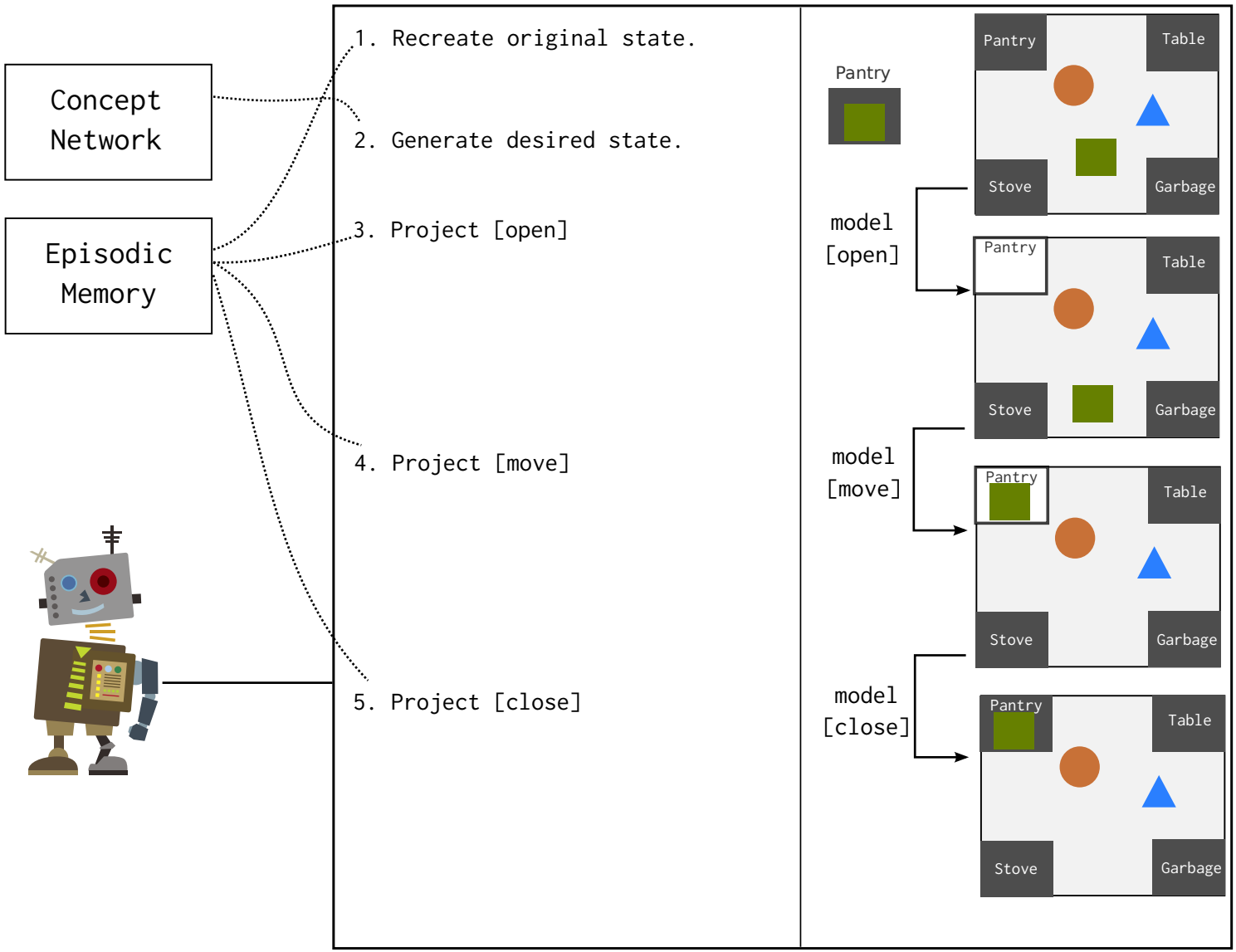
# 2. Retrospective Explanation



# 2. Retrospective Explanation

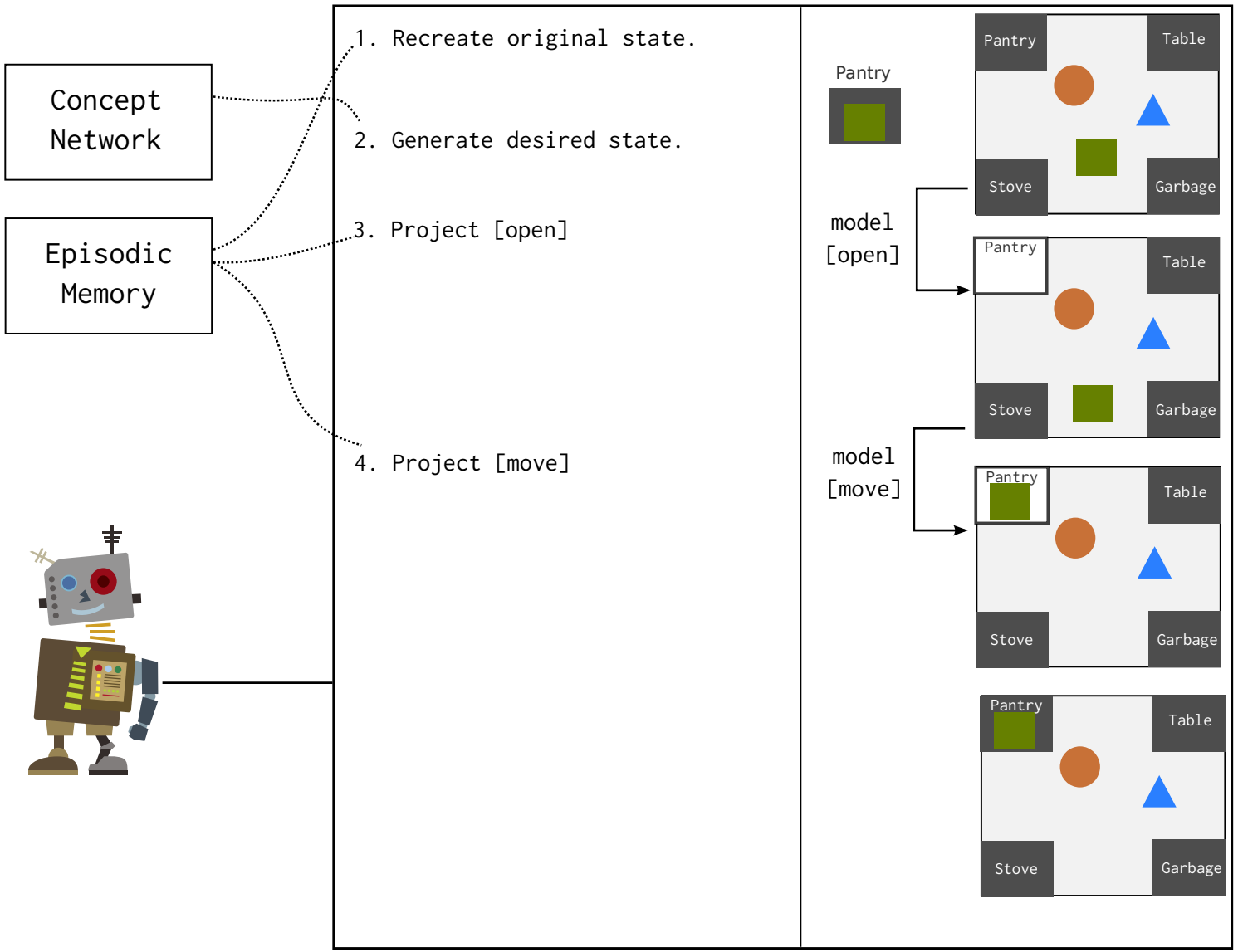


# 2. Retrospective Explanation




**select close**:
If `store([x])` and `IN([x],PANTRY)` and `OPEN(PANTRY)`
-->
select `close(PANTRY)`

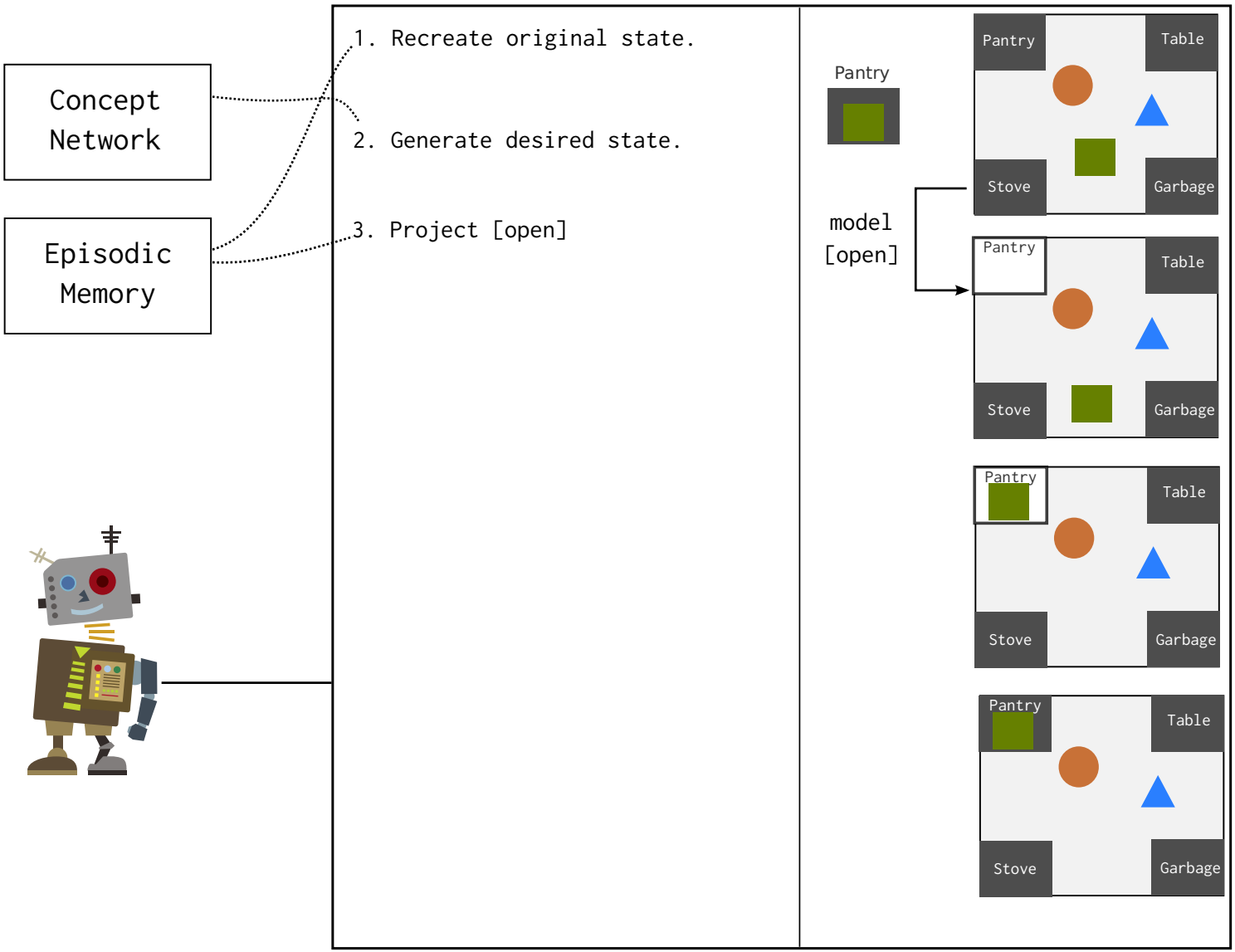
# 2. Retrospective Explanation




**select close**:
If `store([x])` and `IN([x],PANTRY)` and `OPEN(PANTRY)`
-->
select `close(PANTRY)`

**select move**:
If `store([x])` and `-IN([x],PANTRY)` and `OPEN(PANTRY)`
-->
select `move([x],PANTRY)`

# 2. Retrospective Explanation




**select close**:
If `store([x])` and `IN([x],PANTRY)` and `OPEN(PANTRY)`
-->
select `close(PANTRY)`

**select move**:
If `store([x])` and `-IN([x],PANTRY)` and `OPEN(PANTRY)`
-->
select `move([x],PANTRY)`

**select open**:
If `store([x])` and `-IN([x],PANTRY)` and `CLOSED(PANTRY)`
-->
select `open(PANTRY)`

# Evaluation Overview

3 dimensions

- Variety of tasks taught
- Generalization
- Transfer

# 8 Tasks

`place([x],[rel],[y])`,
`move([x],[y])`, `discard([x])`

---

`store([x])`, `cook([x])`,
`serve([x])`

---

`stack-3([x],[y],[z])`,
`stack-4([x],[y],[z],[w])`

# 8 Tasks

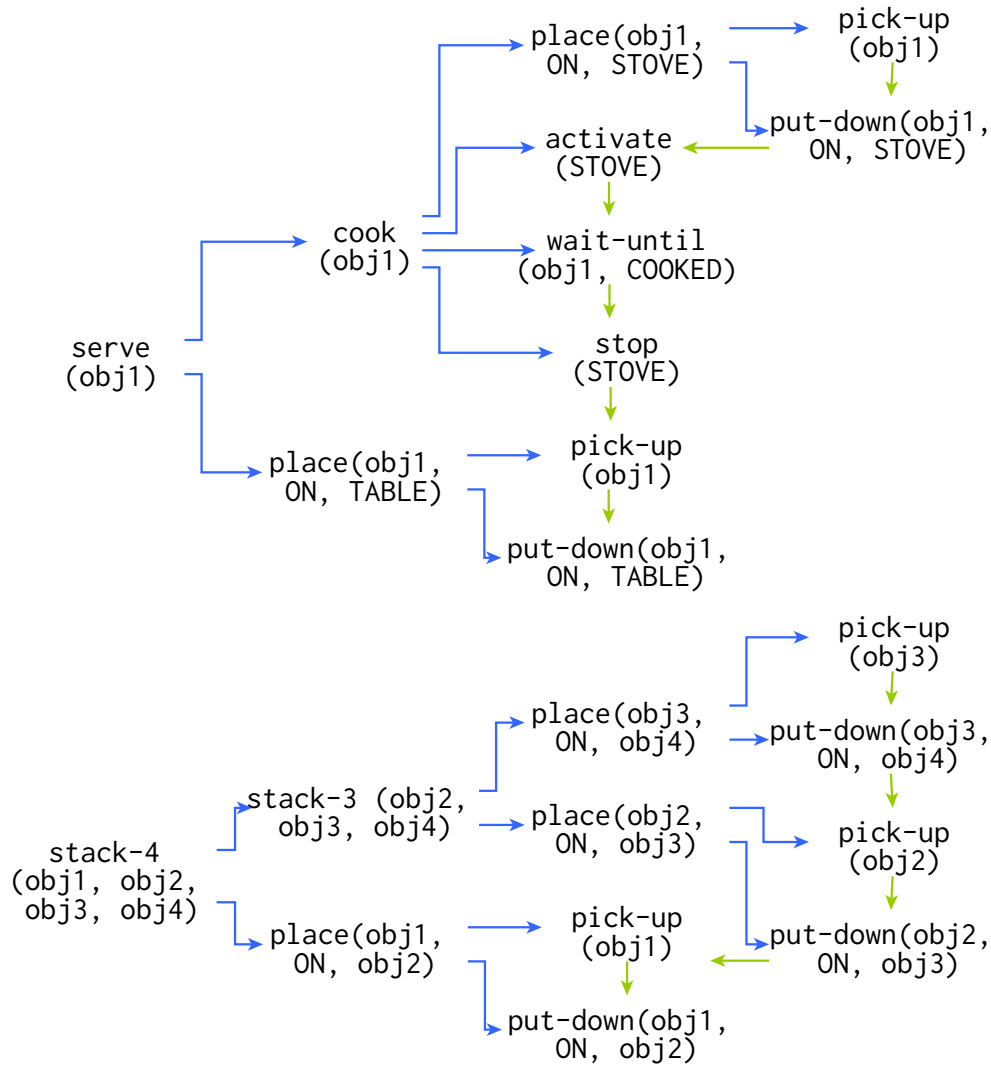
`place([x],[rel],[y])`,
`move([x],[y])`, `discard([x])`

---

`store([x])`, `cook([x])`,
`serve([x])`

---

`stack-3([x],[y],[z])`,
`stack-4([x],[y],[z],[w])`



# Generalization

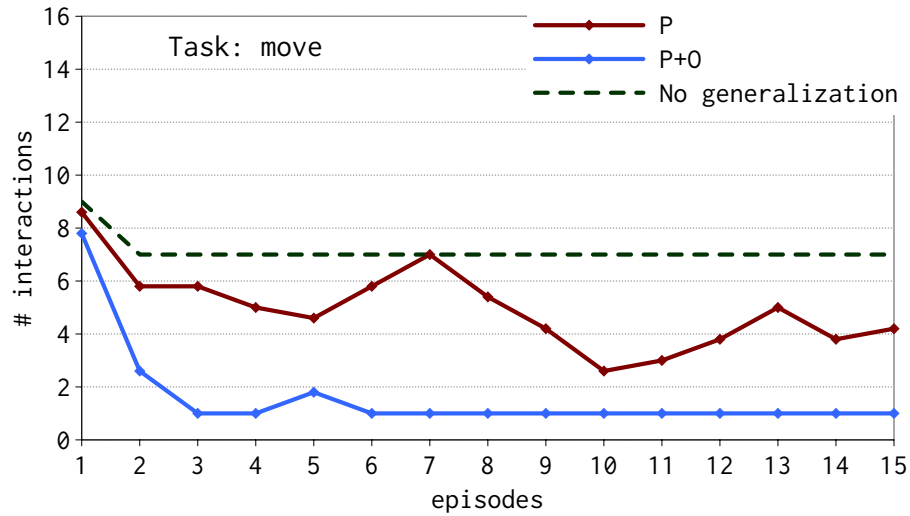
- Predicate selection
  **select open**:
  If `store(O1)` and `-IN(O1,PANTRY)` and `CLOSED(PANTRY)` and ~~`CLOSED(STOVE)` and `OFF(STOVE)` and `-ON(O2,STOVE)` and ...~~
  -->
  select `open(PANTRY)`

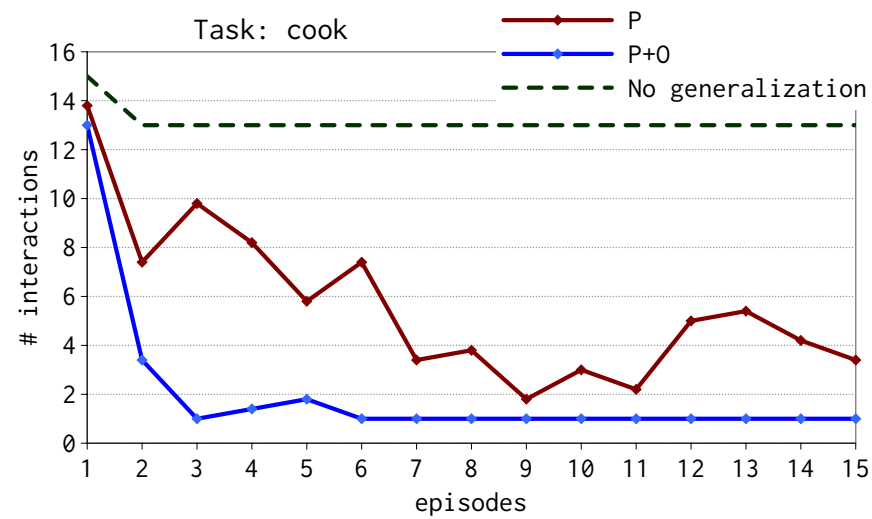
- Object variablization
  Store the green rectangle.
  The goal is the green rectangle in the pantry and the pantry is closed.
  Open the pantry.
  Move the green rectangle to the pantry.

  ...

