

Project 2: SATPlan and the Tower of Hanoi

CSCI 561

Fall 2018

The *Tower of Hanoi* is a classic puzzle consisting of round disks stacked on pegs. One must disk_move all disks to the final peg, subject to the following constraints:

1. Only one disk can be disk_moved at a time.
2. Each disk_move consists of taking the upper disk from one of the stacks and placing it on top of another stack.
3. No disk may be placed on top of a smaller disk.

Answer the following questions:

1. (15 pts) Create PDDL domains (operators and facts) for the following Tower of Hanoi instances (it is possible that the PDDL operators will be the same):

Solutions:

```
(define (domain hanoi)
  (:requirements :strips)
  (:predicates (clear ?x) (on ?x ?y) (smaller ?x ?y))

  (:action disk_move
    :parameters (?x ?y-1 ?y-2)
    :precondition (and (smaller ?y-2 ?x) (on ?x ?y-1)
                       (clear ?x) (clear ?y-2))
    :effect (and (clear ?y-1)
                 (on ?x ?y-2)
                 (not (on ?x ?y-1))
                 (not (clear ?y-2))))
)
```

Figure 1: Tower of Hanoi Domain in pddl

```
(define (problem hanoi-a)
  (:domain hanoi)
  (:objects p1 d1 p2 p3 d2)
  (:init
    (smaller p1 d1) (smaller p2 d1)
    (smaller p3 d1) (smaller p1 d2)
    (smaller p2 d2) (smaller p3 d2)
    (smaller d2 d1)
    (on d1 d2)
    (clear d1) (clear p2) (clear p3)
    (on d2 p1))
  (:goal (and (on d2 p3) (on d1 d2)))
)
```

Figure 2: Facts for Two Disks problem in pddl

```
(define (problem hanoi-a)
  (:domain hanoi)
  (:objects p1 d1 p2 p3 d2 d3 d4)
  (:init
    (smaller p1 d1)
    (smaller p2 d1)
    (smaller p3 d1)
    (smaller p1 d2)
    (smaller p2 d2)
    (smaller p3 d2)

    (smaller p1 d3)
    (smaller p2 d3)
    (smaller p3 d3)

    (smaller p1 d4)
    (smaller p2 d4)
    (smaller p3 d4)

    (smaller d4 d3)
    (smaller d4 d2)
    (smaller d4 d1)
    (smaller d3 d2)
    (smaller d3 d1)
    (smaller d2 d1)

    (on d1 d2)
    (on d2 d3)
    (on d3 d4)
    (on d4 p1)
    (clear d1)
    (clear p2)
    (clear p3))
  (:goal (and (on d4 p3)
               (on d1 d2)
               (on d2 d3)
               (on d3 d4)))
)
```

Figure 3: Facts for Four Disks problem in pddl

- (10 pts) Download one or more of the following planners and use them to produce plans for your PDDL domains:

- Blackbox: <https://www.cs.rochester.edu/u/kautz/satplan/blackbox/>
- Madagascar: <http://research.ics.aalto.fi/software/sat/madagascar/>
- TMKit: <http://tmkit.dyalab.org/>

What plans are produced by each planner for each instance (two and four disks)?

Solutions:

Using blackbox

Two disk:

```
Begin plan
1 (disk_move d1 d2 p2)
2 (disk_move d2 p1 p3)
3 (disk_move d1 p2 d2)
End plan
```

four Disk:

```
Begin plan
1 (disk_move d1 d2 p2)
2 (disk_move d2 d3 p3)
```

```
3 (disk_move d1 p2 d2)
4 (disk_move d3 d4 p2)
5 (disk_move d1 d2 d4)
6 (disk_move d2 p3 d3)
7 (disk_move d1 d4 d2)
8 (disk_move d4 p1 p3)
9 (disk_move d1 d2 d4)
10 (disk_move d2 d3 p1)
11 (disk_move d1 d4 d2)
12 (disk_move d3 p2 d4)
13 (disk_move d1 d2 p2)
14 (disk_move d2 p1 d3)
15 (disk_move d1 p2 d2)
End plan
```

For Madagascar:
Two Disk:

```
STEP 0: disk_move(d1,d2,p2)
STEP 1: disk_move(d2,p1,p3)
STEP 2: disk_move(d1,p2,p1)
STEP 4: disk_move(d1,p1,d2)
```

Four Disk:

```
STEP 0: disk_move(d1,d2,p2)
STEP 1.0: disk_move(d2,d3,p3)
STEP 1.1: disk_move(d1,p2,d2)
STEP 2: disk_move(d3,d4,p2)
STEP 3: disk_move(d1,d2,d4)
STEP 4.0: disk_move(d2,p3,d3)
STEP 4.1: disk_move(d1,d4,d2)
STEP 5.0: disk_move(d4,p1,p3)
STEP 5.1: disk_move(d1,d2,d4)
STEP 6.0: disk_move(d2,d3,p1)
STEP 6.1: disk_move(d1,d4,d2)
STEP 7: disk_move(d3,p2,d4)
STEP 8: disk_move(d1,d2,p2)
STEP 9.0: disk_move(d2,p1,d3)
STEP 9.1: disk_move(d1,p2,d2)
```

3. (15 pts) Encode the two-disk instance as a Boolean formula using the SATPlan method.

