A guide to the literature on handling dynamic systems with ASP and alike

KRR@UP

May 20, 2022

README

The following aims at giving a survey of the literature.

The idea is that first level references constitute common literature that anyone working on the subject should be aware of. Accordingly, next levels go more and more in detail and become thus less relevant for people not on the very same subject.

Notes can be posted in the folder `notes` using the bib-key as the file name (plus `.tex`). The inclusion of notes can be switched on and off in the LaTeX source.

1 Linear temporal logic

1. **degola16a**

2 Linear temporal logic over finite traces

1. **giavar13a**

   (a) **gimamo14a**; **giavar15a**; **giavar16a**

3 Automata for temporal logic

1. **degola16a**

3.1 Linear Temporal logic

3.1.1 Infinite traces

1. **vardi95a**; **vardi97a**

   Susana This old paper explains in more detail concepts used in the newer papers by Vardi for Automata and temporal logics.

   (a) **helnie03a**

Susana Does bounded LTL model checking by computing stable models with a Petri net in ASP using Smodels.

(b) **pursah21a**

Susana Uses an SMT-based algorithm and Büchi Automaton for the *LTL path planning problem*, considering obstacles that are dynamically introduced. Agents must complete a task in a cycle as many times as possible.

(c) **bogigeparu20a**

Susana Synthesizes a finite-state controller from a LTL specification, which is then translated it into a Python program.

3.1.2 Finite traces

1. **giavar13a**

   Susana They define a finite version of LDL and LTL, namely LDL_f and LTL_f, and show their translation to Alternating Automata.

   Susana Even though it is not intuitive, the set of final states of the AWF in the translation is really $\emptyset$. Note that when *last* is in the trace, then all transitions are *true* or *false*, so there will be no states in the last level of the tree (run).

   Susana There is an error in the transition function of LDL_f to AFW. A case analysis is missing, to check if it is the last time point of the trace. This is corrected in **giavar15a**.

2. **hejejoklparasa95a**

   Susana Tool to translate Monadic Second Order Formulas into finite state automata called MONA. See also the system LTLf2DFA [1] where they use MONA to translate LTL_f formulas to DFAs.

   (a) **gimamo14a**; **giavar15a**; **giavar16a**

   (b) **zhpuva19a**

   (c) **zhtapuva20**

   Susana Investigates DFA-minimization algorithms adapted to start from temporal logic formulas in LTL_f and Past LTL_f.

   (d) **baimci06a**

   Susana Explores planning with temporally extended goals by transforming LTL_f with FO Quantifiers to parameterized NFAs.

   (e) **rozvar07a**

   Susana Approaches LTL_f satisfiability problem using Boolean SAT solvers based on a transition system.

   (f) **balitava20a**

[1] https://github.com/whitemech/LTLf2DFA

Susana Hybrid approach for the conversion of LTL_f formulas into DFAs using explicit and symbolic representations.

(g) **cabamumc18a**

Susana Reduces LTL_f synthesis to finding a strong plan for a (FOND) planning problem using a NFA. They improve efficiency by exploiting techniques of symbolic automata decomposition, lazy determinization, certificates of unrealizability, and state abstraction.

3.2 Linear Dynamic logic

1. **giavar13a**

   (a) **giafav21a**

   Susana Constructs a DFA of an LDL_f formula by applying composition of previously computed DFAs. For the Kleen Star they use the translation to AWF **giavar13a** instead.

   (b) **sobanamc03a**

   Susana Represents temporal an procedural knowledge in ASP by encoding temporal formulas and GOLOG programs as predicates. The semantics are directly encoded in a logic program. They include the constraints $\leftarrow \mathit{not}\ \mathit{htf}(\phi, 0)$ and $\leftarrow \mathit{not}\ \mathit{trans}(\rho, 0, n)$ to work in the classical case.

   (c) **ryan14a**

   Susana Compares different ASP encodings for GOLOG. The best performance was the representation as Finite State Machine previously compiled from a regular expression (Thompson 1968).

4 Dynamic temporal logic

1. **hatiko00a**

   (a) **fislad79a**

5 Dynamic temporal logic over finite traces

1. **giavar13a**

   (a) **giavar15a**; **giavar16a**

2. **cadilasc20a**

3. See also:

   (a) Section **??**

   (b) **sobanamc03a**

6 Temporal Equilibrium Logic

1. **agcadipevi13a**

   Klaus Good introduction on temporal here and there and temporal equilibrium logic. It also provides translations of TEL programs into logic programs. Finally, they also consider an automata based method to compute models.

7 Temporal Equilibrium Logic on Finite Traces

1. **cakascsc18a**

   Klaus Explains TEL for finite traces. Translates the TEL_f program into a logic program using a Tseitin-style reduction. It also goes into a translation for incremental solving(implemented in telingo). This translation only uses what they call "present-centered rules".

   (a) **cakamosc19a**

   Klaus This is about the telingo implementation. Most importantly, it explains how a temporal formula(in past-future format) can be translated into a logic program.

8 Metric temporal logic

1.

9 Metric temporal logic over finite traces

1.

10 Temporal logic in general

1.

11 Splitting

1. **liftur94a**

   Francois Original paper on splitting logic programs. Parts 3 and 6 introducing splitting set and splitting sequence are a must read.

12 Module Theory

1. **oikjan06a**

   Francois Great introduction on modules and their composition. Must read for anyone interested in incremental solving.

13 Incremental ASP solving

1. **gekakasc17a**

   (a) **gekakaosscth08a**

   (b) **gekasc12a**

14 Action languages

1. **gellif98a**

   (a) **gilelimctu03a**

   (b) **leelif03b**

2. **leliya13a**

   (a) **khyalelist14a**

   (b) **bablee15a**

15 ASP planning

1. **lifschitz02a**

   Torsten The seminal paper on using ASP for planning. Things may have evolved meanwhile but it still gives the basic intuitions in the admirable style of Vladimir.

   (a) **gellif93a**

   (b) **bargel00a**

2. **sobanamc03a**

16 ASP scheduling

(This is meant to include actions with durations.)

1. **sobatu04a**

2. **baboge05a**

3. See also

   (a) **leelif03b**; **khyalelist14a**

17 SAT planning

1. **rintanen09a**

18 Temporal planning

1. **backab00a**
2. **kvhedo08a**[1]

[1] T2F: Is this the best reference, or just a specialized one. . . ?

19 Agent programming

1. **lereleli97a**
2. **gilele00a**
   (a) **honiclla16a**
   (b) **ryan14a**

20 Planning in general

1.

This article was processed using the comments style on May 20, 2022.
There remain 1 comments to be processed.