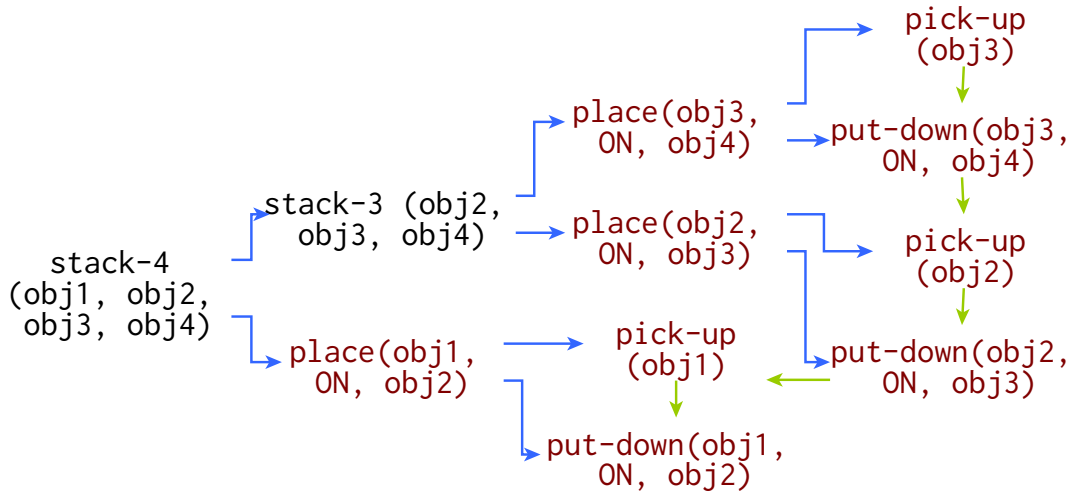
# Generalization

Environment: 4 objects, 4 locations
Variations: P, P+O
Episode: randomly assigned location states and object positions
Command: randomly instantiated arguments

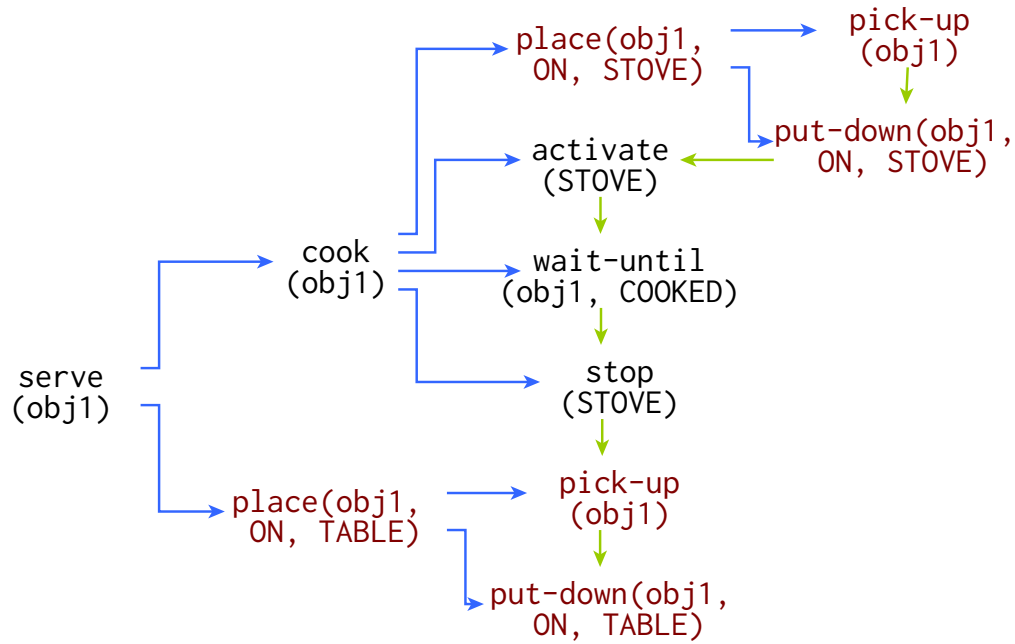


# Common Policy Space

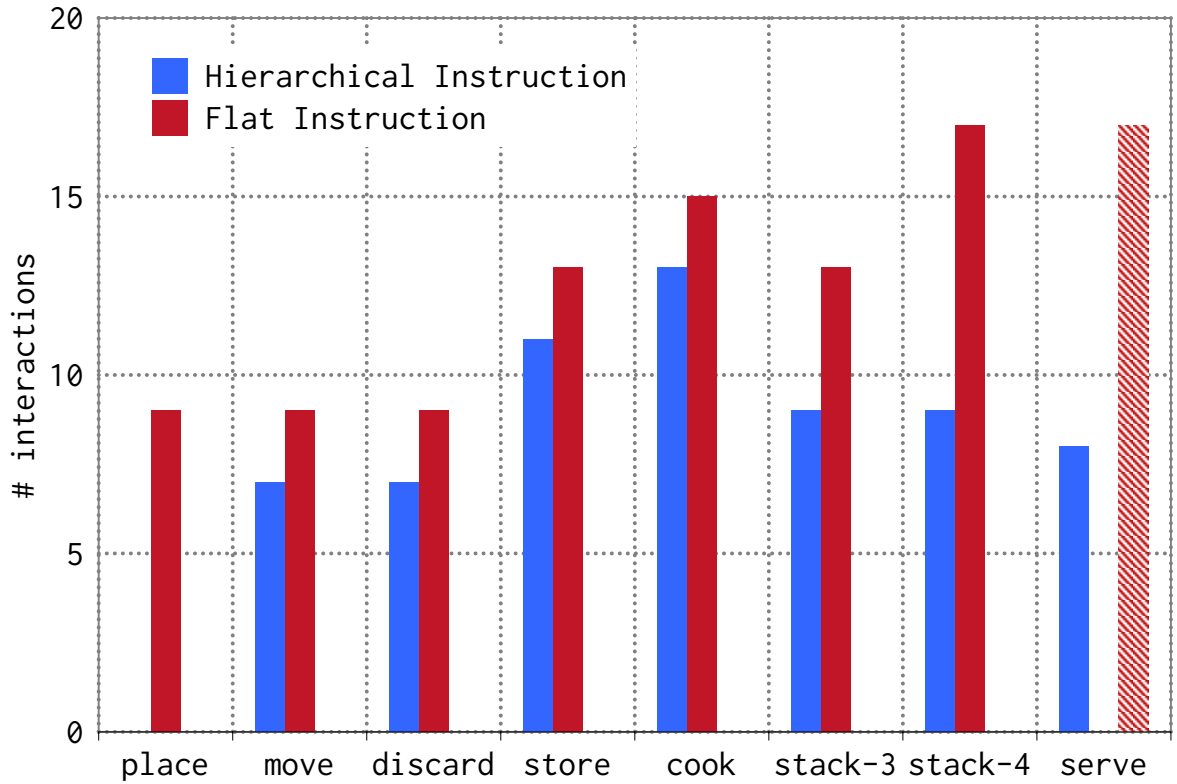



---



# Transfer

- Flat instruction: primitive actions
- Hierarchical instruction: previously taught tasks



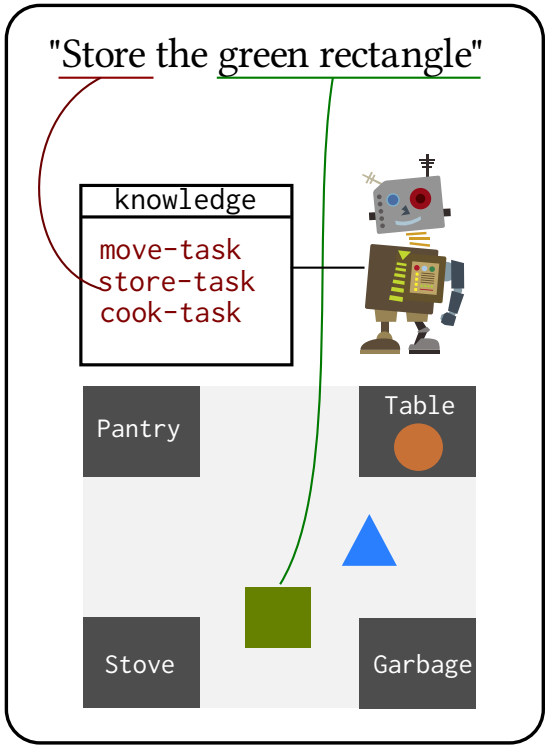
# Summary

- New task $\rightarrow$ diverse knowledge
  - parameters, decomposition structure, policy, model, availability, termination
- Situated Interactive Instruction (SII)
  - *knowledge-level* interactions
  - natural language refers to concepts known to the agent
  - instructor can compose new knowledge
- SII may help task learning
  - generalization: identifying useful features (predicates), variablization
  - transfer: tasks may have common substructure

# Situated Comprehension

## Outline

1. Problem Description
2. Approach - the Indexical Model
3. Addressing complexities
   1. Referring expression resolution
   2. Unexpressed argument alternation



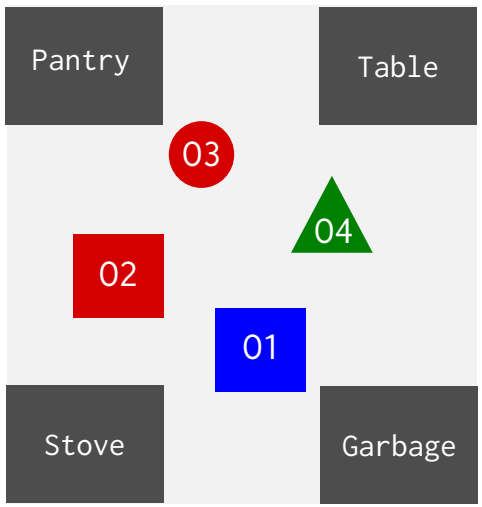

---

Mohan, Mininger, and Laird. *Towards an Indexical Model of Situated Language Comprehension for Real-World Cognitive Agents.* Advances in Cognitive Systems. 2013.

# Situated Comprehension

- Communication
  - is situated
  - is contextual
    - efficient, effective
    - linguistically ambiguous
  - is interactive
- Comprehension model
  - exploit non-linguistic context
  - inform interaction
- Challenges
  - mixed representations
  - continual knowledge acquisition

*Store the blue rectangle.*
*The goal is the rectangle is in the pantry.*
...
*Pick it up.*
...



# Hypothesis

Embodied Psycholinguistics: Glenberg and Robertson (1999), Barsalou (1999), Zwaan (2003)

- Linguistic communication is reference
- Speaker/hearer have a common ground
  - shared perceptions
  - common-sense knowledge
  - similar experiences
- Linguistic features are cues to search common ground
- Language specifies scene, knowledge fills up gaps

# The Indexical Model

inspired by the Indexical Hypothesis: Glenberg and Robertson (1999)

task-oriented comprehension
three steps

1. Index words and phrases to referents
   - *NN/ADJ* $\rightarrow$ perceptual classification
   - *NP* $\rightarrow$ set of objects
   - ...
2. Extract domain-knowledge associated with referents
   - pre-encoded or learned (Mohan et al. 2012)
3. Mesh syntactical and environmental constraints

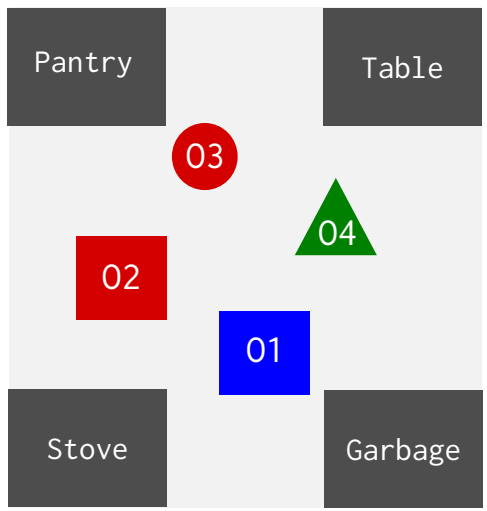
# The Indexical Model

## Step 1: Index components

*Move the blue object to the right of the pantry.*

indexical maps

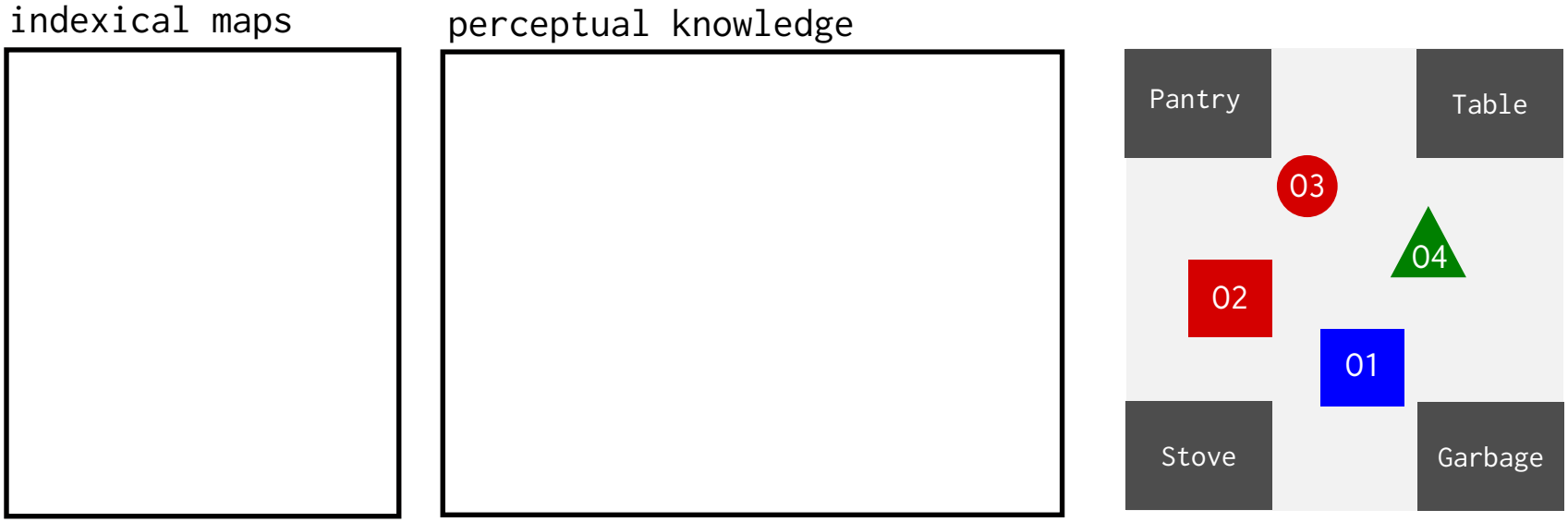
perceptual knowledge



# The Indexical Model

## Step 1: Index components

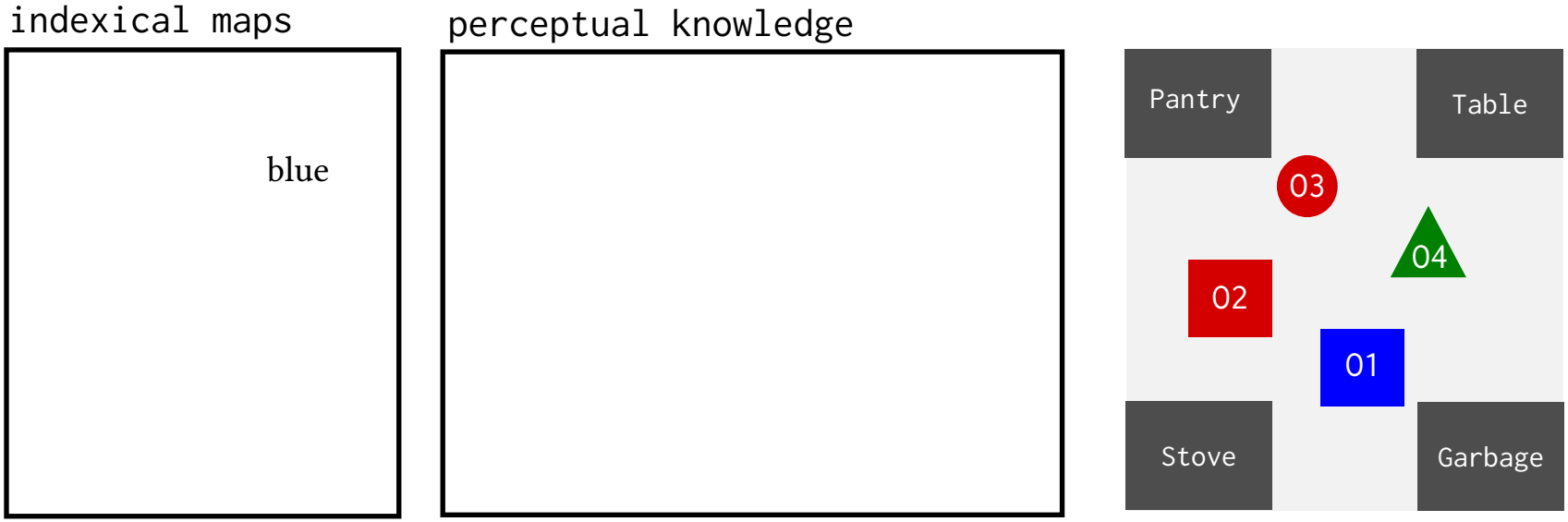
*Move the blue object to the right of the pantry.*



# The Indexical Model

## Step 1: Index components

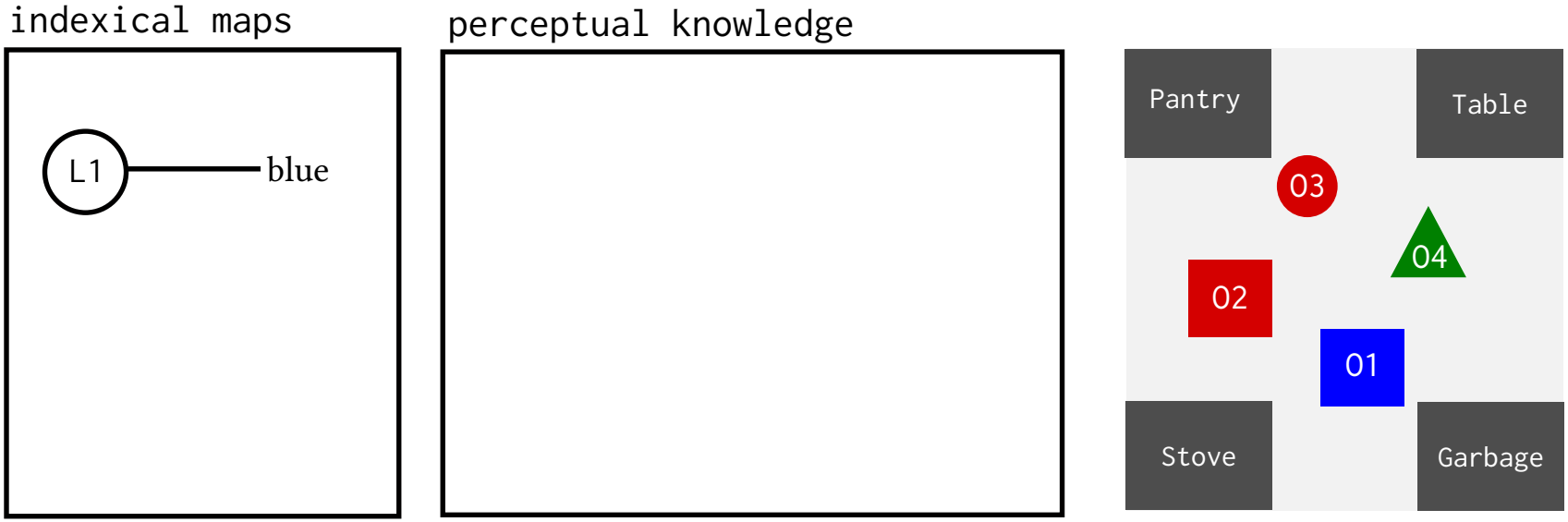
*Move the blue object to the right of the pantry.*



# The Indexical Model

## Step 1: Index components

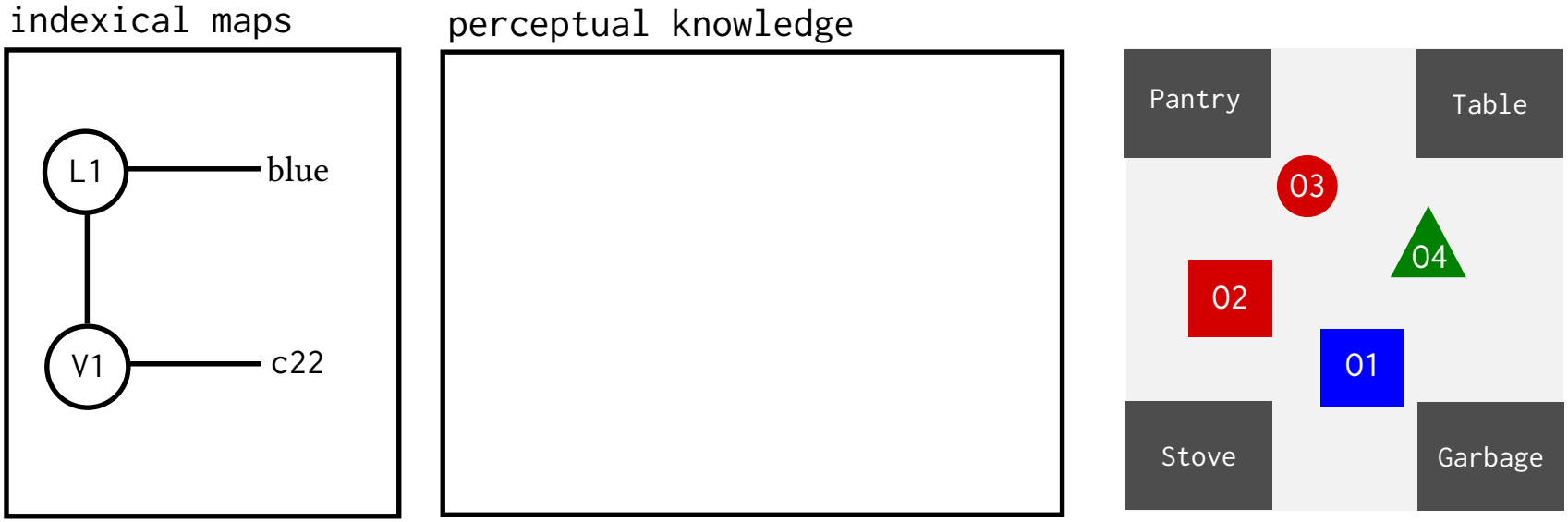
*Move the blue object to the right of the pantry.*