Solutions:

Start:

```
(AND
  smaller-d2-d1-0
  smaller-p1-d1-0
  smaller-p1-d2-0
  smaller-p2-d1-0
  smaller-p2-d2-0
  smaller-p3-d1-0
```

```
smaller -p3-d2-0
clear -d1-0
clear -p2-0
clear -p3-0
on-d1-d2-0
on-d2-p1-0
(NOT smaller -d2-d2-0)
(NOT clear -d2-0)
(NOT clear -p1-0)
(NOT on-d1-d1-0)
(NOT on-d1-p1-0)
(NOT on-d1-p2-0)
(NOT on-d1-p3-0)
(NOT on-d2-d1-0)
(NOT on-d2-d2-0)
(NOT on-d2-p2-0)
(NOT on-d2-p3-0))
```

Goal:

```
(AND on-d1-d2-3 on-d2-p3-3)
```

Operator Encoding:

```
(OR (NOT disk.move-d1-p1-d2-0)
(AND smaller -d2-d1-0
on-d1-p1-0
clear -d1-0
clear -d2-0
clear -p1-1
on-d1-d2-1
(NOT on-d1-p1-1)
(NOT clear -d2-1)))
```

```
(OR (NOT disk.move-d1-p1-p2-0)
(AND smaller -p2-d1-0
on-d1-p1-0
clear -d1-0
clear -p2-0
clear -p1-1
on-d1-p2-1
(NOT on-d1-p1-1)
(NOT clear -p2-1)))
```

```
(OR (NOT disk.move-d1-p1-p3-0)
(AND smaller -p3-d1-0
on-d1-p1-0
clear -d1-0
clear -p3-0
clear -p1-1
on-d1-p3-1
(NOT on-d1-p1-1)
```

```
(NOT clear -p3 -1)))

(OR (NOT disk_move -d1 -p2 -d2 -0)
(AND smaller -d2 -d1 -0
on -d1 -p2 -0
clear -d1 -0
clear -d2 -0
clear -p2 -1
on -d1 -d2 -1
(NOT on -d1 -p2 -1)
(NOT clear -d2 -1)))

(OR (NOT disk_move -d1 -p2 -p1 -0)
(AND smaller -p1 -d1 -0
on -d1 -p2 -0
clear -d1 -0
clear -p1 -0
clear -p2 -1
on -d1 -p1 -1
(NOT on -d1 -p2 -1)
(NOT clear -p1 -1)))

(OR (NOT disk_move -d1 -p2 -p3 -0)
(AND smaller -p3 -d1 -0
on -d1 -p2 -0
clear -d1 -0
clear -p3 -0
clear -p2 -1
on -d1 -p3 -1
(NOT on -d1 -p2 -1)
(NOT clear -p3 -1)))

(OR (NOT disk_move -d1 -p3 -d2 -0)
(AND smaller -d2 -d1 -0
on -d1 -p3 -0
clear -d1 -0
clear -d2 -0
clear -p3 -1
on -d1 -d2 -1
(NOT on -d1 -p3 -1)
(NOT clear -d2 -1)))

(OR (NOT disk_move -d1 -p3 -p1 -0)
(AND smaller -p1 -d1 -0
on -d1 -p3 -0
clear -d1 -0
clear -p1 -0
clear -p3 -1
on -d1 -p1 -1
(NOT on -d1 -p3 -1)
(NOT clear -p1 -1)))

(OR (NOT disk_move -d1 -p3 -p2 -0)
(AND smaller -p2 -d1 -0
```

```
on-d1-p3-0
clear -d1-0
clear -p2-0
clear -p3-1
on-d1-p2-1
(NOT on-d1-p3-1)
(NOT clear -p2-1)))

(OR (NOT disk_move-d1-d2-p1-0)
(AND smaller -p1-d1-0
on-d1-d2-0
clear -d1-0
clear -p1-0
clear -d2-1
on-d1-p1-1
(NOT on-d1-d2-1)
(NOT clear -p1-1)))

(OR (NOT disk_move-d1-d2-p2-0)
(AND smaller -p2-d1-0
on-d1-d2-0
clear -d1-0
clear -p2-0
clear -d2-1
on-d1-p2-1
(NOT on-d1-d2-1)
(NOT clear -p2-1)))

(OR (NOT disk_move-d1-d2-p3-0)
(AND smaller -p3-d1-0
on-d1-d2-0
clear -d1-0
clear -p3-0
clear -d2-1
on-d1-p3-1
(NOT on-d1-d2-1)
(NOT clear -p3-1)))

(OR (NOT disk_move-d2-p1-p2-0)
(AND smaller -p2-d2-0
on-d2-p1-0
clear -d2-0
clear -p2-0
clear -p1-1
on-d2-p2-1
(NOT on-d2-p1-1)
(NOT clear -p2-1)))

(OR (NOT disk_move-d2-p1-p3-0)
(AND smaller -p3-d2-0
on-d2-p1-0
clear -d2-0
clear -p3-0
clear -p1-1
```

```
on-d2-p3-1
(NOT on-d2-p1-1)
(NOT clear-p3-1)))

(OR (NOT disk_move-d2-p2-p1-0)
(AND smaller-p1-d2-0
on-d2-p2-0
clear-d2-0
clear-p1-0
clear-p2-1
on-d2-p1-1
(NOT on-d2-p2-1)
(NOT clear-p1-1)))

(OR (NOT disk_move-d2-p2-p3-0)
(AND smaller-p3-d2-0
on-d2-p2-0
clear-d2-0
clear-p3-0
clear-p2-1
on-d2-p3-1
(NOT on-d2-p2-1)
(NOT clear-p3-1)))

(OR (NOT disk_move-d2-p3-p1-0)
(AND smaller-p1-d2-0
on-d2-p3-0
clear-d2-0
clear-p1-0
clear-p3-1
on-d2-p1-1
(NOT on-d2-p3-1)
(NOT clear-p1-1)))

(OR (NOT disk_move-d2-p3-p2-0)
(AND smaller-p2-d2-0
on-d2-p3-0
clear-d2-0
clear-p2-0
clear-p3-1
on-d2-p2-1
(NOT on-d2-p3-1)
(NOT clear-p2-1)))

(OR (NOT disk_move-d1-p1-d2-1)
(AND smaller-d2-d1-1
on-d1-p1-1
clear-d1-1
clear-d2-1
clear-p1-2
on-d1-d2-2
(NOT on-d1-p1-2)
(NOT clear-d2-2)))
```

```
(OR (NOT disk_move-d1-p1-p2-1)
(AND smaller-p2-d1-1
on-d1-p1-1
clear-d1-1
clear-p2-1
clear-p1-2
on-d1-p2-2
(NOT on-d1-p1-2)
(NOT clear-p2-2)))
```

```
(OR (NOT disk_move-d1-p1-p3-1)
(AND smaller-p3-d1-1
on-d1-p1-1
clear-d1-1
clear-p3-1
clear-p1-2
on-d1-p3-2
(NOT on-d1-p1-2)
(NOT clear-p3-2)))
```

```
(OR (NOT disk_move-d1-p2-d2-1)
(AND smaller-d2-d1-1
on-d1-p2-1
clear-d1-1
clear-d2-1
clear-p2-2
on-d1-d2-2
(NOT on-d1-p2-2)
(NOT clear-d2-2)))
```

```
(OR (NOT disk_move-d1-p2-p1-1)
(AND smaller-p1-d1-1
on-d1-p2-1
clear-d1-1
clear-p1-1
clear-p2-2
on-d1-p1-2
(NOT on-d1-p2-2)
(NOT clear-p1-2)))
```

```
(OR (NOT disk_move-d1-p2-p3-1)
(AND smaller-p3-d1-1
on-d1-p2-1
clear-d1-1
clear-p3-1
clear-p2-2
on-d1-p3-2
(NOT on-d1-p2-2)
(NOT clear-p3-2)))
```

```
(OR (NOT disk_move-d1-p3-d2-1)
(AND smaller-d2-d1-1
on-d1-p3-1
clear-d1-1
```



```
clear -d2-1
clear -p3-2
on-d1-d2-2
(NOT on-d1-p3-2)
(NOT clear -d2-2)))

(OR (NOT disk_move-d1-p3-p1-1)
(AND smaller -p1-d1-1
on-d1-p3-1
clear -d1-1
clear -p1-1
clear -p3-2
on-d1-p1-2
(NOT on-d1-p3-2)
(NOT clear -p1-2)))

(OR (NOT disk_move-d1-p3-p2-1)
(AND smaller -p2-d1-1
on-d1-p3-1
clear -d1-1
clear -p2-1
clear -p3-2
on-d1-p2-2
(NOT on-d1-p3-2)
(NOT clear -p2-2)))

(OR (NOT disk_move-d1-d2-p1-1)
(AND smaller -p1-d1-1
on-d1-d2-1
clear -d1-1
clear -p1-1
clear -d2-2
on-d1-p1-2
(NOT on-d1-d2-2)
(NOT clear -p1-2)))

(OR (NOT disk_move-d1-d2-p2-1)
(AND smaller -p2-d1-1
on-d1-d2-1
clear -d1-1
clear -p2-1
clear -d2-2
on-d1-p2-2
(NOT on-d1-d2-2)
(NOT clear -p2-2)))

(OR (NOT disk_move-d1-d2-p3-1)
(AND smaller -p3-d1-1
on-d1-d2-1
clear -d1-1
clear -p3-1
clear -d2-2
on-d1-p3-2
```

