Compression of Propositional Resolution Proofs by Lowering Subproofs

Joseph Boudou[1] Bruno Woltzenlogel Paleo[2]

[1]Université Paul Sabatier, Toulouse

[2]Vienna University of Technology

SMT Workshop, 2013

Why compressing propositional part of SMT proofs?

SMT solvers are embeded in other tools

- Sledgehammer's extension to SMT solver
- SMTCoq

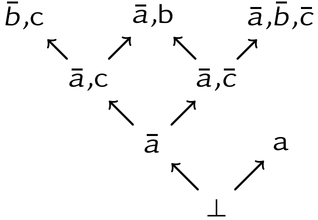
Some tools need the proof to be

- checked;
- tranlated;
- analysed.

Proof as a tree

$$\dfrac{\dfrac{\dfrac{\bar{b}, c \qquad \bar{a}, b}{\bar{a}, c}\,b \qquad \dfrac{\bar{a}, b \qquad \bar{a}, \bar{b}, \bar{c}}{\bar{a}, \bar{c}}\,\bar{b}}{\bar{a}}\,\bar{c} \qquad a}{\bot}\,a$$

Proof as a directed acyclic graph (DAG)



Proof

Definition (Proof)

A proof ψ is a directed acyclic graph

- having a root noted $\rho(\psi)$;
- with nodes labeled with clauses;
- with edges oriented from the resolvent to the premise;
- with edges labeled with the premise's literal removed in the resolvent;
- which is either an axiom or a resolution proof.

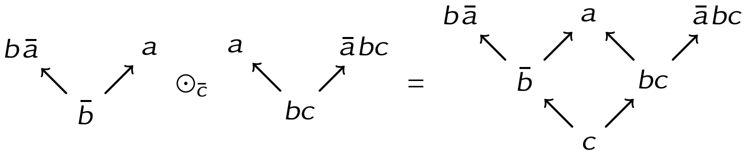
Definition (Axiom)

An axiom is a proof with only one node.

Resolution

Given two proofs φ_L and φ_R with conclusion Γ_L and Γ_R and a literal ℓ s.t. $\bar{\ell} \in \Gamma_L$ and $\ell \in \Gamma_R$, the resolution proof ψ of φ_L and φ_R on ℓ, noted $\psi = \varphi_L \odot_\ell \varphi_R$, is such that:

- ψ's nodes are the union of φ_L and φ_R nodes plus a new root node;
- there is an edge from $\rho(\psi)$ to $\rho(\varphi_L)$ labeled with $\bar{\ell}$;
- there is an edge from $\rho(\psi)$ to $\rho(\varphi_R)$ labeled with ℓ;
- ψ's conclusion is $\left(\Gamma_L \setminus \{\bar{\ell}\}\right) \cup (\Gamma_R \setminus \{\ell\})$.



Deletion

Deletion of an edge

- The resolvent is replaced by the other premise.
- Some subsequent resolutions may have to be deleted too.

Deletion of a subproof φ

- Deletion of every edge coming to $\rho(\varphi)$.
- The operation is commutative and associative.

Notation

$\psi \setminus (\varphi_1, \ldots, \varphi_n)$ denotes the deletions of subproofs $\varphi_1, \ldots, \varphi_n$ from the proof ψ.

Introduction

Redundancies and corresponding algorithms

- Vertical redundancy
- Horizontal redundancy

LowerUnivalents

Conclusion

Regular proof

Definition (Tseitin 1970)

A proof is regular iff on every path from its root to any of its axiom, any literal appears at most once as edge label.

Theorem (Goerdt 1990)

Given a set of axioms and a clause Γ*, the smallest regular proof of* Γ *might be exponentially bigger than the smallest irregular proof of* Γ*.*

RecyclePivotsWithIntersection (RPI)

Partial Regularization

- Delete an outgoing edge labeled with ℓ iff $\bar{\ell}$ appears on **every** path from the root to the node.

