

# HW3

October 5, 2014

## 1 Homework 1: Classical Planning

Robot Intelligence - Planning CS 4649/7649, Fall 2014

Instructor: Sungmoon Joo

9/29/14 ## Team 3 \* Siddharth Choudhary \* Varun Murali \* Yosef Razin \* Ruffin White

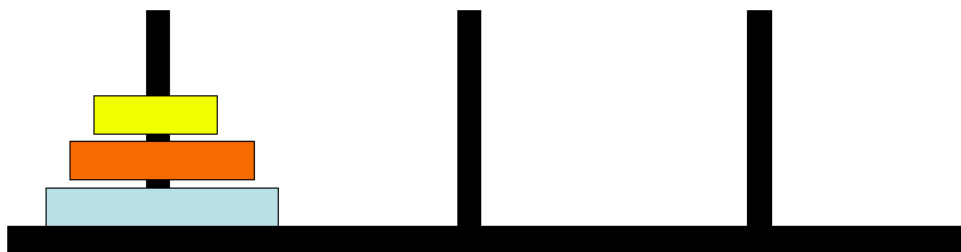
### 1.1 1 ) Towers of Hanoi

A famous problem in classical planning is the Towers of Hanoi. Apparently, some priests in Vietnam are required to stack enormous discs from one tower to another by command of an ancient prophecy. Lets help them out with modern automation. The discs must always be stacked in order of increasing height. The goal is to move all the discs from the first tower to the third. Experiment with at least two different classical planners to solve this problem. Links to the domain are provided on the course web page. The page also contains links to some recommended planners. You are welcome and recommended to try other planners as well.

```
In [1]: from IPython.display import Image
        from IPython.display import FileLink, FileLinks
```

```
In [2]: Image("Figures/Towers of Hanoi.png")
```

Out[2]:



We'll specify the relative path to each of the planners

```
In [3]: planner_path = "../../../Documents/Code/Planners/"
        blackbox = planner_path + "Blackbox43LinuxBinary/blackbox"
        satplan = planner_path + "SatPlan2006_LinuxBin/satplan"
        vhsop = planner_path + "vhsop-2.2.1/vhsop"
        graphplan = planner_path + "Graphplan/graphplan"
        pyperplan = planner_path + "pyperplan/src/pyperplan.py"
        # shop = "../../../Documents/Code/Planners/shop2-2.9.0/"
```

And then the relative path of the domain and problem .pddl files

```
In [4]: tower_path = "/home/tox/git/HW1_Team3/Code/Towers/"
        domain    = tower_path + "hanoi-domain.pddl"
        problem   = tower_path + "hanoi2.pddl"
```

Lets go ahead and try out the blackbox planner

```
In [5]: %%bash -s "$blackbox" "$domain" "$problem"
        $1 -o $2 -f $3 -solver graphplan
```

blackbox version 43

command line: ../../../../Documents/Code/Planners/Blackbox43LinuxBinary/blackbox -o /home/tox/git/HW1\_Team3/Code/Towers/hanoi2.pddl

Begin solver specification

-maxint 0 -maxsec 0.000000 graphplan

End solver specification

Loading domain file: /home/tox/git/HW1\_Team3/Code/Towers/hanoi-domain.pddl

Loading fact file: /home/tox/git/HW1\_Team3/Code/Towers/hanoi2.pddl

Problem name: hanoi-3

Facts loaded.

time: 1, 24 facts and 6 exclusive pairs.

time: 2, 27 facts and 17 exclusive pairs.

time: 3, 31 facts and 35 exclusive pairs.

Goals reachable at 3 steps but mutually exclusive.

time: 4, 33 facts and 42 exclusive pairs.

Goals first reachable in 4 steps.

297 nodes created.

#####

goals at time 5:

on\_d1.d2 on\_d2.d3 on\_d3.p1

-----  
Invoking solver graphplan

Result is Unsat

Iteration was 2

-----  
Can't solve in 4 steps

time: 5, 33 facts and 41 exclusive pairs.

104 new nodes added.

#####

goals at time 6:

on\_d1.d2 on\_d2.d3 on\_d3.p1

-----  
Invoking solver graphplan

Result is Unsat

Iteration was 20

Can't solve in 5 steps  
time: 6, 33 facts and 41 exclusive pairs.  
104 new nodes added.

#####  
goals at time 7:  
on\_d1\_d2 on\_d2\_d3 on\_d3\_p1

-----  
Invoking solver graphplan  
Result is Unsat  
Iteration was 70  
-----

Can't solve in 6 steps  
time: 7, 33 facts and 41 exclusive pairs.  
104 new nodes added.

#####  
goals at time 8:  
on\_d1\_d2 on\_d2\_d3 on\_d3\_p1

-----  
Invoking solver graphplan  
Result is Sat  
Iteration was 239  
Performing plan justification:  
0 actions were pruned in 0.00 seconds  
-----

Begin plan  
1 (move-disk d1 d2 p1)  
2 (move-disk d2 d3 p2)  
3 (move-disk d1 p1 d2)  
4 (move-disk d3 p3 p1)  
5 (move-disk d1 d2 p3)  
6 (move-disk d2 p2 d3)  
7 (move-disk d1 p3 d2)  
End plan  
-----

7 total actions in plan  
25 entries in hash table, 13 hash hits, avg set size 8  
44 total set-creation steps (entries + hits + plan length - 1)  
38 actions tried

#####  
Total elapsed time: 0.00 seconds  
Time in milliseconds: 2  
#####

Ok, so that worked out. Now lets try VHPOP.

```
In [6]: %%bash -s "$vhpop" "$domain" "$problem"
$1 -f LCFR -l 10000 -f MW -l unlimited $2 $3

;hanoi-3
1:(move-disk d1 d2 p1)
2:(move-disk d2 d3 p2)
3:(move-disk d1 p1 d2)
4:(move-disk d3 p3 p1)
5:(move-disk d1 d2 p3)
6:(move-disk d2 p2 d3)
7:(move-disk d1 p3 d2)
Time: 684
```

### 1.1.1 a ) Explain the method by which each of the two planners finds a solution.

The first planner applied, Blackbox, is a planning system that combines satplan and graphplan. Basicly, it parses the PDDL files specified with STRIPS notation into a Boolean, or propositional, satisfiability problem and then applies several diffrent types of satisfiability engines. The name blackbox refers to that the plan generator and the SAT solver know nothing about eachother, thus premitting a flexible sytem to try out diffrent engines to use. The perticular solver used here is just graphplan with default parameters.

The second planner applied, VHPOP is a partial order causal link (POCL) planner loosely based on UCPOP from University of Washington. Written by Håkan L. S. Younes, VHPOP gained reconition durring 3rd International Planning Competition (2002) as Best Newcomer and thus reviving the study partial order planning.

