Difference Logic

Satisfiability Checking Seminar

Alex Ryndin
Supervision: Gereon Kremer

WS 2016/2017

Abstract

This report describes the difference logic (DL) and a graph-based approach for solving satisfiability (SAT) problem of DL formulas. There is of course a simplex-based algorithm which solves SAT problem of any linear arithmetic (LA) constraints. However, it is not efficient compared to the algorithm described in the report because simplex does not utilize the simple structure of DL constraints.

Efficiently solving SAT problem of DL constraints is very important because a lot of timing related problems can be described by this logic e.g. scheduling problems, detecting race conditions in digital circuits etc.

The report is organized as follows. Chapter 1 introduces difference logic and SAT checking. Chapter 2 gives theoretical background. Chapter 3 describes a graph-based approach to solving SAT problem of DL. Chapter 4 draws a conclusion.

1 Introduction

1.1 Difference Logic

DL is a special case of an LA logic in which all LA constraints have the form $x - y \prec c$ where x and y are numerical variables, c is a constant and $\prec \in \{<, \leq\}$ is a comparison operator. A more formal definition of DL is given below [1, 2]:

Definition 1.1 (Difference Logic) *Let $\mathcal{B} = \{b_1, b_2, \dots\}$ be a set of Boolean variables and $\mathcal{X} = \{x_1, x_2, \dots\}$ be a set of numerical variables. The difference logic over $\mathcal{B}$ and $\mathcal{X}$ is called $DL(\mathcal{X}, \mathcal{B})$ and given by the following grammar:*

$$\phi \stackrel{\mathsf{def}}{=} b \mid (x - y < c) \mid \neg\phi \mid \phi \wedge \phi$$

where $b \in \mathcal{B}$, $x, y \in \mathcal{X}$ and $c \in \mathbb{D}$ is a constant. The domain $\mathbb{D}$ is either the integers $\mathbb{Z}$ or the real numbers $\mathbb{R}$.
The remaining Boolean connectives $\vee, \rightarrow, \leftrightarrow, \dots$ can be defined in the usual ways in terms of conjunction and negation.

Examples of DL formulas are given below:

$$f_1 = (p \vee q \vee r) \wedge (p \rightarrow (u - v < 3)) \wedge (q \rightarrow (v - w < -5)) \wedge (r \rightarrow (w - x < 0)) \tag{1}$$

$$f_2 = ()\ land() \tag{2}$$

$$f_3 = () \tag{3}$$

1.2 Solving SAT Problem of Propositional Logic

The satisfiability (SAT) checking problem Describe here the general approach to SAT solving presented in the main paper [1] (procedure Solve, DPLL algorithm, generic scheme of tandem of SAT Solver + Theory Solver). It is a good idea to include some picture/diagram describing the DPLL approach (e.g. a block scheme of the algorithm/procedure). Maybe it is also a good idea to give a simple example. Give definitions: implication graph, unique implication point.

Definition 1.2 (Implication Graph) *def*

Definition 1.3 (Implication Point) *def*

Definition 1.4 (Unique Implication Point) *def*

1.3 Solving SAT Problem of Difference Logic

Shortly describe possible approaches (their core ideas) to solve DL SAT problem. According to the main article [1], they are:

- The lazy approach
- The preprocessing approach
- Incremental approaches

2 Preliminaries

Theoretical stuff, on which the solver from [1] is based on (1-3 pages).

2.1 Constraint Graph (CG)

Give definition of Constraint Graph and an example. Tell also about Difference Bound Matrix (DBM). Allude to the fact that SAT problem can be described in terms of a testing a CG on a negative cycle.

2.2 Bellman-Ford Algorithm

Describe Bellman-Ford algorithm and how is it applied to the DL SAT problem: "We detect cycles using a depth-first variant of the Bellman-Ford-Moore algorithm [GR93] which has much better average case complexity in practice" [1]

3 Topic

Describe how the proposed solver [1] works (3-5 pages).

3.1 Negative Cycles Detection

Show how the satifiability of a DL formula is related to the negative cycle detection.

3.2 Implementation Details

Some implementation details (Numeric Conflict Analysis, Reducing Feasibility Checks).

3.3 Experimental Results

Tell a reader about some experimental results.

4 Conclusion

Conclusion on the topic ($\frac{1}{2}$ of a page).

References

[1] S. Cotton, E. Asarin, O. Maler, and P. Niebert. Some progress in satisfiability checking for difference logic. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 263–276. Springer, 2004.

[2] M. Mahfoudh. *Sur la Vérification de la Satisfaction pour la Logique des Différences*. PhD thesis, Université Joseph Fourier (Grenoble), 2003.