FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Bounded Non-Linear Integer Constraint Solving
Student:	Leonid Burbygin
Supervisor:	doc. Dipl.-Ing. Dr. techn. Stefan Ratschan
Study program:	Informatics
Branch / specialization:	Computer Science
Department:	Department of Theoretical Computer Science
Validity:	until the end of summer semester 2023/2024

Instructions

Solving non-linear integer constraints (i.e., conjunctions and disjunctions of equalities and inequalities over the integers) is an undecidable problem. Still, SAT modulo theory (SMT) solvers contain sophisticated algorithms that can solve many instances of such constraints. Especially, the problem can be made trivially decidable by adding finite bounds to all variables. In this case, the sophisticated algorithms contained in SMT solvers are often an overkill, and the goal of this thesis is to check, how far one can get with algorithms that solve such constraints by simply checking the constraints on the whole finite set of possible values.

1) Write a solver for bounded non-linear integer constraints based on the trivial "check all values" algorithm.
2) Compare its behavior against the SMT solver CVC5 using benchmark examples, for example from the SMTLIB database (https://smtlib.cs.uiowa.edu).
3) Improve the written solver in such a way that it can beat CVC5 in a few more cases.
4) Do systematic computational experiments that document the strong and weak points of the original trivial algorithm, the improved algorithm, and the SMT solver CVC5.

Electronically approved by doc. Ing. Jan Janoušek, Ph.D. on 7 February 2023 in Prague.

Bachelor's thesis

BOUNDED NON-LINEAR INTEGER CONSTRAINT SOLVING

Leonid Burbygin

Faculty of Information Technology
Department of Theoretical Computer Science
Supervisor: doc. Dipl. -Ing. Dr. techn. Stefan Ratschan
April 3, 2023

Citation of this thesis: Burbygin Leonid. *Bounded Non-Linear Integer Constraint Solving.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2023.

Contents

List of Figures

List of Tables

List of code listings

I would like to thank all my teachers for their guidance and support, which has been instrumental in my academic journey. I am also grateful to my family for their unwavering love and encouragement throughout this process. Finally, I am thankful for the support of my friends and colleagues, who have been a constant source of motivation. Without them, this thesis would not have been possible.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of theAct.

In Praze on April 3, 2023 ....

Abstract

Solving non-linear integer constraints (i.e., conjunctions and disjunctions of equalities and inequalities over the integers) is an undecidable problem. Still, SAT modulo theory (SMT) solvers contain sophisticated algorithms that can solve many instances of such constraints. Especially, the problem can be made trivially decidable by adding finite bounds to all variables. In this case, the sophisticated algorithms contained in SMT solvers are often an overkill, and the goal of this thesis is to check, how far one can get with algorithms that solve such constraints by simply checking the constraints on the whole finite set of possible values.

Keywords non-linear integer constraints, SAT modulo theory, SMT solvers, finite bounds, undecidable problem, sophisticated algorithms, constraints

Abstrakt

Řešení nelineárních celočíselných omezení (tj. spojení a disjunkcí rovností a nerovností nad celými čísly) je nedostatečně rozhodnutelný problém. Nicméně SMT (SAT modulo teorie) řešiče obsahují sofistikované algoritmy, které dokáží vyřešit mnoho instancí takovýchto omezení. Zvláště problém může být triviálně rozhodnutelný přidáním konečných hranic ke všem proměnným. V tomto případě jsou sofistikované algoritmy obsažené v SMT řešičích často zbytečné a cílem této práce je zjistit, jak daleko lze s algoritmy dostat, které řeší taková omezení jednoduše kontrolou omezení nad celou konečnou množinou možných hodnot.

Klíčová slova nelineární celočíselná omezení, SAT modulo teorie, SMT řešiči, konečné hranice, nedostatečně rozhodnutelný problém, sofistikované algoritmy, omezení.