# The Indexical Model

## Step 1: Index components

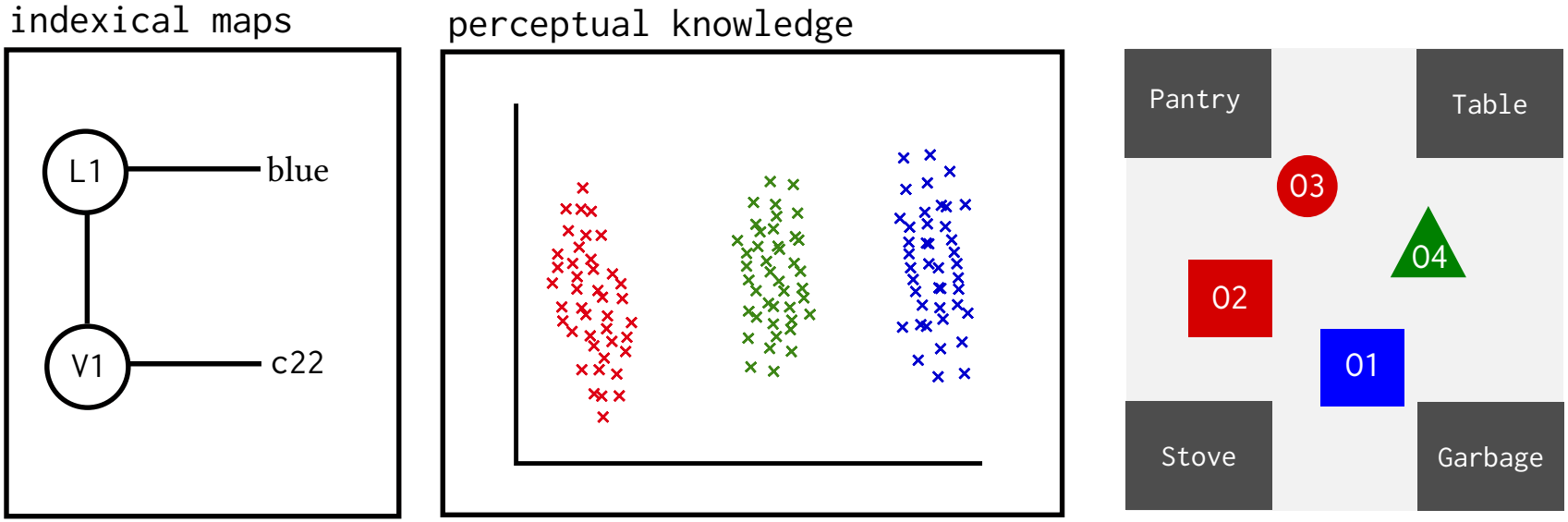
*Move the blue object to the right of the pantry.*



# The Indexical Model

## Step 1: Index components

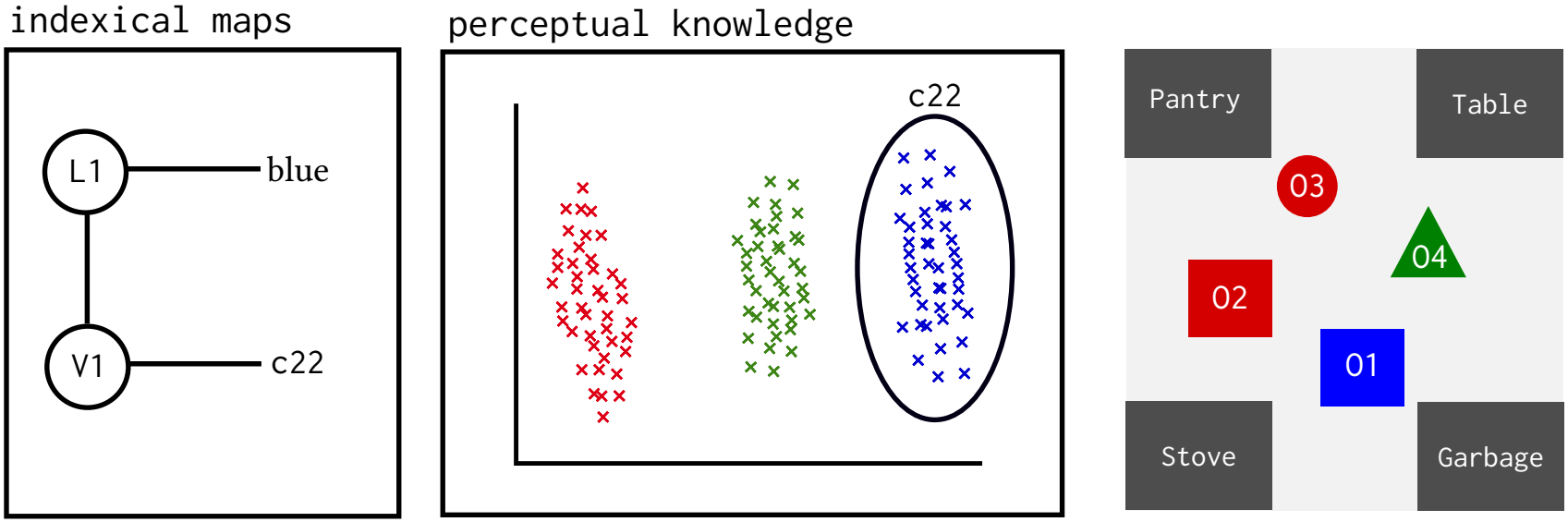
*Move the blue object to the right of the pantry.*



# The Indexical Model

## Step 1: Index components

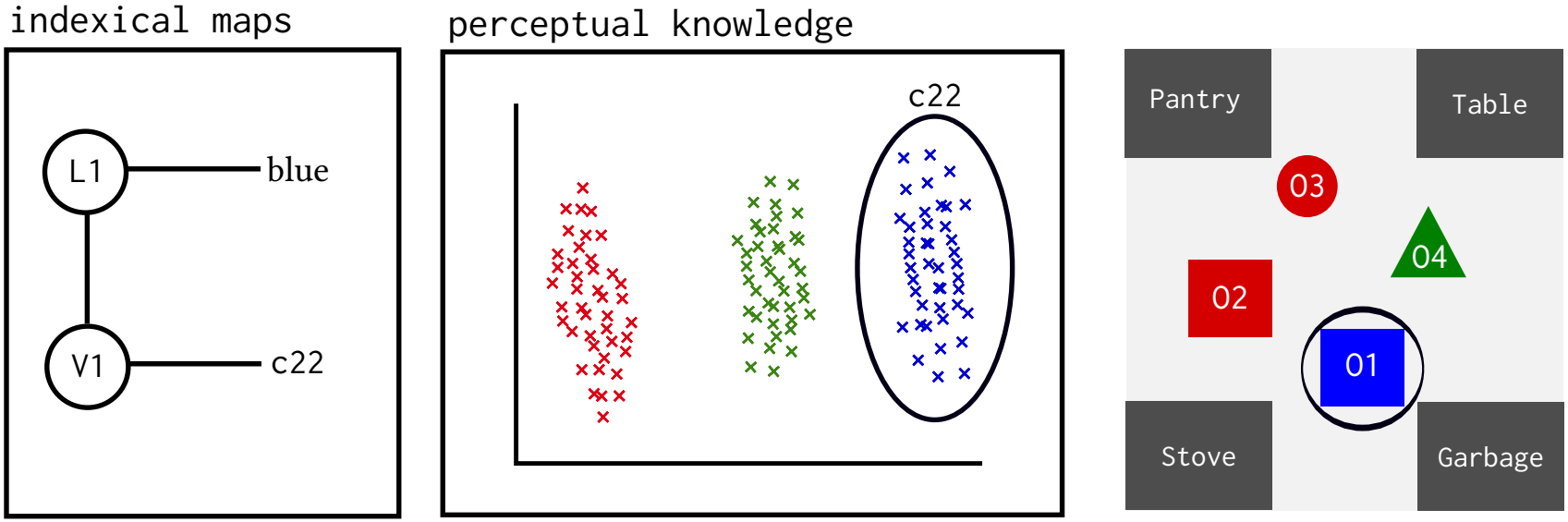
*Move the blue object to the right of the pantry.*



# The Indexical Model

## Step 1: Index components

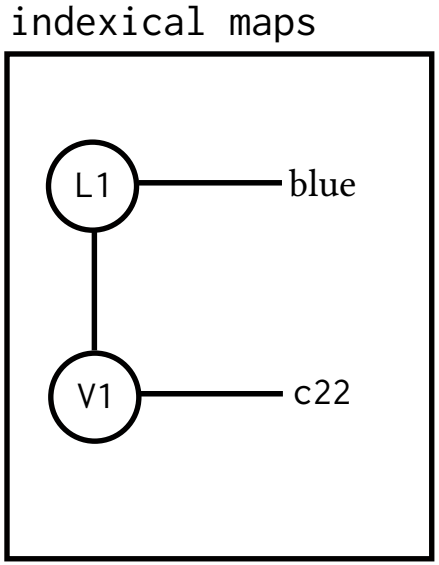
*Move the blue object to the right of the pantry.*