(NOT on-d1-d2-2)
(NOT clear-p3-2)))

(OR (NOT disk_move-d2-p1-p2-1)
(AND smaller-p2-d2-1
on-d2-p1-1
clear-d2-1
clear-p2-1
clear-p1-2
on-d2-p2-2
(NOT on-d2-p1-2)
(NOT clear-p2-2)))

(OR (NOT disk_move-d2-p1-p3-1)
(AND smaller-p3-d2-1
on-d2-p1-1
clear-d2-1
clear-p3-1
clear-p1-2
on-d2-p3-2
(NOT on-d2-p1-2)
(NOT clear-p3-2)))

(OR (NOT disk_move-d2-p2-p1-1)
(AND smaller-p1-d2-1
on-d2-p2-1
clear-d2-1
clear-p1-1
clear-p2-2
on-d2-p1-2
(NOT on-d2-p2-2)
(NOT clear-p1-2)))

(OR (NOT disk_move-d2-p2-p3-1)
(AND smaller-p3-d2-1
on-d2-p2-1
clear-d2-1
clear-p3-1
clear-p2-2
on-d2-p3-2
(NOT on-d2-p2-2)
(NOT clear-p3-2)))

(OR (NOT disk_move-d2-p3-p1-1)
(AND smaller-p1-d2-1
on-d2-p3-1
clear-d2-1
clear-p1-1
clear-p3-2
on-d2-p1-2
(NOT on-d2-p3-2)
(NOT clear-p1-2)))

