Diplomarbeitspräsentation

Algorithms for Quantified Cut-Introduction

Masterstudium:
Computational Intelligence

Christoph W. Spörk

Technische Universität Wien
Theory and Logic Group
Arbeitsbereich: Theoretische Informatik und Logic
BetreuerIn: Univ.-Prof. Dr.phil. Alexander Leitsch

Motivation

Proofs as well as their understanding are of great importance to the scientific community . They can appear in various forms, e.g. as semantic argument or as deviation in a particular calculus. The latter can be represented in **sequent-calculus** and consists of several rules. One of them is the **cut-rule**, which is not necessarily occuring in a derivation, but can be very helpful for compressing such proofs. Proofs with cuts in sequent calculus contain **lemmas** which can give deeper insights into the meaning of a theorem. This thesis implements a method for introducing several quantified cuts in a cut-free proof in sequent caculus.

Method

Implementation of a method for introducing quantified cuts based on

[1] Algorithmic cut-introduction
[2] Algorithmic compression of finite tree languages by rigid acyclic grammars

Integrated into an already existing architecture for proofs (**GAPT**) in Scala.

Compared to an existing approach, able to introduce a single quantified cut.

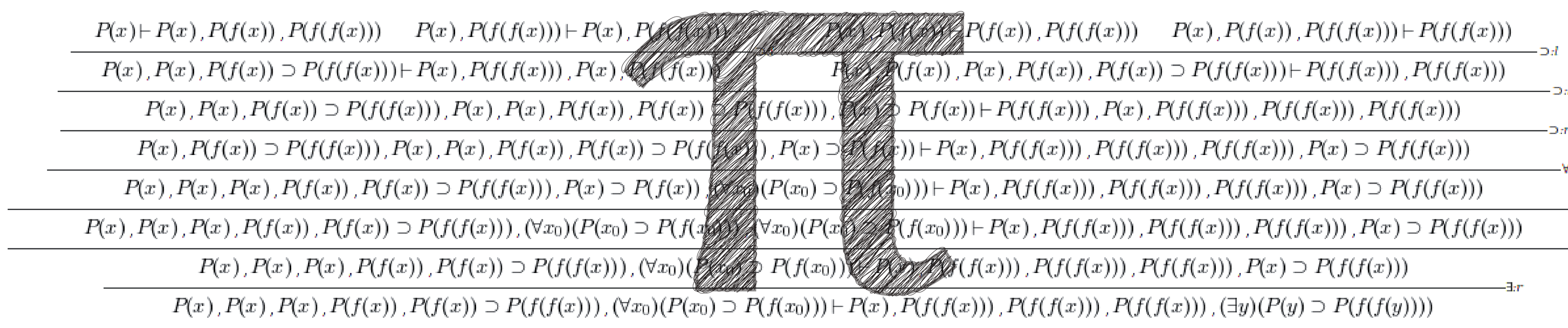
Results

Both the single-cut and the many-cuts approach were **tested** by running a large set of experiments, including ...

... primitive **proof sequences**
... proofs from particular **libraries** (**TPTP**, etc.)

The performance was not improved significantly.

The possibility to **introduce more than one cut** at a time represents a **major improvement** in the field of **cut-introduction**.



1 start at *end sequent* of a cut-free proof

$$P(0), \forall x\, P(x) \supset P(s(x)) \vdash P(s^8(x))$$

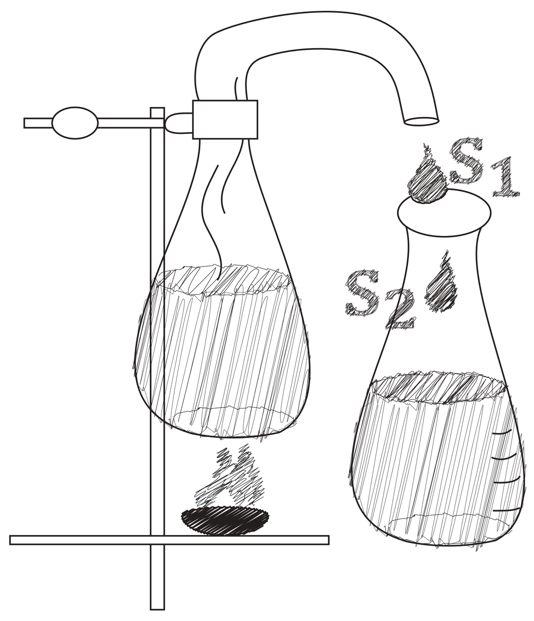
2 extract *Herbrand sequent* and and introduce artificial function symbols

$$P(0), f_1(P(s(0))), \ldots, f_1(P(s^7(0))) \vdash P(s^8(0))$$

3 generate *language*

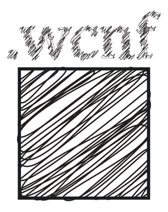
$$T = \{0, f_1(s(0)), \ldots, f_1(s^7(0))\}$$

