Towards the Compression of First-Order Resolution Proofs by Lowering Unit Clauses

J. Gorzny[1] B. Woltzenlogel Paleo[2]

[1]University of Victoria

[2]Vienna University of Technology

6 August 2015

Proof Compression Motivation

an accessible, good motivational example for proof compression

(Propositional) Proofs

Definition (Proof)

A directed acyclic graph $\langle V, E, \Gamma \rangle$, where

- V is a set of nodes
- E is a set of edges labeled by literals
- Γ (the proof clause) is inductively constructible using *axiom* and *resolution* nodes

Definition (Axiom)

A proof with a single node (so $E = \emptyset$)

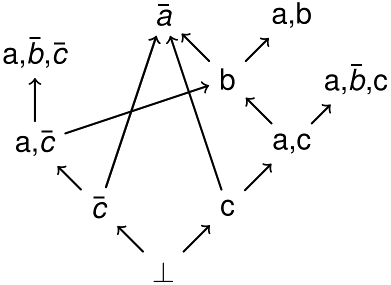
(Propositional) Resolution

Definition (Resolution)

Given two proofs ψ_L and ψ_R with conclusions Γ_L and Γ_R with some literal l such that $\bar{l} \in \Gamma_L$ and $l \in \Gamma_R$, the resolution proof ψ of ψ_L and ψ_R on l, denoted $\psi = \psi_L \psi_R$ is such that:

- ψ's nodes are the union of the nodes of ψ_L and ψ_R, and a new root node
- there is an edge from $\rho(\psi)$ to $\rho(\psi_L)$ labeled with $\bar{l}$
- there is an edge from $\rho(\psi)$ to $\rho(\psi_R)$ labeled with l
- ψ's conclusion is $(\Gamma_L \setminus \{\bar{l}\}) \cup (\Gamma_R \setminus \{l\})$

A Propositional Proof



Deletion

how deleting subproofs or edges in proofs affect them

Redundancy

types of redundancy we hope to remove, small examples (before/after proofs; not animated)

First-Order Proofs

Definition (First-Order Proof)

A directed acyclic graph $\langle V, E, \Gamma \rangle$, where

- V is a set of nodes
- E is a set of edges labeled by literals **and substitutions**
- Γ (the proof clause) is inductively constructible using *axiom*, ***(first order) resolution***, **and *contraction*** nodes

Axioms are unchanged

Substitutions and Unifiers

Definition (Substitution)

A mapping $\{X_1 \setminus t_1, X_2 \setminus t_2, \ldots\}$ from variables $X_1, X_2, \ldots$ to terms $t_1, t_2, \ldots$

Definition (Unifier)

First Order (Unifying) Resolution

Definition (First Order Resolution)

Given two proofs ψ_L and ψ_R with conclusions Γ_L and Γ_R with some literal l such that $l_L \in \Gamma_L$ and $l_R \in \Gamma_R$, and σ_L and σ_R are substitutions usch that $l_L\sigma_L = \overline{l_R}\sigma_R$, and the variables in $(\Gamma_L \setminus l_L)\sigma_L$ and $(\Gamma_R \setminus l_R)\sigma_R$ are disjoint, then the resolution proof ψ of ψ_L and ψ_R on l, denoted $\psi = \psi_L\psi_R$ is such that:

- ψ's nodes are the union of the nodes of ψ_L and ψ_R, and a new root node
- there is an edge from $\rho(\psi)$ to $\rho(\psi_L)$ labeled with l_L and σ_L
- there is an edge from $\rho(\psi)$ to $\rho(\psi_R)$ labeled with l_R and σ_R
- ψ's conclusion is $(\Gamma_L \setminus l_L)\sigma_L \cup (\Gamma_R \setminus l_R)\sigma_R$

Contraction

Definition (Contraction)

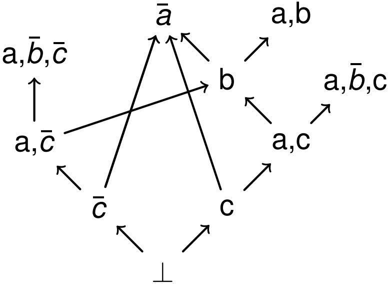
If ψ' is a proof and σ is a unifier of $\{l_1, \ldots, l_n\} \subset \Gamma'$, then a contraction ψ is a proof where

