Massachusetts Institute of Technology

6.038 Artificial Intelligence in Practice

Assignment 4: Thursday, February 27, 1997

1 Objective

The objective of this assignment is to give you experience in using a a state of the art AI planner. A key part of the experience is figuring out how to represent the domain.

2 Files

The UCPOP planner runs in Lucid Lisp (I recommend that you run the following commands in a shell under Emacs). Lucid Lisp only runs on Athena Suns.

```
add lucidcl
lisp
(lcl:cd "/mit/6.038/ucpop")
(load "loader")
(load-ucpop) ;; loads source code
(in-package 'ucpop)
```

Now you're ready to read in your domain definitions and build plans.

You can see examples of many domain definitions in `/mit/6.038/ucpop/domains/`. You will benefit from looking through these files.

The other files mentioned below can be found in `/mit/6.038/assign-4/`.

3 Planning your life

Build a UCPOP domain that encodes some small domain of expertise that appeals to you. Below are a couple of possibilities:

- Travelling in Cambridge: Encode various travel options, e.g. walking, taking the bus, taking the subway, taking a taxi, driving. You should model constraints such as whether there's snow on the ground, (simulated) bus schedules, parking availability (for cars and bikes), cost, etc. Given an initial time and location (and budget), e.g. at home in the morning, build plans to get to a sequence of desired locations with specified time constraints (e.g. class at 11am, lunch sometime, buy a memory card before the 9pm, dinner, and a movie at 8pm).

- Travelling abroad: Encode what would be needed to construct plans for traveling abroad, e.g. a vacation in Bermuda or a conference in France. You should model buying airline tickets, making hotel reservations, choosing ground transportation, decide what clothes to pack, etc. Of course, you needn't try to get the details right, but try to put in enough detail to get a feeling for the modeling issues.
- Machine learning: Given a data set with some number of samples, possibly all numeric or some symbolic, have the plan choose a learning system (e.g. SNNS, C4.5), decide whether to split the data into training and test or whether to do cross-validation. Have the system issue (simulated) commands to make files, run the system, click buttons, etc.

In all of these cases, you should start with simple domains and simple problems and gradually expand them. At some point you will run into combinatorial explosions that cause the planner to take very long to find an answer. You should try to push it to that limit (it won't be hard to do that).

Hand in your domain definition, problem definitions and trace of UCPOP output.

Discuss any (practical or conceptual) difficulties you encountered in the process.