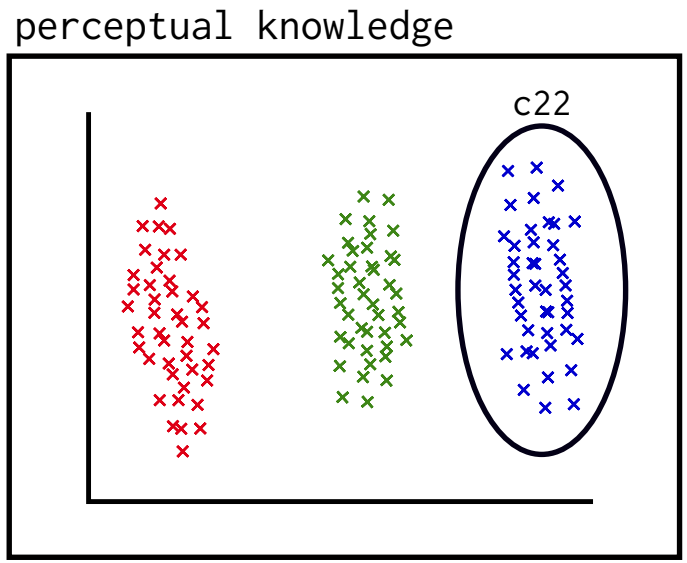


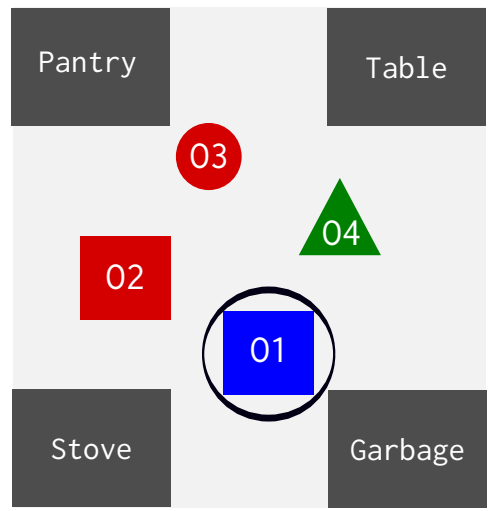
# The Indexical Model

## Step 1: Index components

*Move the blue object to the right of the pantry.*
01



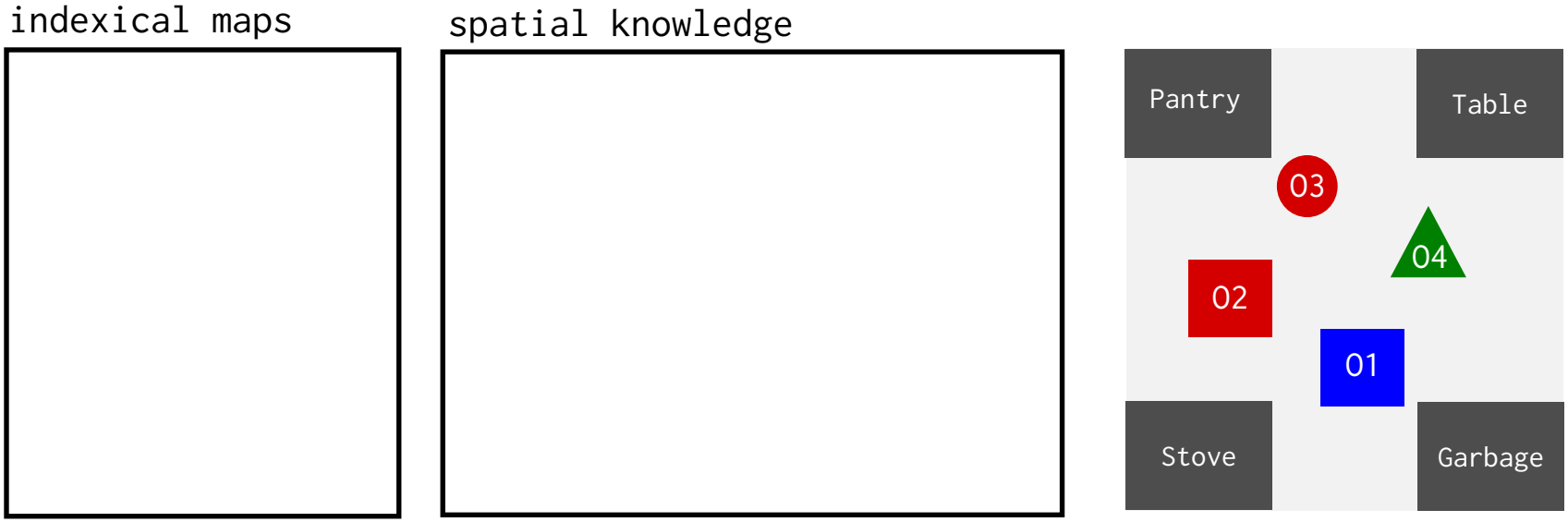



# The Indexical Model

## Step 1: Index components

*Move the blue object to the right of the pantry.*

01

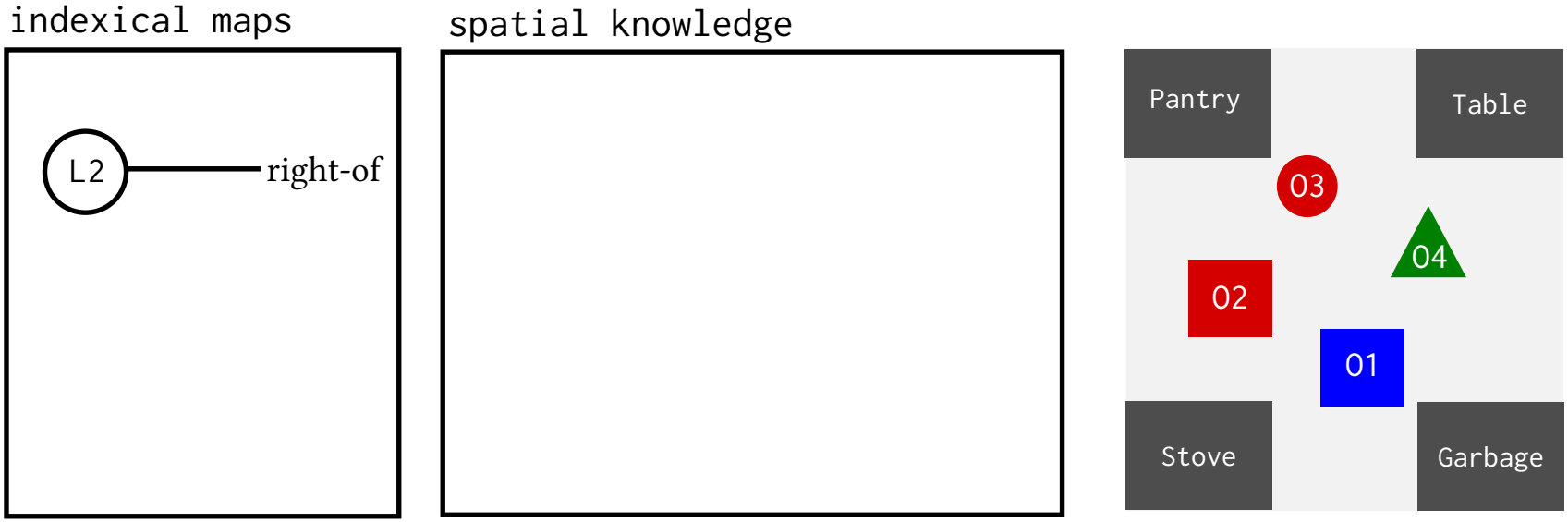


# The Indexical Model

## Step 1: Index components

*Move the blue object to the right of the pantry.*

01

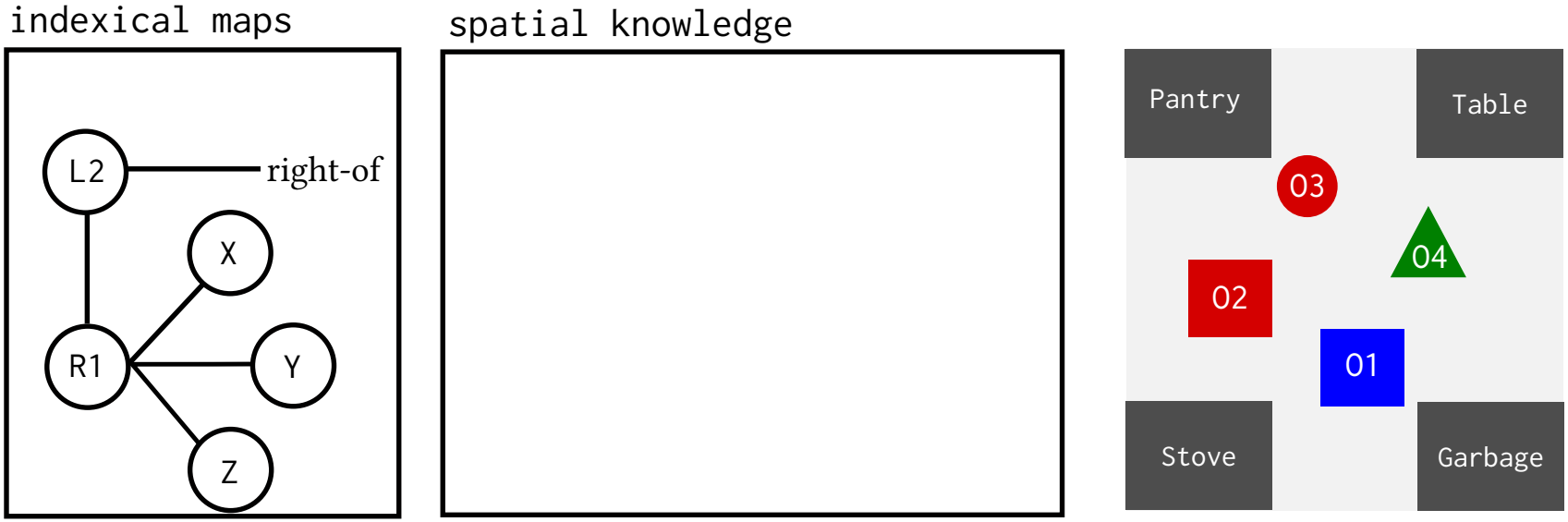


# The Indexical Model

## Step 1: Index components

*Move the blue object to the right of the pantry.*

01

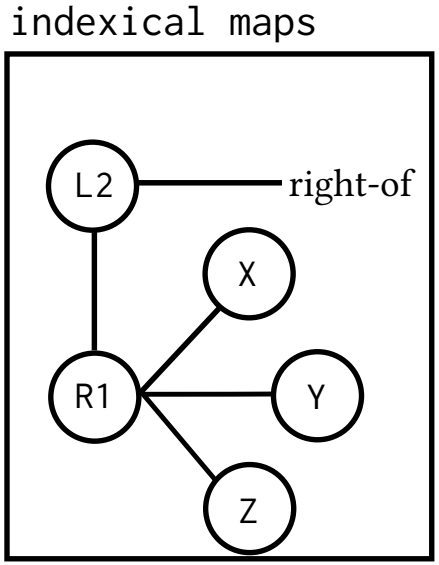


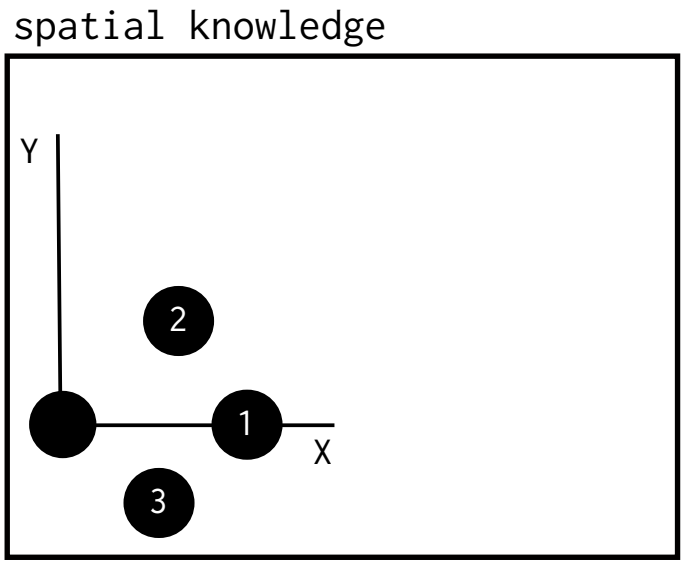
# The Indexical Model

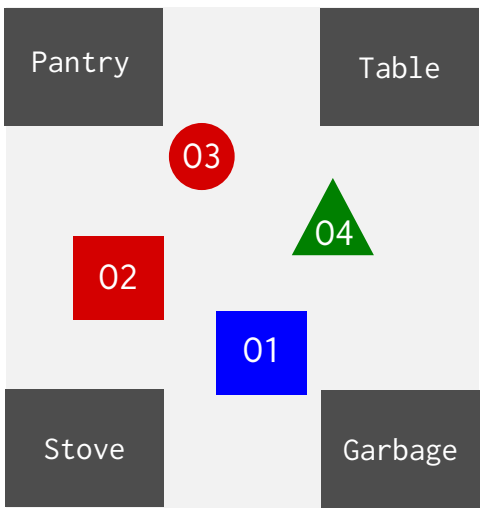
## Step 1: Index components

*Move the blue object to the right of the pantry.*

01



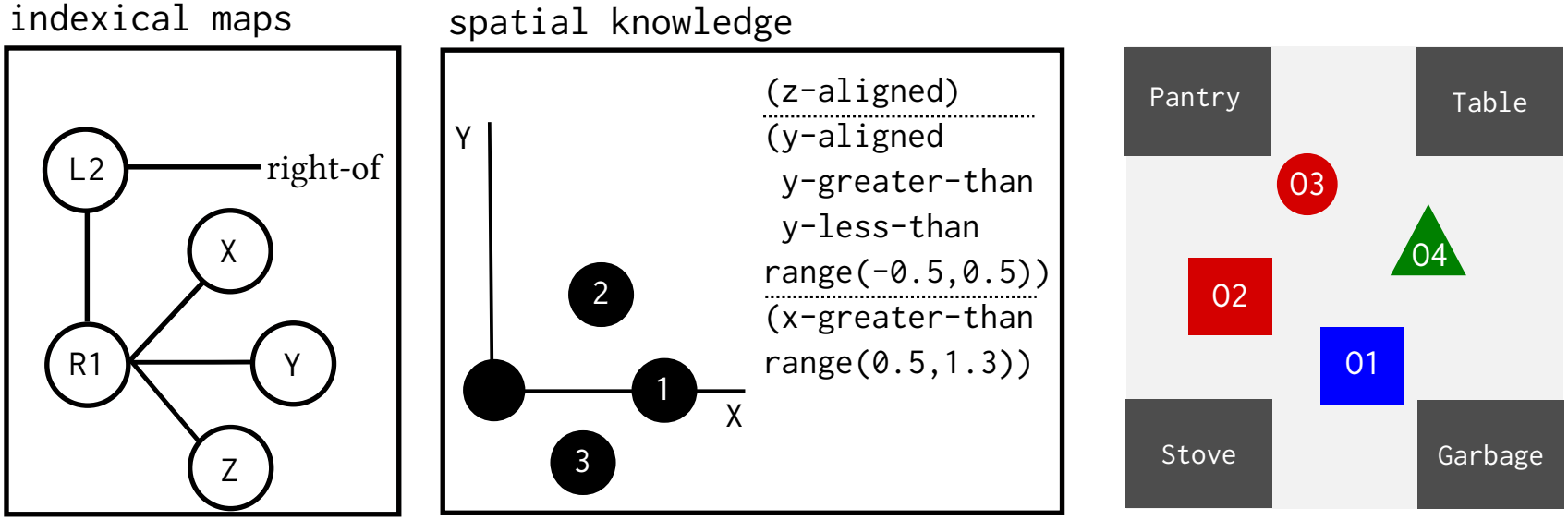



# The Indexical Model

## Step 1: Index components

*Move the blue object to the right of the pantry.*

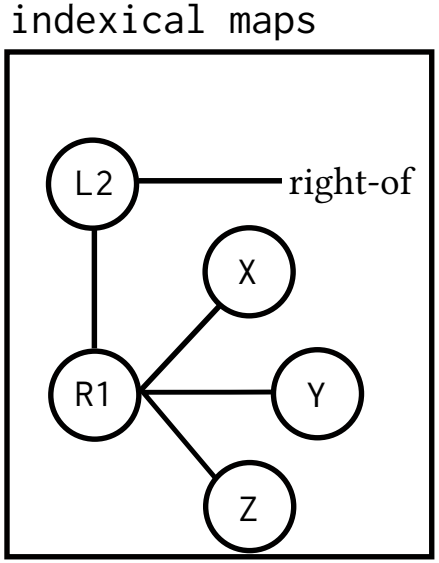
01

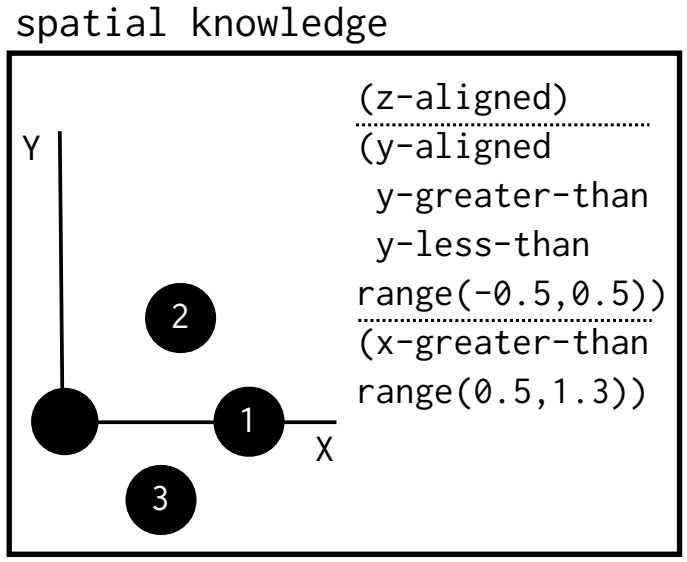


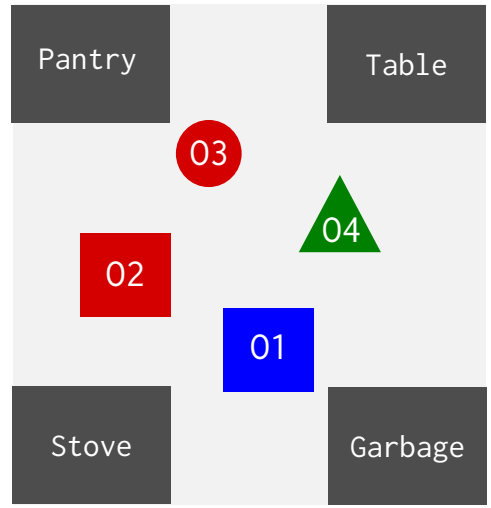
# The Indexical Model

## Step 1: Index components

*Move the blue object to the right of the pantry.*
O1 R1



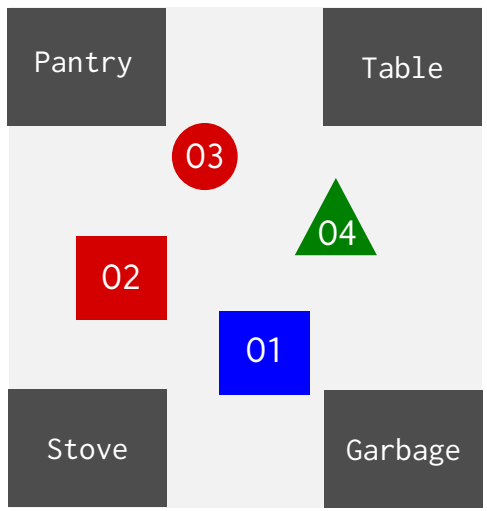
# The Indexical Model

## Step 1: Index components

*Move* the blue object to the right of the pantry.
O1 R1

indexical maps

task knowledge

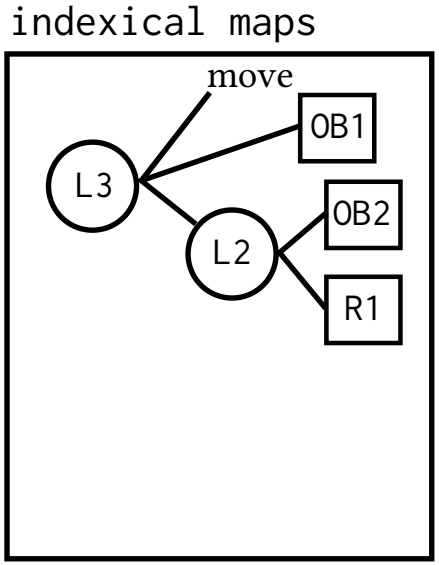


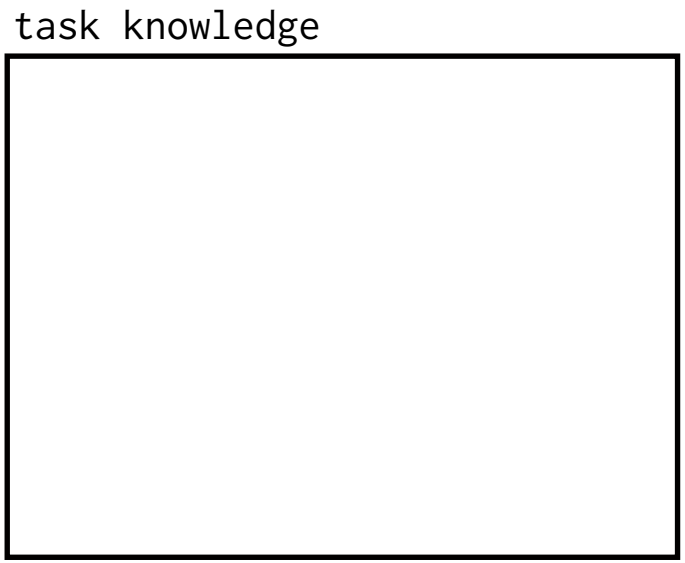
# The Indexical Model

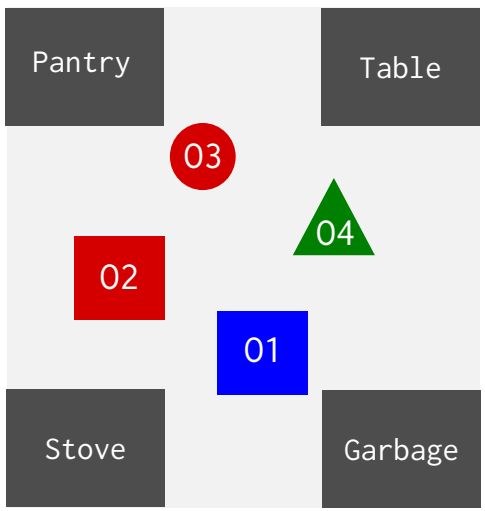
## Step 1: Index components

*Move the blue object to the right of the pantry.*

O1 R1



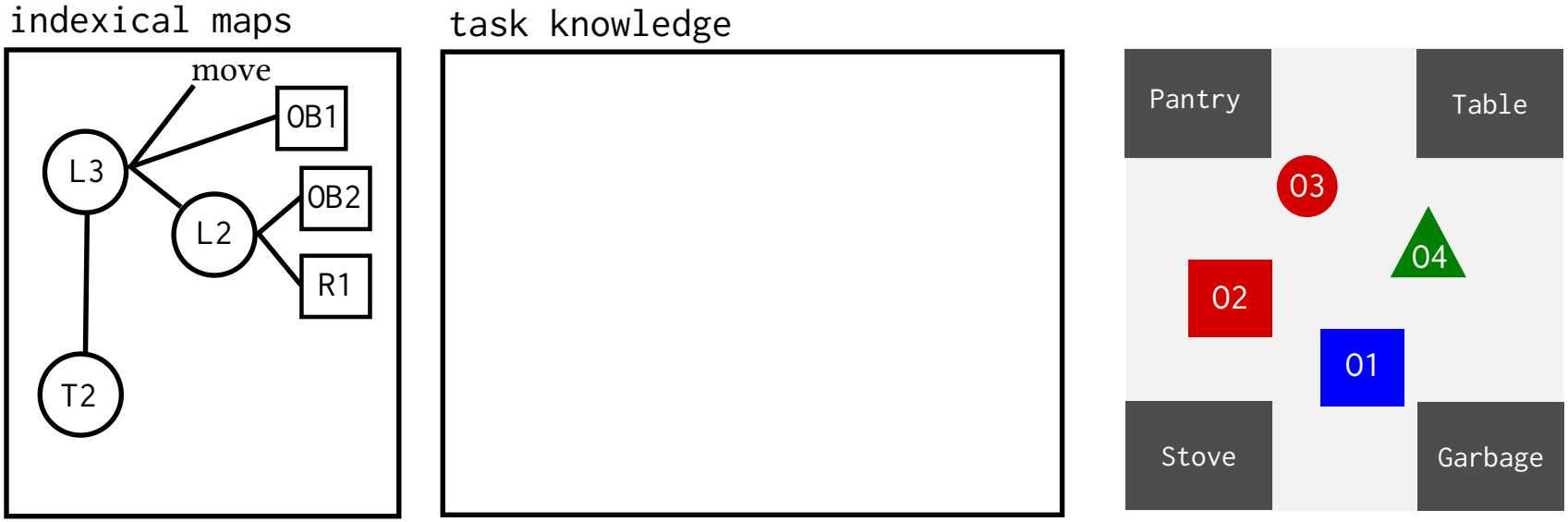



# The Indexical Model

## Step 1: Index components

*Move the blue object to the right of the pantry.*

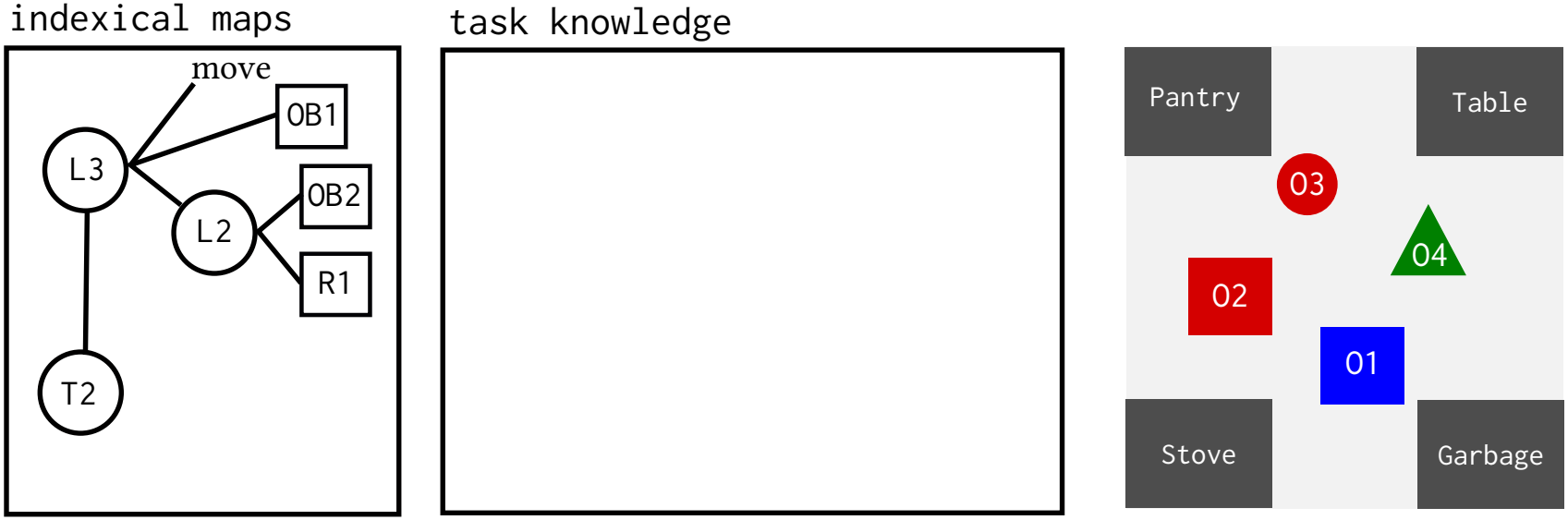
O1 R1



# The Indexical Model

## Step 1: Index components

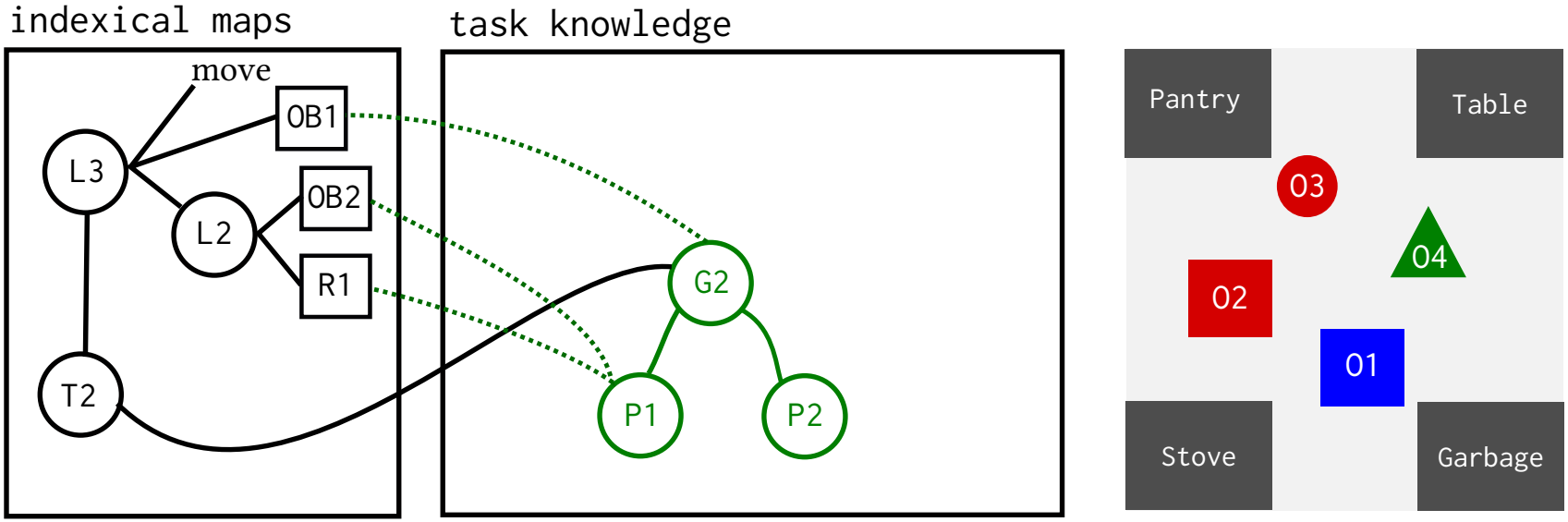
*Move the blue object to the right of the pantry.*
T2 O1 R1



# The Indexical Model

## Step 1: Index components

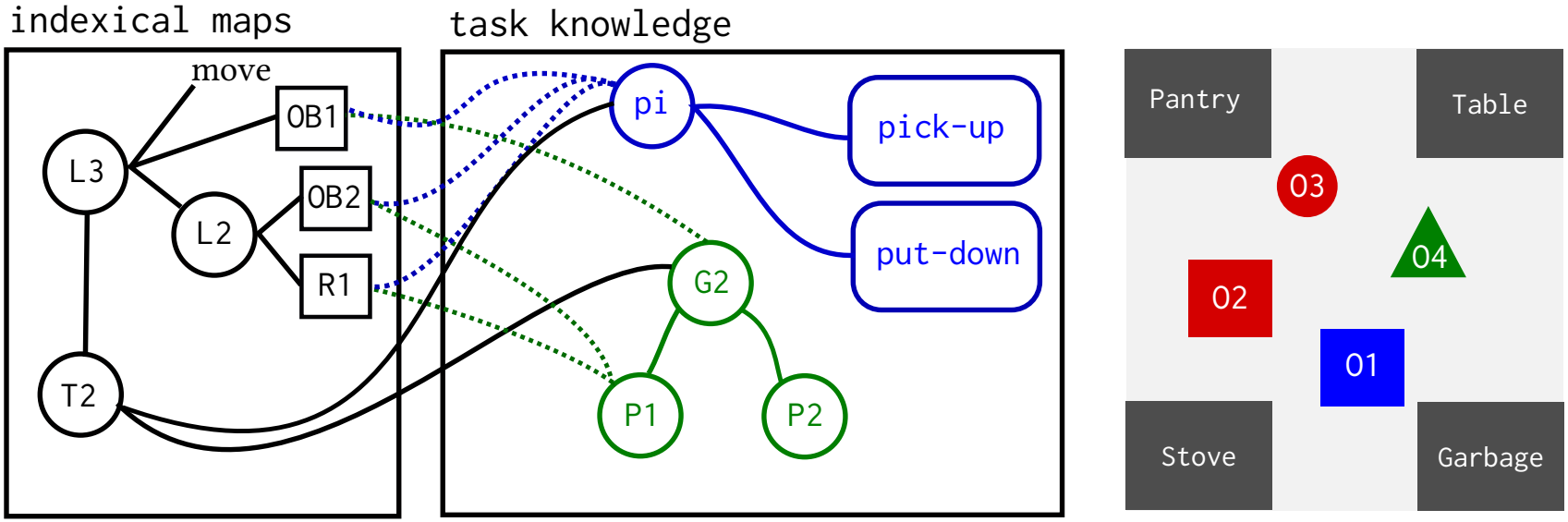
*Move the blue object to the right of the pantry.*
T2 O1 R1