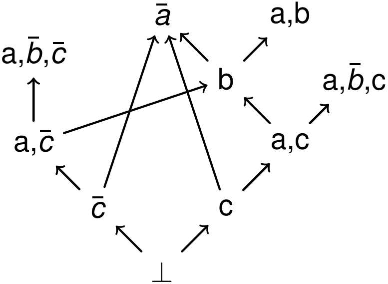
- ψ's nodes are the union of the nodes of ψ' and a new node v
- There is an edge from $\rho(\psi')$ to v labeled with $\{l_1, \ldots, l_n\}$ and σ
- The conclusion is $(\Gamma' \setminus \{l_1, \ldots, l_n\})\sigma \cup \{l\}$, where $l = l_k\sigma$ for $k \in \{1, \ldots, n\}$

LowerUnits

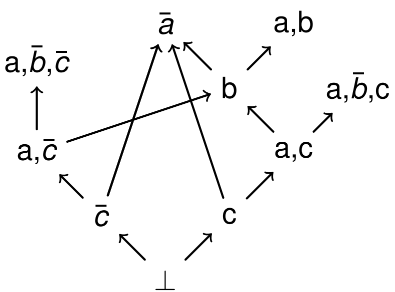
brief high level description; complexity
probably not pseudo-code

Propositional Example

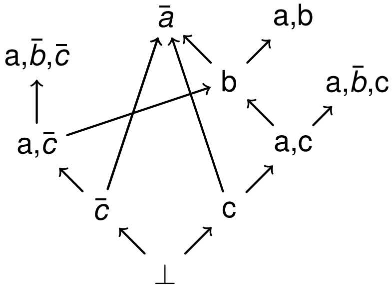


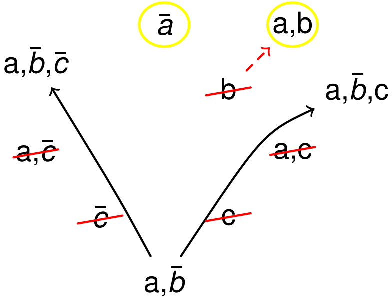


Propositional Example

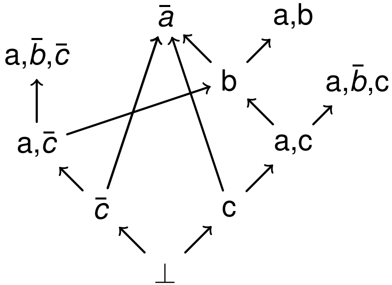


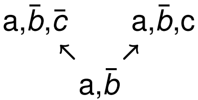
Propositional Example



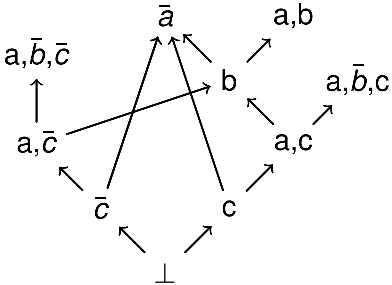


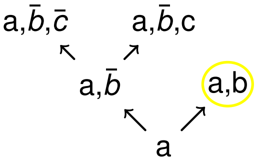
Propositional Example



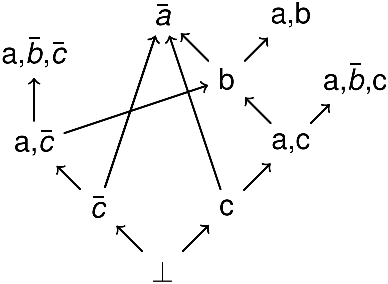


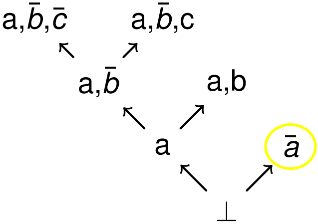
Propositional Example





Propositional Example





First Order Change: Helpful Contractions

$$
\begin{tikzcd}
\eta_2\colon \vdash p(Y) & & \eta_1\colon p(W) \vdash q(Z) & \\
& \eta_3\colon \vdash q(Z) \arrow[ul] \arrow[ur] & & \eta_4\colon p(X), q(Z) \vdash \\
& & \eta_5\colon p(X) \vdash \arrow[ul] \arrow[ur] & \\
& \bot \arrow[uuul] \arrow[ur] & &
\end{tikzcd}
$$

First Order Change: Helpful Contractions

$$
\begin{tikzcd}
\eta_2 \colon \vdash p(Y) & & \eta_1 \colon p(W) \vdash q(Z) & \\
& \eta_3 \colon \vdash q(Z) \arrow[ul] \arrow[ur] & & \eta_4 \colon p(X), q(Z) \vdash \\
& & \eta_5 \colon p(X) \vdash \arrow[ul] \arrow[ur] & \\
& \bot \arrow[uuul, bend left] \arrow[ur] & &
\end{tikzcd}
$$