(OR (NOT disk_move-d2-p3-p2-1)

```
(AND smaller -p2-d2-1
on-d2-p3-1
clear -d2-1
clear -p2-1
clear -p3-2
on-d2-p2-2
(NOT on-d2-p3-2)
(NOT clear -p2-2)))

(OR (NOT disk_move-d1-p1-d2-2)
(AND smaller -d2-d1-2
on-d1-p1-2
clear -d1-2
clear -d2-2
clear -p1-3
on-d1-d2-3
(NOT on-d1-p1-3)
(NOT clear -d2-3)))

(OR (NOT disk_move-d1-p1-p2-2)
(AND smaller -p2-d1-2
on-d1-p1-2
clear -d1-2
clear -p2-2
clear -p1-3
on-d1-p2-3
(NOT on-d1-p1-3)
(NOT clear -p2-3)))

(OR (NOT disk_move-d1-p1-p3-2)
(AND smaller -p3-d1-2
on-d1-p1-2
clear -d1-2
clear -p3-2
clear -p1-3
on-d1-p3-3
(NOT on-d1-p1-3)
(NOT clear -p3-3)))

(OR (NOT disk_move-d1-p2-d2-2)
(AND smaller -d2-d1-2
on-d1-p2-2
clear -d1-2
clear -d2-2
clear -p2-3
on-d1-d2-3
(NOT on-d1-p2-3)
(NOT clear -d2-3)))

(OR (NOT disk_move-d1-p2-p1-2)
(AND smaller -p1-d1-2
on-d1-p2-2
clear -d1-2
clear -p1-2
```

```
clear -p2-3
on-d1-p1-3
(NOT on-d1-p2-3)
(NOT clear -p1-3)))

(OR (NOT disk_move-d1-p2-p3-2)
(AND smaller -p3-d1-2
on-d1-p2-2
clear -d1-2
clear -p3-2
clear -p2-3
on-d1-p3-3
(NOT on-d1-p2-3)
(NOT clear -p3-3)))

(OR (NOT disk_move-d1-p3-d2-2)
(AND smaller -d2-d1-2
on-d1-p3-2
clear -d1-2
clear -d2-2
clear -p3-3
on-d1-d2-3
(NOT on-d1-p3-3)
(NOT clear -d2-3)))

(OR (NOT disk_move-d1-p3-p1-2)
(AND smaller -p1-d1-2
on-d1-p3-2
clear -d1-2
clear -p1-2
clear -p3-3
on-d1-p1-3
(NOT on-d1-p3-3)
(NOT clear -p1-3)))

(OR (NOT disk_move-d1-p3-p2-2)
(AND smaller -p2-d1-2
on-d1-p3-2
clear -d1-2
clear -p2-2
clear -p3-3
on-d1-p2-3
(NOT on-d1-p3-3)
(NOT clear -p2-3)))

(OR (NOT disk_move-d1-d2-p1-2)
(AND smaller -p1-d1-2
on-d1-d2-2
clear -d1-2
clear -p1-2
clear -d2-3
on-d1-p1-3
(NOT on-d1-d2-3)
```

(NOT clear-p1-3)))

(OR (NOT disk_move-d1-d2-p2-2)
(AND smaller-p2-d1-2
on-d1-d2-2
clear-d1-2
clear-p2-2
clear-d2-3
on-d1-p2-3
(NOT on-d1-d2-3)
(NOT clear-p2-3)))

(OR (NOT disk_move-d1-d2-p3-2)
(AND smaller-p3-d1-2
on-d1-d2-2
clear-d1-2
clear-p3-2
clear-d2-3
on-d1-p3-3
(NOT on-d1-d2-3)
(NOT clear-p3-3)))

(OR (NOT disk_move-d2-p1-p2-2)
(AND smaller-p2-d2-2
on-d2-p1-2
clear-d2-2
clear-p2-2
clear-p1-3
on-d2-p2-3
(NOT on-d2-p1-3)
(NOT clear-p2-3)))

(OR (NOT disk_move-d2-p1-p3-2)
(AND smaller-p3-d2-2
on-d2-p1-2
clear-d2-2
clear-p3-2
clear-p1-3
on-d2-p3-3
(NOT on-d2-p1-3)
(NOT clear-p3-3)))

(OR (NOT disk_move-d2-p2-p1-2)
(AND smaller-p1-d2-2
on-d2-p2-2
clear-d2-2
clear-p1-2
clear-p2-3
on-d2-p1-3
(NOT on-d2-p2-3)
(NOT clear-p1-3)))

(OR (NOT disk_move-d2-p2-p3-2)

```
(AND smaller -p3-d2-2
on-d2-p2-2
clear -d2-2
clear -p3-2
clear -p2-3
on-d2-p3-3
(NOT on-d2-p2-3)
(NOT clear -p3-3)))
```

```
(OR (NOT disk_move-d2-p3-p1-2)
(AND smaller -p1-d2-2
on-d2-p3-2
clear -d2-2
clear -p1-2
clear -p3-3
on-d2-p1-3
(NOT on-d2-p3-3)
(NOT clear -p1-3)))
```

```
(OR (NOT disk_move-d2-p3-p2-2)
(AND smaller -p2-d2-2
on-d2-p3-2
clear -d2-2
clear -p2-2
clear -p3-3
on-d2-p2-3
(NOT on-d2-p3-3)
(NOT clear -p2-3)))
```

Operator Exclusion: Presented at the last

Frame Axioms:= Presented at the last.

4. (10 pts) Find the satisfying assignments for two-disk boolean formula using your DPLL implementation.

(a) What is the satisfying assignment?