# The Indexical Model

## Step 1: Index components

*Move the blue object to the right of the pantry.*
T2 O1 R1



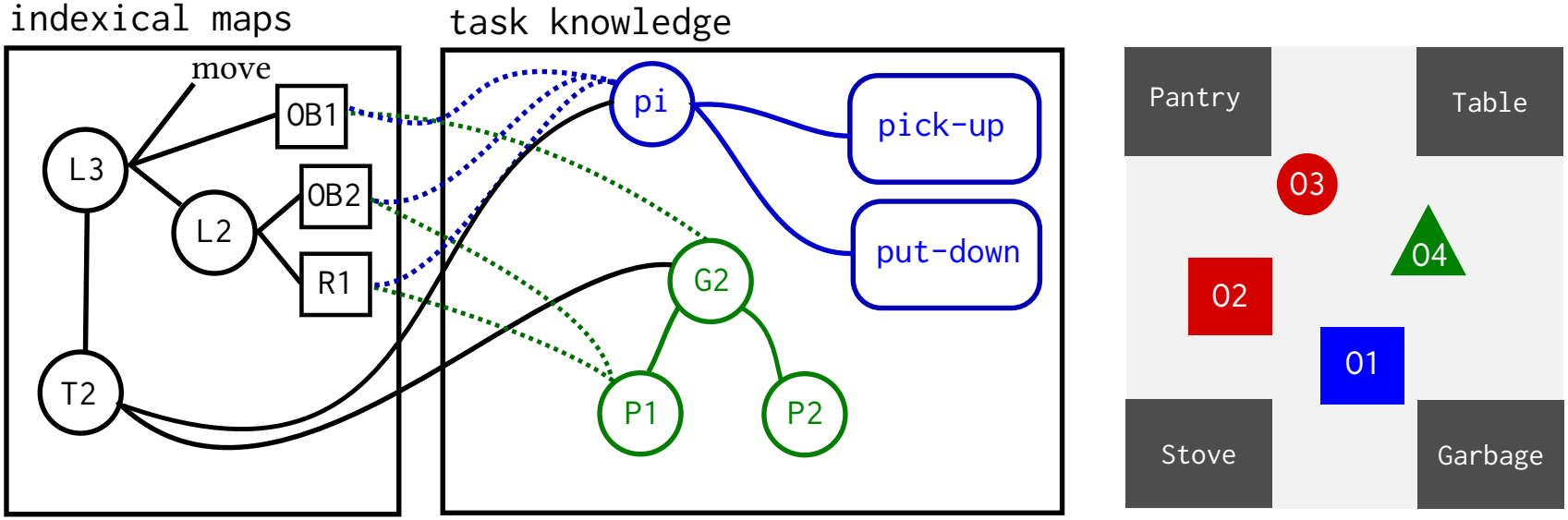
# The Indexical Model

## Step 2: Extract and instantiate domain knowledge

*Move the blue object to the right of the pantry.*

T2 O1 R1

```
task: T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
```



# The Indexical Model

## Step 3: Mesh constraints

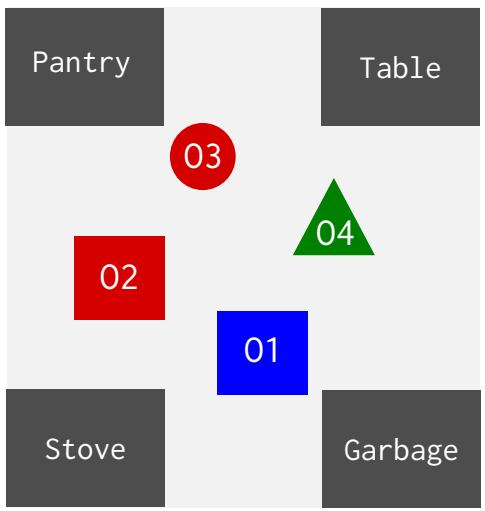
*Move the blue object to the right of the pantry.*

T2 O1 R1

```
task: T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
```

```
available:
T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
T3(O1; G3(O1,(IN,pantry)); pi(O1,IN,pantry))
T4 ...
...


execute:
T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
```



# Addressing Complexities

Natural language is ambiguous, does not completely specify information.

1. Referring Expressions
2. Unexpressed Verb Arguments

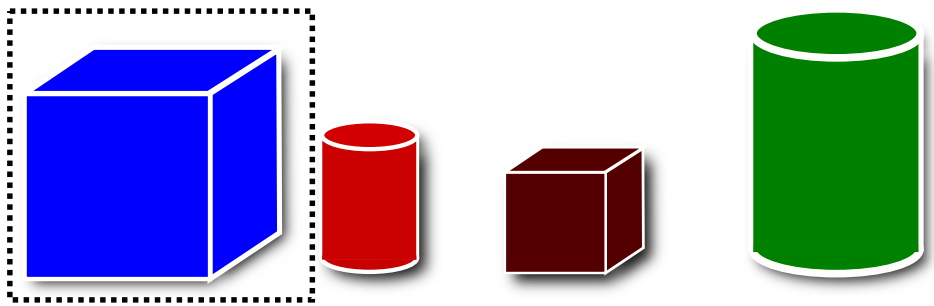
# Referring Expressions

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube* }

# Referring Expressions

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube*}

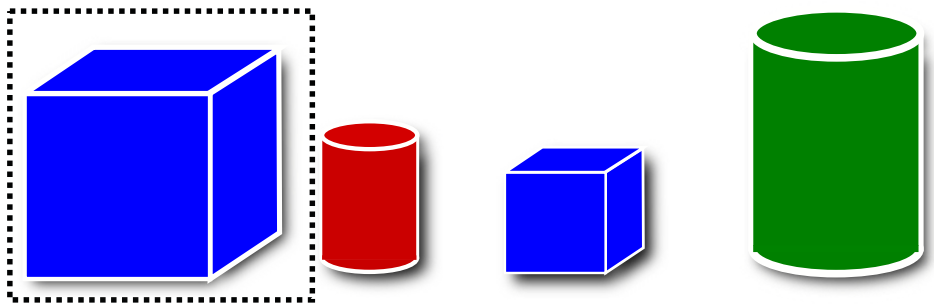
*Pick up the blue cube.*



# Referring Expressions

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube*}

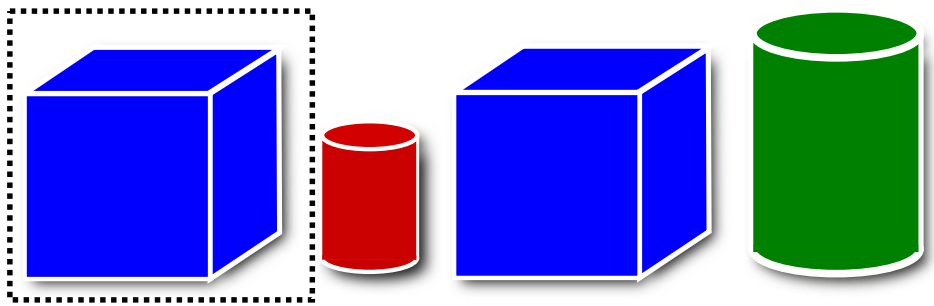
*Pick up the large, blue cube.*



# Referring Expressions

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube*}

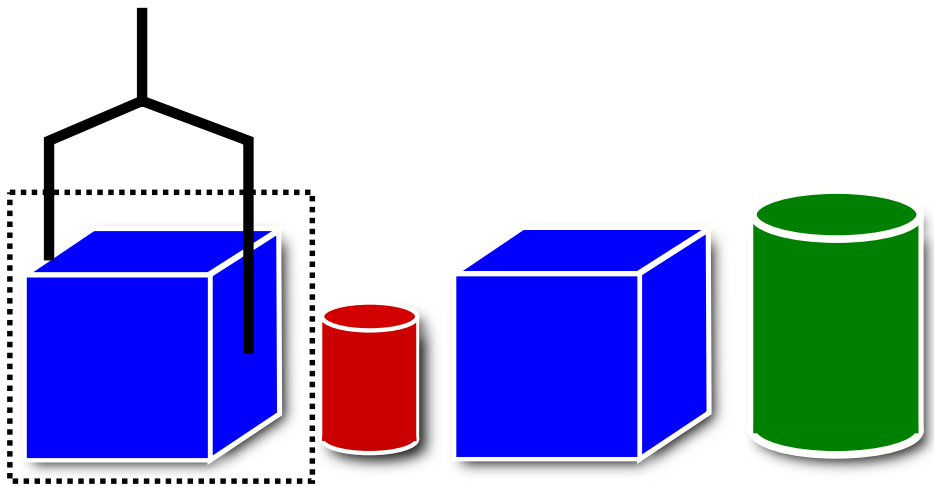
*Pick up the cube on the left of the red cylinder.*



# Referring Expressions

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube*}

*Put it down.*



# Use of Referring Expressions

## The Givenness Hierarchy: Gundel (1993)

- Salience based cognitive-status of referent objects
  - discourse, action, surprise
- Use of referring expression (RE) is dependent on salience
  - more informative RE for less salient objects and vice-versa
  - efficient communication
- RE and cognitive status

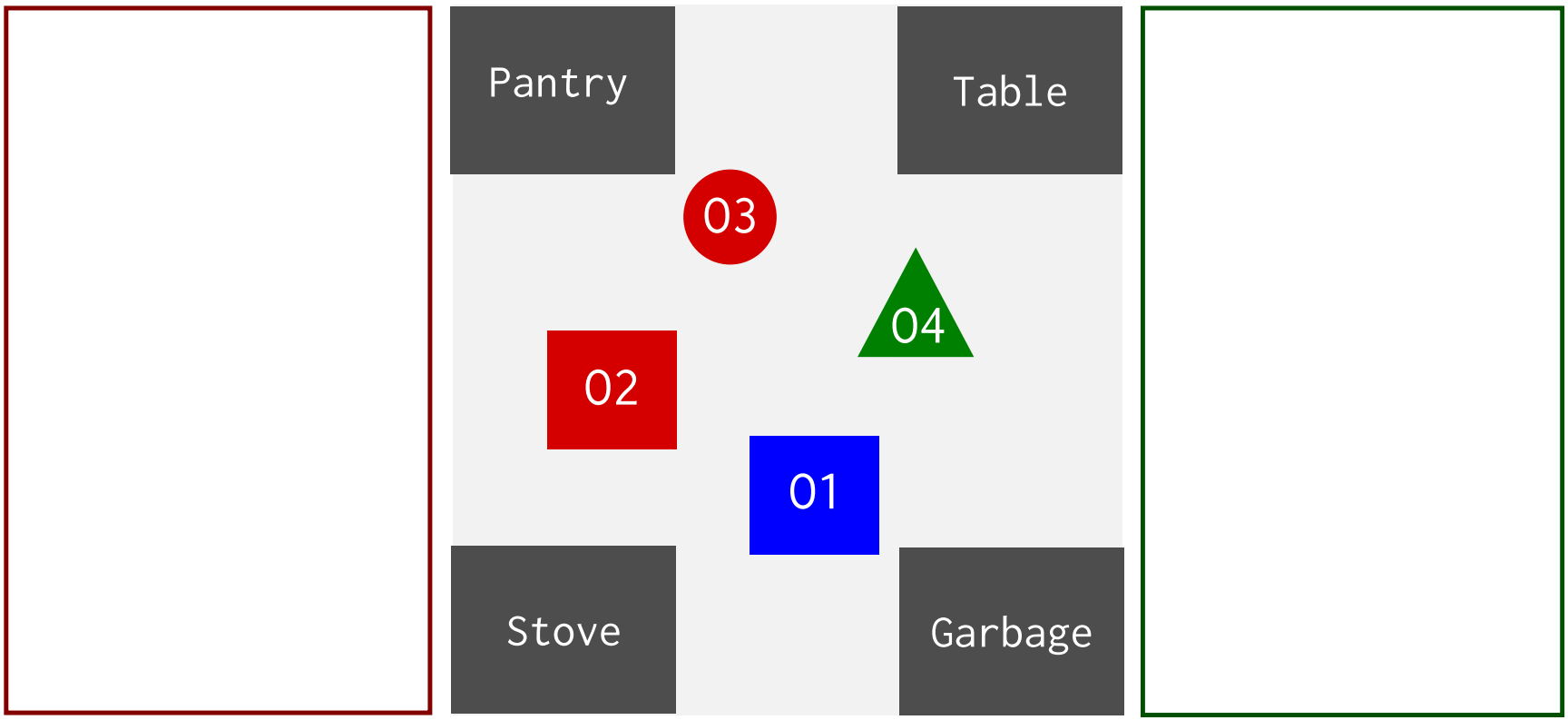
  `in-focus` (*it*) > `activated` (*this, that rectangle*) > `uniquely-identifiable`(*the green rectangle*) > `type-identifiable` (*a green rectangle*)

# RE Resolution Model

Non-linguistic contexts: 2 dimensions

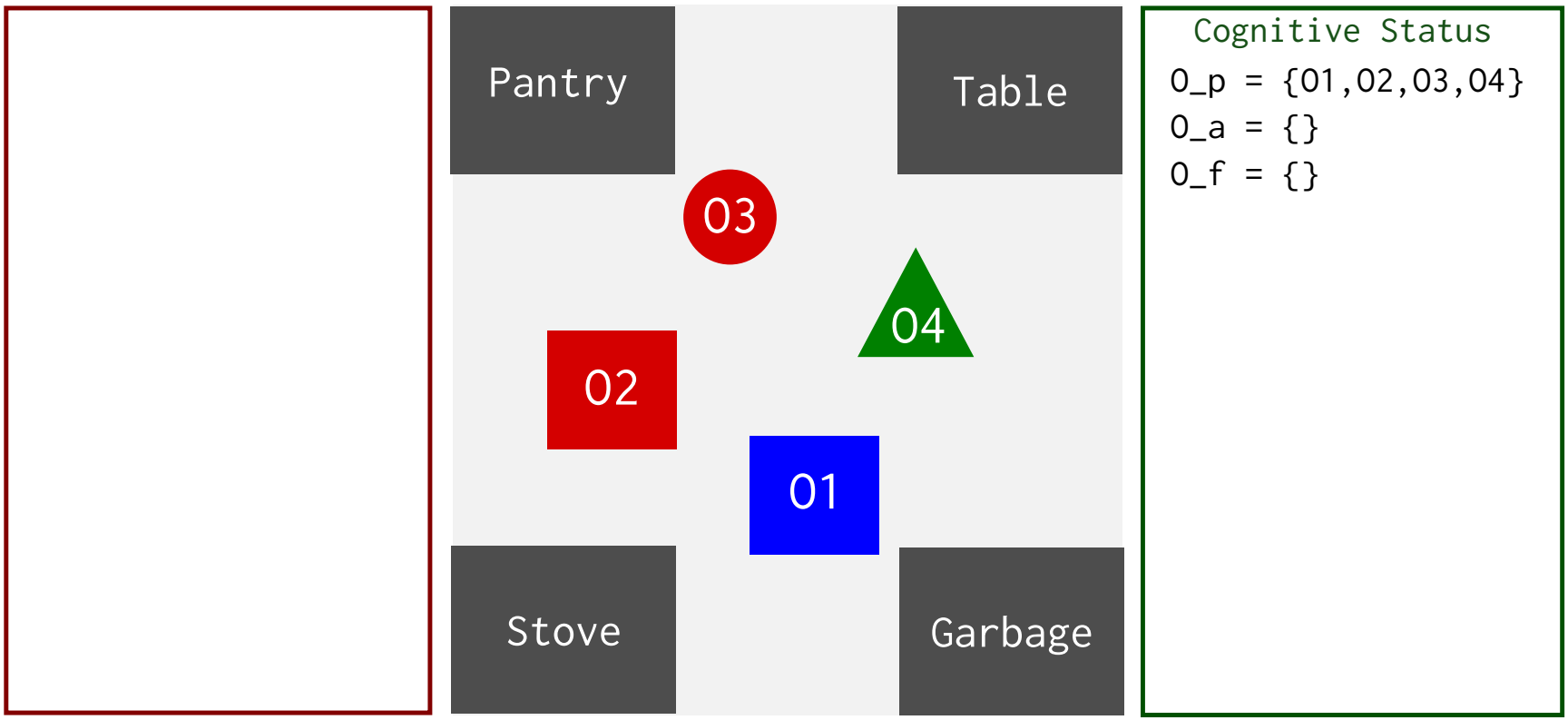
- informational: perceptual, spatial, task knowledge
- temporal: recency, dialog attention

# RE Resolution in the Indexical Model



# RE Resolution in the Indexical Model

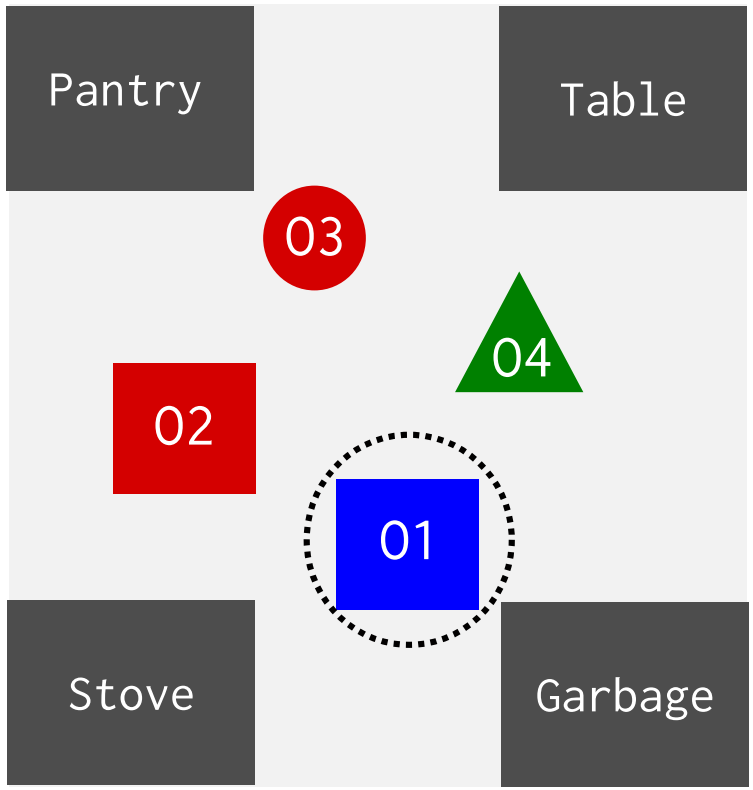
maintain cognitive status



# RE Resolution in the Indexical Model

maintain cognitive status

Instructor: This is a blue rectangle.



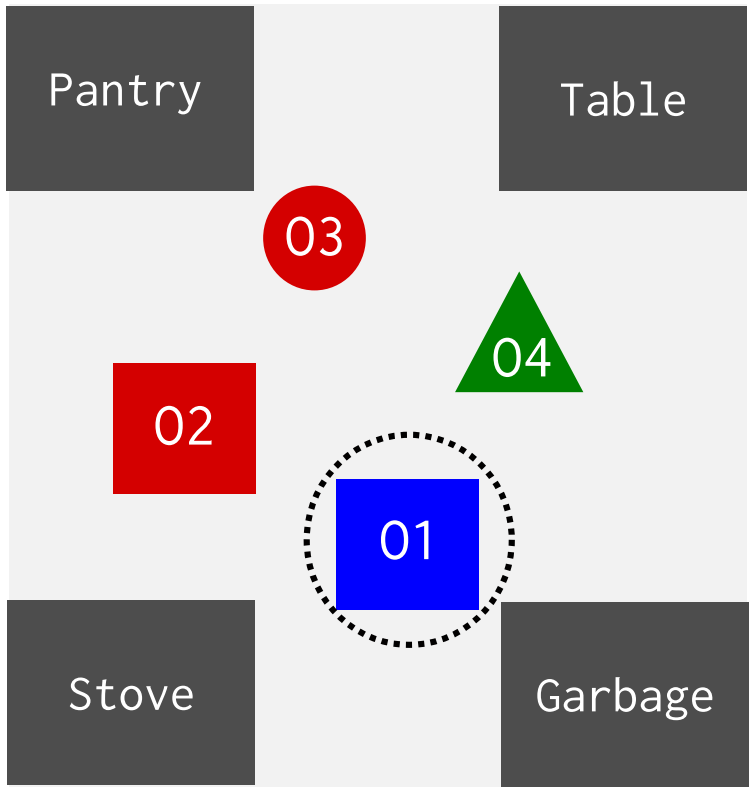

Cognitive Status

O_p = {01,02,03,04}
O_a = {}
O_f = {}

# RE Resolution in the Indexical Model

maintain cognitive status

Instructor: This is a blue rectangle.