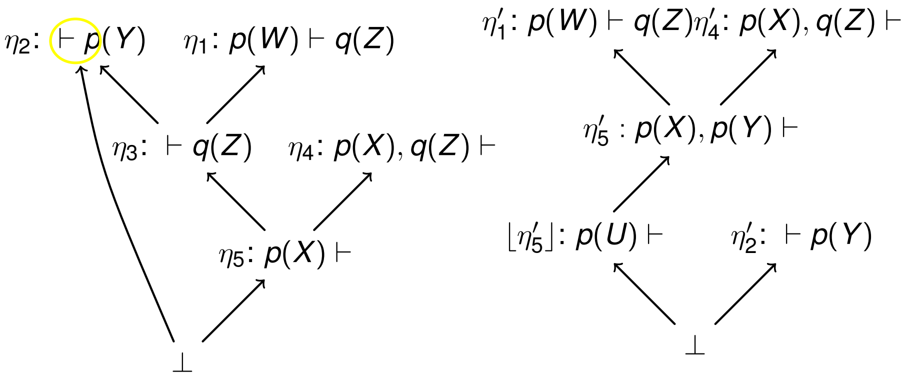
First Order Change: Helpful Contractions

$$
\begin{tikzcd}
\eta_2 \colon \vdash p(Y) & & \eta_1 \colon p(W) \vdash q(Z) & \\
& \eta_3 \colon \vdash q(Z) \arrow[ul] \arrow[ur] & & \eta_4 \colon p(X), q(Z) \vdash \\
& & \eta_5 \colon p(X) \vdash \arrow[ul] \arrow[ur] & \\
& \bot \arrow[uuul] \arrow[ur] & &
\end{tikzcd}
\qquad
\begin{tikzcd}
\eta_1' \colon p(W) \vdash q(Z) & & \eta_4' \colon p(X), q(Z) \vdash \\
& \eta_5' \colon p(X), p(Y) \vdash \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

First Order Change: Helpful Contractions

$$
\begin{tikzcd}[column sep=small]
\eta_2\colon \vdash p(Y) & & \eta_1\colon p(W) \vdash q(Z) & \\
& \eta_3\colon \vdash q(Z) \arrow[ul] \arrow[ur] & & \eta_4\colon p(X), q(Z) \vdash \\
& & \eta_5\colon p(X) \vdash \arrow[ul] \arrow[ur] & \\
& \bot \arrow[uuul, bend left] \arrow[ur] & &
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
\eta_1'\colon p(W) \vdash q(Z) & & \eta_4'\colon p(X), q(Z) \vdash \\
& \eta_5' : p(X), p(Y) \vdash \arrow[ul] \arrow[ur] & \\
\lfloor \eta_5' \rfloor\colon p(U) \vdash \arrow[ur] & &
\end{tikzcd}
$$

First Order Change: Helpful Contractions



First Order Challenge: Pre-Deletion Check

$$
\begin{tikzcd}
\eta_5\colon q(Y) \vdash p(a) & & \eta_4\colon p(X) \vdash & & \eta_3\colon \vdash p(b), q(Y) \\
& \eta_1\colon q(Y) \vdash \arrow[ul] \arrow[ur] & & \eta_2\colon \vdash q(Y) \arrow[ul] \arrow[ur] & \\
& & \bot \arrow[ul] \arrow[ur] & &
\end{tikzcd}
$$

First Order Challenge: Pre-Deletion Check

$$
\begin{tikzcd}[column sep=small]
\eta_5\colon q(Y) \vdash p(a) & & \eta_4\colon p(X) \vdash & & \eta_3\colon \vdash p(b), q(Y) \\
& \eta_1\colon q(Y) \vdash \arrow[ul] \arrow[ur] & & \eta_2\colon \vdash q(Y) \arrow[ul] \arrow[ur] & \\
& & \bot \arrow[ul] \arrow[ur] & &
\end{tikzcd}
$$

First Order Challenge: Pre-Deletion Check

$$
\begin{tikzcd}
\eta_5\colon q(Y) \vdash p(a) & & \eta_4\colon p(X) \vdash & & \eta_3\colon \vdash p(b), q(Y) \\
& \eta_1\colon q(Y) \vdash \arrow[ul] \arrow[ur] & & \eta_2\colon \vdash q(Y) \arrow[ul] \arrow[ur] & \\
& & \bot \arrow[ul] \arrow[ur] & &
\end{tikzcd}
$$

$$
\begin{tikzcd}
\eta_5'\colon q(Y) \vdash p(a) & & \eta_3'\colon \vdash p(b), q(Y) \\
& \eta\colon \vdash p(a), p(b) \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

