An Efficient and Flexible Approach to Resolution Proof Reduction

N. Sharygina

Formal Verification and Security Group

University of Lugano

March 9, 2011

Outline

Outline

Outline

Outline

Outline

Background

Formal Verification in Lugano, Switzerland

- Program Verification

Background

Formal Verification in Lugano, Switzerland

- Program Verification
  - Model checking code (LoopFrog, Synergy, SatAbs (with Oxford), FunFrog), ANSI-C
  - Efficient decision procedures as computational engines of verification (OpenSMT)

Background

Formal Verification in Lugano, Switzerland

- Program Verification
  - Model checking code (LoopFrog, Synergy, SatAbs (with Oxford), FunFrog), ANSI-C
  - Efficient decision procedures as computational engines of verification (OpenSMT)
- Abstractions

Background

Formal Verification in Lugano, Switzerland

- Program Verification
  - Model checking code (LoopFrog, Synergy, SatAbs (with Oxford), FunFrog), ANSI-C
  - Efficient decision procedures as computational engines of verification (OpenSMT)

- Abstractions
  - Program Summarization [ATVA'08], [ASE'09]
    - Avoids fix-point computation by constructing symbolic abstract transformers instead
    - Performs sound over-approximation of (unbounded) loops
    - Precision is tuned by selection of abstract domains
    - Exploits efficiency of SAT/SMT solvers

- Program Termination [CAV'10, TACAS'11]
  - Integration of Loop Summarization with Termination Analysis
  - Compositional Transition Invariants avoid all paths computation of termination checks
  - Simple abstract domains are used for termination checks

Background

Formal Verification in Lugano, Switzerland

- Program Termination [CAV'10, TACAS'11]
  - Integration of Loop Summarization with Termination Analysis
  - Compositional Transition Invariants avoid all paths computation of termination checks
  - Simple abstract domains are used for termination checks
- Synergy of Abstractions [STTT'10]
  - Interleaves precise and over-approximated abstractions
  - Reduces CEGAR iterations
  - Removes multiple counterexamples within a single refinement step
  - Localizes precise abstraction/refinement to relevant parts of the program

Background

Formal Verification in Lugano, Switzerland

- Model checking mobile code [IFM'08], [JFAC'10]
  - Specification language for security policies
  - Formalization of mobile code distribution net
  - Location-specific abstractions and model checking of security policies

Background

Formal Verification in Lugano, Switzerland

- Model checking mobile code [IFM'08], [JFAC'10]
  - Specification language for security policies
  - Formalization of mobile code distribution net
  - Location-specific abstractions and model checking of security policies
- Boolean and Theory Reasoning (SMT)
  - Procedure for bit-vector extraction and concatenation [ICCAD'09]
    - Reduces formulae to the theory of equality to avoid, when possible, expensive reduction to SAT

Background

Formal Verification in Lugano, Switzerland

- Model checking mobile code [IFM'08], [JFAC'10]
  - Specification language for security policies
  - Formalization of mobile code distribution net
  - Location-specific abstractions and model checking of security policies
- Boolean and Theory Reasoning (SMT)
  - Procedure for bit-vector extraction and concatenation [ICCAD'09]
    - Reduces formulae to the theory of equality to avoid, when possible, expensive reduction to SAT
  - Generation of explanations in theory propagation [MEMOCODE'10]
    - Computes explanations on demand by reusing the consistency check algorithm for a generic theory T.

Background

Formal Verification in Lugano, Switzerland

- Boolean and Theory Reasoning (SMT)
  - Generation of interpolants (for QF EUF, RDL)
  - Proof manipulation for interpolation [ICCAD'10]
  - Proof reduction [HVC'10]

- Boolean and Theory Reasoning (SMT)
  - Generation of interpolants (for QF EUF, RDL)
  - Proof manipulation for interpolation [ICCAD'10]
  - Proof reduction [HVC'10]
  - Solver, *OpenSMT*, combines MiniSAT2 SAT-Solver with state-of-the-art decision procedures for QF EUF, LRA, BV, RDL, IDL
    - *Extensible*: the SAT-to-theory interface facilites design and plug-in of new decision procedures
    - *Incremental*: suitable for incremental verification
    - *Open-source*: available under GPL license
    - *Efficient*: currently the fastest open-source SMT Solver for QF UF, IDL, RDL, LRA according to SMT-Comp'10.