Summary

Introduction

In this thesis, I aim to address the problem of solving non-linear integer constraints, which is an undecidable problem. Despite its inherent difficulty, there are sophisticated algorithms contained in SMT solvers that can solve many instances of such constraints. However, in some cases, adding finite bounds to all variables can make the problem trivially decidable, and the use of such algorithms may be an overkill. Therefore, I explore an alternative approach that solves such constraints by checking the constraints against the whole finite set of possible values. The goal of this thesis is to investigate the effectiveness of this approach and compare it to existing methods. To achieve this, I develop a methodology for solving non-linear integer constraints, which involves parsing the constraints, building an AST, and estimating the bounds of the constraints to generate a finite interval of possible values. I then try all values in that interval to determine whether the constraint is satisfiable. Through my research, I aim to contribute to the development of new methods for solving non-linear integer constraints that are effective, efficient, and have potential practical applications.

Literature review

The literature review for this thesis involves an examination of existing research and documentation related to the problem of solving non-linear integer constraints. Specifically, I have reviewed the official documentation of the SMT-LIB standard, version 2.6, which was released in 2017 by Clark Barrett, Pascal Fontaine, and Cesare Tinelli. This standard provides a set of guidelines and specifications for SMT solvers, which are widely used for solving non-linear integer constraints. Our review also includes an examination of the limitations of existing approaches, including the potential overuse of sophisticated algorithms in SMT solvers, and the benefits of using algorithms that check constraints against the whole finite set of possible values. Through this review, we aim to identify gaps in existing research and develop a deeper understanding of the challenges and opportunities associated with solving non-linear integer constraints.

Methodology

The methodology used in this thesis involves a process for solving non-linear integer constraints by checking them against the whole finite set of possible values. Specifically, this involves reading a constraint, parsing it, and building an AST that represents the structure of the constraint. Then, using the AST, I estimate the bounds of the constraint and generate a finite interval of possible values. Finally, I try all values in that interval to determine whether the constraint is satisfiable. This approach does not require the use of sophisticated algorithms contained in SMT solvers, and instead focuses on a simple yet effective method of solving non-linear integer constraints. By using this methodology, I aim to explore the effectiveness of this approach and compare it to other existing methods of solving non-linear integer constraints.

Results

TODO

Conclusion

TODO

Seznam zkratek

SMT	Satisfiability modulo theories
SAT	Boolean satisfiability problem or propositional satisfiability problem
AST	Abstract syntax tree
NP-complete	Nondeterministic polynomial-time complete

Introduction

In this chapter I describe Motivation, Literature Review and tested Methods

0.1 Motivation

In recent years, Satisfiability Modulo Theories (SMT) solvers have become an important tool for solving complex mathematical problems in various fields, including computer science, mathematics, and engineering. These solvers are used to solve a variety of problems, including non-linear integer constraints, which involve conjunctions and disjunctions of equalities and inequalities over integers.

While SAT modulo theory (SMT) solvers contain sophisticated algorithms that can solve many instances of non-linear integer constraints, the problem is still undecidable. However, by adding finite bounds to all variables, the problem can be made trivially decidable. This approach can significantly improve the quality of SMT solvers in some cases, potentially decreasing computing time and making them more efficient.

The goal of this bachelor thesis is to develop a simple SMT solver for integer non-linear constraints that will try all values in some interval. By doing this, it is possible to improve the quality of the CVC5 solver in a few more cases and potentially decrease the computing time. To accomplish this, benchmarks will be used to compare the performance of the developed solver with the original CVC5 solver.

0.2 Literature Review

CVC5 is a highly sophisticated SMT solver that has been developed for many years by a team of researchers. It is a widely used program in the field of automated reasoning and has a strong reputation for its reliability and efficiency. One of the reasons for its popularity is the availability of comprehensive documentation that covers all aspects of the program's design and implementation.

For the development of a simple SMT solver for integer non-linear constraints that tries all values in some interval, the official documentation of CVC5 provides a valuable source of information. It describes the algorithms and techniques used in CVC5 and provides detailed explanations of the various functions and modules of the program. This documentation is a key resource for understanding the inner workings of CVC5 and for developing a custom solver that can be compared to the original program.