First Order Challenge: Pre-Deletion Check

$$
\begin{tikzcd}[column sep=small]
\eta_5\colon q(Y) \vdash p(a) & & \eta_4\colon p(X) \vdash & & \eta_3\colon \vdash p(b), q(Y) \\
& \eta_1\colon q(Y) \vdash \arrow[ul] \arrow[ur] & & \eta_2\colon \vdash q(Y) \arrow[ul] \arrow[ur] & \\
& & \bot \arrow[ul] \arrow[ur] & & \\
& \eta'_5\colon q(Y) \vdash p(a) & & \eta'_3\colon \vdash p(b), q(Y) & \\
& & \eta\colon \vdash p(a), p(b) \arrow[ul] \arrow[ur] & & \\
& & \cancel{\lfloor \eta \rfloor} \arrow[u] & &
\end{tikzcd}
$$

First Order Challenge: Pre-Deletion Check

$$\begin{tikzcd}
\eta_5\colon q(Y) \vdash p(a) & & \eta_4\colon p(X) \vdash & & \eta_3\colon \vdash p(b), q(Y) \\
& \eta_1\colon q(Y) \vdash \arrow[ul] \arrow[ur] & & \eta_2\colon \vdash q(Y) \arrow[ul] \arrow[ur] & \\
& & \bot \arrow[ul] \arrow[ur] & &
\end{tikzcd}$$

Definition (Pre-Deletion Property)

η unit, $l \in \eta$, such that l is resolved with literals $l_1, \ldots, l_n$ in a proof ψ. η satisfies the *pre-deletion unifiability* property in ψ if $l_1, \ldots, l_n$ and $\bar{l}$ are unifiable.

First Order Challenge: Post-Deletion Check

$$
\begin{tikzcd}[column sep=small]
 & & \eta_1\colon r(Y), p(X, q(Y, b)), p(X, Y) \vdash & & \eta_2\colon \vdash p(U, V) \\
\eta_4\colon \vdash r(W) & & \eta_3\colon r(V), p(U, q(V, b)) \vdash \arrow[u] \arrow[urr] & & \\
 & \eta_5\colon p(U, q(W, b)) \vdash \arrow[ul] \arrow[ur] & & & \\
 & & & \perp \arrow[ull] \arrow[uuur, bend right] &
\end{tikzcd}
$$

First Order Challenge: Post-Deletion Check

$$
\begin{tikzcd}
 & \eta_1\colon r(Y), p(X, q(Y, b)), p(X, Y) \vdash & & & \eta_2\colon \vdash p(U, V) \\
\eta_4\colon \vdash r(W) & & \eta_3\colon r(V), p(U, q(V, b)) \vdash \arrow[ul] \arrow[urr] & & \\
 & \eta_5\colon p(U, q(W, b)) \vdash \arrow[ul] \arrow[ur] & & & \\
 & & & \perp \arrow[ull] \arrow[uuur, bend right] &
\end{tikzcd}
$$

First Order Challenge: Post-Deletion Check

$$
\begin{tikzcd}
 & \eta_1\colon r(Y), p(X, q(Y, b)), p(X, Y) \vdash & & \eta_2\colon \vdash p(U, V) \\
\eta_4\colon \vdash r(W) & & \eta_3\colon r(V), p(U, q(V, b)) \vdash \arrow[ul] \arrow[ur] & \\
 & \eta_5\colon p(U, q(W, b)) \vdash \arrow[ul] \arrow[ur] & & \\
 & & \perp \arrow[ul] \arrow[uuur, bend right] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
\eta_1'\colon r(Y), p(X, q(Y, b)), p(X, Y) \vdash & & \eta_4'\colon \vdash r(W) \\
 & \eta_5'\colon p(X, q(W, b)), p(X, W) \vdash \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

First Order Challenge: Post-Deletion Check

$$
\begin{tikzcd}
 & \eta_1\colon r(Y), p(X, q(Y, b)), p(X, Y) \vdash & & \eta_2\colon \vdash p(U, V) \\
\eta_4\colon \vdash r(W) & & \eta_3\colon r(V), p(U, q(V, b)) \vdash \arrow[ul] \arrow[ur] & \\
 & \eta_5\colon p(U, q(W, b)) \vdash \arrow[ul] \arrow[ur] & & \\
 & & \bot \arrow[ul] \arrow[uuur, bend right] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
\eta'_1\colon r(Y), p(X, q(Y, b)), p(X, Y) \vdash & & \eta'_4\colon \vdash r(W) \\
 & \eta'_5\colon p(X, q(W, b)), p(X, W) \vdash \arrow[ul] \arrow[ur] & \\
 & \lfloor \cancel{\eta'_5} \rfloor \arrow[u] &
\end{tikzcd}
$$

