Learning from failure:
Improving task execution with experience

Andrey Kurenkov

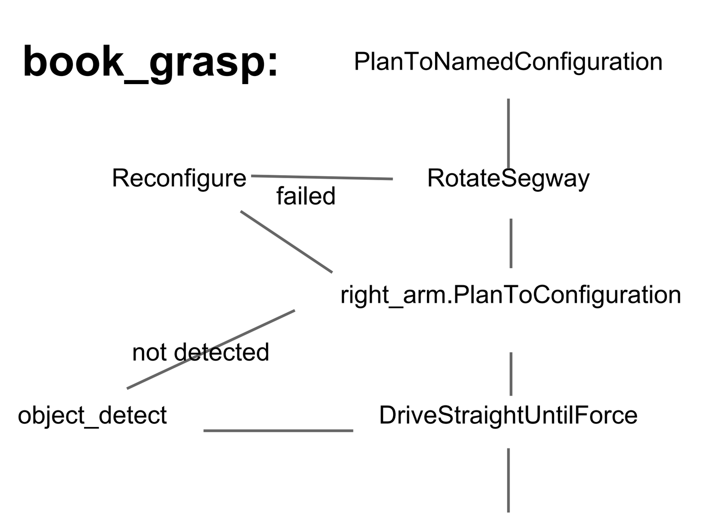
Carnegie Mellon Institute

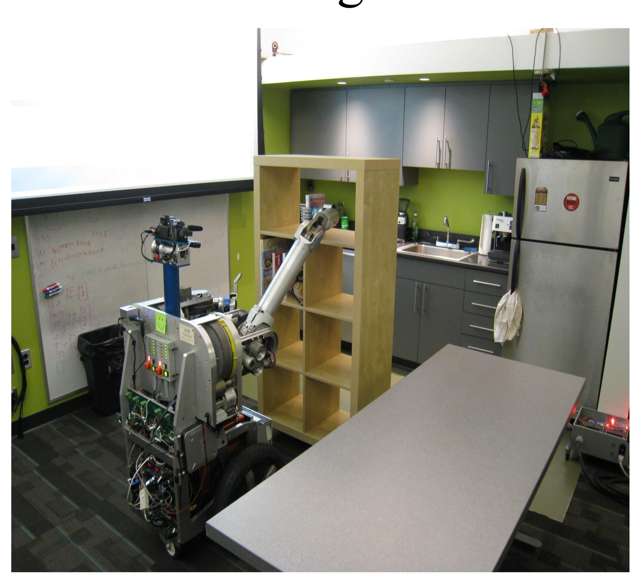
Abstract

A machine learning approach that enables experience-based error detection and recovery during robotic task execution is developed. A task execution framework that uses classical task planning is used and integrated with persistent database storage of world state. Feature extraction is then done on the stored data to create classification trees that differentiate faulty from successful executions and impose additional constraints on future executions to avoid past errors.

Problem

The goal of the Personal Robotics Lab is enabling robots to assist people with a wide range of everyday tasks. This research focused on the following two related problems associated with that goal:





Representation and execution of varied tasks

- Motivation
  - Tasks require a series of steps specified explicitly and formally to the robot
  - Even simple tasks require complex state machines to account for variations in state
- Need
  - A representation that is naturally adaptive and reuses the same actions smartly
- Related
  - STRIPS[1] and more advanced planners[2]

Error handling during task execution

- Motivation
  - Many sources of uncertainty make it hard to account for every possible error
  - Costly to reprogram the task every time an unexpected error occurs
- Need
  - A means of having the robot learn to avoid errors it has already encountered
- Related
  - Prior error detection and recovery[4]