Solutions:

```
((ON-D1-P1-3) (CLEAR-D2-3) (DISK_MOVE-D1-P2-P1-2) (ON-D1-P2-3) (CLEAR-D1-1 . T)
(CLEAR-P2-3 . T) (CLEAR-P3-3) (CLEAR-D2-2 . T) (ON-D1-P3-3) (CLEAR-D1-2 . T)
(DISK_MOVE-D1-P3-P2-2) (ON-D1-P2-2 . T) (DISK_MOVE-D1-P2-D2-2 . T) (DISK_MOVE-
D1-P3-P1-2) (ON-D2-P1-3) (DISK_MOVE-D1-P3-D2-2) (ON-D2-P2-3) (DISK_MOVE-D1-P2-P3-
2) (DISK_MOVE-D2-P1-P2-2) (CLEAR-P3-2) (DISK_MOVE-D2-P2-P1-2) (ON-D2-P1-2) (ON-
D1-P3-2) (CLEAR-P1-2 . T) (ON-D1-D2-2) (DISK_MOVE-D1-D2-P3-1) (CLEAR-P2-2) (DISK_MOVE-
D1-D2-P2-1) (DISK_MOVE-D1-P2-D2-1) (CLEAR-P2-1) (DISK_MOVE-D1-P2-P3-1) (ON-D1-
D2-1) (DISK_MOVE-D1-P3-P2-1) (ON-D1-P2-1 . T) (DISK_MOVE-D1-D2-P2-0 . T) (DISK_MOVE-
D1-P3-D2-1) (ON-D1-P3-1) (DISK_MOVE-D1-D2-P3-0) (ON-D2-P2-2) (CLEAR-P3-1 . T) (DISK_MOVE-
D2-P1-P2-1) (ON-D2-P1-1 . T) (DISK_MOVE-D2-P1-P3-1 . T) (ON-D2-P3-2 . T) (DISK_MOVE-
D2-P1-P3-2) (CLEAR-D2-1 . T) (DISK_MOVE-D2-P2-P3-2) (SMALLER-P1-D1-1 . T) (SMALLER-
P1-D1-2 . T) (SMALLER-P2-D1-1 . T) (SMALLER-P2-D1-2 . T) (SMALLER-P3-D1-1 . T)
(SMALLER-P3-D1-2 . T) (SMALLER-D2-D1-1 . T) (SMALLER-D2-D1-2 . T) (SMALLER-
P1-D2-1 . T) (SMALLER-P1-D2-2 . T) (SMALLER-P2-D2-1 . T) (SMALLER-P2-D2-2 . T)
(SMALLER-P3-D2-1 . T) (SMALLER-P3-D2-2 . T) (DISK_MOVE-D1-P1-P2-2) (DISK_MOVE-
D1-P1-P3-2) (DISK_MOVE-D1-P1-D2-2) (ON-D1-P1-2) (DISK_MOVE-D1-P2-P1-1) (DISK_MOVE-
D1-P3-P1-1) (DISK_MOVE-D1-P1-P3-1) (DISK_MOVE-D1-D2-P1-1) (DISK_MOVE-D1-P1-P2-
1) (CLEAR-P1-1) (DISK_MOVE-D1-P1-D2-1) (ON-D1-P1-1) (DISK_MOVE-D1-P3-D2-0) (DISK_MOVE-
```

D1-P3-P1-0) (DISK_MOVE-D1-P2-P3-0) (DISK_MOVE-D1-P3-P2-0) (DISK_MOVE-D1-P2-P1-0) (DISK_MOVE-D1-D2-P1-0) (DISK_MOVE-D1-P2-D2-0) (DISK_MOVE-D2-P2-P1-1) (DISK_MOVE-D1-P1-P3-0) (DISK_MOVE-D2-P2-P3-1) (DISK_MOVE-D1-P1-P2-0) (ON-D2-P2-1) (DISK_MOVE-D1-P1-D2-0) (DISK_MOVE-D2-P1-P2-0) (ON-D2-D2-0) (DISK_MOVE-D2-P3-P1-1) (ON-D2-D1-0) (DISK_MOVE-D2-P3-P2-1) (ON-D1-P3-0) (ON-D2-P3-1) (ON-D1-P2-0) (DISK_MOVE-D2-P1-P3-0) (ON-D1-P1-0) (DISK_MOVE-D2-P2-P1-0) (ON-D1-D1-0) (SMALLER-D2-D2-3) (CLEAR-P1-0) (SMALLER-D2-D2-2) (CLEAR-D2-0) (SMALLER-D2-D2-1) (SMALLER-D2-D2-0) (DISK_MOVE-D2-P2-P3-0) (ON-D2-P1-0 . T) (ON-D2-P2-0) (ON-D1-D2-0 . T) (DISK_MOVE-D2-P3-P1-0) (CLEAR-P3-0 . T) (DISK_MOVE-D2-P3-P2-0) (CLEAR-P2-0 . T) (ON-D2-P3-0) (CLEAR-D1-0 . T) (DISK_MOVE-D1-D2-P1-2) (SMALLER-P3-D2-0 . T) (DISK_MOVE-D1-D2-P2-2) (SMALLER-P3-D1-0 . T) (DISK_MOVE-D1-D2-P3-2) (SMALLER-P2-D2-0 . T) (ON-D1-D2-3 . T) (SMALLER-P2-D1-0 . T) (DISK_MOVE-D2-P3-P1-2) (SMALLER-P1-D2-0 . T) (DISK_MOVE-D2-P3-P2-2) (SMALLER-P1-D1-0 . T) (ON-D2-P3-3 . T) (SMALLER-D2-D1-0 . T))

(b) What is the corresponding plan?

Solutions:

(DISK_MOVE-D1-D2-P2-0 . T)
(DISK_MOVE-D2-P1-P3-1 . T)
(DISK_MOVE-D1-P2-D2-2 . T)

5. **Extra Credit:** Compare the performance/scalability of your DPLL implementation with one or more state-of-the-art SMT solvers such as:

- Z3: <https://github.com/Z3Prover/z3>
- CVC4: <http://cvc4.cs.stanford.edu/web/>

(Note: you might find the Lisp TIME macro and SBCL's statistical profiler (<http://www.sbcl.org/manual/#Statistical-Profiler>) useful to evaluate performance).