### 1.1.2 b ) Which planner was fastest?

```
In [6]:
```

### 1.1.3 c ) Explain why the winning planner might be more effective on this problem.

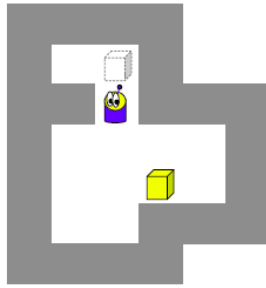
```
In [6]:
```

## 1.2 2 ) Sokoban PDDL

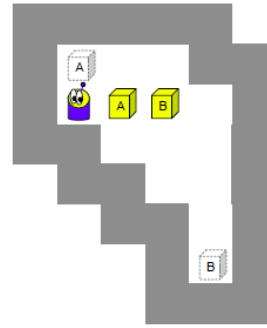
During the times of Pong, Pac-Man and Tetris, Hiroyuki Imabayashi created an complex game that tested the human abilities of planning: Sokoban. Many folks are still addicted to solving Sokoban puzzles and you can join them by playing any of the versions freely distributed on the web. The goal is for the human, or robot, to push all the boxes into the desired locations. The robot can move horizontally and vertically and push one box at a time.

```
In [7]: Image("Figures/Sokoban 1.png")
```

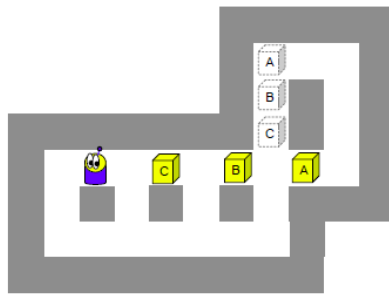
```
Out[7]:
```



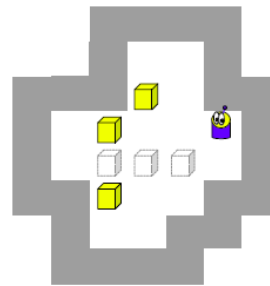
(a) Problem 2.1



(b) Problem 2.2



(c) Problem 2.3



(d) Sokoban Challenge

Describe the Sokoban domain in PDDL. For each of the problems in Figure 2, define the problem in PDDL. You can either use the target lettering given in the picture or let the planner move any box to any target square. For the challenge problem, any box in any location is a solution. In the challenge, PDDL should NOT inform the robot which box should go to which location. In addition you may also try other problems you invent or find on the web. How well do your two planners perform on these problems? If no planner seems to be solving it, perhaps you should consider a different method for defining your problems.

**1.2.1 a ) Show successful plans from at least one planner on the three Sokoban problems in Figure 2(1-3). The challenge problem is optional.**

```
In [8]: %run 'Scratch Book.ipynb'
```

```
In [9]: world_path = "/home/tox/git/HW1_Team3/Code/Worlds/"
```

```
domain    = world_path + "sokoban_domain.pddl"
world     = world_path + "world1.txt"
problem   = world_path + "world1_problem.pddl"
solution  = world_path + "world1_problem.pddl.soln"
graph     = world_path + "world1_problem.png"
```

```
sokoban = Sokoban(world, labeled_boxes = False)
sokoban.writeProblem(problem)
```

```
In [10]: %%bash -s "cat" "$domain"
$1 $2
```

```

; The domain for our sokoban problem is simply named sokoban
(define (domain sokoban)

; Our sokoban domain declares the most basic subset of PDDL, consisting of STRIPS as the only requirements
(:requirements :strips)

; For our predicates, we'll specify the name and number of arguments used for each
(:predicates
  (navigable ?location)           ; Is the location navigable
  (direction ?direct)             ; Is the object a direction
  (block-at ?box ?location)       ; Is the box at the given location
  (robot-at ?location)            ; Is the robot at the given location
  (block ?box)                    ; Is the given box a block or crate
  (adjacent ?location1 ?location2 ?direct) ; Are the two locations adjacent
  (empty ?location)               ; Is the location empty or available
)

; We'll need to be able to move our robot through locations
(:action move

  ; To start, we need a from and to location need to be specified
  :parameters
    (?from ?to ?dir)

  ; Before we effect the current state, we need to make sure
  ; that both the first two given parameters are navigable,
  ; the we have infact a direction
  ; and that given the the direction
  ; that our to location is both reachable (adjacent) and currently empty
  :precondition
    (and
      (navigable ?from)
      (navigable ?to)
      (direction ?dir)
      (robot-at ?from)
      (adjacent ?from ?to ?dir)
      (empty ?to)
    )

  ; The effect on the current state of the world is that
  ; the location we previosly ocupied will now be vacant with no robot,
  ; the robot now at the "to" location so "to" now ocupied
  :effect
    (and
      (empty ?from)
      (robot-at ?to)
      (not (empty ?to))
      (not (robot-at ?from))
    )
)

; To menipulate boxes, we'll need to have an action to push them
(:action push

```

```

; To start, we need locations of the robot, block, destination
; plus the corresponding location and block object to be specified
:parameters
    (?robotLocation ?blockLocation ?freeLocation ?dirr ?activeBlock)

; Before we effect the current state, we need to make sure
; that all the first three given parameters are navigable,
; and that the block can be pushed into an empty location
; that is adjacent in the proper location
:precondition
    (and
        (navigable ?robotLocation)
        (navigable ?blockLocation)
        (navigable ?freeLocation)
        (direction ?dirr)
        (block ?activeBlock)
        (robot-at ?robotLocation)
        (block-at ?activeBlock ?blockLocation)
        (adjacent ?robotLocation ?blockLocation ?dirr)
        (adjacent ?blockLocation ?freeLocation ?dirr)
        (empty ?freeLocation)
    )

; The effect on the current state of the world is that
; the location we previously occupied will now be vacant with no robot,
; the location the box previously occupied will now be occupied by the robot,
; and the free location will now be occupied by the box
:effect
    (and
        (robot-at ?blockLocation)
        (block-at ?activeBlock ?freeLocation)
        (empty ?robotLocation)
        (not (robot-at ?robotLocation))
        (not (block-at ?activeBlock ?blockLocation))
        (not (empty ?freeLocation))
    )
)
)

```

```
In [11]: %%bash -s "cat" "$world"
```

```
$1 $2
```

```

XXXXXX
X_gXXX
XXrXXX
X___X
X__oX
X__XX
XXXXXX

```

```
In [12]: %%bash -s "cat" "$problem"
```

```
$1 $2
```

```

; XXXXXX
; X_gXXX

```

```

; XXrXXX
; X____X
; X__o_X
; X__XXX
; XXXXXX

; This is an auto generated sokoban problem
(define (problem sokoban_problem )

; The domain for our sokoban problem is simply named sokoban
(:domain sokoban)

; Specify list of objects
(:objects

; Specify list of directions
up down left right

; Specify list of blocks
o

; Specify list of locations
r0-c0 r0-c1 r0-c2 r0-c3 r0-c4 r0-c5
r1-c0 r1-c1 r1-c2 r1-c3 r1-c4 r1-c5
r2-c0 r2-c1 r2-c2 r2-c3 r2-c4 r2-c5
r3-c0 r3-c1 r3-c2 r3-c3 r3-c4 r3-c5
r4-c0 r4-c1 r4-c2 r4-c3 r4-c4 r4-c5
r5-c0 r5-c1 r5-c2 r5-c3 r5-c4 r5-c5
r6-c0 r6-c1 r6-c2 r6-c3 r6-c4 r6-c5

)

; Init direction objects
(:init

; Init direction objects
(direction up)
(direction down)
(direction left)
(direction right)

; Init block objects and locations
(block o)
(block-at o r4-c3)

; Init robot location
(robot-at r2-c2)

; Init navigable objects
(navigable r1-c1)
(navigable r1-c2)
(navigable r2-c2)

