UNIVERSITAT POLITÈCNICA DE CATALUNYA

DELIVERABLE 1: CONTEXT AND SCOPE OF THE PROJECT

Design of an environment for solving pseudo-boolean optimization problems

---

Author:
Marc BENEDÍ

Supervisor:
Dr. Jordi CORTADELLA

GEP

March 4, 2018
Edinburgh, UK

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Abstract

Facultat Informàtica de Barcelona
Department of Computer Science

Computer Science Degree

Deliverable 1: Context and scope of the project

by Marc BENEDÍ

In this deliverable, a first introduction into the project context1 will be made. *Boolean Satisfiability Problems* are explained together with other important concepts for this project like *Boolean Formula, Pseudo-Boolean Formula, Conjunctive Normal Form, Minimization, ...* The background for this project and motivations will also be detailed.
Next, the Project Formulation2 will be exposed, where the general objectives of it will be defined.
Later, the Scope3 of the project will be discussed together with what requirements the project should meet, how are they going to be made and what obstacles could be found.
Finally, the used methodology, the tools and the rigor will be exposed in Methodology and Rigor4.

Glossary

LAH **L**ist **A**bbreviations **H**ere
WSF **W**hat (it) **S**tands **F**or

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter a first introduction ...

1.1 Context

Boolean satisfiability problems *(SAT from now on)* is the problem of finding a model[1] for a *Boolean Formula*. In other words, it is the result of evaluating the *Boolean Formula* after replacing its variables for *true* or *false*.
SAT is widely used in Computer Science because it was the first problem proved to be NP-Complete[1][2] which allowed a lot of NP[3] to be reduced to it.

1.1.1 What is a Pseudo-Boolean Formula?

In propositional logic, a boolean formula is defined as following[2]:
Let P be a set of predicate symbols like $p, q, r, ...$

- All predicate symbol of P is a formula.
- If F and G are formulae, then $(F \wedge G)$ and $(F \vee G)$ are formulae to.
- If F is a formula, then $(\neg F)$ is a formula.
- Nothing else is a formula.

This representation has some limitations because it can only express properties which are *true* or *false*.

1.2 Background

During the past semester (Q1 2017/2018) I had been developing a C++ library called _____.
This tool allows the users to represent *Boolean Formulas* in a C++ program in a intuitive way, do operations between them and convert them into *BDDs*. However, the main functionality of this library is the conversion from a *Boolean Formula* to a *CNF*. As previously explained, *CNF* is a particular type of a *Boolean Formula*, a conjunction of disjunctions. *CNF* is an important format because it is the standard input for *SAT Solvers*A.1.

[1] An interpretation which satisfies the formula.
[2] NP and NP-hard.
[3] Nondeterministic polynomial time.

As shown in this paper, *Mitchell, Selman, and Levesque[3]*, there is a correlation between the number of variables, the number of clauses and the hardness of solving the *CNF*.

