Time Window Temporal Logic (TWTL)

Overview

**T**ime **W**indow **T**emporal **L**ogic (TWTL) is a bounded temporal logic used to specify rich properties[[1]](https://sites.bu.edu/hyness/twtl/#1). Relaxed versions of TWTL formulae are also considered in the sense of extending the deadlines of time windows. An automata based approach is proposed to solve synthesis, verification and learning problems. The key ingredient is a construction algorithm of annotated Deterministic Finite State Automata (DFA) from TWTL properties. See [[1]](https://sites.bu.edu/hyness/twtl/#1) for more details.

PyTWTL is a Python 2.7 implementation of the algorithms proposed in [[1]](https://sites.bu.edu/hyness/twtl/#1) based on LOMAP [[2]](https://sites.bu.edu/hyness/twtl/#2), ANTLRv3 [[3]](https://sites.bu.edu/hyness/twtl/#3) and networkx [[4]](https://sites.bu.edu/hyness/twtl/#4) libraries. PyTWTL implementation is released under the GPLv3 license.

The library can be used to:

* construct DFAs and annotated DFAs from TWTL formulae;
* monitor the satisfaction of a TWTL formula;
* monitor the satisfaction of an arbitrary relaxation of a TWTL formula;
* compute the temporal relaxation of a trace with respect to a TWTL formula;
* compute a satisfying control policy with respect to a TWTL formula;
* compute a minimally relaxed control policy with respect to a TWTL formula;
* verify if all traces of a system satisfy some relaxed version of a TWTL formula;
* learn the parameters of a TWTL formula, i.e. the deadlines.

The parsing of TWTL formulae is performed using ANTLRv3 framework. The package provides grammar files which may be used to generate lexers and parsers for other programming languages such as Java, C/C++, Ruby. To support Python 2.7, we used version 3.1.3 of ANTLRv3 and the corresponding Python runtime ANTLR library, which we included in our distribution for convenience.

Citation

If you use TWTL or PyTWTL, then please consider citing the reference paper:

Cristian-Ioan Vasile, Derya Aksaray, and Calin Belta. *“Time Window Temporal Logic”*, arXiv preprint, [arXiv:1602.04294](http://arxiv.org/abs/1602.04294), 2016.

Download

[[Download](https://sites.bu.edu/hyness/files/2016/02/pytwtl.zip)](https://calinbelta.com/wp-content/uploads/2024/01/pytwtl.zip)

Requirements

The package is written for python 2.7. The following python packages are required:

* NumPy
* NetworkX
* ParallelPython
* matplotlib
* setuptools
* ANTLRv3 python runtime

You can install the packages using:
pip install networkx, numpy, matplotlib, pp, antlr-python-runtime, setuptools

How to Use

See examples\_tcs.py for examples of the algorithms and the PyTWTL API.
An ANT build file build.xml is provided to generate the lexer and parser from the ANTLR3 grammar files.

License & Copying

Copyright (C) 2015-2016 Cristian Ioan Vasile

Hybrid and Networked Systems (HyNeSs) Group, BU Robotics Lab,

Boston University

This program is free software: you can redistribute it and/or modify

it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

GNU General Public License for more details.

You should have received a copy of the GNU General Public License

along with this program. If not, see <http://www.gnu.org/licenses/>.

A copy of the GNU General Public License is included in the source folder in a file called ‘gpl.txt’.

TWTL uses the ANTLR (version 3.1.3) third party Java library to generate the lexers and parsers. It is included for convenience in the ‘lib’ folder of the distribution source tree. The library can be downloaded from <https://github.com/antlr/website-antlr3/tree/gh-pages/download>.
See the file names ‘license.txt’ for copyright notices and license information of packages used in PyTWTL.

References

[1] Cristian-Ioan Vasile, Derya Aksaray, and Calin Belta. *“Time Window Temporal Logic.”* arXiv preprint, [arXiv:1602.04294](http://arxiv.org/abs/1602.04294), 2016.

[2] Alphan Ulusoy, Stephen L. Smith, Xu Chu Ding, Calin Belta, and Daniela Rus. *“Optimality and robustness in multi-robot path planning with temporal logic constraints.”* The International Journal of Robotics Research 32, no. 8 (2013): 889-911.

[3] Terence Parr. *“The Definitive ANTLR Reference: Building Domain-Specific Languages.”* Pragmatic Bookshelf, 2007. ISBN 978-0978739256.

[4] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. *“Exploring network structure, dynamics, and function using NetworkX.”* 2008.