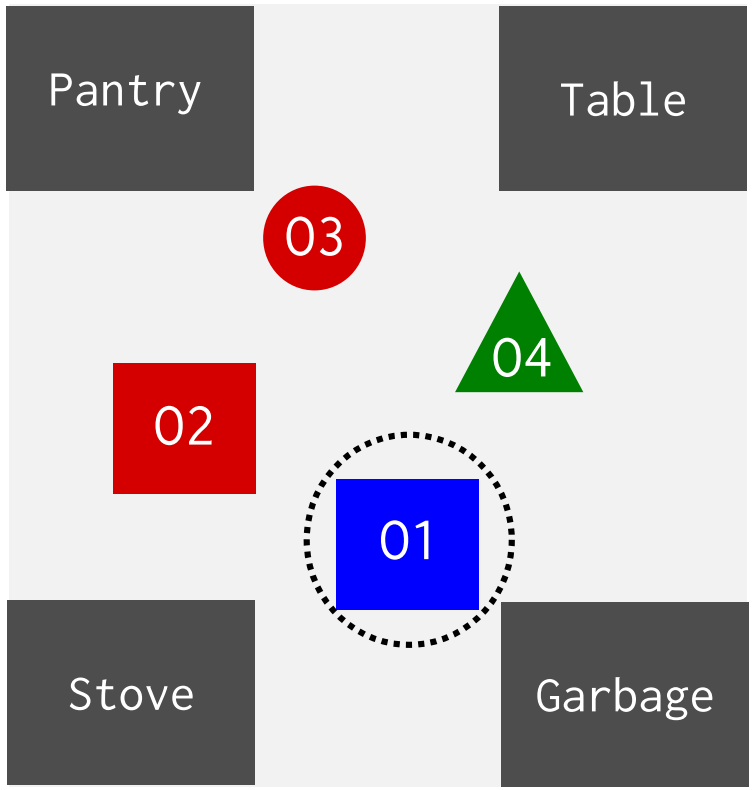

Cognitive Status

O_p = {01,02,03,04}
O_a = {01}
O_f = {}

# RE Resolution in the Indexical Model

use GH heuristics to identify the candidates

Instructor: This is a blue rectangle.




Cognitive Status

O_p = {01,02,03,04}

O_a = {01}

O_f = {}

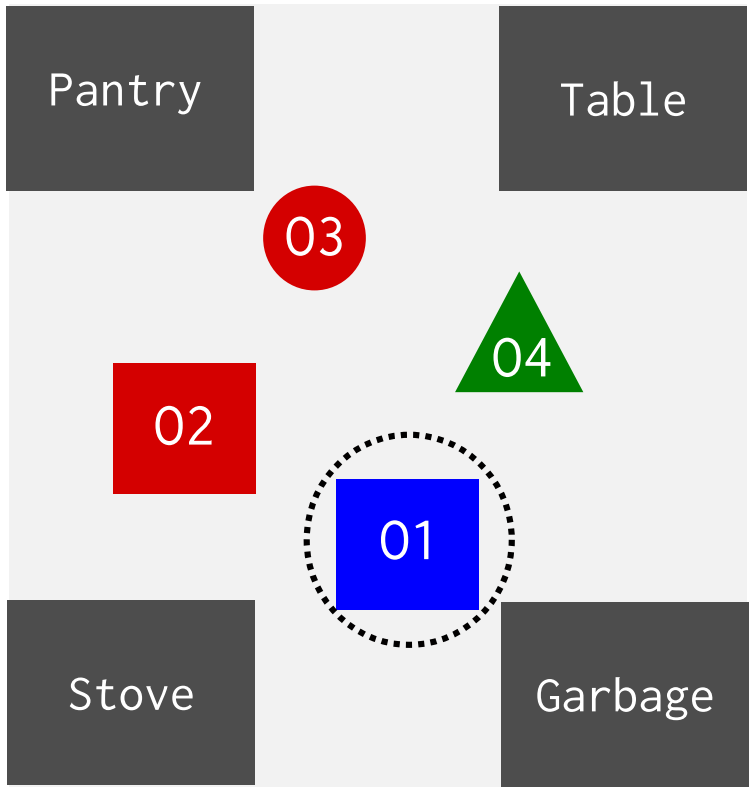
GH Heuristic

this/that N ⟶ active

# RE Resolution in the Indexical Model

use GH heuristics to identify the candidates

Instructor: This is a blue rectangle.




Cognitive Status

O_p = {O1,O2,O3,O4}

O_a = {O1}

O_f = {}

GH Heuristic

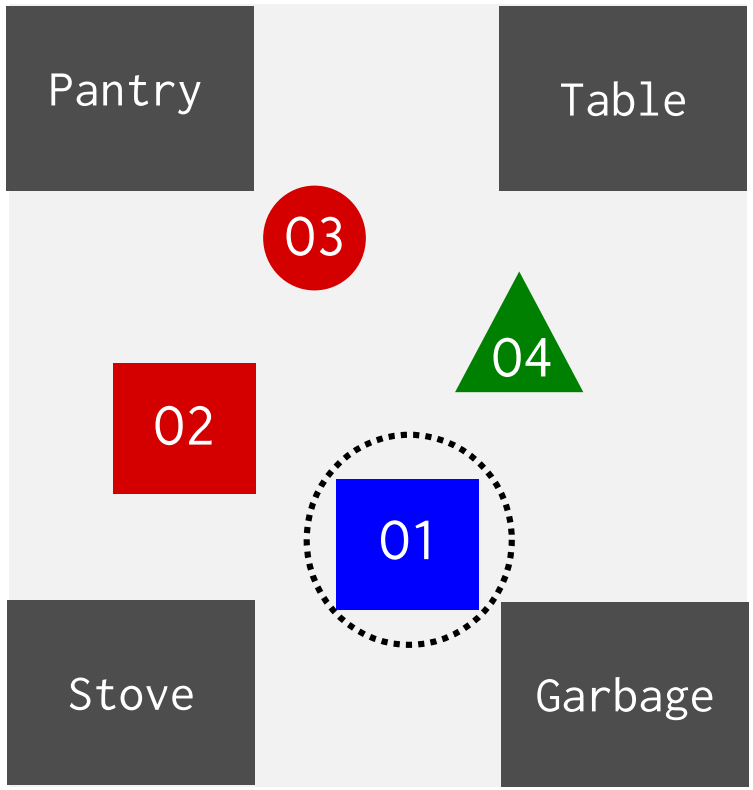
this/that N ⟶ active

O_c = O_a

# RE Resolution in the Indexical Model

resolve

Instructor: This is a blue rectangle.




Cognitive Status

O_p = {O1,O2,O3,O4}

O_a = {O1}

O_f = {}

GH Heuristic

this/that N ⟶ active

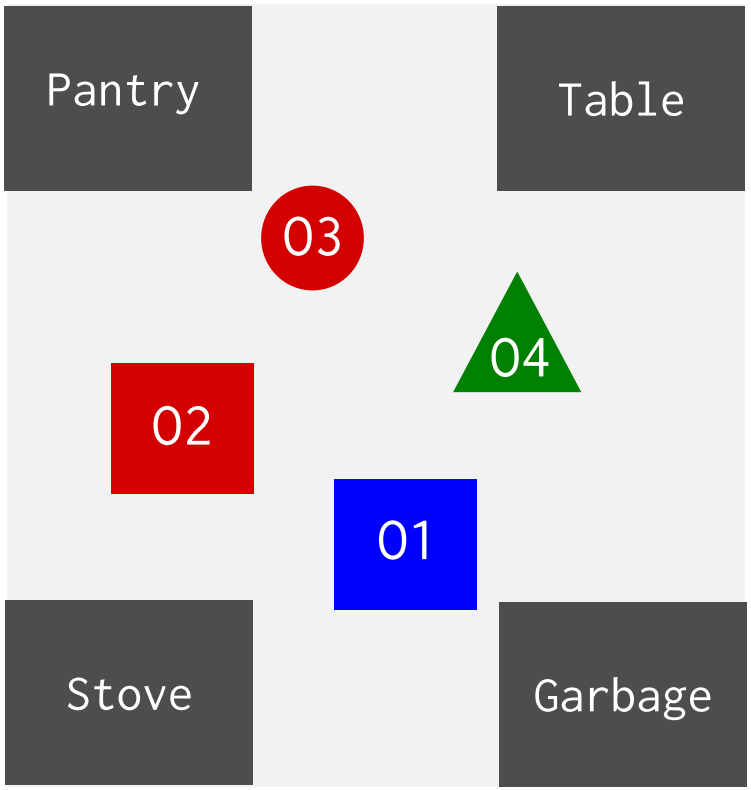
O_c = O_a

Resolution

this = O1

# RE Resolution in the Indexical Model

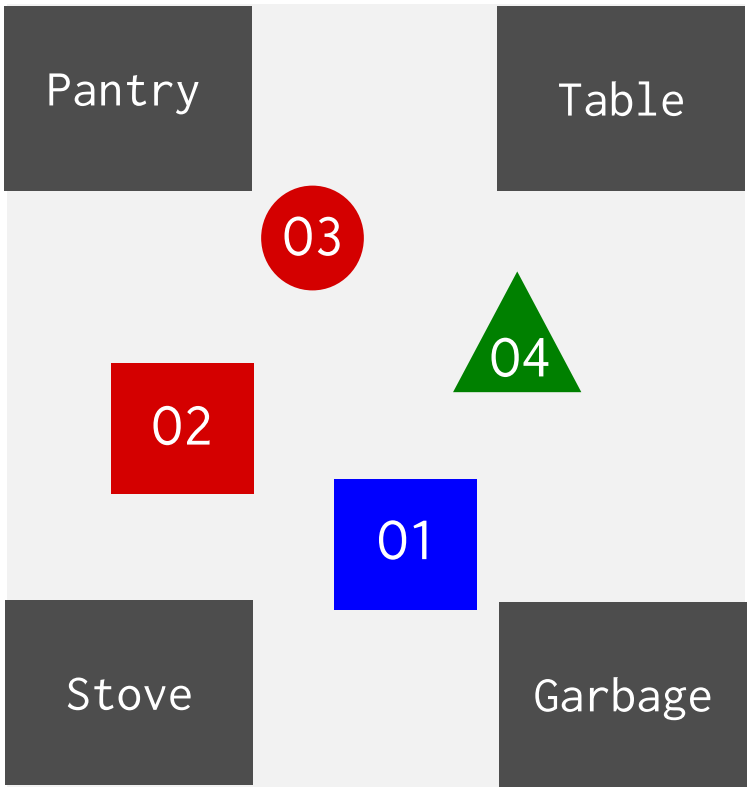
Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.




Cognitive Status
O_p = {01,02,03,04}
O_a = {01}
O_f = {}

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.




Cognitive Status

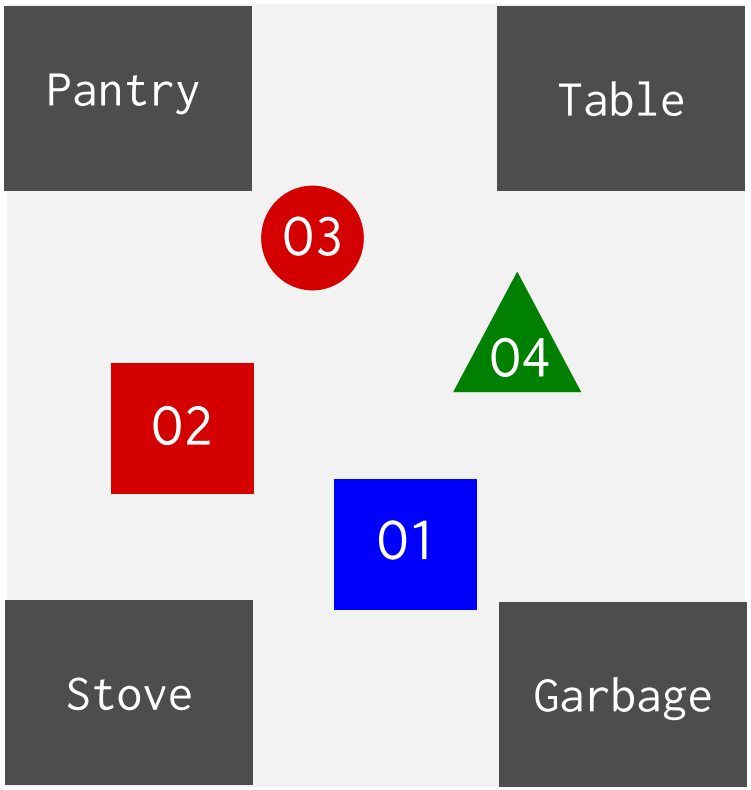
O_p = {01,02,03,04}
O_a = {01}
O_f = {}

GH Heuristic

this/that N ⟶ active

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.




Cognitive Status

O_p = {O1,O2,O3,O4}
O_a = {O1}
O_f = {}

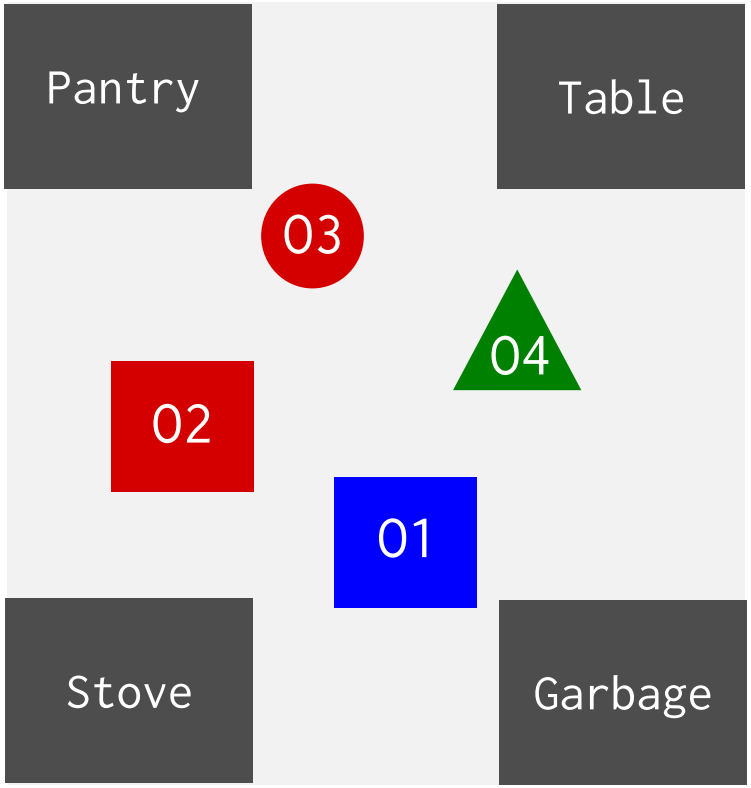
GH Heuristic

this/that N ⟶ active

O_c = O_a

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.




Cognitive Status

O_p = {O1,O2,O3,O4}
O_a = {O1}
O_f = {}

GH Heuristic

this/that N ⟶ active

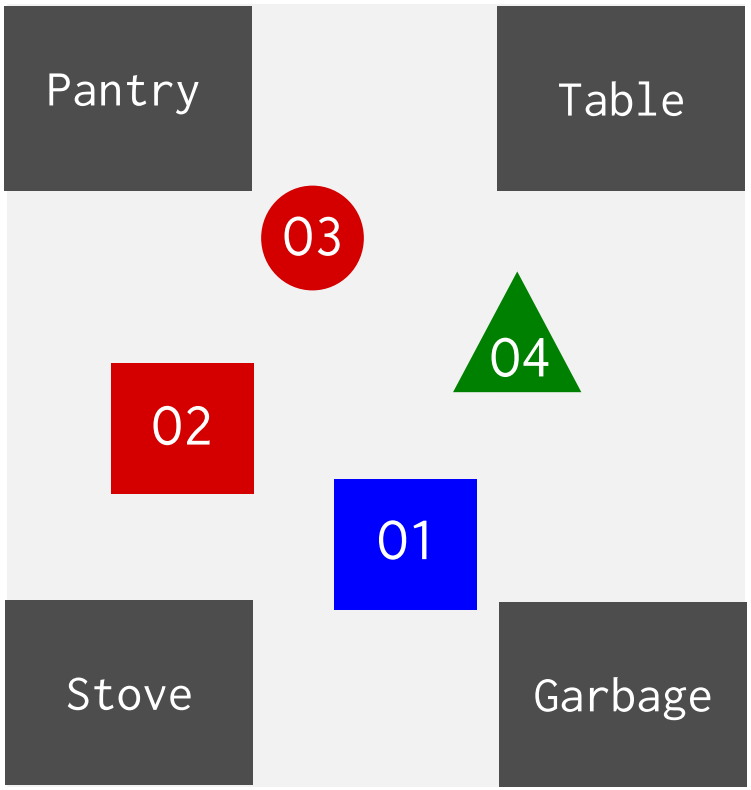
O_c = O_a

Resolution

this = O1

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.




Cognitive Status

O_p = {O1,O2,O3,O4}
O_a = {O1}
O_f = {}

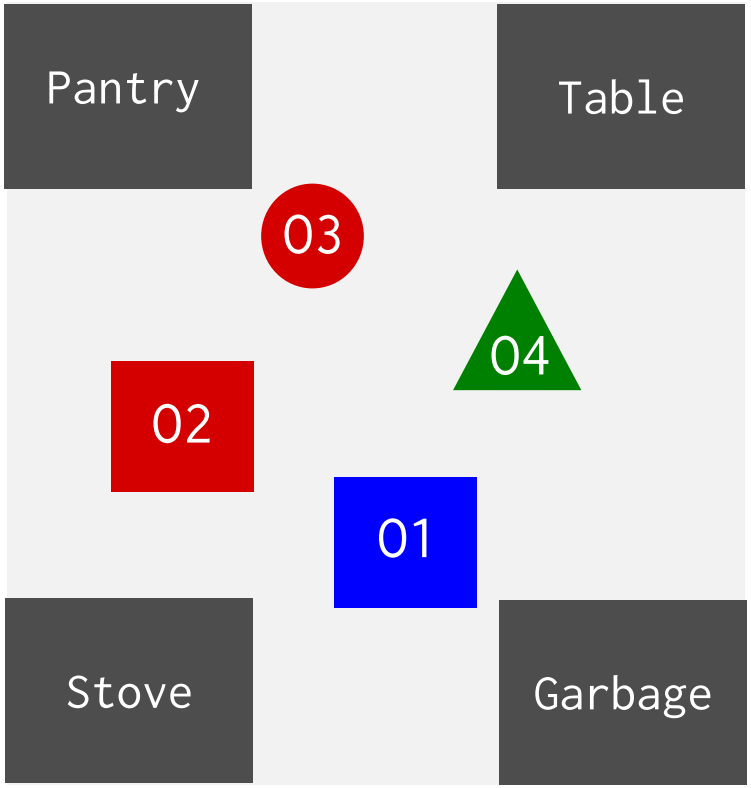
GH Heuristic

the N ⟶ percept

O_c = O_p

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.




Cognitive Status

O_p = {O1,O2,O3,O4}
O_a = {O1}
O_f = {}

GH Heuristic

the N ⟶ percept

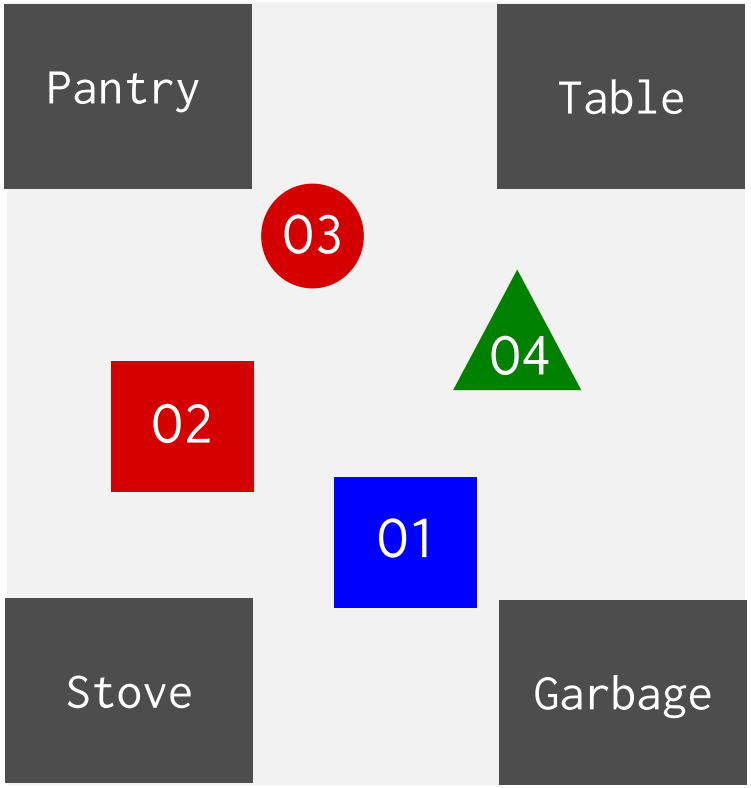
O_c = O_p

Filter(red)

O_c = {O2, O3}

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.




