Partial Regularization of First-Order Resolution Proofs

Jan Gorzny[1*]   jgorzny@uwaterloo.ca
Bruno Woltzenlogel Paleo[2,3]   bruno@logic.at

[1]University of Waterloo
[2]Vienna University of Technology
[3]Australian National University
[*]Supported by the Google Summer of Code Program 2014

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS
David R. Cheriton School of Computer Science

SUMMER of CODE 2014

Australian National University

TU WIEN — TECHNISCHE UNIVERSITÄT WIEN

Introduction

► As the ability of automated deduction has improved, it has been applied to new application domains; e.g. Furbach et al. [2] used it in natural language reasoning.

► Resolution proof production is a key feature of modern theorem provers; the best, most-efficient provers do not necessarily generate the best, least redundant proofs.

► For proofs using propositional resolution generated by SAT- and SMT-solvers, there are many proof compression techniques.

► One approach to compressing first-order logic proofs is to lift ideas used in propositional logic.

Propositional RecyclePivotsWithIntersection [1]

► Traverse a proof from the bottom up: store for every node a set of *safe literals*: literals resolved in all paths below the node

► For any node whose resolved literals are safe, replace it with one of its parents (*regularizing* it.)

The set of safe literals for a node η will be denoted $\mathcal{S}(\eta)$.

A Propositional Example

Consider the proof ψ shown below:

$$\frac{\frac{\eta_2 : a, c, \neg b \quad \frac{\eta_1 : \neg a \quad \eta_3 : a, b}{\eta_4 : b} a}{\eta_5 : a, c} a \quad \frac{\eta_4 \quad \frac{\eta_7 : a, \neg b, \neg c}{\eta_8 : a, \neg c} b}{\eta_9 : \neg c} \eta_1}{\psi : \bot}}{\psi : \bot}$$

The algorithm RPI assigns $\mathcal{S}(\eta_5) \leftarrow \{a, c\}$, $\mathcal{S}(\eta_8) \leftarrow \{a, \neg c\}$, and $\mathcal{S}(\eta_4) \leftarrow \{a, c, b\} \cap \{a, \neg c, b\} = \{a, b\}$. Since $a \in \mathcal{S}(\eta_4)$ where a is a pivot of η_4, η_4 is detected as a redundant node and regularalized by replacing it by its right parent η_3:

$$\frac{\eta_1 : \neg a \quad \frac{\eta_2 : a, c, \neg b \quad \eta_3 : a, b}{\eta_5 : a, c} \quad \eta_3 \quad \frac{\eta_7 : a, \neg c, \neg b}{\eta_8 : a, \neg c} \quad \eta_1}{\psi : \bot}}{\psi : \bot}$$

A First-Order Example

Consider the proof ψ below. When computed as in the propositional case, $\mathcal{S}(\eta_3) \leftarrow \{\vdash q(c),\ p(a, X)\}$

$$\frac{\frac{\eta_1 : \ \vdash p(W, X) \quad \eta_2 : p(W, X) \vdash q(c)}{\eta_3 : \ \vdash q(c)} \quad \eta_4 : q(c) \vdash p(a, X)}{\eta_5 : \ \vdash p(a, X)} \quad \eta_6 : p(Y, b) \vdash}{\psi : \bot}$$

Since $p(W, X) \neq p(a, X)$, propositional RPI algorithm would not change ψ. However, η_3's left pivot $p(W, X) \in \eta_1$ is unifiable with the safe literal $p(a, X)$. Regularizing η_3, by deleting the edge between η_2 and η_3 and replacing η_3 by η_1, leads to further deletion of η_4 (because it is not resolvable with η_1) and finally to the following proof:

$$\frac{\eta_1 : \ \vdash p(W, X) \quad \eta_6 : p(Y, b) \vdash}{\psi' : \bot}$$

Unifiability is Not Enough

Consider ψ below. When computed as in the propositional case, $\mathcal{S}(\eta_3) \leftarrow \{\vdash q(c),\ p(a, X)\}$, and as the pivot $p(a, c)$ is unifiable with the safe literal $p(a, X)$, η_3 appears to be a candidate for regularization.

$$\frac{\frac{\eta_1 : \ \vdash p(a, c) \quad \eta_2 : p(a, c) \vdash q(c)}{\eta_3 : \ \vdash q(c)} \quad \eta_4 : q(c) \vdash p(a, X)}{\eta_5 : \ \vdash p(a, X)} \quad \eta_6 : p(Y, b) \vdash}{\psi : \bot}$$

However, if we attempt to regularize the proof, the same series of actions as in the last example would require resolution between η_1 and η_6, which is not possible.

