Combining RecyclePivotsWithIntersection and LowerUnits

Joseph Boudou, Bruno Woltzenlogel Paleo

October 2012

Overview

Benchmarks

Algorithms implemented in Scala for Skeptik

1000 proofs from VeriT

- 100 SAT proofs from the old external SAT solver
- 900 SMT proofs translated to propositional resolution

Propositional resolution calculus

Conventions

- Clauses are sets of literals.
- $\bar{a}$ is the dual of a.
- Let η_Γ be a proof of clause Γ and η_Δ a proof of Δ.
- If $\bar{a} \in \Gamma$ and $a \in \Delta$ then $\eta = \eta_\Gamma \odot_a \eta_\Delta$ is a proof of $(\Gamma \cup \Delta) \setminus \{a, \bar{a}\}$.
- a is the *pivot* of η.
- η_Γ and η_Δ are the *premises* of η.
- η is a *child* of both η_Γ and η_Δ.

Proof as DAG

- A node can have more than one child.

Regular proof

Definition (Regular proof)

A proof is said to be regular if on every path from its root to any of its axiom, each pivot appears only once.

Regular proof

Definition (Regular proof)

A proof is said to be regular if on every path from its root to any of its axiom, each pivot appears only once.

Theorem (Tseitin)

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

Extending Irregularity

Definition (Fully regular proof)

A proof is fully regular if for each variable there is at most one resolution node with this variable as pivot.

Conventions

- Usual irregularities are called *vertical irregularities*.
- Other irregularities are called *horizontal irregularities*.

RecyclePivotsWithIntersection (RPI)

Partial Vertical Regularization

- Delete a branch only if the pivot 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 the proof's conclusion (root's clause).

Two traversals

$\uparrow$ Collect safe literals and mark edges to delete.

$\downarrow$ Delete edges and fix the proof.

LowerUnits (LU)

Lowering

- Moving a node down the proof to resolve it only once.

Lowering Units

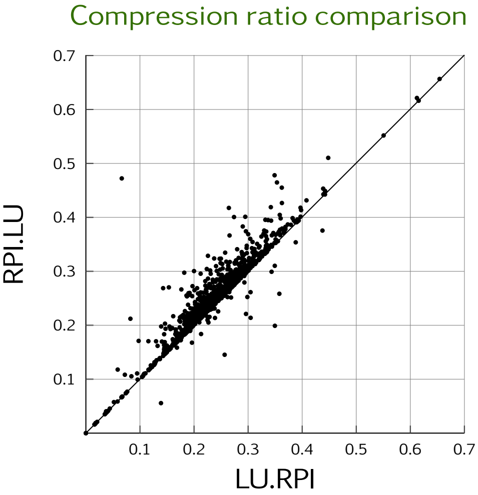
- Units can always be lowered.
- Reduces horizontal irregularities.

Two traversals

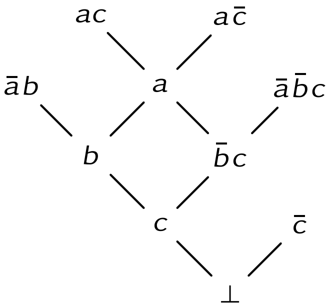
$\updownarrow$ Collect units with more than one child.

$\downarrow$ Delete units, fix the proof and then reintroduce the units at the bottom of the proof.