Cognitive Status

O_p = {O1,O2,O3,O4}
O_a = {O1}
O_f = {}

GH Heuristic

the N ⟶ percept

O_c = O_p

Filter(red)

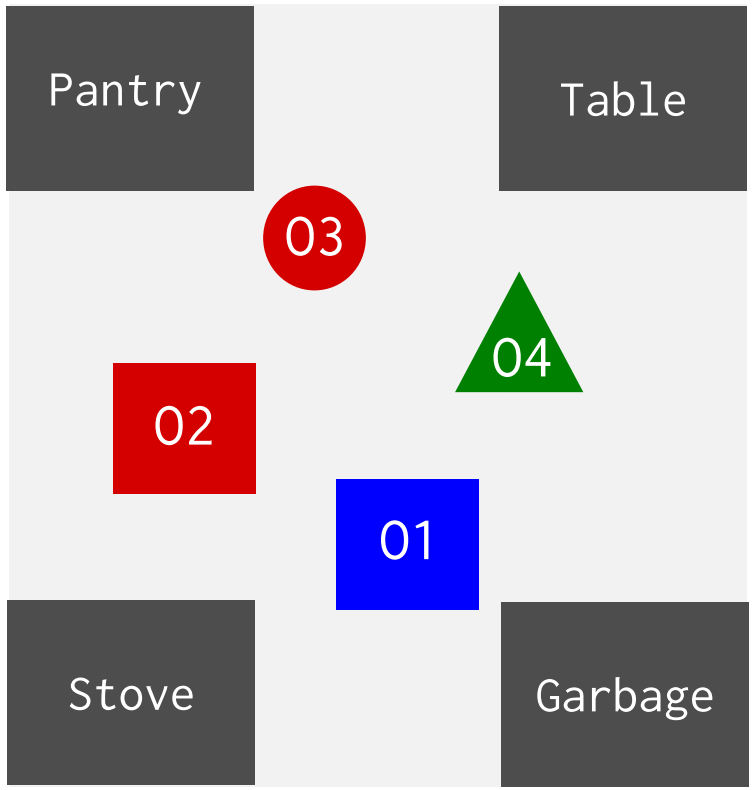
O_c = {O2, O3}

Filter(circle)

O_c = {O3}

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.




Cognitive Status

O_p = {O1,O2,O3,O4}
O_a = {O1}
O_f = {}

GH Heuristic

the N ⟶ percept

O_c = O_p

Filter(red)

O_c = {O2, O3}

Filter(circle)

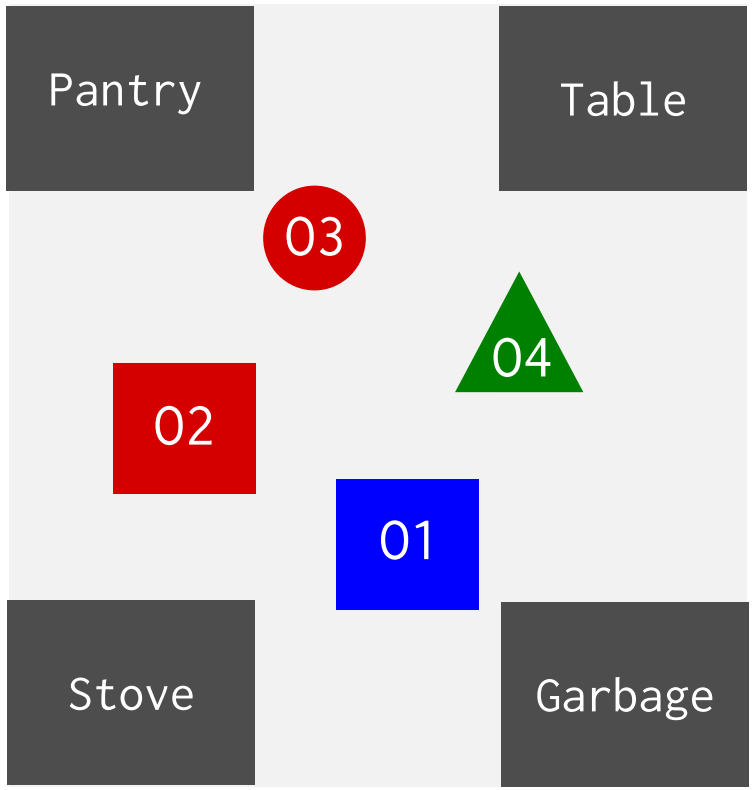
O_c = {O3}

Resolution

'the red circle' = O3

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.




Cognitive Status

O_p = {01,02,03,04}
O_a = {03, 01}
O_f = {}

GH Heuristic

the N ⟶ percept

O_c = O_p

Filter(red)

O_c = {02, 03}

Filter(circle)

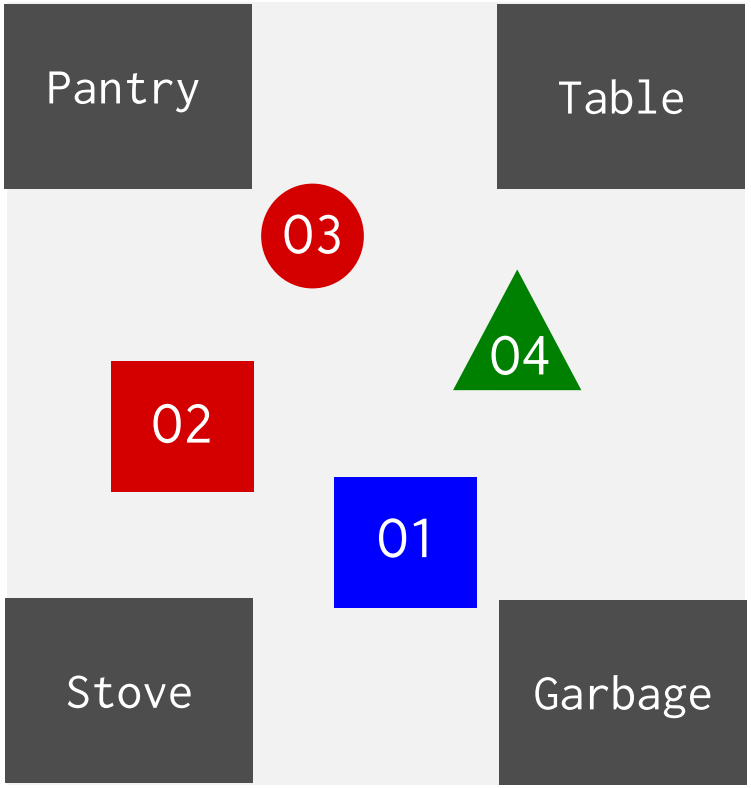
O_c = {03}

Resolution

'the red circle' = 03

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.
Instructor: Move the rectangle to the pantry.




Cognitive Status

O_p = {O1,O2,O3,O4}
O_a = {O3, O1}
O_f = {}

GH Heuristic

the N ⟶ percept

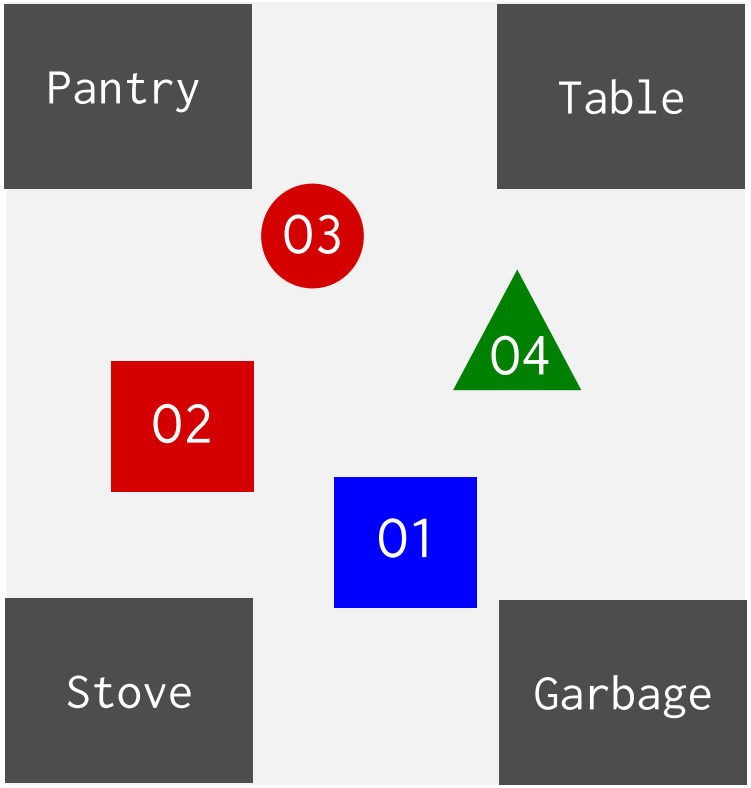
O_c = O_p

Filter(rectangle)

O_c = {O1, O2}

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.
Instructor: Move the rectangle to the pantry.




Cognitive Status

O_p = {01,02,03,04}
O_a = {03, 01}
O_f = {}

GH Heuristic

the N ⟶ percept

O_c = O_p

Filter(rectangle)

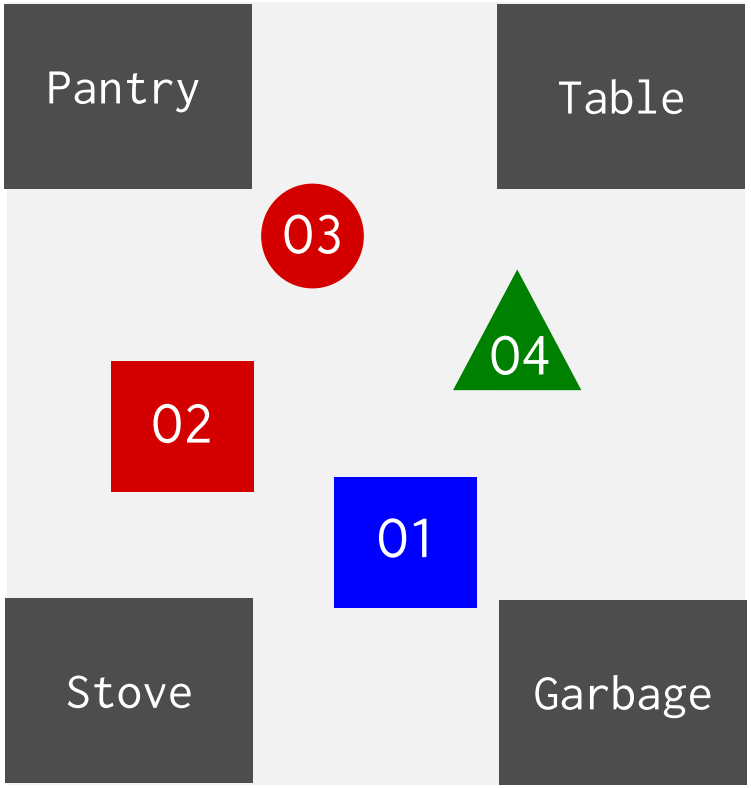
O_c = {01, 02}

Resolution

'the rectangle' = 01

# RE Resolution in the Indexical Model

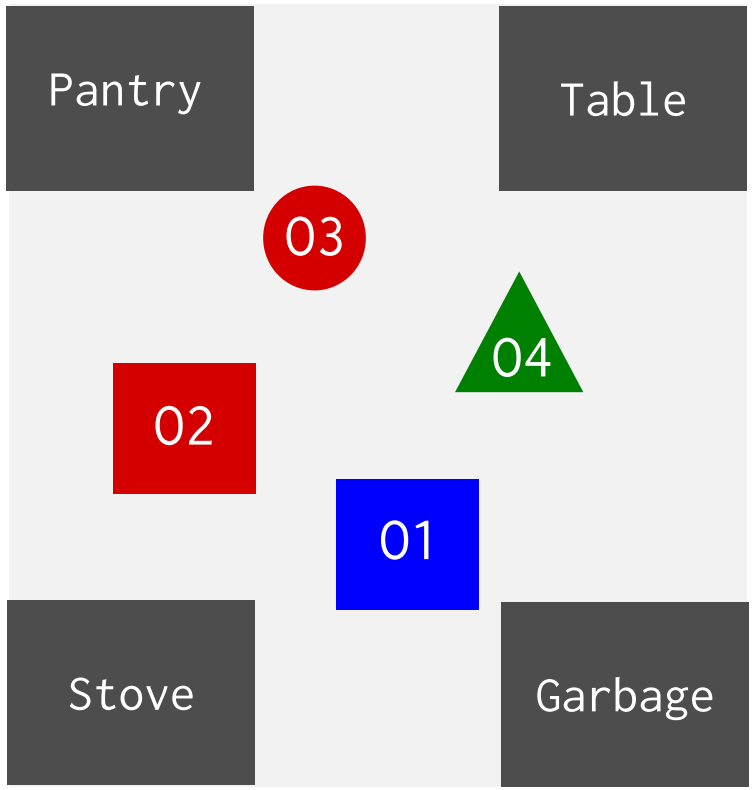
Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.
Instructor: Move the rectangle to the pantry.




Cognitive Status
O_p = {01,02,03,04}
O_a = {03, 01}
O_f = {01, pantry}

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.
Instructor: Move the rectangle to the pantry.
Instructor: Pick it.




Cognitive Status

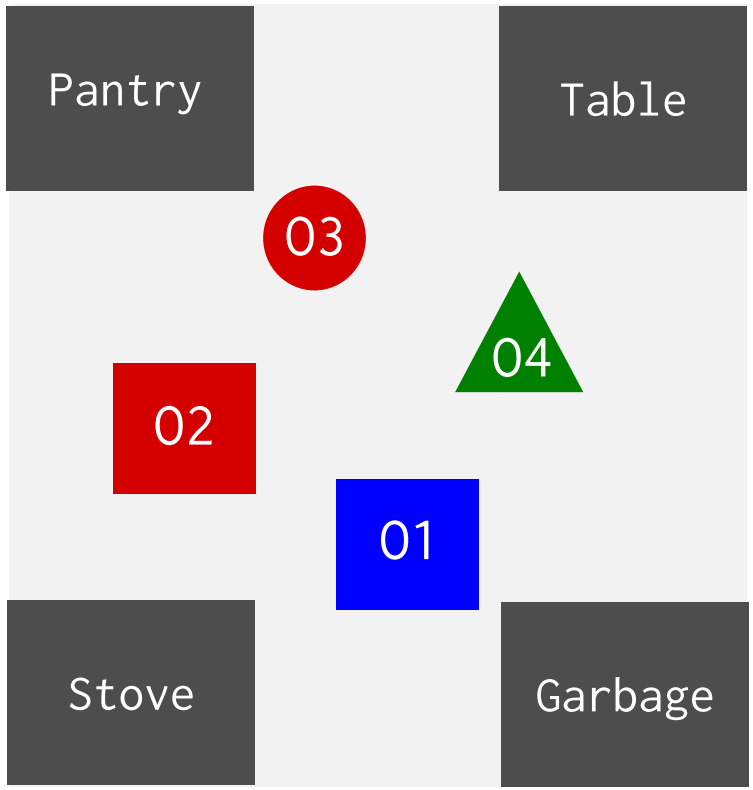
O_p = {01,02,03,04}
O_a = {03, 01}
O_f = {01, pantry}

GH Heuristic

it ⟶ focus
O_c = O_f

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.
Instructor: Move the rectangle to the pantry.
Instructor: Pick it.




Cognitive Status

O_p = {01,02,03,04}
O_a = {03, 01}
O_f = {01, pantry}

GH Heuristic

it ⟶ focus

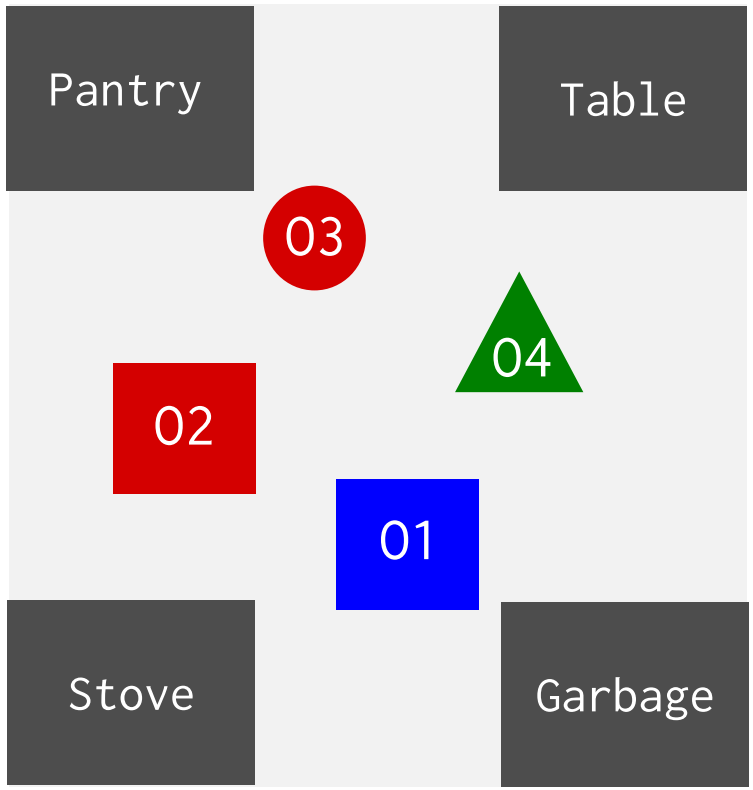
O_c = O_f

Affordance(pick-up)

O_c = {01,pantry}

# RE Resolution in the Indexical Model

Instructor: This is a blue rectangle.
Instructor: This is to the right of the red circle.
Instructor: Move the rectangle to the pantry.
Instructor: Pick it.




Cognitive Status

O_p = {01,02,03,04}
O_a = {03, 01}
O_f = {01, pantry}

GH Heuristic

it ⟶ focus

O_c = O_f

Affordance(pick-up)

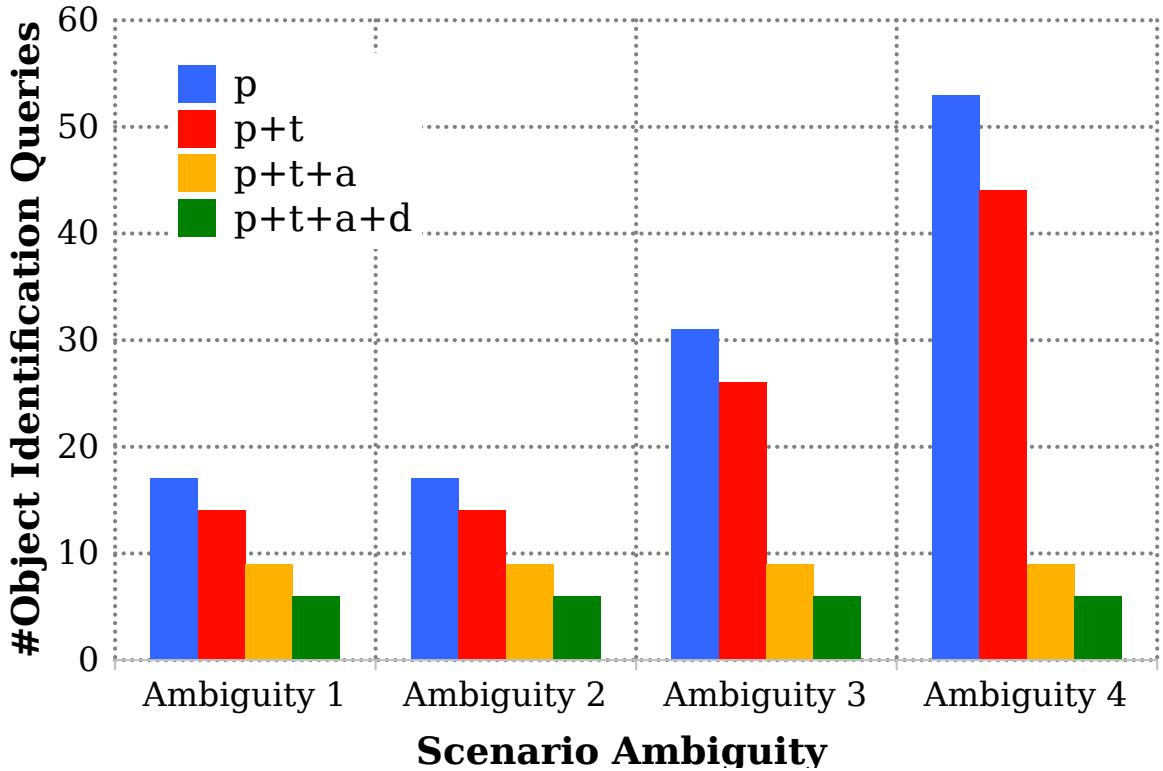
O_c = {01,pantry}

Resolution

'it' = 01

# RE Resolution Performance

**scenarios**: number of distractors
**models**: *p*, *p+t*, *p+t+a*, *p+t+a+d*
**corpus**: instructional dialogs, 12 personal pronouns (*it*), 4 demonstrative pronouns (*this*), 3 demonstrative phrases (*that cylinder*), and 14 noun phrases (*the red cylinder*)




Stanford CoreNLP fails at 28.6% of references.