Pre-Regularization Unifiability

Let η be a node with pivot ℓ' unifiable with safe literal ℓ which is resolved against literals $\ell_1, \ldots, \ell_n$ in a proof ψ. η is said to satisfy the *pre-regularization unifiability* property in ψ if $\ell_1, \ldots, \ell_n$, and $\overline{\ell'}$ are unifiable.

Pre-Regularization Unifiability: Still Not Enough

Consider the proof ψ below. After collecting the safe literals, $\mathcal{S}(\eta_3) \leftarrow \{q(T, V), p(c, d) \vdash q(f(a, e), c)\}$.

$$\frac{\eta_8 : q(f(a,e),c) \vdash \quad \frac{\eta_6 : \ \vdash p(c,d) \quad \frac{\frac{\eta_1 : p(U,V) \vdash q(f(a,V),U) \quad \eta_2 : q(f(a,X),Y), q(T,X) \vdash q(f(a,Z),Y)}{\eta_3 : p(U,V), Q(T,V) \vdash q(f(a,Z),U)} \quad \eta_4 : \ \vdash q(R,S)}{\eta_5 : \ p(U,V) \vdash q(f(a,Z),U)}}{\eta_7 : \ \vdash q(f(a,Z),c)}}{\psi : \bot}$$

η_3's pivot $q(f(a, V), U)$ is unifiable to (and even more general than) the safe literal $q(f(a, e), c)$. Attempting to regularize η_3 would lead to the removal of η_2, the replacement of η_3 by η_1 and the removal of η_4 (because η_1 does not contain the pivot required by η_5), with η_5 also being replaced by η_1. Then resolution between η_1 and η_6 results in η_7', which cannot be resolved with η_8, as shown below.

$$\frac{\eta_8 : Q(f(a,e),c) \vdash \quad \frac{\eta_6 : \ \vdash P(c,d) \quad \eta_1 : \ P(U,V) \vdash Q(f(a,V),U)}{\eta_7' : \ \vdash Q(f(a,d),c)}}{\psi' : ??}$$

η_1's literal $q(f(a, V), U)$, which would be resolved with η_8's literal, was changed to $Q(f(a, d), c)$ due to resolution between η_1 and η_6.
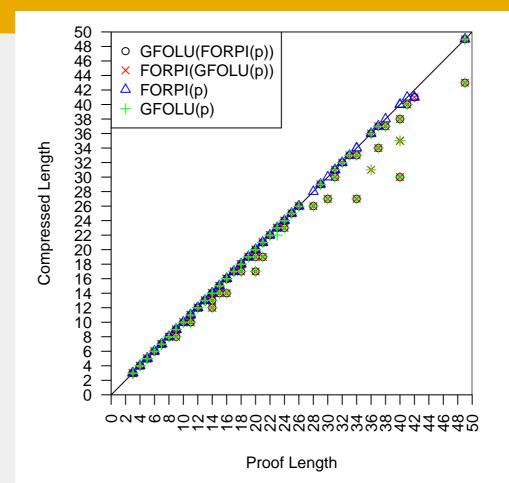
Regularization Unifiability

Let η be a node with safe literals ϕ that is marked for regularization with parents η_1 and η_2, where η_2 is marked as a deletedNode in a proof ψ. η is said to satisfy the *regularization unifiability property* in ψ if there exists a substitution σ such that $\eta_1 \sigma \subseteq \phi$.

The First-Order Algorithm

► Similar idea to the propositional case, but with care taken to ensure proofs satisfy the last two properties.

► First order *factoring* also employed to reduce proof size further, e.g. if $\eta_1 : p(X), p(Y) \vdash$, factor to $\eta_1' : p(X) \vdash$ before performing resolution.

► Does not compress all first-order proofs (yet).

Preliminary Results

► First-order proofs generated by SPASS show some compression by Scala based implementation.

► Data set is small: only 308 proofs, all of which are short

► Evaluation with GFOLU [3], a first-order variant of LowerUnits, another propositional compression algorithm lifted to first-order logic.
  ► Recycle pivots compresses 10x more than lower units in the propositional case

► Algorithm composition may matter less in the first-order case when compared to the propositional case.

► Quick compression: 40 minutes to generate all proofs, 8 seconds to compress all proofs.

Future Directions

► Larger evaluation - more proofs, bigger proofs

► Identify properties that would enable all irregular first-order proofs to be compressed

► Investigate lifting other propositional proof compression techniques.

References

[1] P. Fontaine, S. Merz, and B. Woltzenlogel Paleo.
Compression of propositional resolution proofs via partial regularization. In *CADE-23*. Springer, 2011.

[2] Ulrich Furbach, Björn Pelzer, and Claudia Schon.
Automated reasoning in the wild. In *CADE-25*. Springer, 2015.

[3] J. Gorzny and B. Woltzenlogel Paleo.
Towards the compression of first-order resolution proofs by lowering unit clauses. *CADE-25*. Springer, 2015.