Background

Formal Verification in Lugano, Switzerland

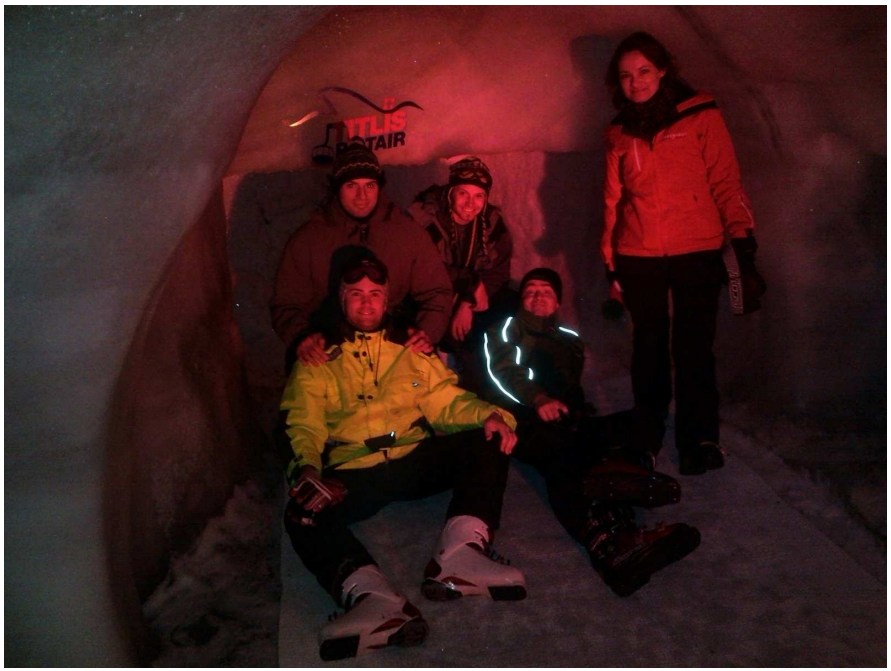


Figure: Working Hard

- Boolean and Theory Reasoning (SMT)
  - Generation of interpolants (for QF EUF, RDL)
  - Proof manipulation for interpolation [ICCAD'10]
  - Resolution proof reduction [S.F. Rollini, R. Bruttomesso, N. Sharygina, HVC'10]

Outline

Proof Reduction

Motivation

- Resolution proofs find application in several ambits

Proof Reduction

Motivation

- Resolution proofs find application in several ambits
  - Interpolation-based model checking
  - Abstraction techniques
  - Unsatisfiable core extraction in SAT/SMT
  - Automatic theorem proving

Proof Reduction

Motivation

- Resolution proofs find application in several ambits
  - Interpolation-based model checking
  - Abstraction techniques
  - Unsatisfiable core extraction in SAT/SMT
  - Automatic theorem proving
- Problems

Proof Reduction

Motivation

- Resolution proofs find application in several ambits
  - Interpolation-based model checking
  - Abstraction techniques
  - Unsatisfiable core extraction in SAT/SMT
  - Automatic theorem proving

- Problems
  - Size affects efficiency
  - Size can be exponential w.r.t. input formula

Proof Reduction

Motivation

- Resolution proofs find application in several ambits
  - Interpolation-based model checking
  - Abstraction techniques
  - Unsatisfiable core extraction in SAT/SMT
  - Automatic theorem proving
- Problems
  - Size affects efficiency
  - Size can be exponential w.r.t. input formula
- Reduction/compression of resolution proofs is important

Related Work

Features

- Post-processing approach

Related Work

Features

- Post-processing approach
- SAT/SMT solving framework

Related Work

Features

- Post-processing approach
- SAT/SMT solving framework

Related Work

Features

- Post-processing approach
- SAT/SMT solving framework
- Compression techniques

Related Work

Features

- Post-processing approach
- SAT/SMT solving framework
- Compression techniques
  - Clauses subsumption checking [Amjad07]
  - Proof reordering based on literals linking [Amjad07]
  - Proof reordering based on variable splitting [Cotton10]
  - Merging of shared substructures in subproofs [Sinz07]
  - Memoization of shared substructures [Amjad08,Cotton10]
  - Algebraic approach, resolution hypergraphs [Fontaine10]
  - Removal pivots redundancies along paths [Bar-Ilan08]

Notation

Resolution System

- Literal $p \quad \overline{p}$

Notation

Resolution System

- Literal $p \quad \overline{p}$
- Clause $p \vee \overline{q} \vee r \vee \ldots \quad \rightarrow \quad p\overline{q}r \ldots$
- Empty clause $\perp$

Notation

Resolution System

- Literal $p \quad \overline{p}$
- Clause $p \vee \overline{q} \vee r \vee \ldots \quad \rightarrow \quad p\overline{q}r \ldots$
- Empty clause $\bot$
- Resolution rule $\dfrac{pC \qquad \overline{p}D}{CD}\, p$