```



```

(navigable r3-c1)
(navigable r3-c2)
(navigable r3-c3)
(navigable r3-c4)
(navigable r4-c1)
(navigable r4-c2)
(navigable r4-c3)
(navigable r4-c4)
(navigable r5-c1)
(navigable r5-c2)

```

```

; Init empty objects

```

```

(empty r1-c1)
(empty r1-c2)
(empty r3-c1)
(empty r3-c2)
(empty r3-c3)
(empty r3-c4)
(empty r4-c1)
(empty r4-c2)
(empty r4-c4)
(empty r5-c1)
(empty r5-c2)

```

```

; Init adjacent objects

```

```

(adjacent r0-c0 r1-c0 down) (adjacent r0-c0 r0-c1 right)
(adjacent r0-c1 r1-c1 down) (adjacent r0-c1 r0-c0 left) (adjacent r0-c1 r0-c2 right)
(adjacent r0-c2 r1-c2 down) (adjacent r0-c2 r0-c1 left) (adjacent r0-c2 r0-c3 right)
(adjacent r0-c3 r1-c3 down) (adjacent r0-c3 r0-c2 left) (adjacent r0-c3 r0-c4 right)
(adjacent r0-c4 r1-c4 down) (adjacent r0-c4 r0-c3 left) (adjacent r0-c4 r0-c5 right)
(adjacent r0-c5 r1-c5 down) (adjacent r0-c5 r0-c4 left)
(adjacent r1-c0 r0-c0 up) (adjacent r1-c0 r2-c0 down) (adjacent r1-c0 r1-c1 right)
(adjacent r1-c1 r0-c1 up) (adjacent r1-c1 r2-c1 down) (adjacent r1-c1 r1-c0 left) (adjacent r1-c1 r1-c2 right)
(adjacent r1-c2 r0-c2 up) (adjacent r1-c2 r2-c2 down) (adjacent r1-c2 r1-c1 left) (adjacent r1-c2 r1-c3 right)
(adjacent r1-c3 r0-c3 up) (adjacent r1-c3 r2-c3 down) (adjacent r1-c3 r1-c2 left) (adjacent r1-c3 r1-c4 right)
(adjacent r1-c4 r0-c4 up) (adjacent r1-c4 r2-c4 down) (adjacent r1-c4 r1-c3 left) (adjacent r1-c4 r1-c5 right)
(adjacent r1-c5 r0-c5 up) (adjacent r1-c5 r2-c5 down) (adjacent r1-c5 r1-c4 left)
(adjacent r2-c0 r1-c0 up) (adjacent r2-c0 r3-c0 down) (adjacent r2-c0 r2-c1 right)
(adjacent r2-c1 r1-c1 up) (adjacent r2-c1 r3-c1 down) (adjacent r2-c1 r2-c0 left) (adjacent r2-c1 r2-c2 right)
(adjacent r2-c2 r1-c2 up) (adjacent r2-c2 r3-c2 down) (adjacent r2-c2 r2-c1 left) (adjacent r2-c2 r2-c3 right)
(adjacent r2-c3 r1-c3 up) (adjacent r2-c3 r3-c3 down) (adjacent r2-c3 r2-c2 left) (adjacent r2-c3 r2-c4 right)
(adjacent r2-c4 r1-c4 up) (adjacent r2-c4 r3-c4 down) (adjacent r2-c4 r2-c3 left) (adjacent r2-c4 r2-c5 right)
(adjacent r2-c5 r1-c5 up) (adjacent r2-c5 r3-c5 down) (adjacent r2-c5 r2-c4 left)
(adjacent r3-c0 r2-c0 up) (adjacent r3-c0 r4-c0 down) (adjacent r3-c0 r3-c1 right)
(adjacent r3-c1 r2-c1 up) (adjacent r3-c1 r4-c1 down) (adjacent r3-c1 r3-c0 left) (adjacent r3-c1 r3-c2 right)
(adjacent r3-c2 r2-c2 up) (adjacent r3-c2 r4-c2 down) (adjacent r3-c2 r3-c1 left) (adjacent r3-c2 r3-c3 right)
(adjacent r3-c3 r2-c3 up) (adjacent r3-c3 r4-c3 down) (adjacent r3-c3 r3-c2 left) (adjacent r3-c3 r3-c4 right)
(adjacent r3-c4 r2-c4 up) (adjacent r3-c4 r4-c4 down) (adjacent r3-c4 r3-c3 left) (adjacent r3-c4 r3-c5 right)
(adjacent r3-c5 r2-c5 up) (adjacent r3-c5 r4-c5 down) (adjacent r3-c5 r3-c4 left)
(adjacent r4-c0 r3-c0 up) (adjacent r4-c0 r5-c0 down) (adjacent r4-c0 r4-c1 right)
(adjacent r4-c1 r3-c1 up) (adjacent r4-c1 r5-c1 down) (adjacent r4-c1 r4-c0 left) (adjacent r4-c1 r4-c2 right)
(adjacent r4-c2 r3-c2 up) (adjacent r4-c2 r5-c2 down) (adjacent r4-c2 r4-c1 left) (adjacent r4-c2 r4-c3 right)
(adjacent r4-c3 r3-c3 up) (adjacent r4-c3 r5-c3 down) (adjacent r4-c3 r4-c2 left) (adjacent r4-c3 r4-c4 right)
(adjacent r4-c4 r3-c4 up) (adjacent r4-c4 r5-c4 down) (adjacent r4-c4 r4-c3 left) (adjacent r4-c4 r4-c5 right)

```

```

(adjacent r4-c5 r3-c5 up) (adjacent r4-c5 r5-c5 down) (adjacent r4-c5 r4-c4 left)
(adjacent r5-c0 r4-c0 up) (adjacent r5-c0 r6-c0 down) (adjacent r5-c0 r5-c1 right)
(adjacent r5-c1 r4-c1 up) (adjacent r5-c1 r6-c1 down) (adjacent r5-c1 r5-c0 left) (adjacent r5-c1 r5-c2
(adjacent r5-c2 r4-c2 up) (adjacent r5-c2 r6-c2 down) (adjacent r5-c2 r5-c1 left) (adjacent r5-c2 r5-c3
(adjacent r5-c3 r4-c3 up) (adjacent r5-c3 r6-c3 down) (adjacent r5-c3 r5-c2 left) (adjacent r5-c3 r5-c4
(adjacent r5-c4 r4-c4 up) (adjacent r5-c4 r6-c4 down) (adjacent r5-c4 r5-c3 left) (adjacent r5-c4 r5-c5
(adjacent r5-c5 r4-c5 up) (adjacent r5-c5 r6-c5 down) (adjacent r5-c5 r5-c4 left)
(adjacent r6-c0 r5-c0 up) (adjacent r6-c0 r6-c1 right)
(adjacent r6-c1 r5-c1 up) (adjacent r6-c1 r6-c0 left) (adjacent r6-c1 r6-c2 right)
(adjacent r6-c2 r5-c2 up) (adjacent r6-c2 r6-c1 left) (adjacent r6-c2 r6-c3 right)
(adjacent r6-c3 r5-c3 up) (adjacent r6-c3 r6-c2 left) (adjacent r6-c3 r6-c4 right)
(adjacent r6-c4 r5-c4 up) (adjacent r6-c4 r6-c3 left) (adjacent r6-c4 r6-c5 right)
(adjacent r6-c5 r5-c5 up) (adjacent r6-c5 r6-c4 left)