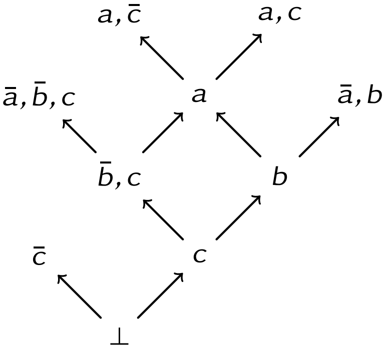
Definition (Safe literal)

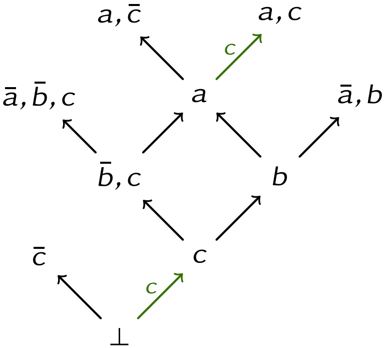
A literal is safe for a node η if it can be added to η's clause without changing proof's conclusion.

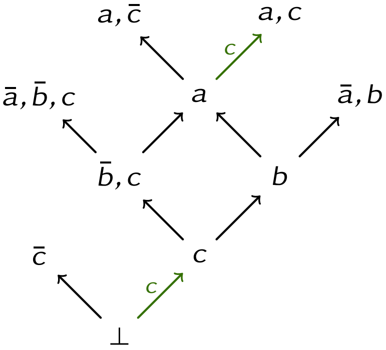
Two traversals

$\uparrow$ Collect safe literals and mark edges to be deleted.

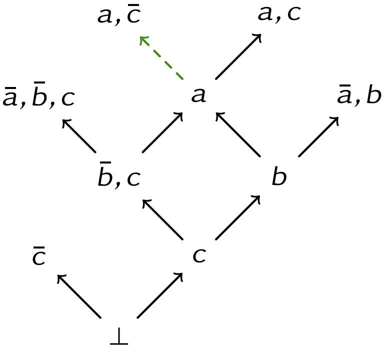
$\downarrow$ Delete edges.


$a, \bar{c}$
a, c
$\bar{a}, \bar{b}, c$
a
$\bar{a}, b$
$\bar{b}, c$
b
$\bar{c}$
c
$\bot$

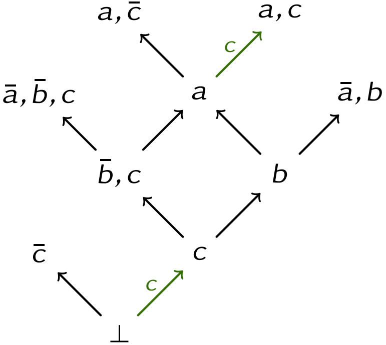

$a, \bar{c}$
a, c
c
$\bar{a}, \bar{b}, c$
a
$\bar{a}, b$
$\bar{b}, c$
b
$\bar{c}$
c
c
$\bot$



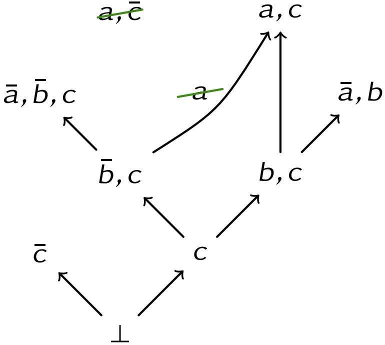
Original proof



Compressed proof



Original proof
5 resolutions



Compressed proof
4 resolutions

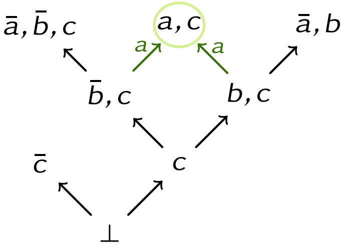
Definition

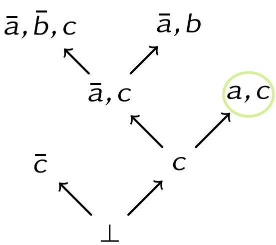
A node is an horizontal redundancy iff it has at least two incoming edges labeled with the same literal.

Reducing horizontal redundancy

- postponing resolution until resolvents are resolved.

Example





LowerUnits (LU)

Definition (Unit)

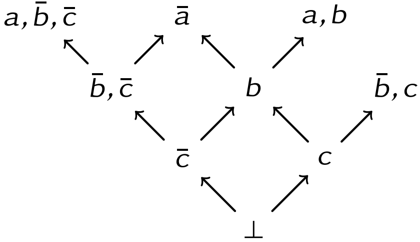
A unit is a subproof with a conclusion clause having exactly one literal.