Solution: We evaluated the performance of our DPLL implementation with the z3 solver. The output from the z3 solver is presented as follows:

```
sat
model validated
1 2 -3 4 -5 -6 7 8 -9 -10 -11 12 13 14 -15 -16 -17 18 -19
-20 -21 -22 -23 24 -25 -26 -27 -28 29 30 -31 -32 -33 34 -35
36 37 38 39 40 41 42 43 44 45 -46 -47 -48 49 50 -51 52 53 -54
-55 56 57 58 -59 60 61 -62 63 -64 -65 -66 -67 68 69 70 71

(:binary-propagations 54
:conflicts 2
:decisions 15
:mk-binary-clause 180
:mk-bool-var 72
:mk-clause 5
:mk-ternary-clause 3
:propagations 5
:ternary-propagations 5
:total-time 0.0003
:units 13)
```

Figure 4: z3 solver on the two disc problem

6. **Extra Credit:** Optimize your DPLL implementation. For example, you could improve the implementation of DPLL-CHOOSE-LITERAL. Discuss the optimizations you implement and characterize the speedup (for example, using TIME or SBCL's statistical profiler).

Solutions:

Original:

Evaluation took:

70.815 seconds of real time

70.801857 seconds of total run time (70.673627 user, 0.128230 system)

[Run times consist of 0.611 seconds GC time, and 70.191 seconds non-GC time.]

99.98% CPU

169,952,795,622 processor cycles

7,375,912,032 bytes consed

Optimal:

Evaluation took:

0.054 seconds of real time

0.053120 seconds of total run time (0.053117 user, 0.000003 system)

98.15% CPU

127,814,358 processor cycles

8,975,968 bytes consed

We tried to modify the DPLL-CHOOSE-LITERAL to optimize our DPLL implementation. But we were not able to get optimal results. Instead we got an increase in execution time as shown above.

disk_move-d1-p2-p1-2) (NOT disk_move-d1-p2-p3-2) (NOT disk_move-d1-p3-d2-2) (NOT
disk_move-d1-p3-p1-2) (NOT disk_move-d1-p3-p2-2) (NOT disk_move-d2-p1-p2-2) (NOT
disk_move-d2-p1-p3-2) (NOT disk_move-d2-p2-p1-2) (NOT disk_move-d2-p2-p3-2) (NOT
disk_move-d2-p3-p1-2)))

Frame Axioms

(OR clear-d1-0 (NOT clear-d1-1))
(OR (OR clear-d2-0 (NOT clear-d2-1)) disk_move-d1-p1-d2-0 disk_move-d1-p2-d2-0
disk_move-d1-p3-d2-0 disk_move-d1-d2-p1-0 disk_move-d1-d2-p2-0 disk_move-d1-d2-p3-0)
(OR (OR clear-p1-0 (NOT clear-p1-1)) disk_move-d1-d2-p1-0 disk_move-d1-p2-p1-0
disk_move-d1-p3-p1-0 disk_move-d2-p2-p1-0 disk_move-d2-p3-p1-0 disk_move-d1-p1-d2-0
disk_move-d1-p1-p2-0 disk_move-d1-p1-p3-0 disk_move-d2-p1-p2-0 disk_move-d2-p1-p3-0)
(OR (OR clear-p2-0 (NOT clear-p2-1)) disk_move-d1-d2-p2-0 disk_move-d1-p1-p2-0
disk_move-d1-p3-p2-0 disk_move-d2-p1-p2-0 disk_move-d2-p3-p2-0 disk_move-d1-p2-d2-0
disk_move-d1-p2-p1-0 disk_move-d1-p2-p3-0 disk_move-d2-p2-p1-0 disk_move-d2-p2-p3-0)
(OR (OR clear-p3-0 (NOT clear-p3-1)) disk_move-d1-d2-p3-0 disk_move-d1-p1-p3-0
disk_move-d1-p2-p3-0 disk_move-d2-p1-p3-0 disk_move-d2-p2-p3-0 disk_move-d1-p3-d2-0
disk_move-d1-p3-p1-0 disk_move-d1-p3-p2-0 disk_move-d2-p3-p1-0 disk_move-d2-p3-p2-0)

(OR (OR on-d1-p1-0 (NOT on-d1-p1-1)) disk_move-d1-p1-d2-0 disk_move-d1-p1-p2-0
disk_move-d1-p1-p3-0 disk_move-d1-d2-p1-0 disk_move-d1-p2-p1-0 disk_move-d1-p3-p1-0)
(OR (OR on-d1-p2-0 (NOT on-d1-p2-1)) disk_move-d1-p2-d2-0 disk_move-d1-p2-p1-0
disk_move-d1-p2-p3-0 disk_move-d1-d2-p2-0 disk_move-d1-p1-p2-0 disk_move-d1-p3-p2-0)
(OR (OR on-d1-p3-0 (NOT on-d1-p3-1)) disk_move-d1-p3-d2-0 disk_move-d1-p3-p1-0
disk_move-d1-p3-p2-0 disk_move-d1-d2-p3-0 disk_move-d1-p1-p3-0 disk_move-d1-p2-p3-0)
(OR (OR on-d1-d2-0 (NOT on-d1-d2-1)) disk_move-d1-d2-p1-0 disk_move-d1-d2-p2-0
disk_move-d1-d2-p3-0 disk_move-d1-p1-d2-0 disk_move-d1-p2-d2-0 disk_move-d1-p3-d2-0)
(OR (OR on-d2-p1-0 (NOT on-d2-p1-1)) disk_move-d2-p1-p2-0 disk_move-d2-p1-p3-0
disk_move-d2-p2-p1-0 disk_move-d2-p3-p1-0)
(OR (OR on-d2-p2-0 (NOT on-d2-p2-1)) disk_move-d2-p2-p1-0 disk_move-d2-p2-p3-0
disk_move-d2-p1-p2-0 disk_move-d2-p3-p2-0)
(OR (OR on-d2-p3-0 (NOT on-d2-p3-1)) disk_move-d2-p3-p1-0 disk_move-d2-p3-p2-0
disk_move-d2-p1-p3-0 disk_move-d2-p2-p3-0)