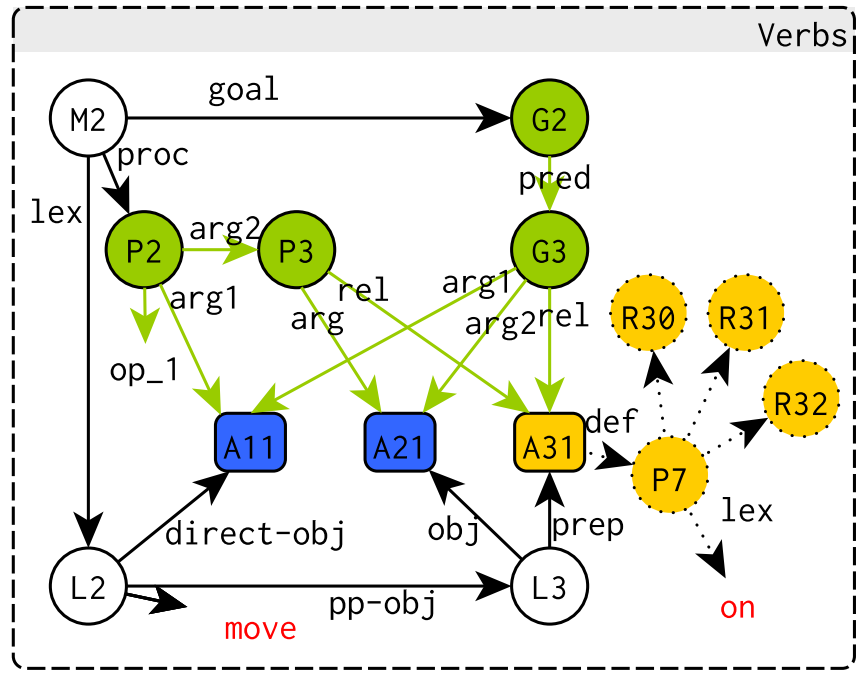
# Unexpressed Argument Alternation

- Goal of verb comprehension - identify and instantiate a task
- Variability in how tasks are described.

  a. *Take the trash out to the curb.*
  b. *Take the trash out.*
- Human speakers and hearers rely on shared experience/knowledge
  - agreement on where the trash is usually deposited (common ground)
  - missing information is filled in from knowledge

# Exploiting the Instructional Experience

| 1 | 2 |
|---|---|
| move the red cylinder to the right of the pantry | move the red cylinder to the pantry |
| the goal is the red cylinder to the right of the pantry | the goal is the red cylinder in the pantry |
| pick up the red cylinder | pick up the red cylinder |
| put the red cylinder to the right of the pantry | put the red cylinder in the pantry |
| you are done | you are done |



# Performance

***pick up***

***put down***

***move***

a. *Move the red cylinder to the right of the table.*

b. *Move the red cylinder to the table.*

***store***

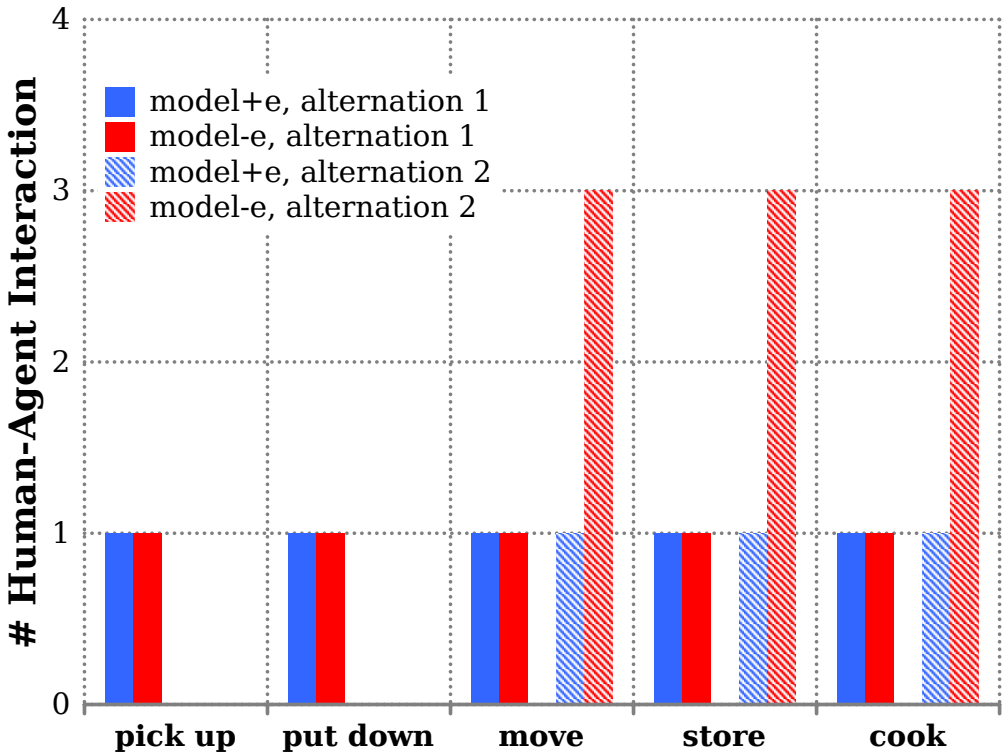
a. *Store the red cylinder in the pantry.*

b. *Store the red cylinder.*

***cook***

a. *Cook the steak on the stove.*

b. *Cook the steak.*



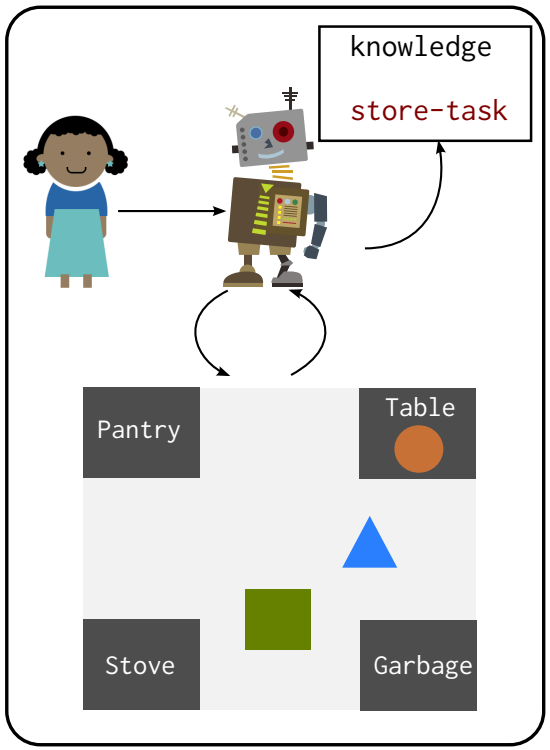
# Summary

- Linguistic features are cues to search perception, knowledge, and experience
- Allows for modality-specific mixed representation
  - classification, reasoning for continuous space, policy
  - integrates with learning
- Natural role of non-linguistic context
  - constrain and guide search
- Addresses complexities due to ambiguities and missing information

# Task Learning

*How can human supervision be integrated in learning?*

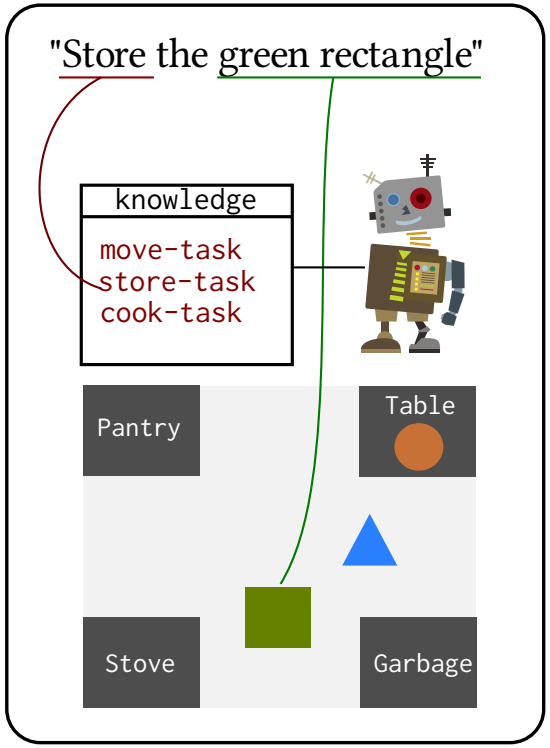
- Complexity
  - disjunctive and negated goal predicates
  - maintenance, performance policies
  - complex hierarchies, complex domains
  - learning value functions
- Autonomy
  - exploring policy space
  - learning goals from demonstrations
- Correction
  - on-fly agent-initiated corrections
  - concept correction, policy correction
- Evaluation
  - in the wild
  - how do humans teach?



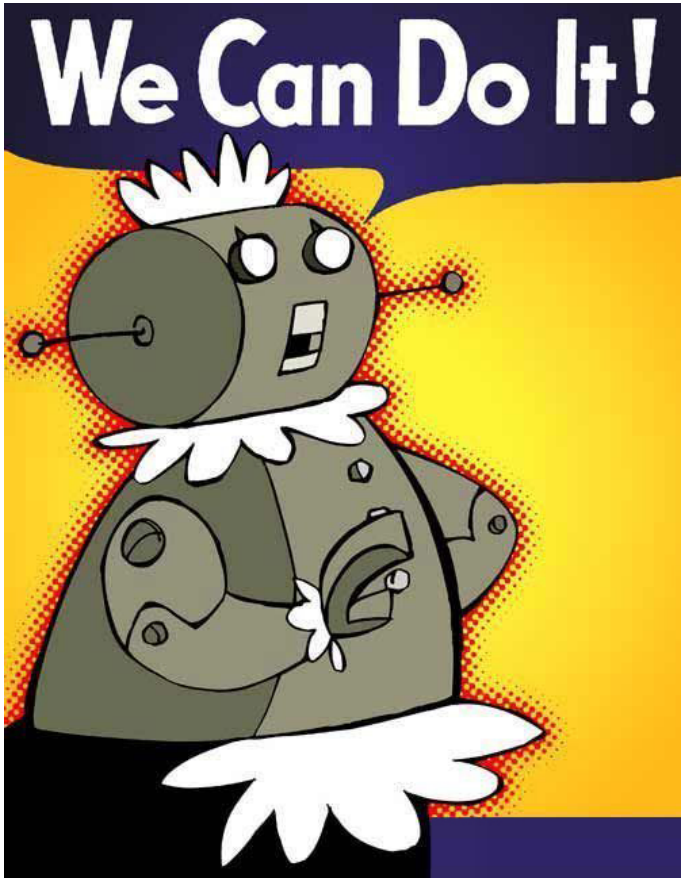
# Situated Communication

*How is non-linguistic knowledge useful for language comprehension?*

- Ambiguity
  - preposition-phrase attachment
  - verb-task polysemy
- Complexity
  - partially-observable environments
  - non-situated comprehension
- Generation
  - task-oriented hearer model
  - most informative response, questions
- Evaluation
  - in the wild
  - information-theoretic models



# Questions




Thanks to:
Soar: John Laird, Aaron Mininger, James Kirk,
APRIL: Edwin Olson, Robert Goeddel, Lauren Hinkle

# Game Learning

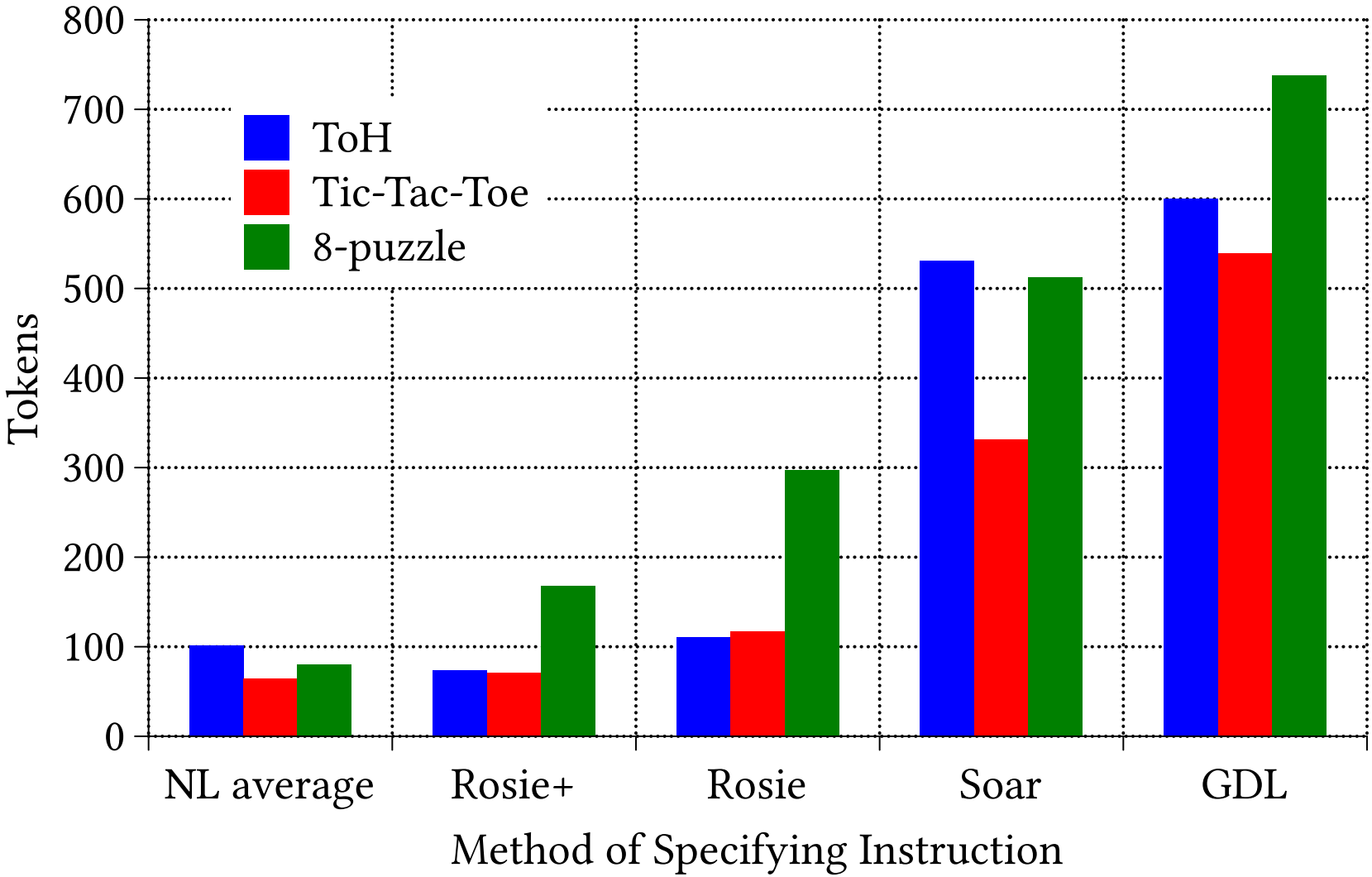
James Kirk, Aaron Mininger, John E. Laird

- Learns the game formulation from interactive instruction
  - Game state: places and types of pieces
  - Legal actions
  - Constraints on actions, Goals, Failure states
- Solves puzzles and games using internal search
  - Simulates task operators internally by imagining executing the underlying primitive actions.
- Games Learned
  Blocks world, Five Puzzle, TicTacToe, Towers of Hanoi, 4 Queens, Connect-3, Frogs and Toads, Hexapawn, Knights Tour, Peg Solitaire, Sokoban, Fox, Goose, and Beans

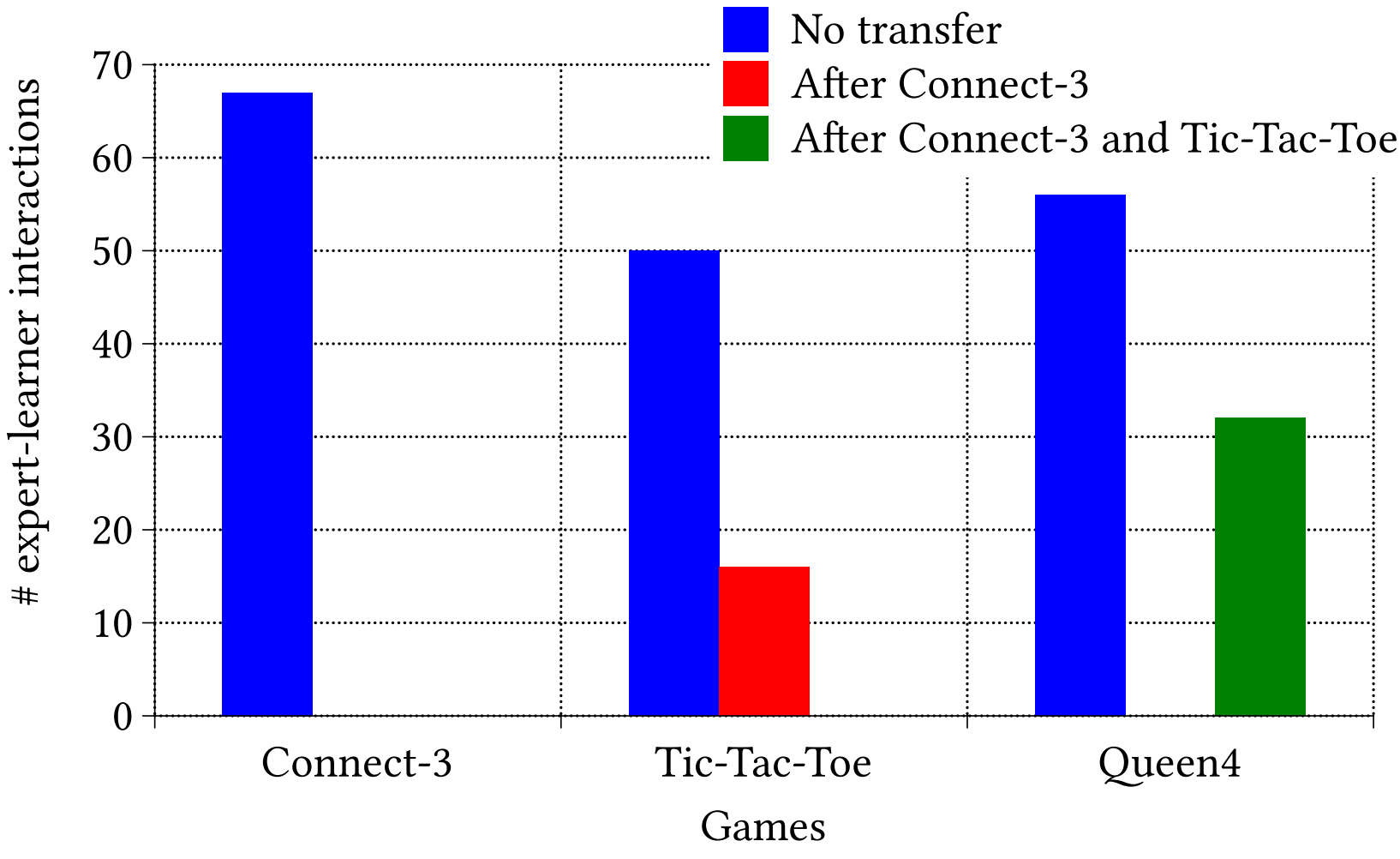
# Sample Interaction

M: The game is tic-tac-toe.
A: I don't know that game, please start by teaching me the name of a legal action in the game.
M: place
A: What is the verb associated with this action?
M: move
A: Ok I know that verb, please tell me the name of a parameter for this action.
M: a block
A: What is a condition for this parameter (or say finished if no more)
M: the block is a cross
A: What is a condition for this parameter (or say finished if no more)
M: the block is not in a location
A: What is a condition for this parameter (or say finished if no more)
M: finished.
A: Please tell me the name of another parameter for this action. (or say finished if no more)
M: a location
A: What is a condition for this parameter (or say finished if no more)
M: the location is not in a block

# Efficiency of Communication



# Transfer


No transfer
After Connect-3
After Connect-3 and Tic-Tac-Toe
# expert-learner interactions
70
60
50
40
30
20
10
0
Connect-3
Tic-Tac-Toe
Queen4
Games