

TP6 - Machine de Turing

PAUL CHAIGNON - CLÉMENT GAUTRAIS

November 19, 2013

1 Simulation d'une machine de Turing en Prolog

1.1 Questions

Listing 1: tp6_turing.pro

```
1 /**
2  * Question 1
3  * next(+Program, +State0, +Symbol0, -Symbol1, -Dir, -State1)
4  */
5 next(program(Start, Finals, Transitions), State0, Symbol0, Symbol1,
6     Dir, State1):-
7     match_next_step(Transitions, State0, Symbol0, Symbol1, Dir, State1).
8 /**
9  * match_next_step(+Transitions, +State0, +Symbol0, -Symbol1, -Dir,
10     -State1)
11 */
12 match_next_step([delta(State0, Symbol0, Symbol1, Dir, State1)|L],
13     State0, Symbol0, Symbol1, Dir, State1).
14 match_next_step([delta(StateInit, SymbolTete, _, _, _)|L], State0,
15     Symbol0, Symbol1, Dir, State1):-
16     match_next_step(L, State0, Symbol0, Symbol1, Dir, State1).
17 /**
18  * Question 2
19  * update_tape(+Tape, +Symbol, +Direction, -UpdatedTape)
20  */
21 update_tape(tape(Left, [Head|Right]), Symbol, left, tape(NewLeft,
22     [NewHead, Symbol|Right])):-
23     last_elem(Left, NewHead),
24     add_blank_left(Left, Left2),
25     remove_last(Left2, NewLeft).
26 update_tape(tape(Left, [Head, NewHead|Right]), Symbol, right,
27     tape(NewLeft, [NewHead|Right])):-
28     insert_last(Symbol, Left, NewLeft).
29 update_tape(tape(Left, [Head]), Symbol, right, tape(NewLeft, [' ', Symbol])):-
30     insert_last(Symbol, Left, NewLeft).
31 /**
32  * last_elem(+List, -NewList)
33  */
34 last_elem([Last], Last).
```

```

33 last_elem([First|List], Last):-
34     last_elem(List, Last).
35
36 /**
37  * remove_last(+List, -NewList)
38  */
39 remove_last([Last], []).
40 remove_last([First|List], [First|NewList]):-
41     remove_last(List, NewList).
42
43 /**
44  * insert_last(+LastElem, +List, -NewList)
45  */
46 insert_last(Last, [], [Last]).
47 insert_last(Last, [First|List], [First|NewList]):-
48     insert_last(Last, List, NewList).
49
50 /**
51  * add_blank_left(+List, -NewList)
52  * We can't have an empty list.
53  */
54 add_blank_left([Last], [' ', Last]).
55 add_blank_left([First, Second|List], [First, Second|List]).
56
57
58 /**
59  * Question 4
60  * run_turing_machine(+Program, +Input, -Output, -FinalState, +Filename)
61  */
62 run_turing_machine(program(Depart, Finals, Trans), BandDroite, Output,
63     FinalState, Filename):-
64     run_turing_machine_tape(program(Depart, Finals, Trans), Depart,
65         tape([' ', BandDroite], tape(Left, Right), FinalState, Dump),
66         concat(Left, Right, Output),
67         dump_to_mpost(Filename, Dump)).
68
69 /**
70  * run_turing_machine_tape(+Program, +CurrentState, +Input, -Output,
71     -FinalState, -Dump)
72  */
73 run_turing_machine_tape(program(Depart, Finals, Trans), CurrentState,
74     Input, Input, CurrentState, [(CurrentState, Input)]):-
75     membre(CurrentState, Finals).
76 run_turing_machine_tape(Prog, CurrentState, tape(Left, [Tete|Right]),
77     Output, FinalState, [(CurrentState, tape(Left,
78         [Tete|Right]))|Dump]):-
79     next(Prog, CurrentState, Tete, NewTete, Direct, NextState),
80     update_tape(tape(Left, [Tete|Right]), NewTete, Direct, UpdatedTape),
81     run_turing_machine_tape(Prog, NextState, UpdatedTape, Output,
82         FinalState, Dump).
83
84 /**
85  * membre(?A, +X)
86  */
87 membre(A, [A|R]).
88 membre(A, [X|R]):-
89     membre(A, R).

```

```

84 /**
85  * concat(+X, +Y, ?T)
86  */
87 concat([], Y, Y).
88 concat([P|R], Y, [P|T]) :-
89     concat(R, Y, T).

```

1.2 Tests

Listing 2: tp6_turing_tests.pro

```

1 copy_prog(Prog), next(Prog, start, 1, X, Y, Z).
2   X = ' ',
3   Y = right
4   Z = s2
5
6 update_tape(tape([' ', ''], [1, ' ', '']), ' ', right, UpdatedTape).
7   UpdatedTape = tape([' ', ' ', ' ', ''], [' ', '']).
8
9 copy_prog(Prog), run_turing_machine(Prog, [1], Output, FinalState).

```