Methods

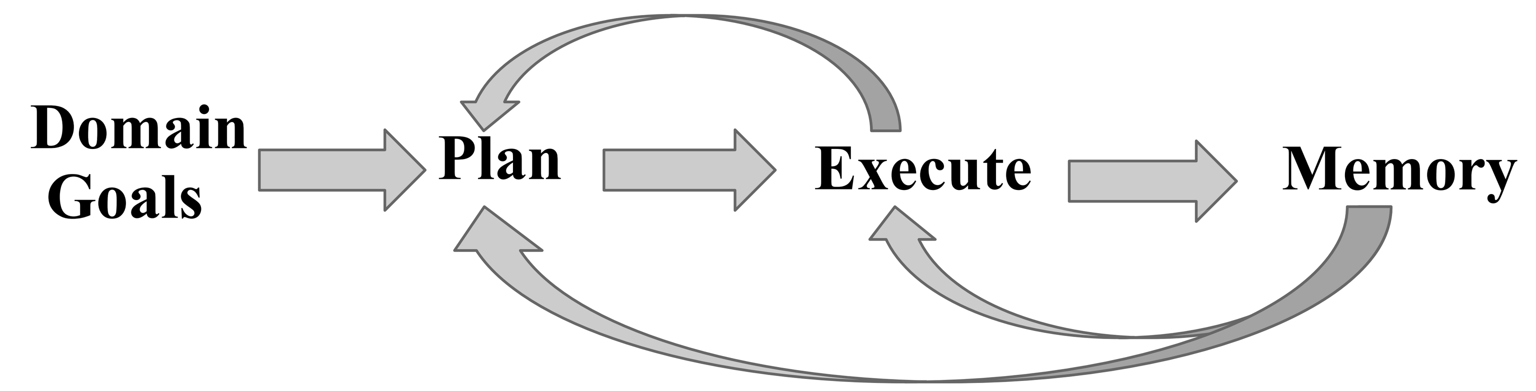


Figure 1. The basic execution loop for execution with error handling

- Interweaving of classical symbolic planning with execution
- Storage of state during both successful and failed executions with [3]
- Modification of subsequent planning and execution