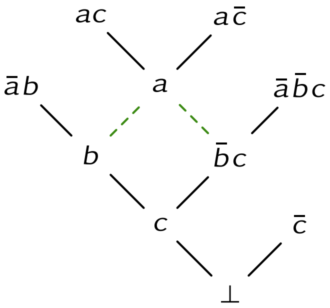
Sequential Composition



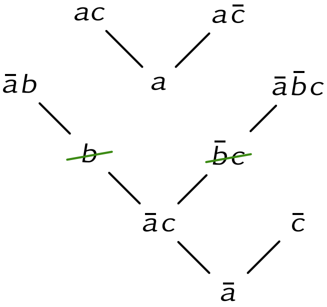
Irregular Units



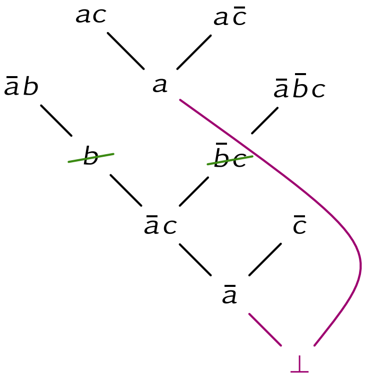
Irregular Units



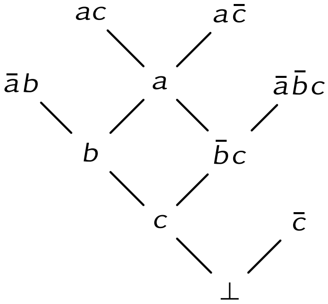
Irregular Units



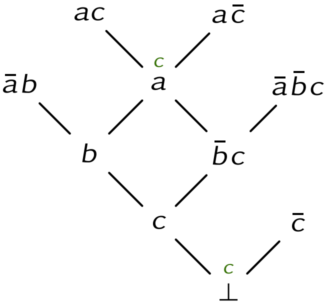
Irregular Units



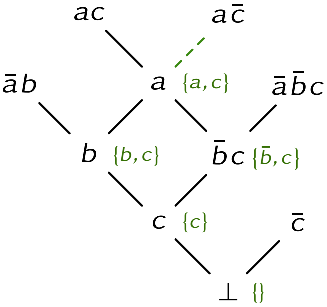
Irregular Units



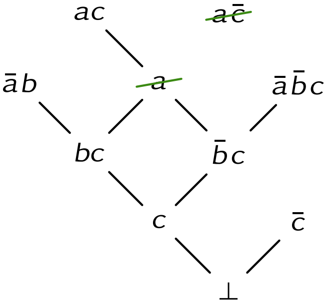
Irregular Units



Irregular Units



Irregular Units



Root and Units Safe Literals

In RPI.LU, after LU :

- the proof is of the form $\eta \odot_{a_0} \eta_0 \odot_{a_1} \cdots \odot_{a_n} \eta_n$;
- $\{\bar{a}_i \mid i \leq n\}$ is the safe literals for η : *the root's safe literals* ;
- $\forall i < n, \left\{\bar{a}_j \mid i < j \leq n\right\}$ is the safe literals for η_i.

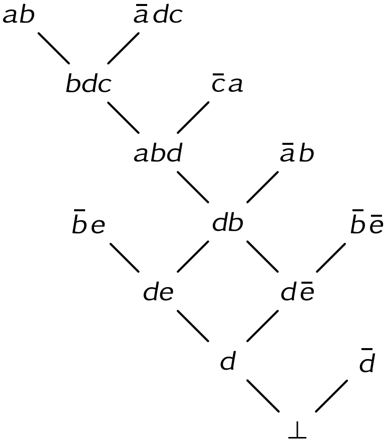
Root and Units Safe Literals

In RPI.LU, after LU :

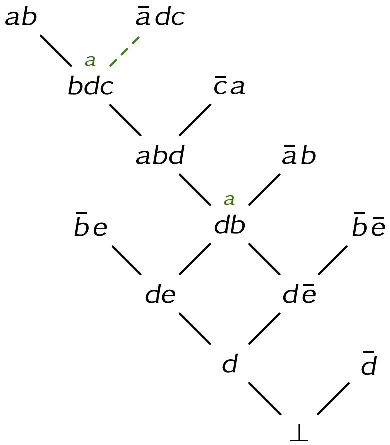
- the proof is of the form $\eta \odot_{a_0} \eta_0 \odot_{a_1} \cdots \odot_{a_n} \eta_n$;
- $\{\bar{a}_i \mid i \leq n\}$ is the safe literals for η : *the root's safe literals* ;
- $\forall i < n, \left\{\bar{a}_j \mid i < j \leq n\right\}$ is the safe literals for η_i.

For a combined algorithm to be always at least as good as RPI.LU it has to compute root and units safe literals.

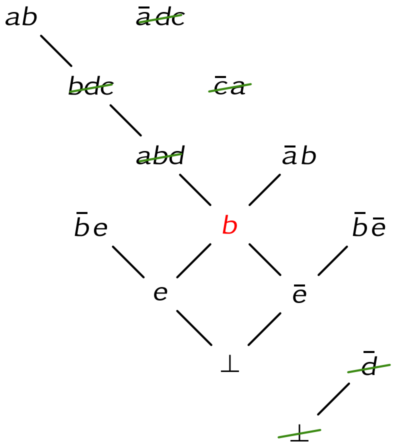
Units introduced by RPI



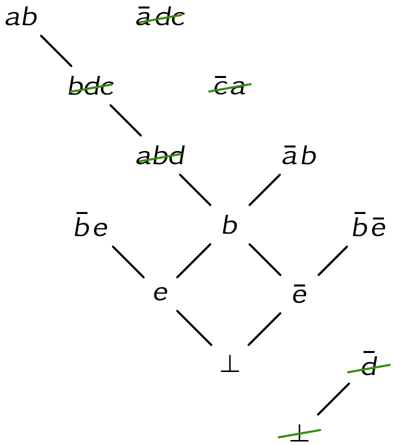
Units introduced by RPI



Units introduced by RPI



Units introduced by RPI



For a combined algorithm to be always at least as good as LU.RPI it has to be able to lower units introduced by RPI.

