Helena 2.3
Example 3 - The towers of Hanoi

Sami Evangelista - (Sami [dot] Evangelista [at] lipn.univ-paris13 [dot] fr)

March 28, 2013

The towers of Hanoi is a well-known mathematical game[1]. It consists of three towers, and a number of disks of different sizes which can slide onto any tower. The puzzle starts with the disks neatly stacked in order of size on one tower, smallest at the top, thus making a conical shape.

The objective of the game is to move the entire stack to another tower, obeying the following rules:

- Only one disk may be moved at a time.
- Each move consists of taking the upper disk from one of the towers and sliding it onto another tower, on top of the other disks that may already be present on that tower.
- No disk may be placed on top of a smaller disk.

This example illustrates the use of lists in Helena.

Listing 1: Helena file of the towers of Hanoi (file `examples/hanoi.lna`)

```
1  /*************************************************************************
2  *
3  *  Example file of Helena distribution
4  *
5  *  File   : hanoi.lna
6  *  Author: Sami Evangelista
7  *  Date   : 15 feb. 2007
8  *
9  *  This file contains the description of the towers of Hanoi game.
10 *
11 *************************************************************************/
12
13 hanoi (N := 3,      /*  N = the number of disks   */
14        M := 3)  {   /*  M = number of towers   */
15
16    //  identifier of a disk
17    type disk: range 1..N;
18
19    //  identifier of a tower
20    type tower: range 1 .. M;
21
22    //  a list of disks
23    type disk_list: list[nat] of disk with capacity N;
24
25    //  construct the list of disks initially present on the first tower
26    function construct_tower1() -> disk_list {
27       disk_list result := empty;
28       for(i in disk)
29          result := i & result;
30       return result;
31    }
```

[1] The description is taken from http://en.wikipedia.org/wiki/Tower_of_Hanoi.

```
32
33    //  this unique place models the state of the towers.
34    //  in the initial marking the first tower contains the list |N, ..., 1|
35    //  and all others are empty
36    place towers {
37       dom : tower * disk_list;
38       init: <( tower'first, construct_tower1() )>
39           + for(t in tower range tower'first + 1 .. tower'last)
40                <( t, empty )>;
41    }
42
43    //  transition move_disk models the move of the disk on top
44    //  of tower src to the tower dest.  the src stack must not be
45    //  (not src_disks'empty).  if the dest tower is not empty, the
46    //  disk on top of src (src_disks'last) must be smaller than the disk
47    //  on top of the dest tower (dest_disks'last).
48    //  the move consists of deleting the last element from the src stack
49    //  (src_disks'prefix is the the list src_disks from which we remove
50    //  the last element) and pushing it onto the dest stack.
51    transition move_disk {
52       in  {
53          towers: <( src,  src_disks  )>
54                + <( dest, dest_disks )>;
55       }
56       out {
57          towers: //  remove the last disk of tower t
58                  <( src,  src_disks'prefix )>
59
60                  //  add the removed disk on top of tower u
61                + <( dest, dest_disks & src_disks'last )>;
62       }
63       guard: not src_disks'empty
64              and (dest_disks'empty or src_disks'last < dest_disks'last);
65       description: "move␣disk␣%d␣from␣tower␣%d␣to␣tower␣%d",
66                    src_disks'last, src, dest;
67    }
68
69    //  in the end state the last tower is full, i.e., it contains the list
70    //  |N, ..., 1|
71    proposition all_moved:
72       exists(t in towers | t->1 = tower'last and t->2'full);
73 }
```

Listing 2: Helena file of the towers of Hanoi properties (file `examples/hanoi.prop.lna`)

```
1  //  we reach a state in which all disks have been moved
2  //  to the last tower
3  state property end_state:
4     reject all_moved;
```