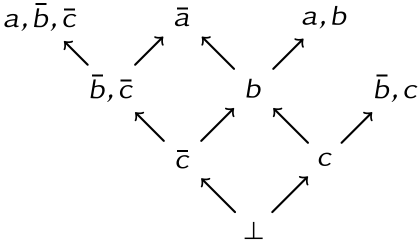
Theorem

A unit can always be lowered.

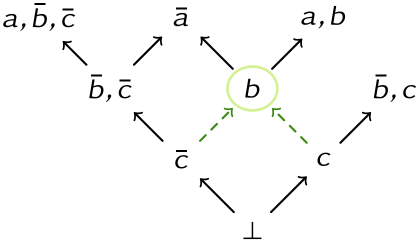
Two traversals

- ↕ Collect units with more than one resolvent.
- ↓ Delete units and reintroduce them at the bottom of the proof.

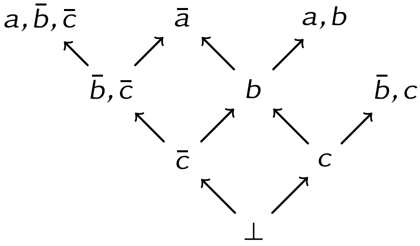

a, b̄, c̄
ā
a, b
b̄, c̄
b
b̄, c
c̄
c
⊥



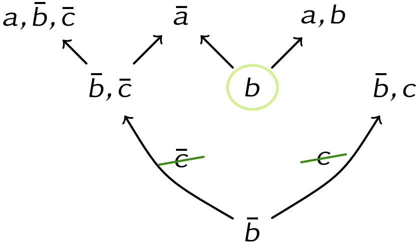
Original proof



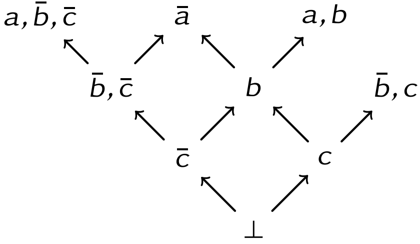
Compressed proof



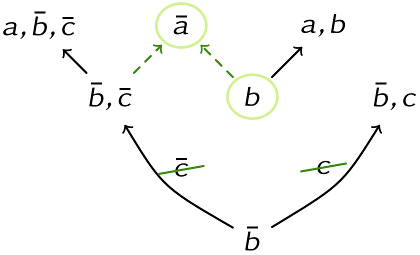
Original proof



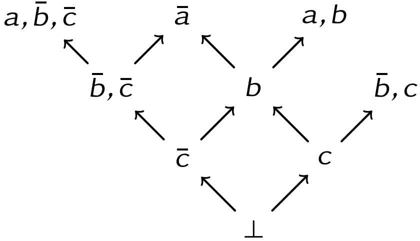
Compressed proof



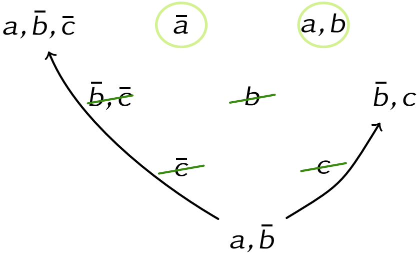
Original proof



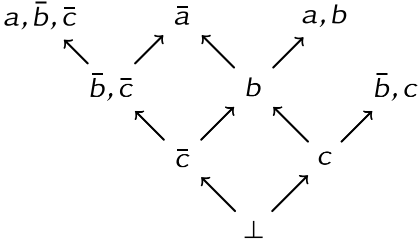
Compressed proof



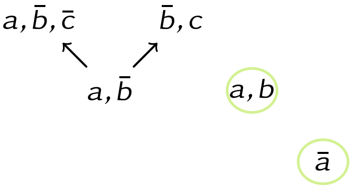
Original proof



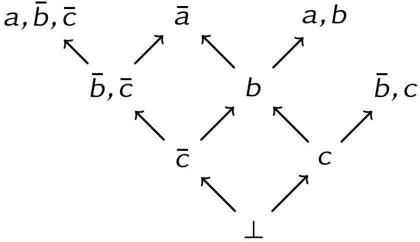
Compressed proof



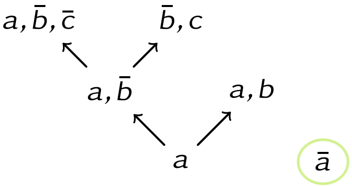
Original proof



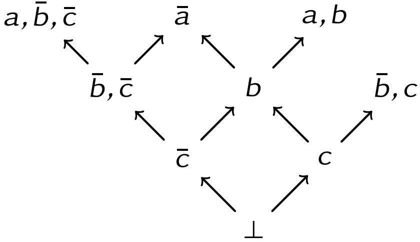
Compressed proof



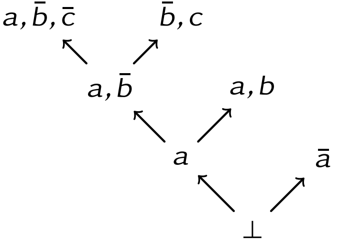
Original proof



Compressed proof



Original proof
5 resolutions



Compressed proof
3 resolutions

Goals

- Lower more subproofs.
- Allow fast combination after RPI.

Idea

- If a unit with conclusion clause $\{a\}$ is already marked for lowering, a subproof with conclusion clause $\{\bar{a}, b\}$ may be lowered too.

Definition (Valent literal)