(OR smaller-p1-d1-0 (NOT smaller-p1-d1-1))
(OR smaller-p2-d1-0 (NOT smaller-p2-d1-1))
(OR smaller-p3-d1-0 (NOT smaller-p3-d1-1))
(OR smaller-d2-d1-0 (NOT smaller-d2-d1-1))
(OR smaller-p1-d2-0 (NOT smaller-p1-d2-1))
(OR smaller-p2-d2-0 (NOT smaller-p2-d2-1))
(OR smaller-p3-d2-0 (NOT smaller-p3-d2-1))
(OR smaller-d2-d2-0 (NOT smaller-d2-d2-1))

(OR clear-d1-1 (NOT clear-d1-2))
(OR (OR clear-d2-1 (NOT clear-d2-2)) disk_move-d1-p1-d2-1 disk_move-d1-p2-d2-1
disk_move-d1-p3-d2-1 disk_move-d1-d2-p1-1 disk_move-d1-d2-p2-1 disk_move-d1-d2-p3-1)

(OR (OR clear-p1-1 (NOT clear-p1-2)) disk_move-d1-d2-p1-1 disk_move-d1-p2-p1-1
disk_move-d1-p3-p1-1 disk_move-d2-p2-p1-1 disk_move-d2-p3-p1-1 disk_move-d1-p1-d2-1
disk_move-d1-p1-p2-1 disk_move-d1-p1-p3-1 disk_move-d2-p1-p2-1 disk_move-d2-p1-p3-1)
(OR (OR clear-p2-1 (NOT clear-p2-2)) disk_move-d1-d2-p2-1 disk_move-d1-p1-p2-1
disk_move-d1-p3-p2-1 disk_move-d2-p1-p2-1 disk_move-d2-p3-p2-1 disk_move-d1-p2-d2-1
disk_move-d1-p2-p1-1 disk_move-d1-p2-p3-1 disk_move-d2-p2-p1-1 disk_move-d2-p2-p3-1)
(OR (OR clear-p3-1 (NOT clear-p3-2)) disk_move-d1-d2-p3-1 disk_move-d1-p1-p3-1
disk_move-d1-p2-p3-1 disk_move-d2-p1-p3-1 disk_move-d2-p2-p3-1 disk_move-d1-p3-d2-1
disk_move-d1-p3-p1-1 disk_move-d1-p3-p2-1 disk_move-d2-p3-p1-1 disk_move-d2-p3-p2-1)

(OR (OR on-d1-p1-1 (NOT on-d1-p1-2)) disk_move-d1-p1-d2-1 disk_move-d1-p1-p2-1
disk_move-d1-p1-p3-1 disk_move-d1-d2-p1-1 disk_move-d1-p2-p1-1 disk_move-d1-p3-p1-1)
(OR (OR on-d1-p2-1 (NOT on-d1-p2-2)) disk_move-d1-p2-d2-1 disk_move-d1-p2-p1-1
disk_move-d1-p2-p3-1 disk_move-d1-d2-p2-1 disk_move-d1-p1-p2-1 disk_move-d1-p3-p2-1)
(OR (OR on-d1-p3-1 (NOT on-d1-p3-2)) disk_move-d1-p3-d2-1 disk_move-d1-p3-p1-1
disk_move-d1-p3-p2-1 disk_move-d1-d2-p3-1 disk_move-d1-p1-p3-1 disk_move-d1-p2-p3-1)
(OR (OR on-d1-d2-1 (NOT on-d1-d2-2)) disk_move-d1-d2-p1-1 disk_move-d1-d2-p2-1
disk_move-d1-d2-p3-1 disk_move-d1-p1-d2-1 disk_move-d1-p2-d2-1 disk_move-d1-p3-d2-1)
(OR (OR on-d2-p1-1 (NOT on-d2-p1-2)) disk_move-d2-p1-p2-1 disk_move-d2-p1-p3-1
disk_move-d2-p2-p1-1 disk_move-d2-p3-p1-1)
(OR (OR on-d2-p2-1 (NOT on-d2-p2-2)) disk_move-d2-p2-p1-1 disk_move-d2-p2-p3-1
disk_move-d2-p1-p2-1 disk_move-d2-p3-p2-1)
(OR (OR on-d2-p3-1 (NOT on-d2-p3-2)) disk_move-d2-p3-p1-1 disk_move-d2-p3-p2-1
disk_move-d2-p1-p3-1 disk_move-d2-p2-p3-1)