```

```

; Init Done
)
; Define goal states
(:goal (and (block-at o r1-c2)))

```

```

; Problem Define Done
)

```

```

In [13]: %%bash -s "$blackbox" "$domain" "$problem"
$1 -o $2 -f $3

```

blackbox version 43

command line: ../../../../Documents/Code/Planners/Blackbox43LinuxBinary/blackbox -o /home/tox/git/HW1\_Team3/Code/Worlds/sokoban\_domain.pddl

Begin solver specification

```

-maxint      0  -maxsec 10.000000  graphplan
-maxint      0  -maxsec 0.000000   chaff

```

End solver specification

Loading domain file: /home/tox/git/HW1\_Team3/Code/Worlds/sokoban\_domain.pddl

Loading fact file: /home/tox/git/HW1\_Team3/Code/Worlds/world1\_proble.pddl

Problem name: sokoban-problem

Facts loaded.

time: 1, 176 facts and 6 exclusive pairs.

time: 2, 180 facts and 28 exclusive pairs.

time: 3, 185 facts and 79 exclusive pairs.

time: 4, 188 facts and 117 exclusive pairs.

time: 5, 189 facts and 124 exclusive pairs.

time: 6, 190 facts and 130 exclusive pairs.

time: 7, 190 facts and 122 exclusive pairs.

time: 8, 191 facts and 131 exclusive pairs.

time: 9, 193 facts and 162 exclusive pairs.

time: 10, 193 facts and 148 exclusive pairs.

time: 11, 193 facts and 139 exclusive pairs.

time: 12, 194 facts and 153 exclusive pairs.

time: 13, 195 facts and 167 exclusive pairs.

time: 14, 196 facts and 180 exclusive pairs.

Goals first reachable in 14 steps.

5912 nodes created.

#####

goals at time 15:

block-at.o\_r1-c2

```
-----
Invoking solver graphplan
Result is Sat
Iteration was 385
Performing plan justification:
    0 actions were pruned in 0.00 seconds
-----
```

```
Begin plan
1 (move r2-c2 r3-c2 down)
2 (move r3-c2 r3-c3 right)
3 (move r3-c3 r3-c4 right)
4 (move r3-c4 r4-c4 down)
5 (push r4-c4 r4-c3 r4-c2 left o)
6 (move r4-c3 r3-c3 up)
7 (move r3-c3 r3-c2 left)
8 (move r3-c2 r3-c1 left)
9 (move r3-c1 r4-c1 down)
10 (move r4-c1 r5-c1 down)
11 (move r5-c1 r5-c2 right)
12 (push r5-c2 r4-c2 r3-c2 up o)
13 (push r4-c2 r3-c2 r2-c2 up o)
14 (push r3-c2 r2-c2 r1-c2 up o)
End plan
-----
```

```
14 total actions in plan
0 entries in hash table,
13 total set-creation steps (entries + hits + plan length - 1)
14 actions tried
```

```
#####
Total elapsed time: 0.32 seconds
Time in milliseconds: 322
```

```
#####
```

```
In [14]: %%bash -s "$vhpop" "$domain" "$problem"
$1 -f LCFR -l 10000 -f MW -l unlimited $2 $3
```

Process is terminated.

Well, VHPOP never seemed to be able to finish this example under 15 min, so lets try something else.

```
In [15]: %%bash -s "$pyperplan" "$domain" "$problem"
$1 $2 $3 $4 --plugins visualizer -s bfs
```

```
2014-10-05 00:18:53,023 INFO      Using plugin: visualizer
2014-10-05 00:18:53,023 INFO      problem: /home/tox/git/HW1_Team3/Code/Worlds/world1_proble.pddl
2014-10-05 00:18:53,023 INFO      using search: breadth_first_search
2014-10-05 00:18:53,023 INFO      using heuristic: None
2014-10-05 00:18:53,023 INFO      using transition reduction: NoneTransitionPruning
```

```

2014-10-05 00:18:53,023 INFO      using state reduction: NoneStatePruning
2014-10-05 00:18:53,036 INFO      Parsing domain /home/tox/git/HW1_Team3/Code/Worlds/sokoban_domain.pddl
domFile: /home/tox/git/HW1_Team3/Code/Worlds/sokoban_domain.pddl
2014-10-05 00:18:53,038 INFO      Parsing problem /home/tox/git/HW1_Team3/Code/Worlds/world1_proble.pddl
2014-10-05 00:18:53,044 INFO      7 Predicates parsed
2014-10-05 00:18:53,045 INFO      2 Actions parsed
2014-10-05 00:18:53,045 INFO      47 Objects parsed
2014-10-05 00:18:53,045 INFO      0 Constants parsed
2014-10-05 00:18:53,045 INFO      Grounding start: sokoban_problem
2014-10-05 00:18:53,148 INFO      Relevance analysis removed 5 facts
2014-10-05 00:18:53,149 INFO      Grounding end: sokoban_problem
2014-10-05 00:18:53,149 INFO      38 Variables created
2014-10-05 00:18:53,149 INFO      48 Operators created
2014-10-05 00:18:53,149 INFO      Search start: sokoban_problem
2014-10-05 00:18:53,191 INFO      Goal reached. Start extraction of solution.
2014-10-05 00:18:53,191 INFO      95 Nodes expanded
2014-10-05 00:18:53,195 INFO      Search end: sokoban_problem
2014-10-05 00:18:53,195 INFO      Wall-clock search time: 0.04592
2014-10-05 00:18:53,904 INFO      Plan length: 14
2014-10-05 00:18:53,905 INFO      validate could not be found on the PATH so the plan can not be validated

```

```

In [16]: %%bash -s "cat" "$solution"
$1 $2

```

```

(move r2-c2 r3-c2 down)
(move r3-c2 r3-c3 right)
(move r3-c3 r3-c4 right)
(move r3-c4 r4-c4 down)
(push r4-c4 r4-c3 r4-c2 left o)
(move r4-c3 r3-c3 up)
(move r3-c3 r3-c2 left)
(move r3-c2 r3-c1 left)
(move r3-c1 r4-c1 down)
(move r4-c1 r5-c1 down)
(move r5-c1 r5-c2 right)
(push r5-c2 r4-c2 r3-c2 up o)
(push r4-c2 r3-c2 r2-c2 up o)
(push r3-c2 r2-c2 r1-c2 up o)