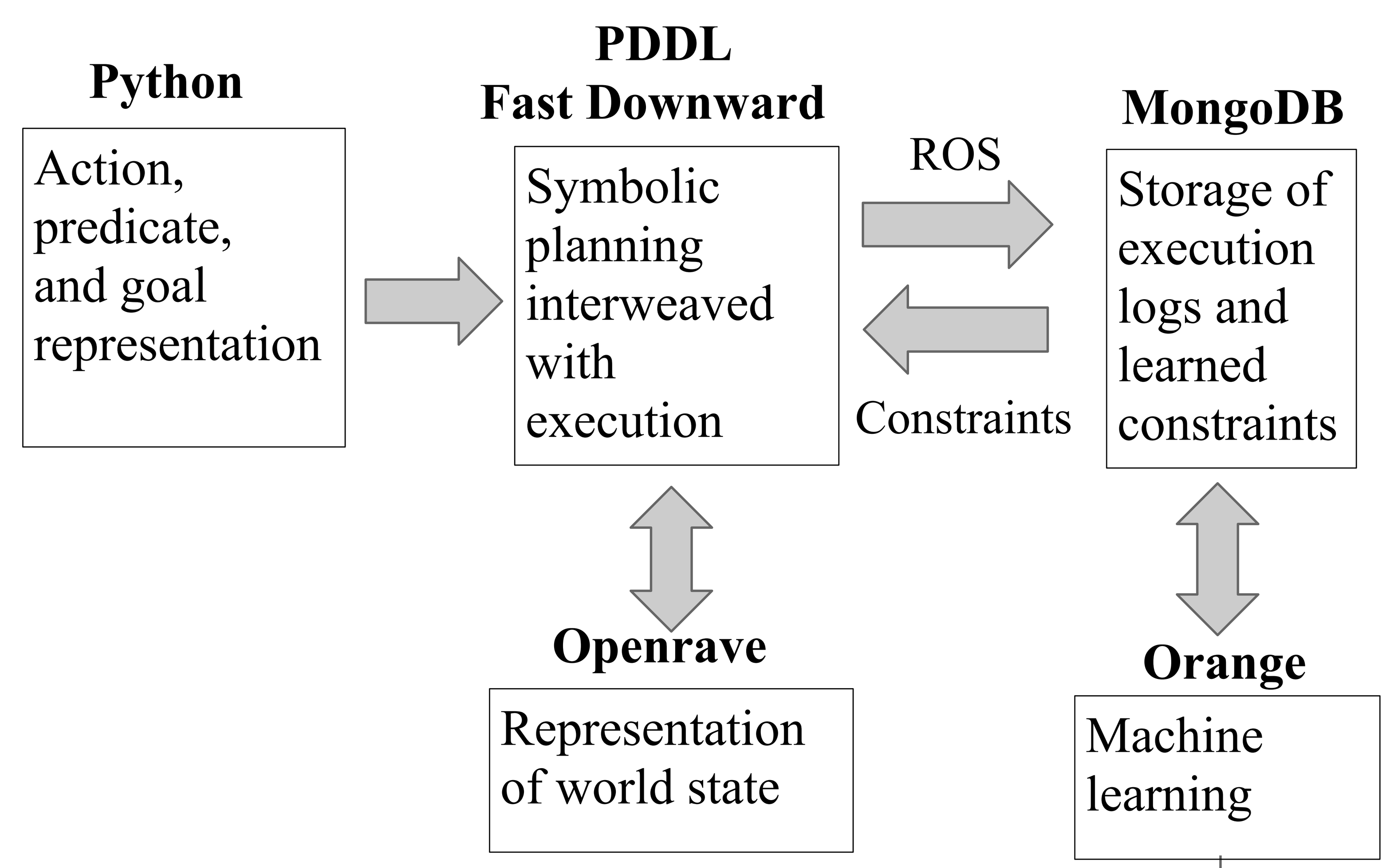


Figure 2. The error detection and recovery framework

1. Decompose actions into subactions
2. Use 1D continuous variables for features
3. Train classification tree at start and end of each subaction
   - simple constraints
   - built-in feature selection
4. Impose constraints during execution

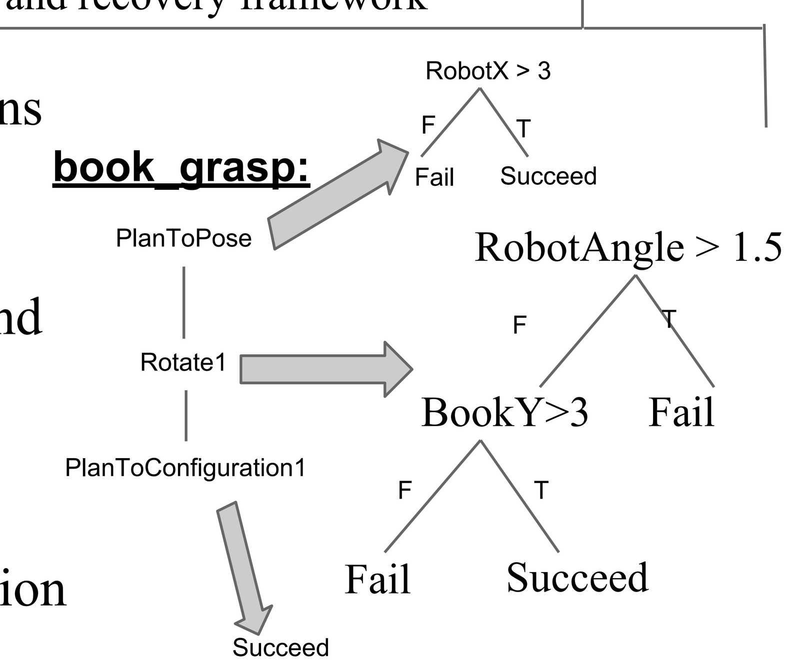


Figure 3. Classification trees for subactions

Results

- Multiple tasks were carried out in simulation with simulated uncertainty of certain parameters to generate a test database
- After being learned, the classification trees are used during execution to check whether the robot is ever in a state that predicts failure.
- In case of a failure statei n such a case provides a goal state by finding a path to success in the pertinent tree.
- Trees with equivalent classification certainty are ranked by testing their effectiveness in simulation; Table 1 contains important trees.