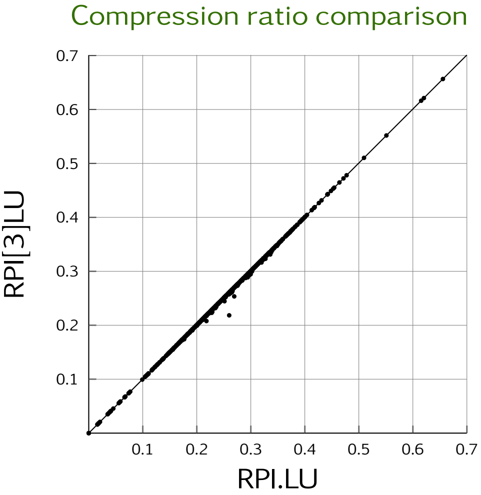
RPI[3]LU

For a combined algorithm to be always at least as good as RPI.LU it has to compute root and units safe literals.

Three traversals

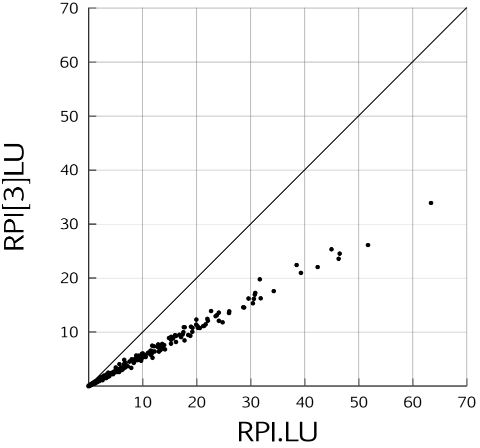
- ↓ collect units and compute root and units safe literals ;
- ↑ compute safe literals and mark edges to be deleted ;
- ↓ fix the proof and reintroduce units.

RPI[3]LU vs RPI.LU



RPI[3]LU vs RPI.LU

Compression time comparison (in seconds)



LUnivRPI

For a combined algorithm to be always at least as good as LU.RPI it has to be able to lower units introduced by RPI.

- Collect units during fixing (top-down traversal).

LUnivRPI

For a combined algorithm to be always at least as good as LU.RPI it has to be able to lower units introduced by RPI.

- Collect units during fixing (top-down traversal).

Problems

- If a unit $\{a\}$ depends on a unit $\{b\}$ it'll be seen as $\{a, \bar{b}\}$.

LUnivRPI

For a combined algorithm to be always at least as good as LU.RPI it has to be able to lower units introduced by RPI.

- Collect units during fixing (top-down traversal).

Problems

- If a unit $\{a\}$ depends on a unit $\{b\}$ it'll be seen as $\{a, \bar{b}\}$.
- What to do if we find $\{a, \bar{b}\}$ after $\{b\}$?

LUnivRPI

For a combined algorithm to be always at least as good as LU.RPI it has to be able to lower units introduced by RPI.

- Collect units during fixing (top-down traversal).

Problems

- If a unit $\{a\}$ depends on a unit $\{b\}$ it'll be seen as $\{a, \bar{b}\}$.
- What to do if we find $\{a, \bar{b}\}$ after $\{b\}$?
- A new algorithm extending LU is needed.

Lowering a node

The (generalized) problem

- Given $\psi[\eta] \odot_{a_0} \eta_0 \odot_{a_1} \cdots \odot_{a_{n-1}} \eta_{n-1}$
- is $\mathsf{Fix}(\psi[]) \odot_a \eta \odot_{a_0} \eta_0 \odot_{a_1} \cdots \odot_{a_{n-1}} \eta_{n-1}$ equivalent ?

Two steps

- Deleting the node : $\mathsf{Fix}(\psi[])$;
- Reintroducing it : $\odot_a \eta$.

Beware of introduced literals

- $\Delta = \{\bar{a}_i \mid i < n\}$ is the safe literals of $\mathsf{Fix}(\psi[]) \odot_a \eta$.

Conditions

Literals introduced by reintroducing the node

- Let Γ_+ be η's clause,
- $\Gamma \setminus \Delta = \{a\}$.

Definition (Active literal)

Let's consider a node η with clause Γ_+. A literal a from Γ_+ is said to be an active literal of η iff a is the pivot of one of η's child.

Literals introduced by deleting the node

- Let Γ_- be the set of the duals of η's active literals,
- $\Gamma_- \setminus \Delta = \{\bar{a}\}$.

Partial regularization

Deletable node

- If $\Gamma_{-} \setminus \Delta = \varnothing$ then delete η.

Partial regularization

Deletable node

- If $\Gamma_{-} \setminus \Delta = \varnothing$ then delete η.

