

Graph-Based Task Libraries for Robots: Generalization and Autocompletion

Steven Klee, Guglielmo Gemignani, Daniele Nardi, and
Manuela Veloso



SAPIENZA
UNIVERSITÀ DI ROMA



Computer Science Department,
Carnegie Mellon University

The Setting

We have a framework for teaching tasks to a robot through language interactions.

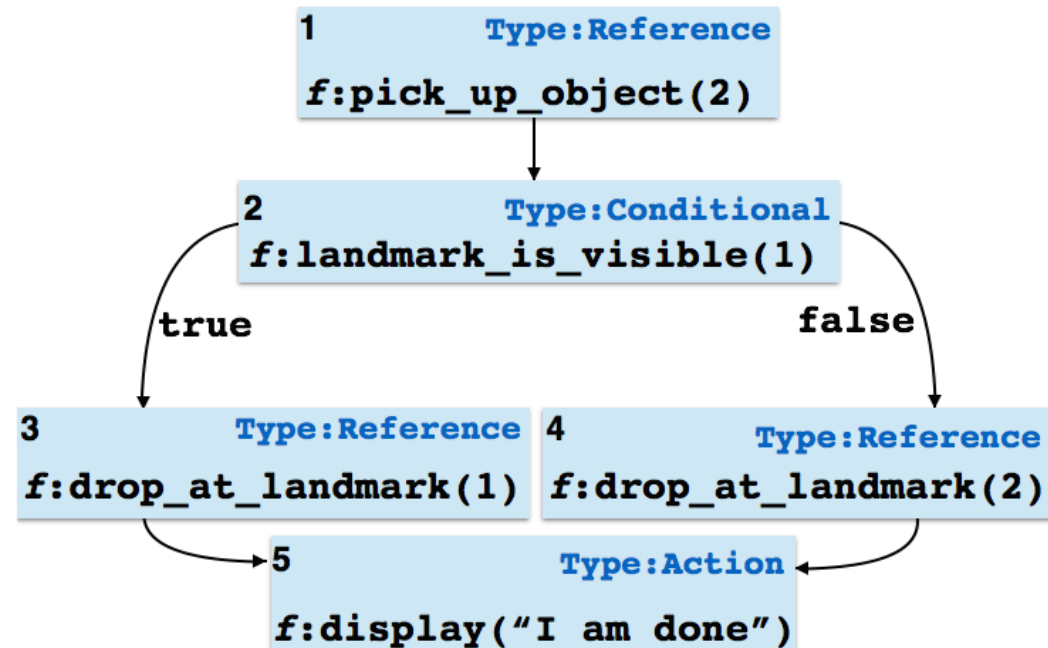
Eg:



- Pick up objects
- Drop Objects
- Display Messages
- Sense Landmarks

The Setting

Example Task: Pick up an object and drop it at one of two locations.



Library of Tasks

Consider a library of such tasks, and a user providing one *additional* task.

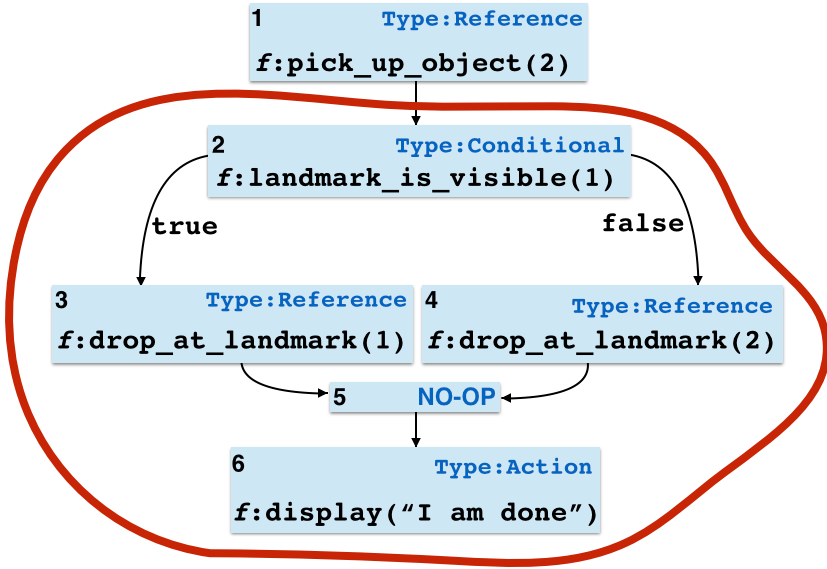
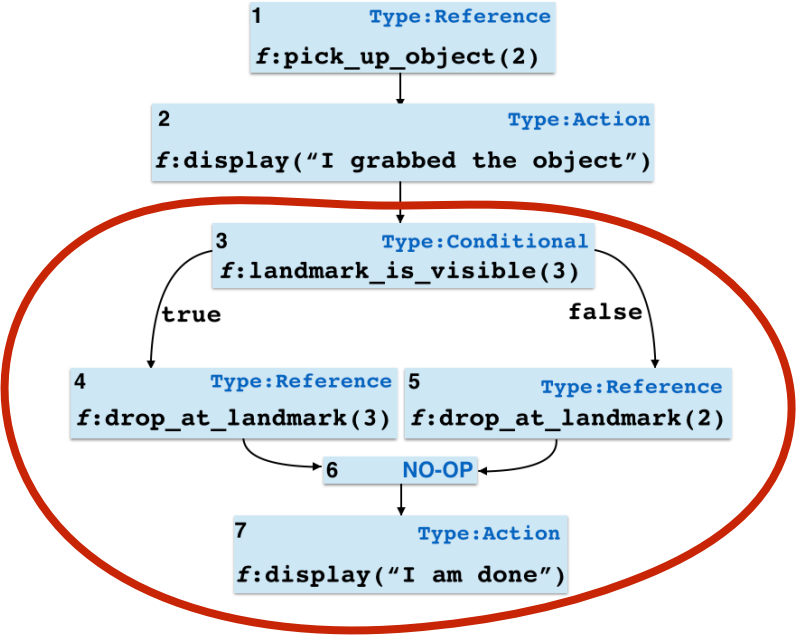
Unlikely: The individual knows all of the tasks the robot knows.