4 generate sufficient set of keys

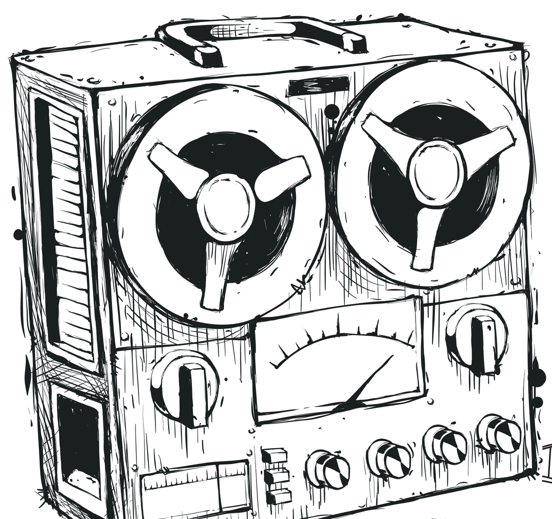


5 transform to *MinCostSAT* formulation

.wcnf



6 *MaxSAT* Solver

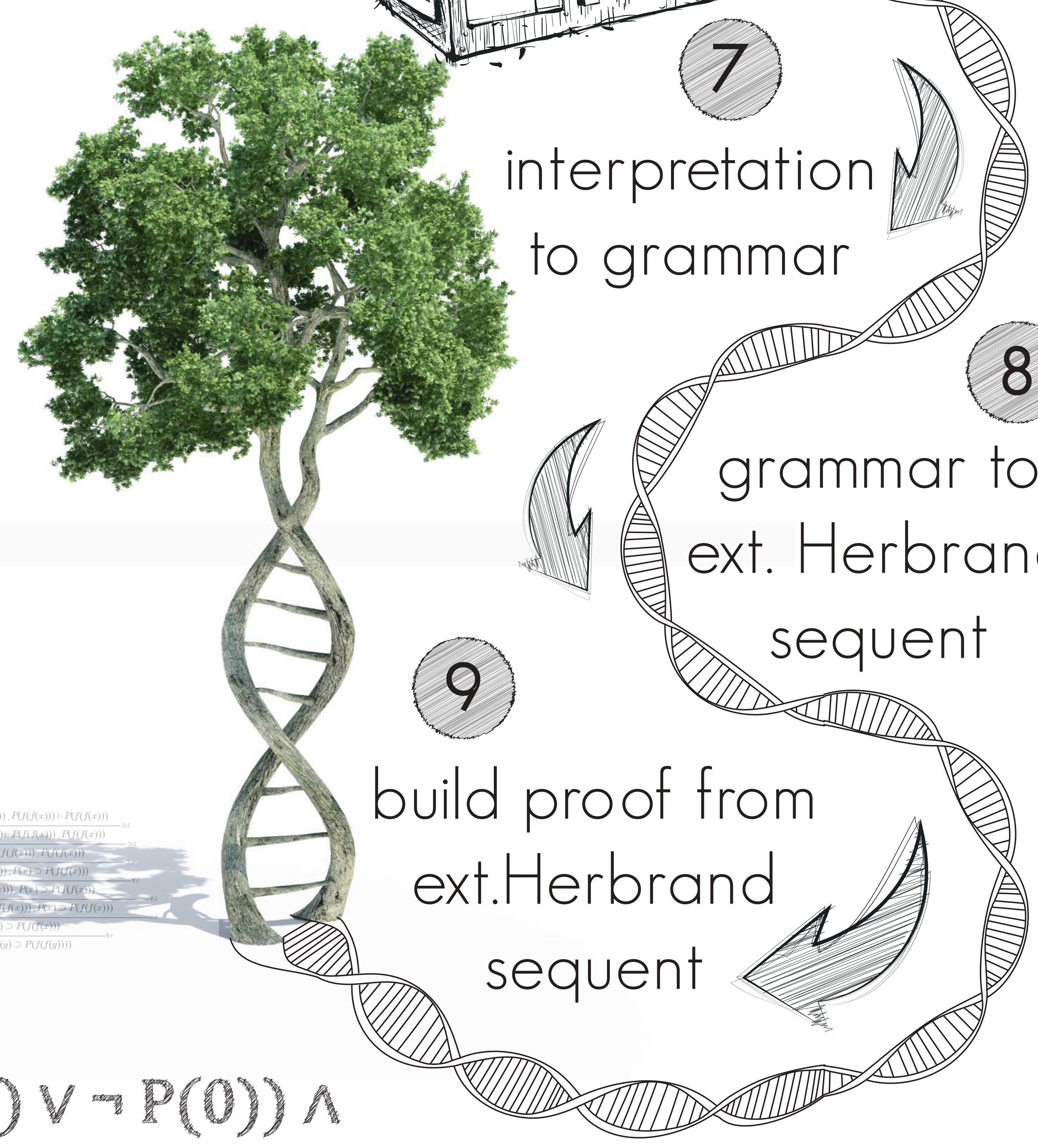


7 interpretation to grammar

8 grammar to ext. Herbrand sequent

9 build proof from ext.Herbrand sequent

Compressed proof with introduced cuts



ex. cut-formula

$$(P(s^4(\alpha)) \vee \neg P(\alpha)) \supset ((P(s^4(0)) \vee \neg P(0)) \wedge (P(s^8(0)) \vee \neg P(s^4(0)))$$

[1] Stefan Hetzl, Alexander Leitsch, Giselle Reis, and Daniel-Weller. Algorithmic introduction of quantified cuts. Theoretical Computer Science, 549, 2014.

[2] Sebastian Eberhard and Stefan Hetzl. Algorithmic compression of finite tree languages by rigid acyclic grammars. 2014.

Kontakt: christoph.spoerk@gmail.com