First Order Challenge: Post-Deletion Check

$$
\begin{tikzcd}
 & \eta_1\colon r(Y), p(X, q(Y, b)), p(X, Y) \vdash & & \eta_2\colon \vdash p(U, V) \\
\eta_4\colon \vdash r(W) & & \eta_3\colon r(V), p(U, q(V, b)) \vdash \arrow[ul] \arrow[ur] & \\
 & \eta_5\colon p(U, q(W, b)) \vdash \arrow[ul] \arrow[ur] & & \\
 & & \perp \arrow[ul] \arrow[uuur, bend right] &
\end{tikzcd}
$$

Definition (Post-Deletion Property)

η unit, $l \in \eta$, such that l is resolved with literals $l_1, \ldots, l_n$ in a proof ψ. η satisfies the *post-deletion unifiability* property in ψ if $l_1^{\dagger\downarrow}, \ldots, l_n^{\dagger\downarrow}$ and $\overline{l^{\dagger}}$ are unifiable, where $l^{\dagger}$ is the literal in $\psi' = \psi \setminus \{\eta\}$ corresponding to l in ψ, and $l^{\dagger\downarrow}$ is the descendant of $l^{\dagger}$ in the roof of ψ'.

First Order Lower Units Ideas/Principles

briefly mention all ideas, e.g. quadratic time naive approach to deal with both properties

Simple/Greedy First Order Lower Units

introduce simpler idea, make compromises explicit and list benefits
high level description
(probably not pseudo-code, but list # of traversals, complexity, etc)

First Order Example

small, animated example

Experiment Setup

- Simple First Order Lower units implemented as part of Skeptik (in Scala)
- 308 real first-order proofs generated by SPASS from problems from TPTP Problem Library
- proofs *generated* on cluster at the University of Victoria
- proofs *compressed* on *this laptop*

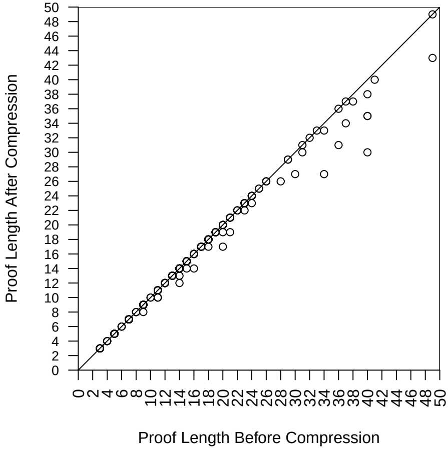
Time to generate proofs: $\approx$ 40 minutes
Time to compress proofs: $\approx$ 5 seconds

Experiment Setup

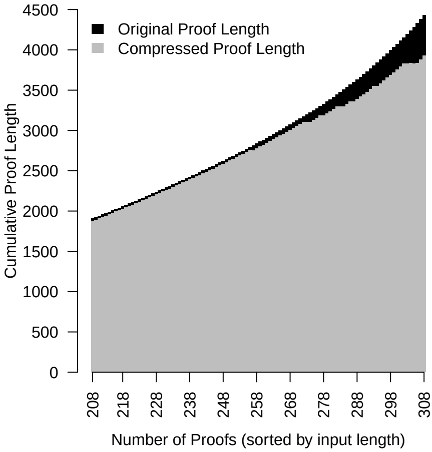
- Simple First Order Lower units implemented as part of Skeptik (in Scala)
- 308 real first-order proofs generated by SPASS from problems from TPTP Problem Library
- proofs *generated* on cluster at the University of Victoria
- proofs *compressed* on *this laptop*

Time to generate proofs: $\approx$ 40 minutes
Time to compress proofs: $\approx$ 5 seconds

Results



Proof Length After Compression
Proof Length Before Compression



Original Proof Length
Compressed Proof Length
Cumulative Proof Length
Number of Proofs (sorted by input length)

Results

Higher compression in longer proofs: 13/18 proofs with length ≥ 30 nodes successfully compressed.

Total compression ratio 11.3%: 4429 vs. 3929 nodes.
18.4% for 100 longest proofs.

Conclusion

- Simple First Order Lower Units is a quick algorithm for first order proof compression
- Future work:
  - Explore other proof compression algorithms, e.g. Recycle Pivots with Intersection
  - Explore ways of dealing with the post-deletion property quickly

Thank you for your attention.
Any questions?

- Source code: `https://github.com/jgorzny/Skeptik`
- Data: `http://www.math.uvic.ca/~jgorzny/data/`