Likely: The robots knows a task similar to one the user is teaching.

Goals

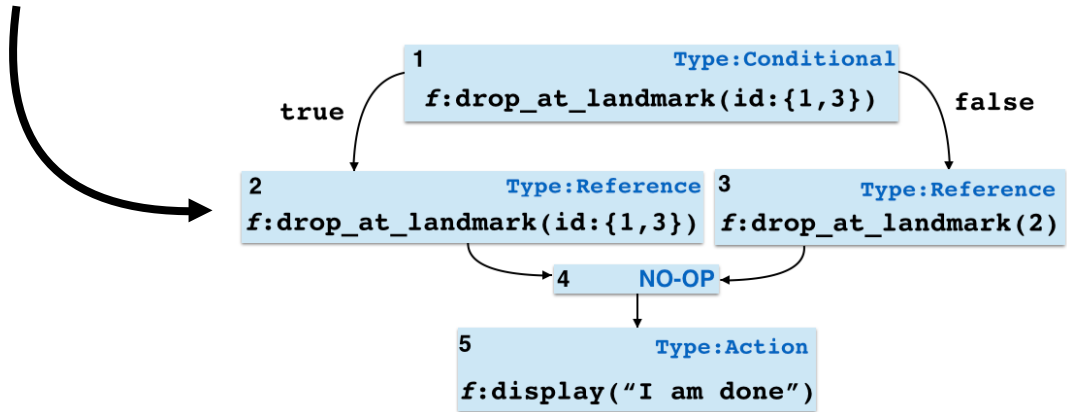
- 1) To **generalize** structurally similar parts of tasks.
- 2) To use these generalizations during future sessions to suggest **autocompletions** to the user.

Example



Find similarities between tasks

Use these to generalize tasks.



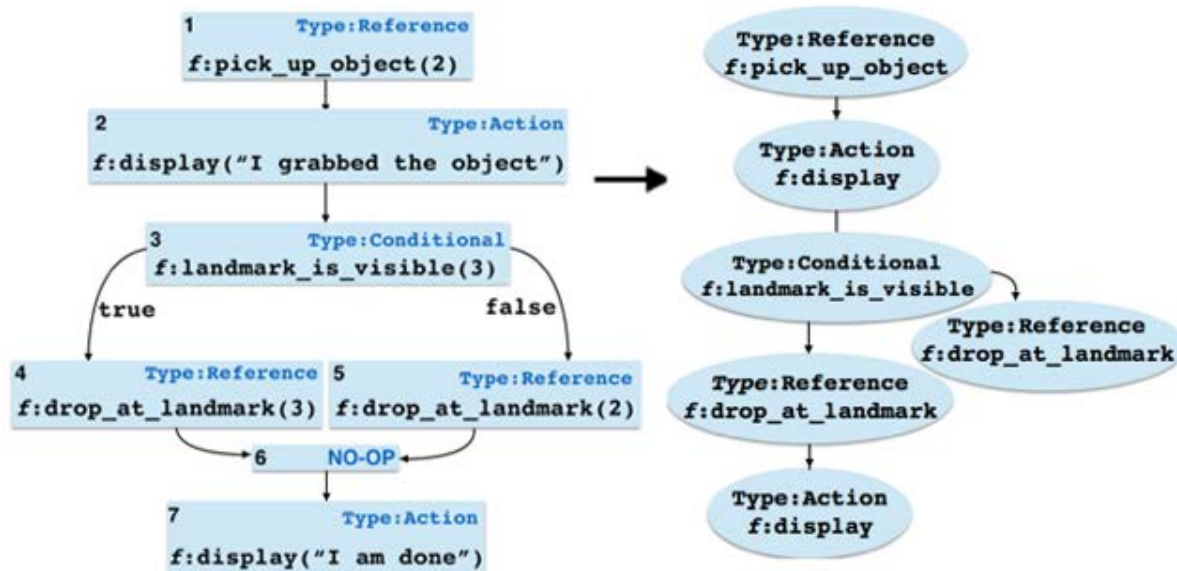
Finding Frequently Occurring Parts of a Task

Enumerating labeled subgraph isomorphisms is **NP-Hard**.

We found a bijection from **executable** Instruction Graphs to trees.

Finding frequently occurring labeled trees is (relatively) easy.

We use a tree mining algorithm.



Task Autocompletion

During teaching, the agent knows a *partial task*.

The robot proposes tasks that are *similar* and have high utility.

To propose a task, the robot provides a demonstration to the user.

Conclusion

- Experiments show that our approach reduces the required number of user interactions.
- Our robots can now:
 - Generalize over structurally similar tasks.
 - Propose autocompletions during future teaching episodes.