In this work, I rely on the official documentation of CVC5 as the main source of knowledge for the development of my own solver. I use the benchmarks provided by CVC5 to compare the performance of my solver with that of the original program. By doing so, I aim to identify cases where my solver can improve the quality of the results and potentially decrease the computing time.

0.3 Methods

The initial approach is to take a specified interval, which was set to range from -100 to 100. If it becomes evident from the constraints that an interval has a boundary, the interval is shortened. Once the intervals are defined, the solver tries all possible integer values within the intervals. This approach has an asymptotic complexity of the interval width to the power of the number of variables, i.e. $O(w^n)$, where w is the interval width and n is the number of variables.

0.4 Benchmarks

The benchmarks were taken from the official repository: `https://clc-gitlab.cs.uiowa.edu:2443/SMT-LIB-benchmarks/QF_NIA/-/tree/master/20220315-MathProblems`.

1. STC_0001.smt2 - STC_1000.smt2 - a collection of problems related to the Sum of Three Cubes. This set of benchmarks was chosen because it is easy to parse and contains simple math operations.
2. MC_01.smt2 - a benchmark for the Magic Square of Cubes problem. This benchmark was chosen because it is a challenging problem that requires the solver to find solutions for a set of equations that involve the sum of cubes of integers.
3. TODO write about complexity, and relevance to the research question.

   TODO finish bib-database like this [1]

Practical part

This chapter focuses on the development, testing and evaluation of results.

0.5 Introduction

The practical part of this Bachelor's thesis aims to implement a simple Satisfiability Modulo Theories (SMT) solver for integer non-linear constraints. All SMT solvers consist of three logical parts: lexer (lexan), parser, and Abstract Syntax Tree (AST). The lexer reads the input string character by character and tokenizes it into a stream of meaningful tokens. These tokens are then passed to the parser, which validates the input and generates an AST. The AST represents the input as a structured tree, making it easier to reason about and manipulate. Finally, the constraints are evaluated using the AST. In this practical part, I will develop each of these logical parts and integrate them to create a working SMT solver. To provide a clear overview of how the parts work together, I will present a simple diagram to illustrate the interactions between the lexer, parser, and AST.

TODO finish

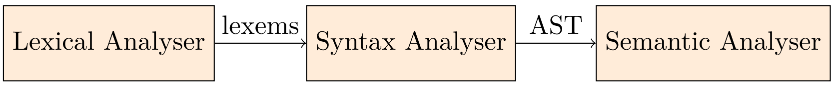

Figure 1 Logical components of a solver

Additionally, structure of the project looks in the following way:

```
code
├── src ............................................ Directory with all source files.
│   └── main.cpp ...................................................................
└── test ................................................... Directory with all tests.
    └── STC_0000.smt2 ........................................... Example test
```

0.6 Development

TODO write about parts of program, code?

0.7 Data Collection

0.7.1 First run

Interval of my solver was set from -100 to 100. 1000 STC test were executed. Total run-time was XXX minutes.

0.7.2 Second run

Interval of my solver was set from -300 to 300. 1000 STC test were executed. Total run-time was XXX minutes.

TODO Why? Other tests?

0.8 Data Analysis

0.8.1 First run

Original CVC5 solver gave XXX result within 1 minute timeout, and 144 tests had the same results.

0.9 Results

0.10 Discussion

Discussion

Conclusions

Appendix A
Nějaká příloha

Sem přijde to, co nepatří do hlavní části.

Bibliography

1. ŠESTÁKOVÁ, Eliška; JANOUŠEK, Jan. Automata Approach to XML Data Indexing. *Information.* 2018, vol. 9, no. 1. ISSN 2078-2489. Available from DOI: `10.3390/info9010012`.

Obsah přiloženého média

- `readme.txt` ... stručný popis obsahu média
- `exe` ... adresář se spustitelnou formou implementace
- `src`
  - `impl` ... zdrojové kódy implementace
  - `thesis` ... zdrojová forma práce ve formátu LaTeX
- `text` ... text práce
  - `thesis.pdf` ... text práce ve formátu PDF