(OR smaller-p1-d1-1 (NOT smaller-p1-d1-2))
(OR smaller-p2-d1-1 (NOT smaller-p2-d1-2))
(OR smaller-p3-d1-1 (NOT smaller-p3-d1-2))
(OR smaller-d2-d1-1 (NOT smaller-d2-d1-2))
(OR smaller-p1-d2-1 (NOT smaller-p1-d2-2))
(OR smaller-p2-d2-1 (NOT smaller-p2-d2-2))
(OR smaller-p3-d2-1 (NOT smaller-p3-d2-2))
(OR smaller-d2-d2-1 (NOT smaller-d2-d2-2))

(OR clear-d1-2 (NOT clear-d1-3))
(OR (OR clear-d2-2 (NOT clear-d2-3)) disk_move-d1-p1-d2-2 disk_move-d1-p2-d2-2
disk_move-d1-p3-d2-2 disk_move-d1-d2-p1-2 disk_move-d1-d2-p2-2 disk_move-d1-d2-p3-2)
(OR (OR clear-p1-2 (NOT clear-p1-3)) disk_move-d1-d2-p1-2 disk_move-d1-p2-p1-2
disk_move-d1-p3-p1-2 disk_move-d2-p2-p1-2 disk_move-d2-p3-p1-2 disk_move-d1-p1-d2-2
disk_move-d1-p1-p2-2 disk_move-d1-p1-p3-2 disk_move-d2-p1-p2-2 disk_move-d2-p1-p3-2)
(OR (OR clear-p2-2 (NOT clear-p2-3)) disk_move-d1-d2-p2-2 disk_move-d1-p1-p2-2
disk_move-d1-p3-p2-2 disk_move-d2-p1-p2-2 disk_move-d2-p3-p2-2 disk_move-d1-p2-d2-2
disk_move-d1-p2-p1-2 disk_move-d1-p2-p3-2 disk_move-d2-p2-p1-2 disk_move-d2-p2-p3-2)
(OR (OR clear-p3-2 (NOT clear-p3-3)) disk_move-d1-d2-p3-2 disk_move-d1-p1-p3-2
disk_move-d1-p2-p3-2 disk_move-d2-p1-p3-2 disk_move-d2-p2-p3-2 disk_move-d1-p3-d2-2
disk_move-d1-p3-p1-2 disk_move-d1-p3-p2-2 disk_move-d2-p3-p1-2 disk_move-d2-p3-p2-2)

(OR (OR on-d1-p1-2 (NOT on-d1-p1-3)) disk_move-d1-p1-d2-2 disk_move-d1-p1-p2-2
disk_move-d1-p1-p3-2 disk_move-d1-d2-p1-2 disk_move-d1-p2-p1-2 disk_move-d1-p3-p1-2)
(OR (OR on-d1-p2-2 (NOT on-d1-p2-3)) disk_move-d1-p2-d2-2 disk_move-d1-p2-p1-2
disk_move-d1-p2-p3-2 disk_move-d1-d2-p2-2 disk_move-d1-p1-p2-2 disk_move-d1-p3-p2-2)
(OR (OR on-d1-p3-2 (NOT on-d1-p3-3)) disk_move-d1-p3-d2-2 disk_move-d1-p3-p1-2
disk_move-d1-p3-p2-2 disk_move-d1-d2-p3-2 disk_move-d1-p1-p3-2 disk_move-d1-p2-p3-2)
(OR (OR on-d1-d2-2 (NOT on-d1-d2-3)) disk_move-d1-d2-p1-2 disk_move-d1-d2-p2-2
disk_move-d1-d2-p3-2 disk_move-d1-p1-d2-2 disk_move-d1-p2-d2-2 disk_move-d1-p3-d2-2)
(OR (OR on-d2-p1-2 (NOT on-d2-p1-3)) disk_move-d2-p1-p2-2 disk_move-d2-p1-p3-2
disk_move-d2-p2-p1-2 disk_move-d2-p3-p1-2)
(OR (OR on-d2-p2-2 (NOT on-d2-p2-3)) disk_move-d2-p2-p1-2 disk_move-d2-p2-p3-2
disk_move-d2-p1-p2-2 disk_move-d2-p3-p2-2)
(OR (OR on-d2-p3-2 (NOT on-d2-p3-3)) disk_move-d2-p3-p1-2 disk_move-d2-p3-p2-2
disk_move-d2-p1-p3-2 disk_move-d2-p2-p3-2)

(OR smaller-p1-d1-2 (NOT smaller-p1-d1-3))
(OR smaller-p2-d1-2 (NOT smaller-p2-d1-3))
(OR smaller-p3-d1-2 (NOT smaller-p3-d1-3))
(OR smaller-d2-d1-2 (NOT smaller-d2-d1-3))
(OR smaller-p1-d2-2 (NOT smaller-p1-d2-3))
(OR smaller-p2-d2-2 (NOT smaller-p2-d2-3))
(OR smaller-p3-d2-2 (NOT smaller-p3-d2-3))
(OR smaller-d2-d2-2 (NOT smaller-d2-d2-3))

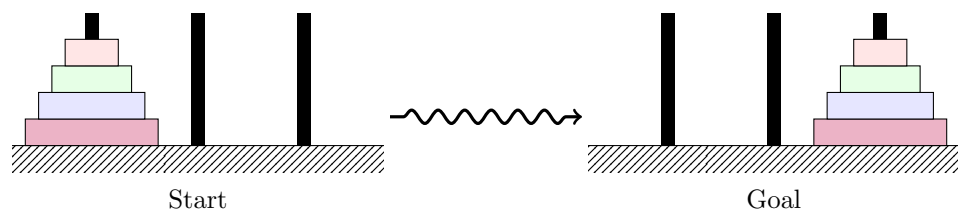


Figure 5: Tower of Hanoi Puzzle with 3 pegs and 4 disks.