

A Software Tool to Help Robots Generalize Skills

E. Topp, elin_anna.topp@cs.lth.se and C. Reichenbach, christoph.reichenbach@cs.lth.se

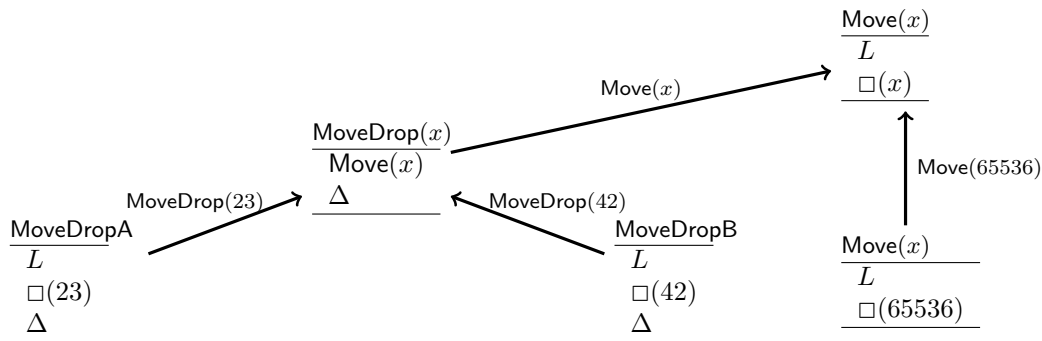
January 19, 2018

Wanted: Two compiler frontend hackers who have taken the Compilers course, or one who has taken Compilers and one who has taken Applied AI and/or Robotics. Both should be interested in software technology and robots. This is an M.Sc. thesis project supervised by Elin Anna Topp and Christoph Reichenbach in the Computer Science department's Robotics and SDE groups.

Programming robots remains challenging. To simplify this process, the Robotics group has developed a domain-specific language for describing and composing robot actions, so-called *skills*. For instance, picking up a block of a fixed size at one fixed position might be such a skill.

However, such skills tend to be highly specific to a particular physical item layout, and adapting it to a new problem (e.g., to blocks of a different shape) takes time and effort. We aim to make this easier and allow robot programmers to copy, paste, and modify existing solutions.

Unfortunately, copy-paste programming has many practical disadvantages, so we want to provide a software tool that can automatically merge and re-use such code. In the (simplified) example below, the developer might provide `MoveDropA`, `MoveDropB`, and `MoveC`, and the tool would automatically learn the skills `Move` and `MoveDrop`:



Moreover, the language currently allows some nonsensical activities, such as dropping an object without picking one up, or dropping an object twice. To avoid confusing future generations of robots, the language should be evolved and the language implementation extended to detect and warn about such problems.

This thesis project entails:

- Programming at least one robot and making it move things around.
- Working with and improving an existing language for controlling the robot.
- Building a JastAdd frontend to parse and analyse the language, and produce warnings.
- Building a tool that can automatically generalise copied-and-pasted code into a more abstract form, and specialise it again afterwards.
- Testing and validating.
- Writing the thesis.

Contact us at elin_anna.topp@cs.lth.se and christoph.reichenbach@cs.lth.se for more information!