```

```

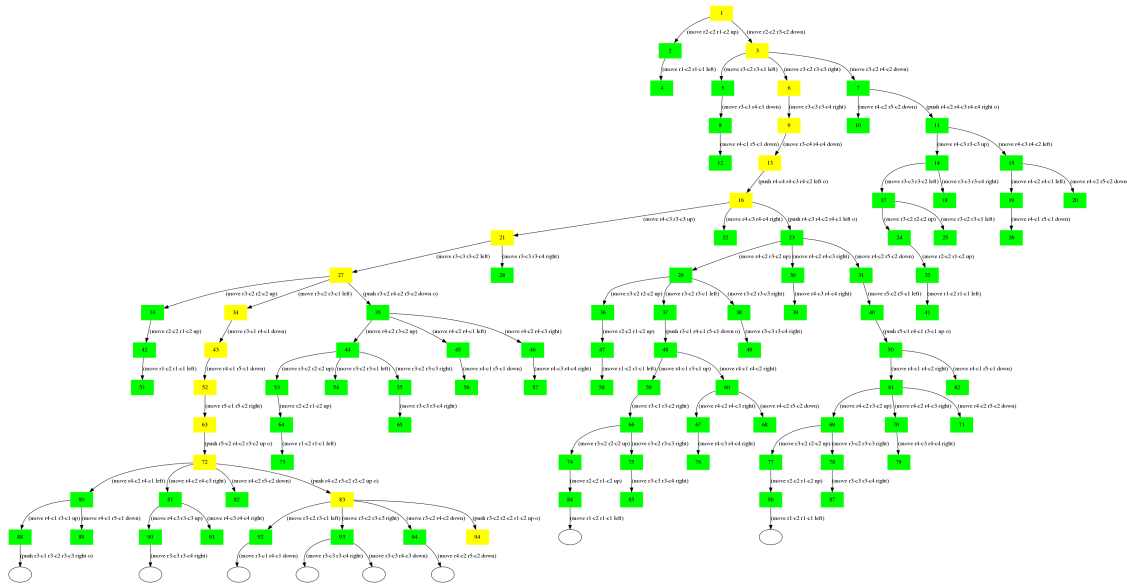
In [17]: Image("Worlds-bfs-hff-world1_proble.png")

```

```

Out[17]:

```



```
In [18]: domain = world_path + "sokoban_domain.pddl"
world = world_path + "world2.txt"
problem = world_path + "world2_problem.pddl"
solution = world_path + "world2_problem.pddl.soln"
```

```
sokoban = Sokoban(world, labeled_boxes = False)
sokoban.writeProblem(problem)
```

```
In [19]: %%bash -s "$pyperplan" "$domain" "$problem"
$1 $2 $3 $4 --plugins visualizer
```

```
2014-10-05 00:18:57,062 INFO Using plugin: visualizer
2014-10-05 00:18:57,062 INFO problem: /home/tox/git/HW1_Team3/Code/Worlds/world2_problem.pddl
2014-10-05 00:18:57,062 INFO using search: breadth_first_search
2014-10-05 00:18:57,062 INFO using heuristic: None
2014-10-05 00:18:57,062 INFO using transition reduction: NoneTransitionPruning
2014-10-05 00:18:57,062 INFO using state reduction: NoneStatePruning
2014-10-05 00:18:57,063 INFO Parsing domain /home/tox/git/HW1_Team3/Code/Worlds/sokoban_domain.pddl
2014-10-05 00:18:57,065 INFO domFile: /home/tox/git/HW1_Team3/Code/Worlds/sokoban_domain.pddl
2014-10-05 00:18:57,073 INFO Parsing problem /home/tox/git/HW1_Team3/Code/Worlds/world2_problem.pddl
2014-10-05 00:18:57,073 INFO 7 Predicates parsed
2014-10-05 00:18:57,073 INFO 2 Actions parsed
2014-10-05 00:18:57,073 INFO 54 Objects parsed
2014-10-05 00:18:57,073 INFO 0 Constants parsed
2014-10-05 00:18:57,073 INFO Grounding start: sokoban_problem
2014-10-05 00:18:57,293 INFO Relevance analysis removed 12 facts
2014-10-05 00:18:57,293 INFO Grounding end: sokoban_problem
2014-10-05 00:18:57,293 INFO 56 Variables created
2014-10-05 00:18:57,293 INFO 76 Operators created
2014-10-05 00:18:57,293 INFO Search start: sokoban_problem
2014-10-05 00:18:57,739 INFO Goal reached. Start extraction of solution.
2014-10-05 00:18:57,739 INFO 953 Nodes expanded
2014-10-05 00:18:57,746 INFO Search end: sokoban_problem
```

```

2014-10-05 00:18:57,746 INFO      Wall-clock search time: 0.45312
2014-10-05 00:19:02,665 INFO      Plan length: 32
2014-10-05 00:19:02,666 INFO      validate could not be found on the PATH so the plan can not be validated

```

```
In [20]: %%bash -s "cat" "$solution"
```

```
$1 $2
```

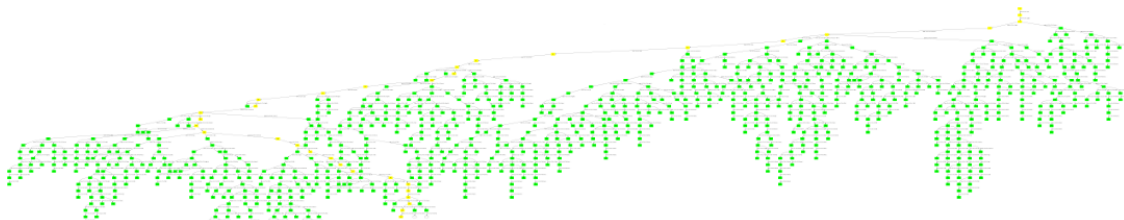
```

(move r2-c1 r1-c1 up)
(move r1-c1 r1-c2 right)
(move r1-c2 r1-c3 right)
(push r1-c3 r2-c3 r3-c3 down b)
(move r2-c3 r1-c3 up)
(move r1-c3 r1-c2 left)
(move r1-c2 r1-c1 left)
(move r1-c1 r2-c1 down)
(push r2-c1 r2-c2 r2-c3 right a)
(move r2-c2 r3-c2 down)
(push r3-c2 r3-c3 r3-c4 right b)
(move r3-c3 r3-c2 left)
(move r3-c2 r2-c2 up)
(move r2-c2 r1-c2 up)
(move r1-c2 r1-c3 right)
(push r1-c3 r2-c3 r3-c3 down a)
(move r2-c3 r2-c4 right)
(push r2-c4 r3-c4 r4-c4 down b)
(push r3-c4 r4-c4 r5-c4 down b)
(push r4-c4 r5-c4 r6-c4 down b)
(move r5-c4 r4-c4 up)
(move r4-c4 r4-c3 left)
(push r4-c3 r3-c3 r2-c3 up a)
(move r3-c3 r3-c4 right)
(move r3-c4 r2-c4 up)
(push r2-c4 r2-c3 r2-c2 left a)
(move r2-c3 r3-c3 down)
(move r3-c3 r3-c2 left)
(push r3-c2 r2-c2 r1-c2 up a)
(move r2-c2 r2-c3 right)
(move r2-c3 r1-c3 up)
(push r1-c3 r1-c2 r1-c1 left a)

