FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

Space & Congruence Compression of Proofs

Master- /Diplomstudium:
European Master in Computational Logic

Andreas Fellner

Technische Universität Wien
Institut für Computersprachen
Arbeitsbereich: Programmiersprachen und Übersetzer
BetreuerIn: Univ. Prof. Dr.phil. Alexander Leitsch
Mitwirking: Bruno Woltzenlogel Paleo, Dr.

Proof Compression

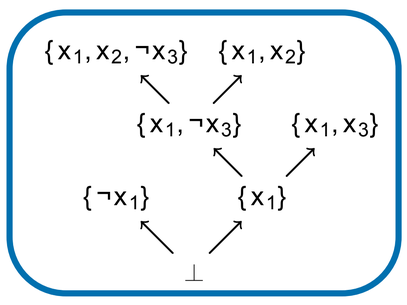
We present two methods for compression of formal proofs. Formal proofs are of great importance to modern computer scienece. They can be used to combine deductive systems, e.g. via the use of SAT Solvers. Problems tackled by automated systems are huge and so are the produced proofs. Proof files easily reach many gigabytes in size. Processing such proofs even brings computer systems to their limits. Usually there are many different proofs of one problem and the goal of proof compression is to manipulate proofs to obtain proofs, using the same or less axioms, that are better with respect to measures like length or space.

Two proofs for the unsatisfiability of
$(x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (\neg x_1)$

The Propositional Logic Resolution Calclulus is a simple calculus with a single inference rule that is very popular in automated deduction. Its simplicity comes at the price that proofs tend to become large. These two properties make it a good target for proof compression.

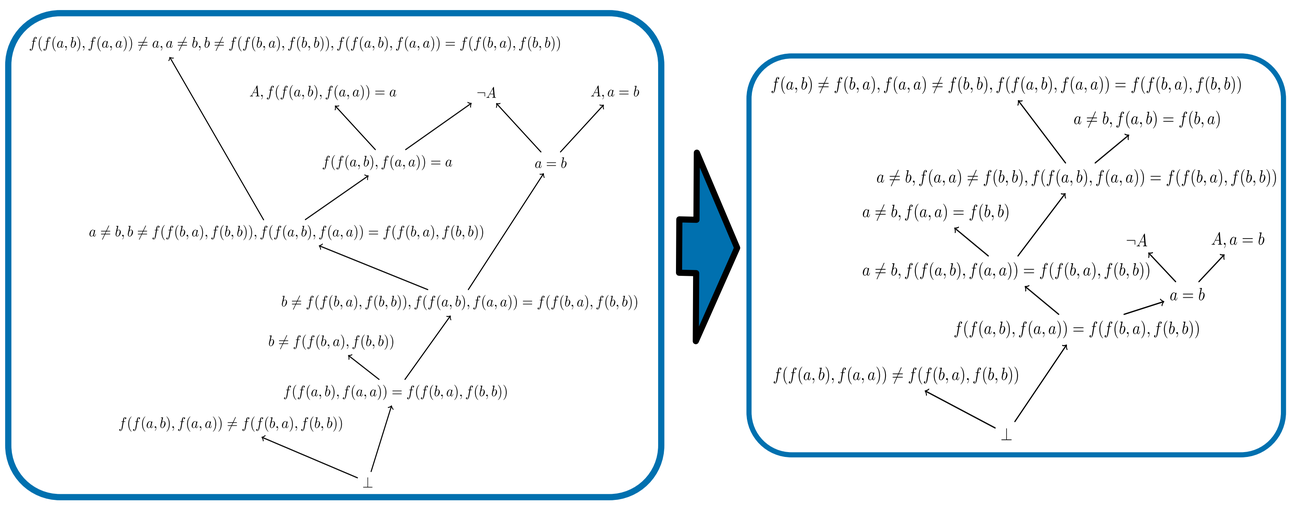
Resolution Rule

$$\frac{C \vee x \quad D \vee \neg x}{C \vee D}$$



Congruence

Equality is a well studied and important topic in logic. Congruence reasoning derives new equalities on terms from a set of equations, using the axioms of equality reflexivity, symmetry, transitivity and compatibility.
A set of equations implying some target equality is an explanation. By replacing large by small explanations in proofs, we achieve compression in proof length. The following example illustrates our method. The proof on the left reasons with a long explanation using transivity, whereas the proof on the right reasons with a shorter explanation using compatability. Even though the replaced subproof is longer, the overall proof is shorter. In our work, we discuss the tradeoff between explanation and subproof size.



We prove that finding the shortest explanation for some equality is an NP-complete problem by reducing the well known NP-complete decision problem SAT to ours.
We propose a novel explanation producing congruence closure algorithm that was designed to produce short explanations in runtime that is the best known among congruence closure algorithms. Our algorithm is described in such a way that it can be implemented into a functional programming language easily.

Space

The space of a proof is the maximal amount of proof nodes that have to be kept in memory at once while processing the proof. To process huge proofs, it is critical to drop parts of the proof from memory that are not needed anymore. The order in which proofs nodes are traveresed during processing is essential for space requirements. We propose two algorithms, corresponding to travesing the proof bottom-up resp. top-down, together with a collection of heuristics to construct orders.

Experiments

The Congruence- and Space compression methods were implemented in the proof compression tool Skeptik and evaluated on 3965 respectively 7555 proofs, produced by the SMT solver VeriT and the SAT solver PicoSAT. The experiments were carried out on the Vienna Scientific Cluster.
The congruence method shows an 2% effective proof length- and a 28% explanation size compression.
The experiments showed that traversing proofs in a bottom-up fashion memorywise is much better than top-down. On average only 44% memory is required when removing proof nodes from memory during proof processing.

Kontakt: fellner.a@gmail.com