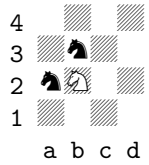
CS 510: Assignment 3

Fall 2012

Consider the following variant of Solitaire Chess[1]:

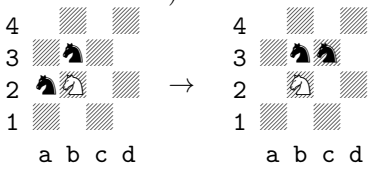
- You are given a 4×4 chess board.
- There are one-to-fifteen black knight pieces placed on the board.
- In addition, there is a single white knight.
- Any knight can capture another knight, regardless of their color.
- The object is, with as few moves possible, to have the white knight be the last night in play, *and* for that piece to end up in the upper-leftmost corner of the board.

Your task is to encode this problem in the STRIPS planning language and solve it using the GraphPlan planning algorithm. For example, given the following initial conditions:

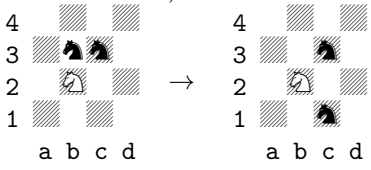


your STRIPS representation should result in a plan like the following:

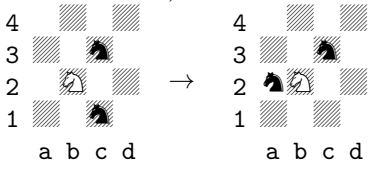
1. (`MOVE k1 a2 c3`)



2. (`MOVE k2 b3 c1`)

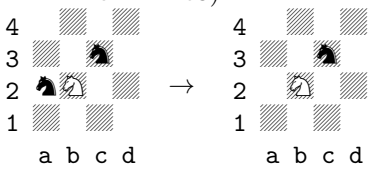


3. (`MOVE k2 c1 a2`)

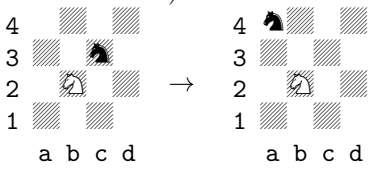


[1] `http://en.wikipedia.org/wiki/Chess_puzzle`

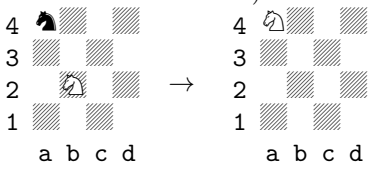
4. (TAKE k2 a2 k1 c3)



5. (MOVE k2 c3 a4)



6. (TAKE white b2 k2 a4)



GraphPlan is a relatively difficult algorithm to implement, though, so you may use a preexisting implementation. I recommend using the original authors' reference implementation[2]. I can attest to the fact that their code compiles and runs (just by executing the Makefile) on both OSX and Linux. It will likely also compile easily on Cygwin using `gcc`.

To run this implementation of graphplan:

```
./graphplan -o knight_ops -f knight_facts -O IL -d -M 1024
```

As you can see, this implementation takes two files: one that contains the operator/action definitions, and another that contains the initial state of the world and goal conditions. Take a look at some of the examples from the GraphPlan website[3] for inspiration. The `fixit` examples are particularly good.

If—for some reason that is hard for me to fathom—you would prefer to compile and run the planner from Visual Studio, the following instructions should work[4]. First, add the `timeofday.c` and `timeofday.h` files that have been included with this assignment. Next, make the following changes:

`graphplan.h`

1. Remove `#include <strings.h>`
2. Add `#include`s for `<string.h>`, `<windows.h>`, and `<malloc.h>`

`graphplan.c`

1. Remove `#include <sys/time.h>`
2. Add `#include "timeofday.h"`
3. Replace all instances of "`random()`" with "`rand()`"

[2] `http://www.cs.cmu.edu/~avrim/graphplan.html`

[3] `http://www.cs.cmu.edu/afs/cs.cmu.edu/usr/avrim/Planning/Graphplan/`

[4] The authors' license to the code prohibits me from giving you the modified version.

`lex.yy.c`

1. Remove `#include <osfcn.h>`
2. Remove all instances of `#include <stdlib.h>`
3. Add `#include <io.h>`
4. Remove this block of three lines:
   ```
   char *malloc();
   int free();
   int read();
   ```

`y.tab.c`

1. Remove `#include <stdlib.h>`
2. Change
   ```
   foo->item = malloc(4*sizeof(char));
   ```
   to
   ```
   foo->item = (char*)malloc(4*sizeof(char));
   ```