```

```
In [2]: Image("Figures/Worlds-bfs-hff-world2_proble.png", width=1080)
```

```
Out[2]:
```



```
In [22]: %%bash -s "$pyperplan" "$domain" "$problem"
$1 $2 $3 $4 --plugins visualizer -H hff -s gbf
```

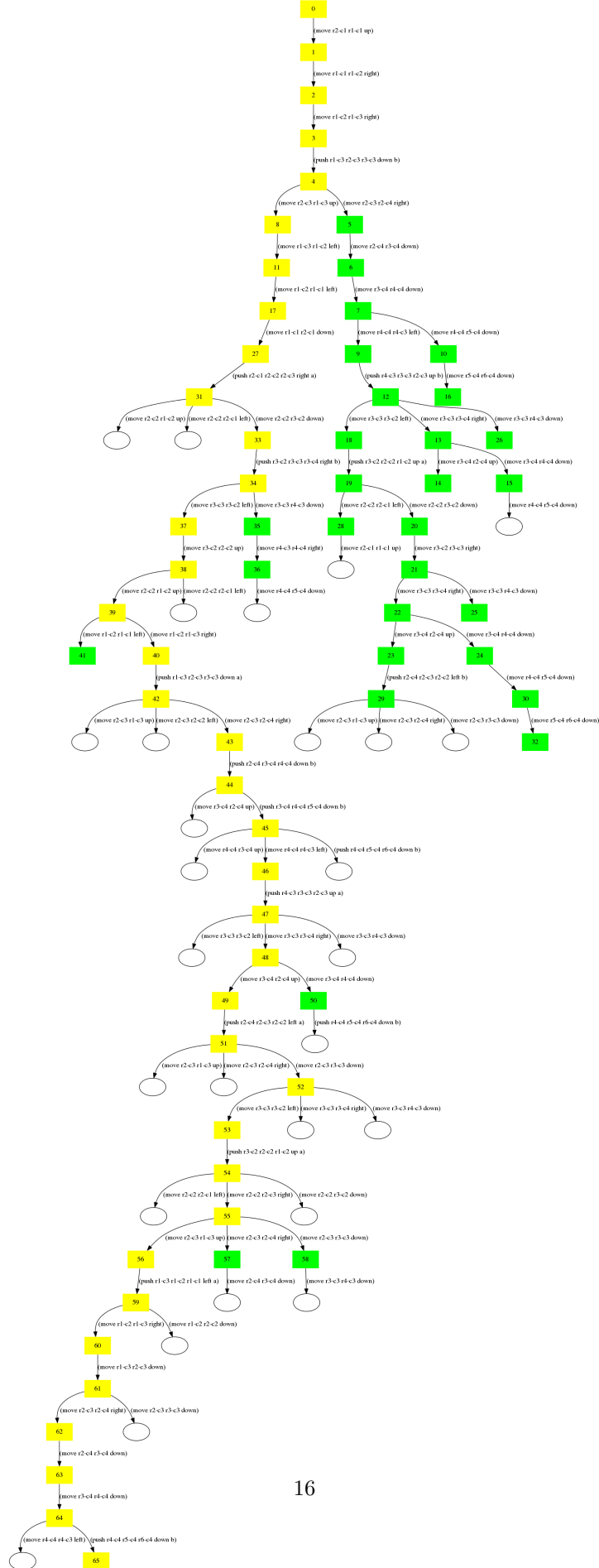
```

2014-10-05 00:19:03,068 INFO      Using plugin: visualizer
2014-10-05 00:19:03,068 INFO      problem: /home/tox/git/HW1_Team3/Code/Worlds/world2_proble.pddl
2014-10-05 00:19:03,068 INFO      using search: greedy_best_first_search
2014-10-05 00:19:03,068 INFO      using heuristic: hFFHeuristic
2014-10-05 00:19:03,068 INFO      using transition reduction: NoneTransitionPruning
2014-10-05 00:19:03,068 INFO      using state reduction: NoneStatePruning
2014-10-05 00:19:03,068 INFO      Parsing domain /home/tox/git/HW1_Team3/Code/Worlds/sokoban_domain.pddl
domFile: /home/tox/git/HW1_Team3/Code/Worlds/sokoban_domain.pddl
2014-10-05 00:19:03,070 INFO      Parsing problem /home/tox/git/HW1_Team3/Code/Worlds/world2_proble.pddl
2014-10-05 00:19:03,078 INFO      7 Predicates parsed
2014-10-05 00:19:03,078 INFO      2 Actions parsed
2014-10-05 00:19:03,078 INFO      54 Objects parsed
2014-10-05 00:19:03,078 INFO      0 Constants parsed
2014-10-05 00:19:03,079 INFO      Grounding start: sokoban_problem
2014-10-05 00:19:03,310 INFO      Relevance analysis removed 12 facts
2014-10-05 00:19:03,310 INFO      Grounding end: sokoban_problem
2014-10-05 00:19:03,310 INFO      56 Variables created
2014-10-05 00:19:03,310 INFO      76 Operators created
2014-10-05 00:19:03,310 INFO      Search start: sokoban_problem
2014-10-05 00:19:03,311 INFO      Initial h value: 15.000000
2014-10-05 00:19:03,388 INFO      Goal reached. Start extraction of solution.
2014-10-05 00:19:03,388 INFO      66 Nodes expanded
2014-10-05 00:19:03,396 INFO      Search end: sokoban_problem
2014-10-05 00:19:03,396 INFO      Wall-clock search time: 0.08562
2014-10-05 00:19:03,815 INFO      Plan length: 36
2014-10-05 00:19:03,816 INFO      validate could not be found on the PATH so the plan can not be validated