  Antecedent Resolvent Pivot

Notation

Resolution System

- Literal $p \quad \overline{p}$
- Clause $p \vee \overline{q} \vee r \vee \ldots \quad \rightarrow \quad p\overline{q}r \ldots$
- Empty clause $\bot$
- Resolution rule

$$\frac{pC \qquad \overline{p}D}{CD}\, p$$

Antecedent Resolvent Pivot

- Resolution proof of unsatisfiability of a set of clauses S

Notation

Resolution System

- Literal $p \quad \overline{p}$
- Clause $p \vee \overline{q} \vee r \vee \ldots \quad \rightarrow \quad p\overline{q}r \ldots$
- Empty clause $\bot$
- Resolution rule $\dfrac{pC \qquad \overline{p}D}{CD}\, p$

  Antecedent Resolvent Pivot

- Resolution proof of unsatisfiability of a set of clauses S
  - Tree
  - Leaves as clauses of S
  - Intermediate nodes as resolvents
  - Root as unique empty clause

Resolution Proofs

Example

- Set of clauses $\{\overline{pq}, p\overline{q}, q\overline{r}, qr\}$
- Proof of unsatisfiability

$$\dfrac{\dfrac{\dfrac{\mathbf{\overline{pq}} \qquad \mathbf{p\overline{q}}}{\overline{q}}\, p \qquad \dfrac{\mathbf{q\overline{r}} \qquad \mathbf{qr}}{q}\, r}{q}\, q}{\bot}$$

Pivots Redundancies [Bar-Ilan08]

- No need to resolve more than once on a pivot in a path leaf-root

Pivots Redundancies [Bar-Ilan08]

- No need to resolve more than once on a pivot in a path leaf-root

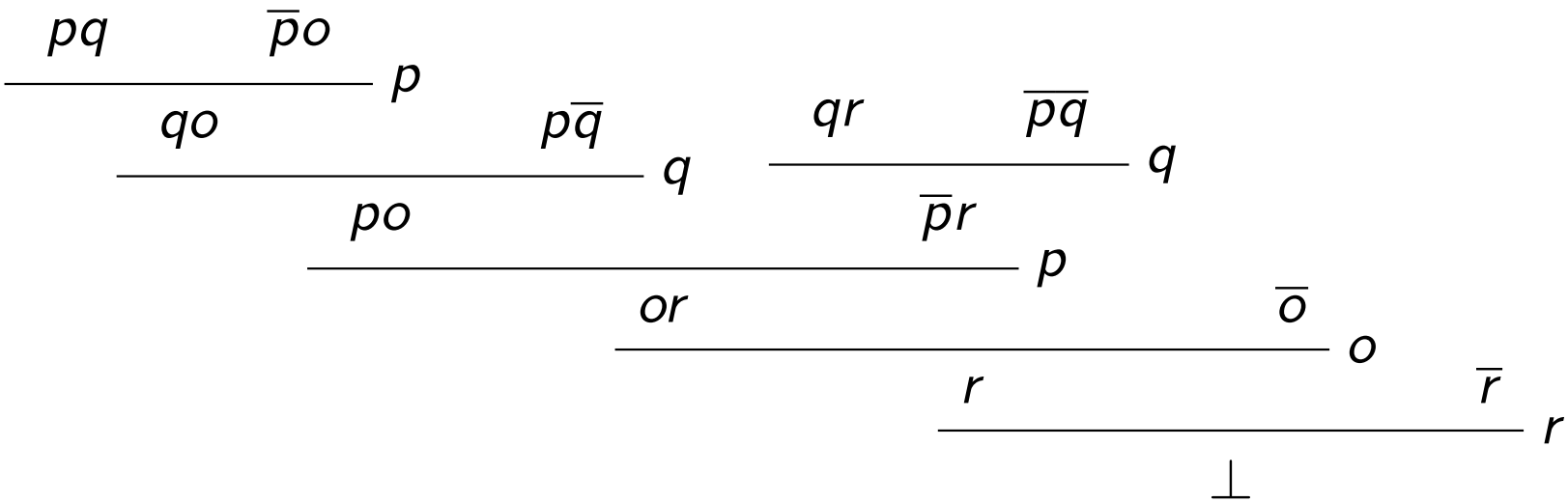
- O.Bar-Ilan, O.Fuhrmann, S.Hoory, O.Shacham and O.Strichman: RecyclePivots

Pivots Redundancies [Bar-Ilan08]

