state (s7).

```
%% Get the translated interpreter:
:- [ 'verifier.tlp' ].
\mbox{\$---} An example: the microwave oven, from:
      Edmund M. Clarke, Jr., Orna Grumberg, and Doron A. Peled:
응
응
      "Model Checking"
응
      The MIT Press (Cambridge, Massachussets and London, England) 1999
응
                                 S1
응
응
응
응
응
응
응
응
응
응
응
                     S2
                                 S3
                                           - S4
응
응
응
응
응
응
응
응
응
                      S5
                                S6 ---->S7
proposition( start ).
proposition(close).
proposition( heat ).
proposition( error ).
state (s1).
state (s2).
state (s3).
state(s4).
state(s5).
state(s6).
```

```
trans(s1, s2). % start oven
trans( s1, s3 ).
                 % close door
trans( s2, s5 ).
                 % close door
trans( s3, s1 ).
                 % open door
trans( s3, s6 ).
                 % start oven
trans( s4, s1 ).
                  % open door
trans( s4, s3 ).
                  % done
trans( s4, s4 ).
                 % cook
trans (s5, s2).
               % open door
trans( s5, s3 ).
                % reset
trans( s6, s7 ).
               % warm up
trans(s7,s4). % start cooking
holds( s2, start ). holds( s2, error ).
holds (s3, close).
holds(s4, close). holds(s4, heat).
holds( s5, start ). holds( s5, close ). holds( s5, error ).
holds( s6, start ). holds( s6, close ).
holds(s7, start). holds(s7, close). holds(s7, heat).
:- top q1/0, q2/0, q3/0, q4/0.
                                        Expected
                                                  Prolog
                                                            Tabling
q1 :- check(s1, g(^{\sim} heat u close)).
                                      % yes
                                                   loops
                                                            yes
q2 :- check(s1, ~g(~heat u close)).
                                       % no
                                                   no
                                                            no
q3 :- check(s1, f(close)).
                                       % yes
                                                   yes
                                                            yes
q4 :- check( s1, f( error ) ).
                                       % no
                                                   no
                                                            no
```