```

In [23]: `Image("Worlds-gbf-hff-world2_proble.png")`

Out[23]:





```
In [24]: domain    = world_path + "sokoban_domain.pddl"
        world      = world_path + "world3.txt"
        problem    = world_path + "world3_proble.pddl"
        solution   = world_path + "world3_proble.pddl.soln"
```

```
sokoban = Sokoban(world, labeled_boxes = False)
sokoban.writeProblem(problem)
```

```
In [25]: %%bash -s "$pyperplan" "$domain" "$problem"
        $1 $2 $3 $4 --plugins visualizer
```

```
2014-10-05 00:19:04,184 INFO      Using plugin: visualizer
2014-10-05 00:19:04,184 INFO      problem: /home/tox/git/HW1_Team3/Code/Worlds/world3_proble.pddl
2014-10-05 00:19:04,184 INFO      using search: breadth_first_search
2014-10-05 00:19:04,184 INFO      using heuristic: None
2014-10-05 00:19:04,184 INFO      using transition reduction: NoneTransitionPruning
2014-10-05 00:19:04,184 INFO      using state reduction: NoneStatePruning
2014-10-05 00:19:04,185 INFO      Parsing domain /home/tox/git/HW1_Team3/Code/Worlds/sokoban_domain.pddl
domFile: /home/tox/git/HW1_Team3/Code/Worlds/sokoban_domain.pddl
2014-10-05 00:19:04,187 INFO      Parsing problem /home/tox/git/HW1_Team3/Code/Worlds/world3_proble.pddl
2014-10-05 00:19:04,204 INFO      7 Predicates parsed
2014-10-05 00:19:04,204 INFO      2 Actions parsed
2014-10-05 00:19:04,204 INFO      95 Objects parsed
2014-10-05 00:19:04,204 INFO      0 Constants parsed
2014-10-05 00:19:04,204 INFO      Grounding start: sokoban_problem
2014-10-05 00:19:06,341 INFO      Relevance analysis removed 17 facts
2014-10-05 00:19:06,341 INFO      Grounding end: sokoban_problem
2014-10-05 00:19:06,341 INFO      135 Variables created
2014-10-05 00:19:06,341 INFO      192 Operators created
2014-10-05 00:19:06,341 INFO      Search start: sokoban_problem
2014-10-05 00:19:11,207 INFO      Goal reached. Start extraction of solution.
2014-10-05 00:19:11,207 INFO      7867 Nodes expanded
2014-10-05 00:19:11,233 INFO      Search end: sokoban_problem
2014-10-05 00:19:11,233 INFO      Wall-clock search time: 4.89248
dot: graph is too large for cairo-renderer bitmaps. Scaling by 0.335329 to fit
```

```
2014-10-05 00:20:17,593 INFO      Plan length: 89
2014-10-05 00:20:17,599 INFO      validate could not be found on the PATH so the plan can not be validated
```

```
In [26]: %%bash -s "cat" "$solution"
        $1 $2
```

```
(move r4-c2 r4-c3 right)
(move r4-c3 r5-c3 down)
(move r5-c3 r6-c3 down)
(move r6-c3 r6-c4 right)
(move r6-c4 r6-c5 right)
(move r6-c5 r5-c5 up)
(move r5-c5 r4-c5 up)
(push r4-c5 r4-c4 r4-c3 left c)
(push r4-c4 r4-c3 r4-c2 left c)
(move r4-c3 r4-c4 right)
```

```

(move r4-c4 r4-c5 right)
(move r4-c5 r5-c5 down)
(move r5-c5 r6-c5 down)
(move r6-c5 r6-c6 right)
(move r6-c6 r6-c7 right)
(move r6-c7 r5-c7 up)
(move r5-c7 r4-c7 up)
(move r4-c7 r3-c7 up)
(move r3-c7 r2-c7 up)
(move r2-c7 r1-c7 up)
(move r1-c7 r1-c8 right)
(move r1-c8 r1-c9 right)
(move r1-c9 r2-c9 down)
(move r2-c9 r3-c9 down)
(move r3-c9 r4-c9 down)
(push r4-c9 r4-c8 r4-c7 left a)
(move r4-c8 r4-c9 right)
(move r4-c9 r3-c9 up)
(move r3-c9 r2-c9 up)
(move r2-c9 r1-c9 up)
(move r1-c9 r1-c8 left)
(move r1-c8 r1-c7 left)
(move r1-c7 r2-c7 down)
(move r2-c7 r3-c7 down)
(push r3-c7 r4-c7 r5-c7 down a)
(push r4-c7 r4-c6 r4-c5 left b)
(push r4-c6 r4-c5 r4-c4 left b)
(move r4-c5 r5-c5 down)
(move r5-c5 r6-c5 down)
(move r6-c5 r6-c6 right)
(move r6-c6 r6-c7 right)
(push r6-c7 r5-c7 r4-c7 up a)
(push r5-c7 r4-c7 r3-c7 up a)
(push r4-c7 r3-c7 r2-c7 up a)
(push r3-c7 r2-c7 r1-c7 up a)
(move r2-c7 r3-c7 down)
(move r3-c7 r4-c7 down)
(move r4-c7 r4-c6 left)
(move r4-c6 r4-c5 left)
(move r4-c5 r5-c5 down)
(move r5-c5 r6-c5 down)
(move r6-c5 r6-c4 left)
(move r6-c4 r6-c3 left)
(move r6-c3 r5-c3 up)
(move r5-c3 r4-c3 up)
(push r4-c3 r4-c4 r4-c5 right b)
(push r4-c4 r4-c5 r4-c6 right b)
(push r4-c5 r4-c6 r4-c7 right b)
(move r4-c6 r4-c5 left)
(move r4-c5 r5-c5 down)
(move r5-c5 r6-c5 down)
(move r6-c5 r6-c6 right)
(move r6-c6 r6-c7 right)
(move r6-c7 r5-c7 up)

```

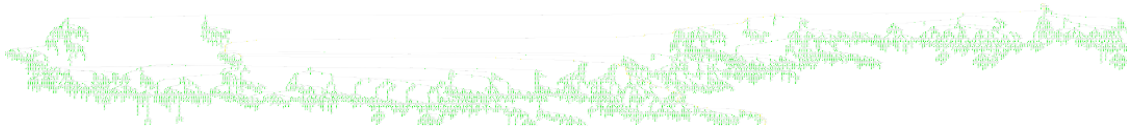
```

(push r5-c7 r4-c7 r3-c7 up b)
(push r4-c7 r3-c7 r2-c7 up b)
(move r3-c7 r4-c7 down)
(move r4-c7 r4-c6 left)
(move r4-c6 r4-c5 left)
(move r4-c5 r4-c4 left)
(move r4-c4 r4-c3 left)
(move r4-c3 r5-c3 down)
(move r5-c3 r6-c3 down)
(move r6-c3 r6-c2 left)
(move r6-c2 r6-c1 left)
(move r6-c1 r5-c1 up)
(move r5-c1 r4-c1 up)
(push r4-c1 r4-c2 r4-c3 right c)
(push r4-c2 r4-c3 r4-c4 right c)
(push r4-c3 r4-c4 r4-c5 right c)
(push r4-c4 r4-c5 r4-c6 right c)
(push r4-c5 r4-c6 r4-c7 right c)
(move r4-c6 r4-c5 left)
(move r4-c5 r5-c5 down)
(move r5-c5 r6-c5 down)
(move r6-c5 r6-c6 right)
(move r6-c6 r6-c7 right)
(move r6-c7 r5-c7 up)
(push r5-c7 r4-c7 r3-c7 up c)

```

In [3]: Image("Figures/Worlds-bfs-hff-world3\_proble.png", width=1080)

Out[3]:



In [28]: `%%bash -s "$pyperplan" "$domain" "$problem"`

`$1 $2 $3 $4 --plugins visualizer -H hff -s gbf`

```

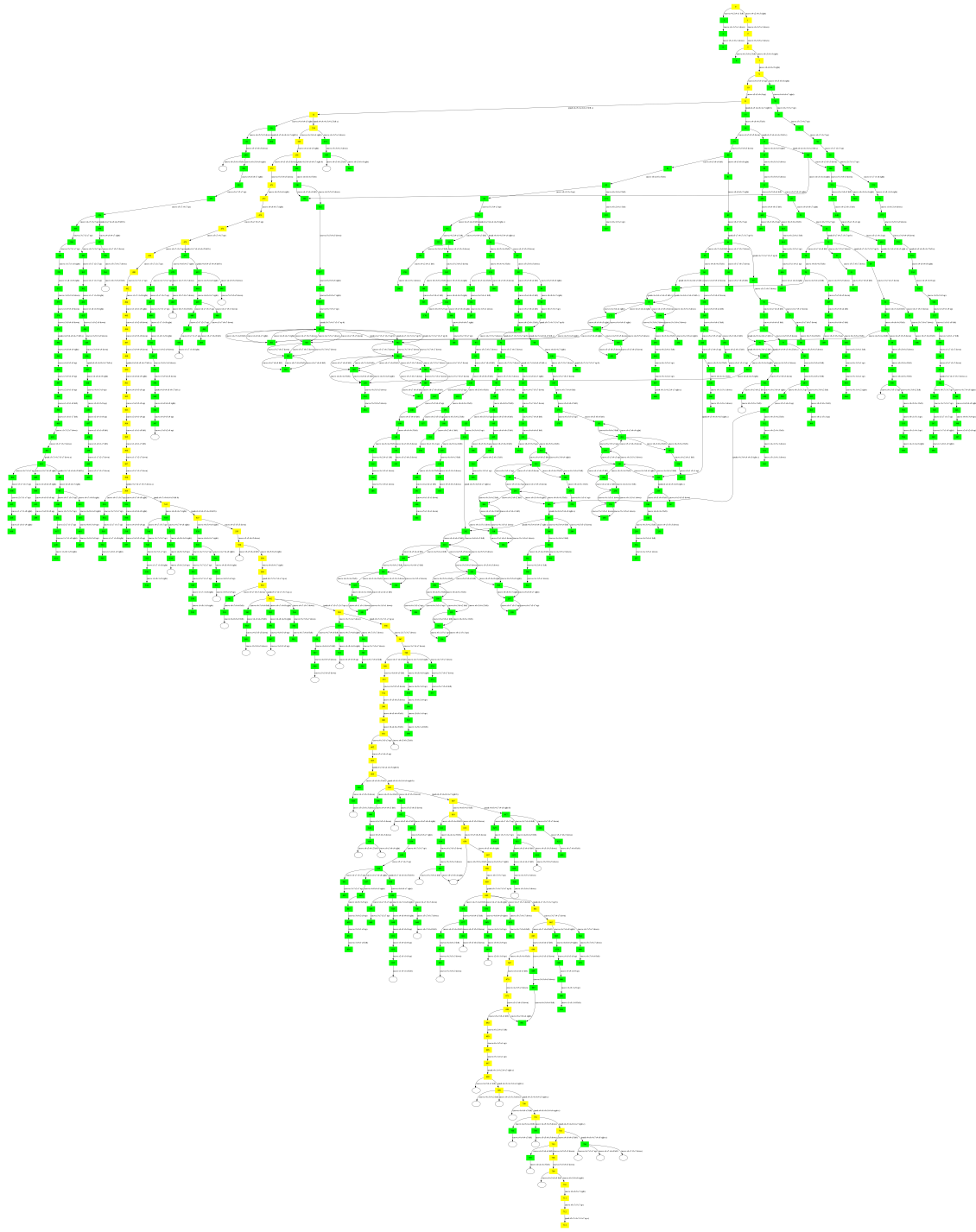
2014-10-05 00:20:18,758 INFO      Using plugin: visualizer
2014-10-05 00:20:18,758 INFO      problem: /home/tox/git/HW1_Team3/Code/Worlds/world3_proble.pddl
2014-10-05 00:20:18,758 INFO      using search: greedy_best_first_search
2014-10-05 00:20:18,758 INFO      using heuristic: hFFHeuristic
2014-10-05 00:20:18,758 INFO      using transition reduction: NoneTransitionPruning
2014-10-05 00:20:18,758 INFO      using state reduction: NoneStatePruning
2014-10-05 00:20:18,758 INFO      Parsing domain /home/tox/git/HW1_Team3/Code/Worlds/sokoban_domain.pddl
2014-10-05 00:20:18,760 INFO      Parsing problem /home/tox/git/HW1_Team3/Code/Worlds/world3_proble.pddl
2014-10-05 00:20:18,776 INFO      7 Predicates parsed
2014-10-05 00:20:18,776 INFO      2 Actions parsed
2014-10-05 00:20:18,776 INFO      95 Objects parsed
2014-10-05 00:20:18,776 INFO      0 Constants parsed
2014-10-05 00:20:18,776 INFO      Grounding start: sokoban_problem
2014-10-05 00:20:20,810 INFO      Relevance analysis removed 17 facts

```

```
2014-10-05 00:20:20,810 INFO      Grounding end: sokoban_problem
2014-10-05 00:20:20,810 INFO      135 Variables created
2014-10-05 00:20:20,810 INFO      192 Operators created
2014-10-05 00:20:20,810 INFO      Search start: sokoban_problem
2014-10-05 00:20:20,811 INFO      Initial h value: 19.000000
2014-10-05 00:20:22,436 INFO      Goal reached. Start extraction of solution.
2014-10-05 00:20:22,436 INFO      714 Nodes expanded
2014-10-05 00:20:22,481 INFO      Search end: sokoban_problem
2014-10-05 00:20:22,481 INFO      Wall-clock search time: 1.67087
2014-10-05 00:20:26,493 INFO      Plan length: 89
2014-10-05 00:20:26,494 INFO      validate could not be found on the PATH so the plan can not be validated
```

```
In [29]: Image("Worlds-gbf-hff-world3_proble.png")
```

```
Out[29]:
```



```
In [30]: world      = "Worlds/world4.txt"
         problem    = "Worlds/world4_problem.pddl"
         solution    = "Worlds/world4_problem.pddl.soln"
         # sokoban = Sokoban(world, labeled_boxes = False)
         # sokoban.writeProblem(problem)
```

```
In [31]: %bash -s "cat" "$solution"  
$1 $2
```

```
cat: Worlds/world4.proble.pddl.soln: No such file or directory
```

```
In [31]:
```

```
In [31]:
```

**1.2.2 b ) Compare the performance of two planners on this domain. Which one works better? Does this make sense, why?**

```
In [31]:
```

**1.2.3 c ) Clearly PDDL was not intended for this sort of application. Discuss the challenges in expressing geometric constraints in semantic planning.**

```
In [31]:
```

**1.2.4 d ) In many cases, geometric and dynamic planning are insufficient to describe a domain. Give an example of a problem that is best suited for semantic (classical) planning. Explain why a semantic representation would be desirable.**

```
In [31]:
```

```
In []:
```