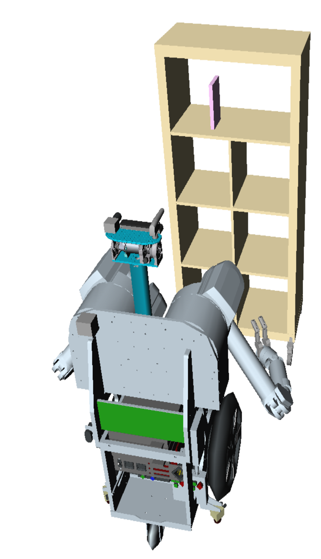


Figure 4. Task start state

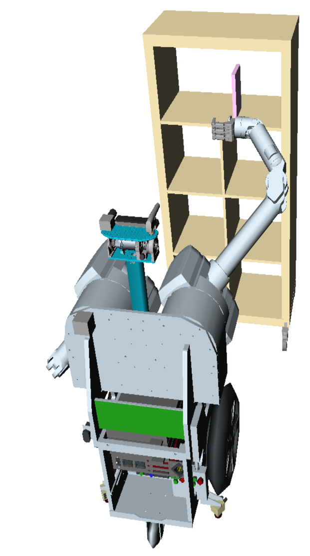


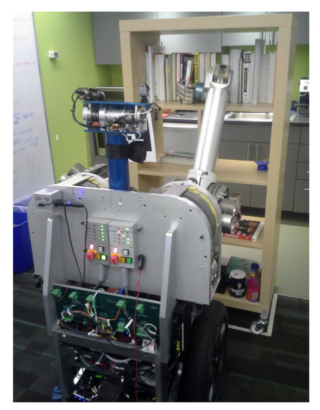
Figure 5. Failure due to book location

Variable varied	**Classification Trees Learned**	**Effect of constraints**
Robot heading	At DriveSegway1_start: if robot | locYaw>0.025: Failure elif robot | locYaw<=0.025: if robot | locYaw<=-0.020: Failure elif robot | locYaw>-0.020: Success	Robot correctly rotated to face the book
Book location on Y axis	At RotateSegway1_start: if robot |relative | dracula | locY<=0.165: Failure elif robot | relative | dracula| locY>0.165: Success	Robot aborted action if the book was violating its position constraint

Table 1. Results of learning for a task with two variations

Future Work

- Implement on-line learning for constraint refinement and selection
- Perform experiments outside simulation
- Test refining real-world results with subsequent batch simulations



References

1. Richard E. Fikes and Nils J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. Articial Intelligence, 2(3-4):189-208, 1971.
2. Seabra Lopes, L.; Camarinha-Matos, L.M. (1995) A Machine Learning Approach to Error Detection and Recovery in Assembly, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'95)*, Pittsburgh, Pennsylvania, August 1995, vol. 3, pp. 197-203.
3. Thomas Keller, Patrick Eyerich, and Bernhard Nebel. Task planning for an autonomous service robot. In Proceedings of the 33rd annual German con-ference on Advances in articial intelligence, KI'10, pages 358{365, Berlin,Heidelberg, 2010. Springer-Verlag.
4. Tim D Niemueller, Gerhard Lakemeyer, and Siddhartha Srinivasa. A generic robot database and its application in fault analysis and performance evaluation. In IEEE, editor, IEEE International Conference on Intelligent Robots and Systems, October 2012.