In a proof ψ, a literal ℓ is *valent* for the subproof φ iff $\bar{\ell}$ belongs to the conclusion of $\psi \setminus (\varphi)$ but not to the conclusion of ψ.

Definition (Univalent subproof)

A subproof φ with conclusion Γ is *univalent* w.r.t. a set Δ of literals iff φ has exactly one valent literal ℓ, $\ell \notin \Delta$ and $\Gamma \subseteq \Delta \cup \{\ell\}$. ℓ is called the *univalent literal* of φ w.r.t. Δ.

Theorem

Given a proof ψ, if there is a sequence $U = (\varphi_1 \ldots \varphi_n)$ of ψ's subproofs and a sequence $(\ell_1 \ldots \ell_n)$ of literals such that $\forall i \in [1 \ldots n]$, ℓ_i is the univalent literal of φ_i w.r.t. $\Delta_{i-1} = \{\bar{\ell}_1 \ldots \bar{\ell}_{i-1}\}$, then the conclusion of

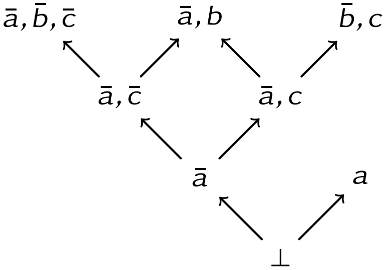
$$\psi' = \psi \setminus (U) \odot_{\ell_n} \varphi_n \ldots \odot_{\ell_1} \varphi_1$$

subsumes the conclusion of ψ.

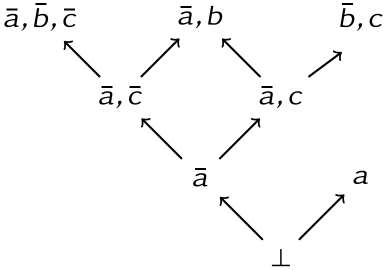
Input: a proof ψ
Output: a compressed proof ψ'

```
Univalents ← ∅ ;
Δ ← ∅ ;

for every subproof φ, in a top-down traversal do
    ψ' ← φ\Univalents ;
    if ψ' is univalent w.r.t. Δ then
        let ℓ be the univalent literal ;
        push ℓ̄ onto Δ ;
        push ψ' onto Univalents ;

// At this point, ψ' = ψ \ Univalents
while Univalents ≠ ∅ do
    φ ← pop from Univalents;
    ℓ ← pop from Δ ;
    if ℓ in the conclusion of ψ' then ψ' ← φ ⊙_ℓ ψ' ;
```