Partial regularization

- If the dual of any of η's active literal belongs to Δ then delete the edge.

Partial regularization

Deletable node (implemented)

- If $\Gamma_{-} \setminus \Delta = \varnothing$ then delete η.

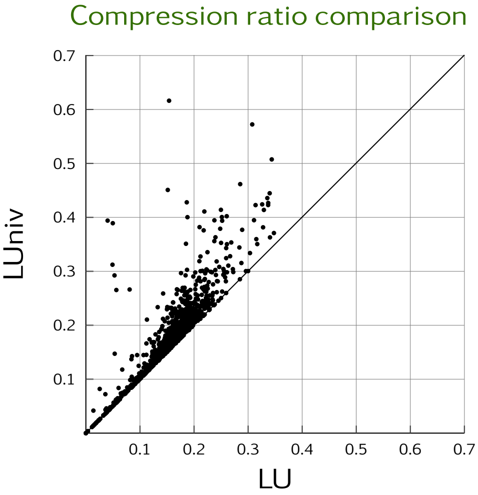
Partial regularization (not implemented)

- If the dual of any of η's active literal belongs to Δ then delete the edge.

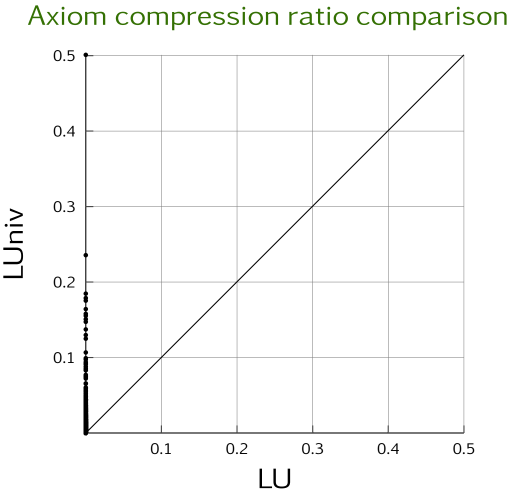
Algorithm

$\Delta \leftarrow \emptyset$;
for *every node* η *in a top-down traversal* **do**
    Fix η ;
    Compute $\Gamma_{-} \setminus \Delta$;
    if $\Gamma_{-} \setminus \Delta = \emptyset$ **then**
        Delete η ;
    else if $\Gamma_{-} \setminus \Delta = \{\bar{a}\}$ **and** $\Gamma_{+} \setminus \Delta = \{a\}$ **then**
        Lower η ;
        $\Delta \leftarrow \Delta \cup \{a\}$;

Comparison with LU

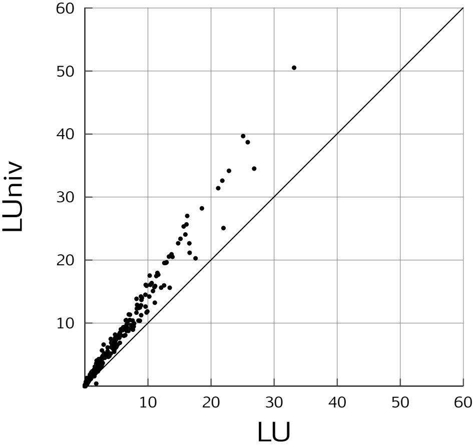


Comparison with LU



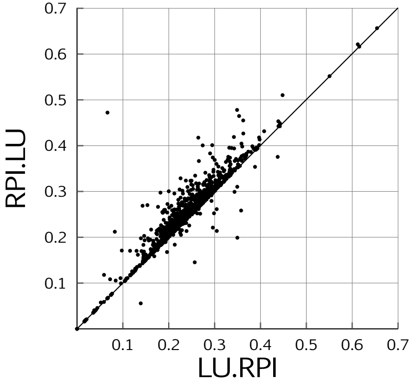
Comparison with LU

Compression time comparison (in seconds)



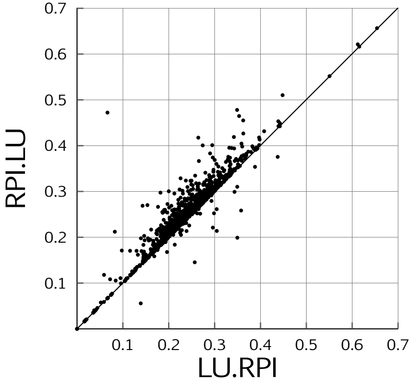
Conclusion

Compression ratio comparison



Conclusion

Compression ratio comparison



References

Skeptik

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

Bibliography

- Fontaine, P., Merz, S., Woltzenlogel Paleo, B.: Compression of propositional resolution proofs via partial regularization. In: CADE. Lecture Notes in Computer Science, vol. 6803, pp. 237–251. Springer (2011)