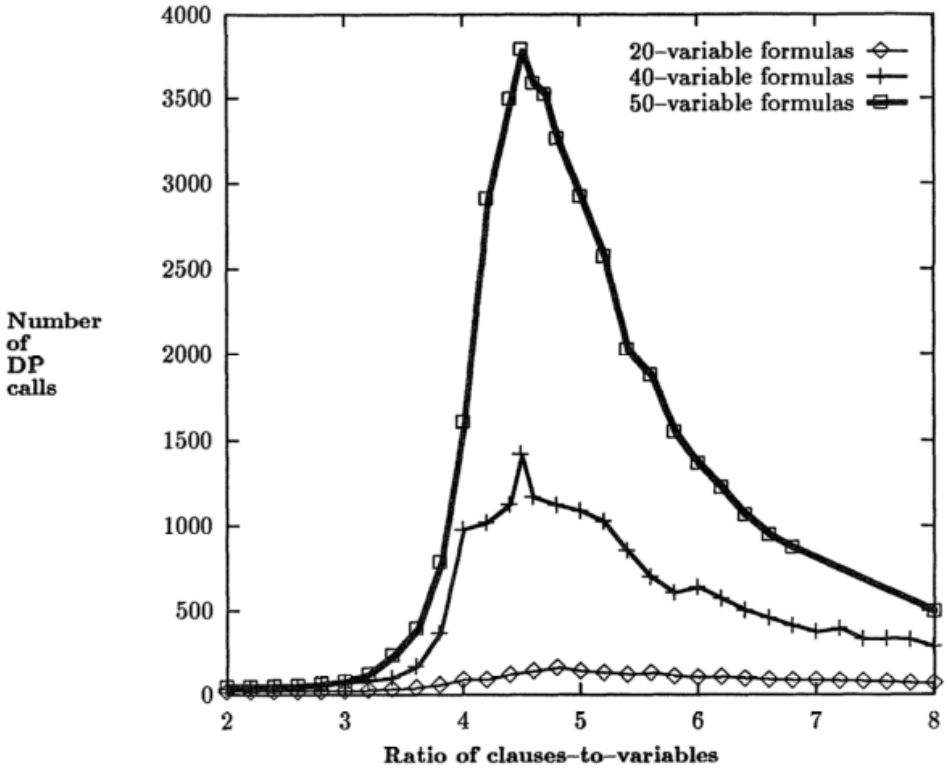

FIGURE 1.1: Median number of recursive DP calls for Random 3-SAT formulas, as a function of the ratio of clauses-to-variables. Extracted from Mitchell, Selman, and Levesque[3]

Therefore, an improvement of the input *CNF* of the *SAT Solver* can reduce a lot the hardness of the problem.
This is the main goal of the library, try to reduce the size of the final *CNF* resulting from applying different converting methods on the original *Boolean Formula*.

1.3 Sate-of-art

1.4 Motivation

Informatics Logic is taught in this[4] faculty. In this course I realized how important is *logic* through its lecturer, Dr.Robert Nieuwenhuis, and its activities.

In the first coursework we had to code a *SAT Solver* which used *Unit Propagation*A.2. With this activity I comprehended how hard and substantial is the study of *logic* and all its context. For example, how *logic* is used in Artificial Intelligence and Planners.

When the time of deciding the *TFG*, I contacted my actual supervisor, Dr. Jordi Cortadella, and he proposed me some topics and ideas for projects. Finally, we agreed on doing this project.

[4]Facultat Informàtica de Barcelona

Chapter 2

Project Formulation

Explicar en que consisteix el projecte

2.1 General objectives

Explicar quins son els objectius generals del treball

Chapter 3

Scope

3.1 What and how?

Parlar del que es vol fer i com es fara
requirements the project should meet
what to do? meet the requirements established by the client in particular, and by the rest of stakeholders

Chapter 4

Methodology and Rigor

explicar el perque es defineix una metodologia

4.1 Methodology

TDD i agile
short cicles
weekly scrum

4.2 Tools

In this chapter the development tools for this project will be introduced.

4.2.1 Git

Git is a well known version control system developed by Linus Torvalds[1].
Git will be used in this project because it allows to maintain a tracking of all the changes made (commits), and what is more important, return to them at any time. In addition to this, it enforces a short cycle development (because commits are small units of work) and the developer has to document them.

4.2.2 Trello

Trello is lala

4.2.3 Mendeley

4.2.4 CLion

4.3 Communication

4.4 Rigor and Validation

[1] Linux creator. (more)

Appendix A

More information

A.1 Why SAT Solvers use CNF as input format?

There are two mains reasons for this: Equisatisfiability and Computational Complexity. Let us start with the first one:

Two *Boolean Formulas* are **equisatisfiable** if and only if both have the same *models*. This may seem the same as equality but it is not because in an equality relationship both *Boolean Formulas* have to have the same variables.
This is important because between a *Boolean Formula* and its result from a *CNF* transformation the equisatisfiability is preserved which means that if the *SAT Solver* finds a *model* for the *CNF*, then this *interpretation* will be also a *model* for the original *Boolean Formula*.

The second reason is computational complexity. Let us have a look at the following table:

	DNF	CNF
TAUT	NP	P
SAT	P	NP

TABLE A.1: Complexity of deciding if a *Boolean Formula* is SAT or TAUT depending of its format.

So as a *Boolean Formula* can be converted into a *CNF* in linear time while preserving equisatisfiability, *SAT Solvers* will use them to target satisfiability.

A.2 What is Unit Propagation?

Unit propagation

Bibliography

[1] Stephen A. Cook. "The complexity of theorem-proving procedures". In: *Proceedings of the third annual ACM symposium on Theory of computing - STOC '71*. New York, New York, USA: ACM Press, 1971, pp. 151–158. DOI: 10.1145/800157.805047. URL: http://portal.acm.org/citation.cfm?doid=800157.805047.

[2] Rafael Farré et al. *Notas de Clase para IL - 2.Definición de la Lógica Proposicional*. Barcelona, 2009. URL: https://app.box.com/file/225148187559.

[3] David Mitchell, Bart Selman, and Hector Levesque. "Hard and Easy Distributions of SAT Problems". In: (). URL: https://aaai.org/Papers/AAAI/1992/AAAI92-071.pdf.