$$\Delta = \varnothing$$

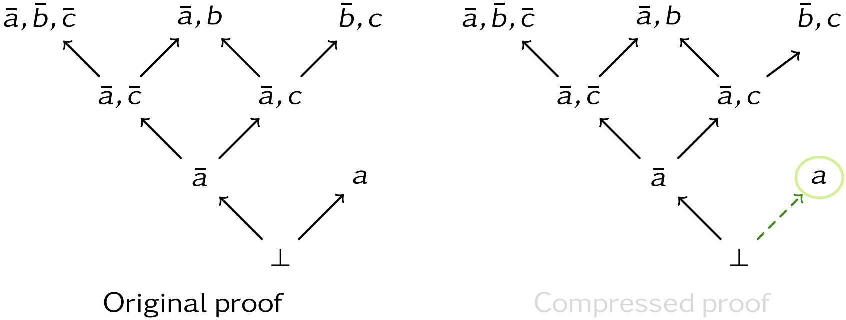


Original proof

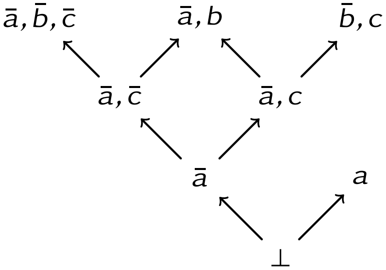


Compressed proof

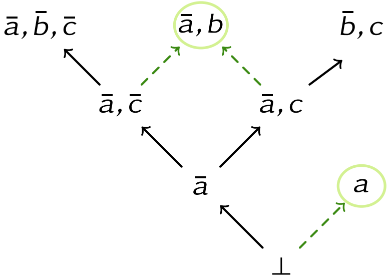
$$\Delta = \{\bar{a}\}$$


$\bar{a}, \bar{b}, \bar{c}$
$\bar{a}, b$
$\bar{b}, c$
$\bar{a}, \bar{c}$
$\bar{a}, c$
$\bar{a}$
a
$\bot$
Original proof
$\bar{a}, \bar{b}, \bar{c}$
$\bar{a}, b$
$\bar{b}, c$
$\bar{a}, \bar{c}$
$\bar{a}, c$
$\bar{a}$
a
$\bot$
Compressed proof

