LTL Control in Uncertain Environments with Probabilistic Satisfaction Guarantees

Xu Chu (Dennis) Ding, Stephen L. Smith, Calin Belta, and Daniela Rus

Hybrid and Networked Systems Laboratory, Boston University {xcding(at)bu.edu, cbelta(at)bu.edu}  
Department of Electrical and Computer Engineering, University of Waterloo {stephen.smith@uwaterloo.ca}  
Distributed Robotics Laboratory, Massachusetts Institute of Technology {rus@csail.mit.edu}

The MSL (Maximal Satisfiability LTL) toolbox automatically generates an optimal control policy for a Markov Decision Process (MDP) that maximizes the probability of satisfying a Linear Temporal Logic (LTL) formula. For more detail of the algorithm that produces the policy, please see [[1]](https://sites.bu.edu/hyness/maximal_sat-ltl/#1). Note in [[1]](https://sites.bu.edu/hyness/maximal_sat-ltl/#1) we model the system with probabilistic observations and convert the system model to an MDP, we then use this software to generate the policy. This software calls open source software [[2]](https://sites.bu.edu/hyness/maximal_sat-ltl/#2) and [[3]](https://sites.bu.edu/hyness/maximal_sat-ltl/#3).

**Requirements**

– Mac OS X 10.6, Linux and Windows (Linux and Windows should work, but let me know if there is any issue)  
– MATLAB 2010b or later with Bioinformatics toolbox (for SCC computation)

**Download**

[[MSL package](https://sites.bu.edu/hyness/files/2014/07/Maximal_Sat-LTL_code.zip)](https://calinbelta.com/wp-content/uploads/2024/01/Maximal_Sat-LTL_code.zip)

**How to use**

**[optProb\_init, maxProb, policy, policyForAMECstates] = LTLmaxProb(MDPdata, formula)**

Input:

* MDPdata: string containing the name of file where the MDP data is stored. Exepcted data: struct M with fields:
  + M.P: cell array of sparse transition matrices, one for each action (matrices must be same dimension, which is state-size times state-size)
  + M.Q0: initial state
  + Rest of fields are for observation labels. E.g., if there are two observations A and B, then M.A is the vector of states with observation A and M.B is the vector of states with observation B. A state can have multiple observations
* formula: string of LTL formula in prefix format. See [here](http://www.ltl2dstar.de/docs/ltl2dstar.html#ltl-formulas) for syntax. Note: The set of atomic propositions in the formula must be the same set as the set of observations of the MDP

Output:

* optProb\_init: maximal probablity of satisfying the LTL formula from the initial state
* maxProb: the matrix containing maximal probabilities of satisfying the LTL formula. **maxProb(Mstate, Rstate)** is the maximal probability of satisfying the formula if the MDP state is Mstate and the DRA state is Rstate
* policy: the matrix containing optimal policies maximizing the probability of satisfying the LTL formula. **policy(Mstate, Rstate)** is the optimal action to take if the MDP state is Mstate and the DRA state is Rstate. Note: if policy(Mstate, Rstate)=0, then it means that a random action with a subset of enabled action can be take at this state (reaching this state means that an AMEC is reached), this subset is stored in policyForAMECstates (see below)
* policyForAMECstates: Stores the set of actions that can be taken randomly at an AMEC state under the optimal policy. If **policy(Mstate, Rstate)==0**, then **policyForAMECstates{Mstate, Rstate}** returns a subset of enabled action at MDP state Mstate (if DRA state is Rstate) that should be taken randomly under the optimal action. Note: To get the set of such states in some AMEC, simply do: **[Mstates, Rstates]=find(policy==0)**. To get the subset of enabled action at these states under the optimal policy, do: **getCells(policyForAMECstates, find(policy==0))**

**Example**

We show a test case as an example.

A computer screen with green text

Description automatically generated

This MDP is stored in **MDP.mat**. This MDP has 608 states, 4 actions and 5 observations “UP”, “UN”, “RI”, “VD” and “RD”. We want to satisfy the LTL formula



We convert the formula to prefix format and run the software, as below. In this case the Deterministic Rabin Automaton has 17 states

A computer screen with green text

Description automatically generated

This example is included in the software package.

**References**

[1] Xu Chu (Dennis) Ding, Stephen L. Smith, Calin Belta, and Daniela. “LTL Control in Uncertain Environments with Probabilistic Satisfaction Guarantees”, IFAC World Congress, Milan, Italy, Aug. 2011 (see technical report [here](http://arxiv.org/abs/1104.1159)).

[2] “LTL2DSTAR”, <http://www.ltl2dstar.de/>.

[3] “LTL2BA,” [http://www.lsv.ens-cachan.fr/˜gastin/ltl2ba/index.php](http://www.lsv.ens-cachan.fr/~gastin/ltl2ba/index.php).