$$\Delta = \{\bar{a}, \bar{b}\}$$

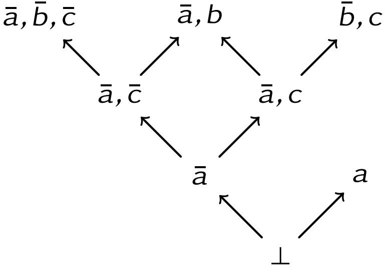


Original proof

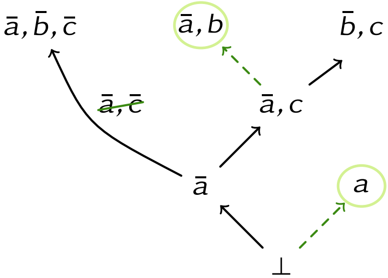


Compressed proof

$$\Delta = \{\bar{a}, \bar{b}\}$$

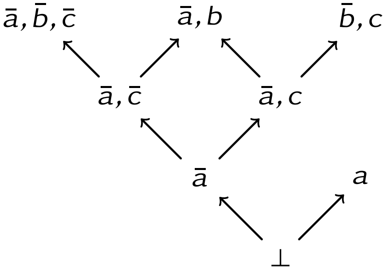


Original proof

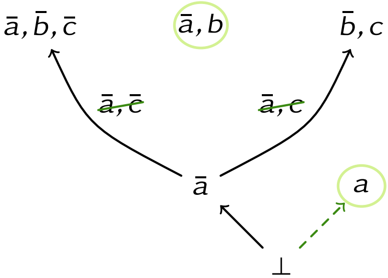


Compressed proof

$$\Delta = \{\bar{a}, \bar{b}\}$$

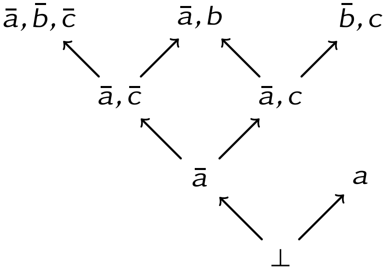


Original proof

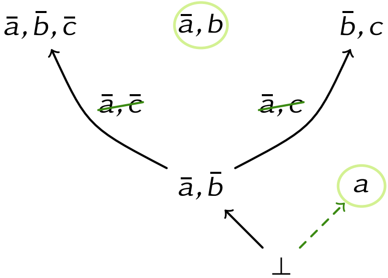


Compressed proof

$$\Delta = \{\bar{a}, \bar{b}\}$$

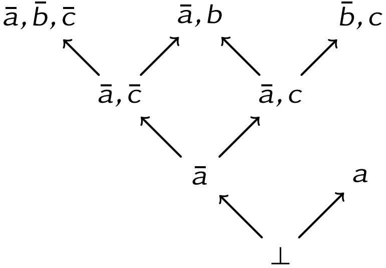


Original proof

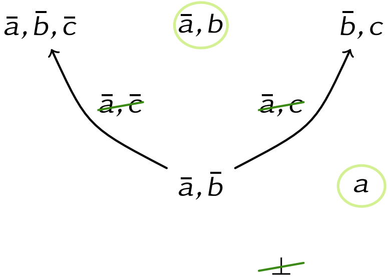


Compressed proof

$$\Delta = \{\bar{a}, \bar{b}\}$$

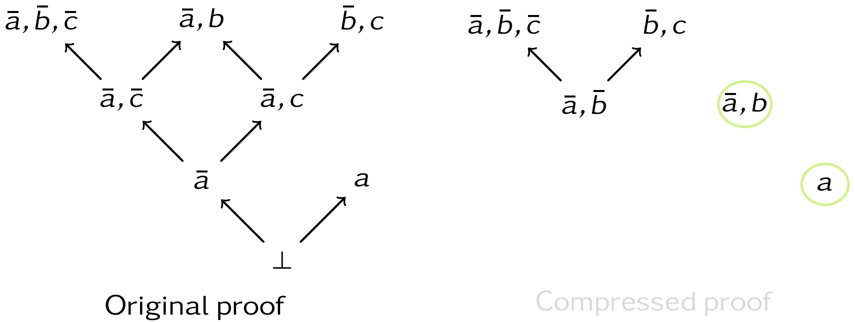


Original proof

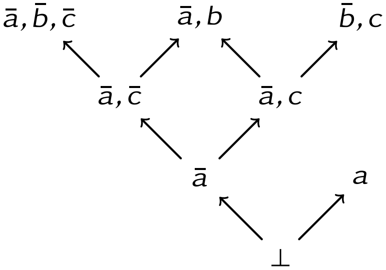


Compressed proof

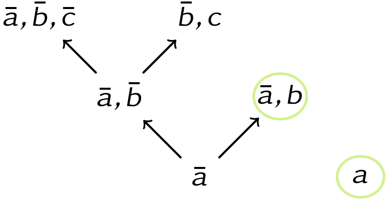
$$\Delta = \{\bar{a}, \bar{b}\}$$



$$\Delta = \{\bar{a}, \bar{b}\}$$

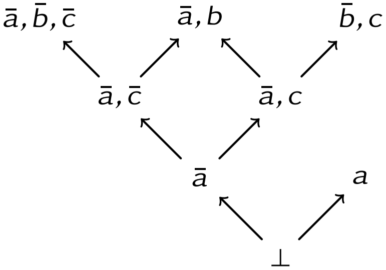


Original proof

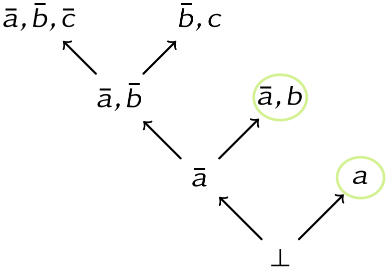


Compressed proof

$$\Delta = \{\bar{a}, \bar{b}\}$$



Original proof
4 resolutions



Compressed proof
3 resolutions

Configuration

- Algorithms implemented in Scala for the Skeptik library.
- 5 000 SMT proofs produced by the VeriT solver.
- Experiments performed on the Vienna Scientific Cluster.

Results

Algorithm	Compression	Speed
LowerUnits	7.5 %	22.4 n/ms
LowerUnivalents	8.0 %	20.4 n/ms
LU composed after RPI	21.7 %	15.1 n/ms
LUniv combined after RPI	22.0 %	17.8 n/ms

Goals achieved

- LowerUnivalents compresses more than LowerUnits.
- LowerUnivalents combines efficiently after RPI.

Future works

- Combine LowerUnivalents after other algorithms.
- Get rid of order dependency.
- Lower subproofs to the middle of the proof.
- Explore other kinds of redundancies.

Thank you for your attention.

Any question ?

Skeptik

- http://github.com/Paradoxika/Skeptik