- No need to resolve more than once on a pivot in a path leaf-root
- O.Bar-Ilan, O.Fuhrmann, S.Hoory, O.Shacham and O.Strichman: RecyclePivots
  - Perform DFS from root to leaves
  - Track pivots occurrences along paths
  - In case of multiple occurrences keep the closest one to root
  - Output regular proof

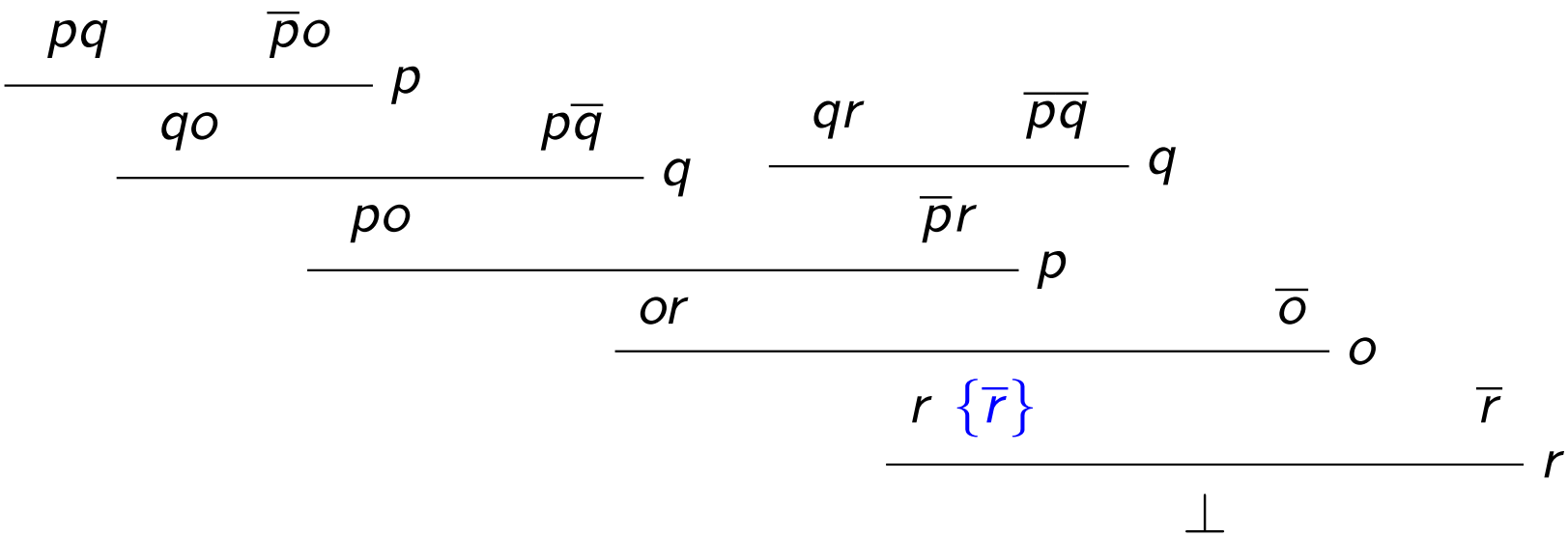
RecyclePivots

Example



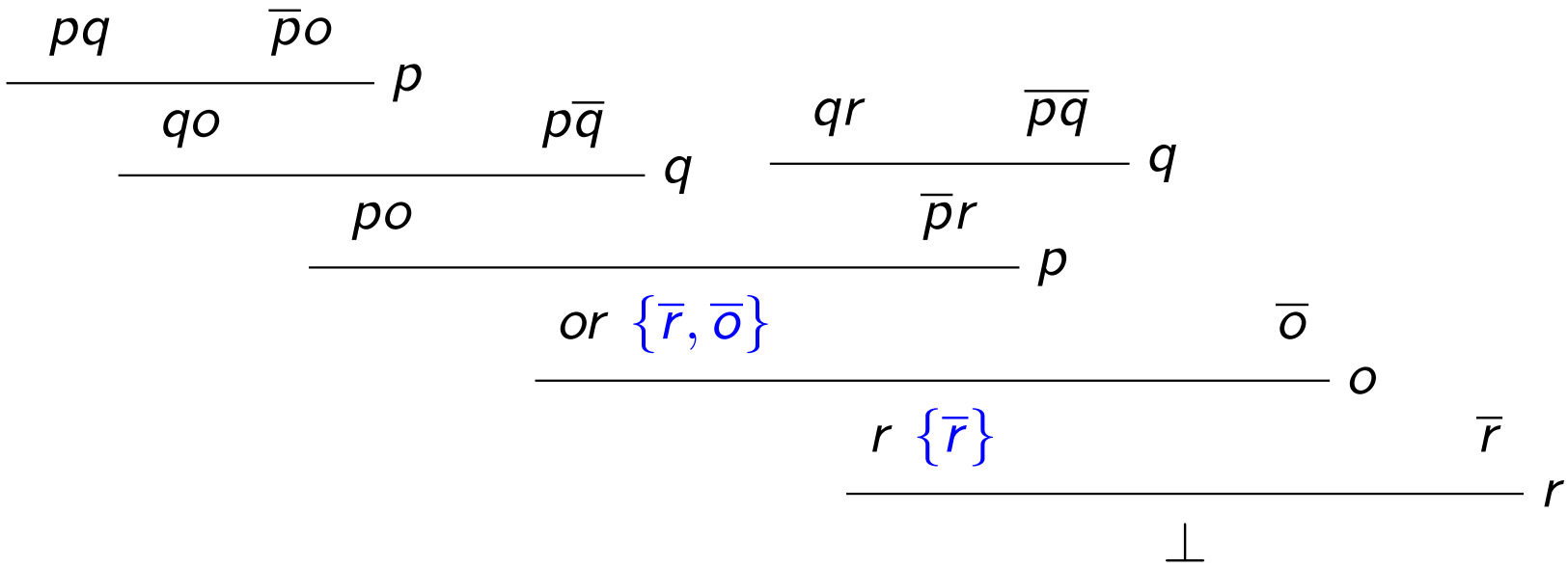
RecyclePivots

Example



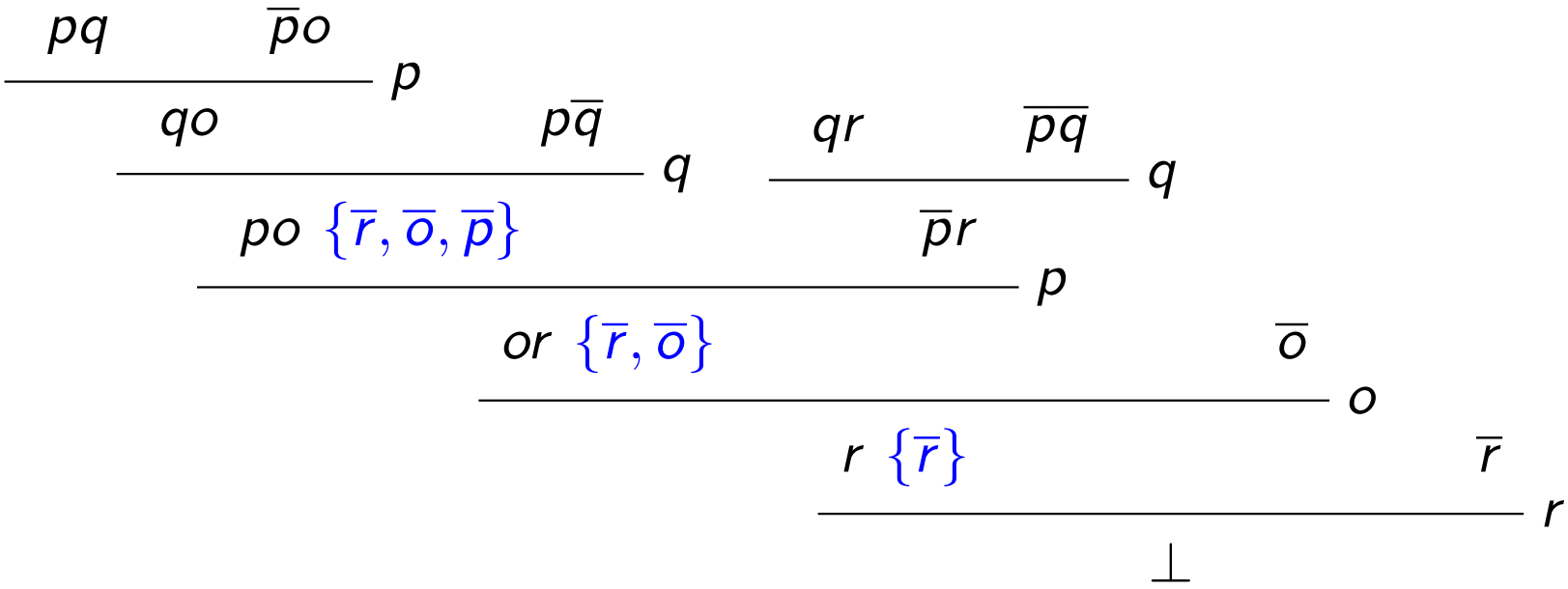
RecyclePivots

Example



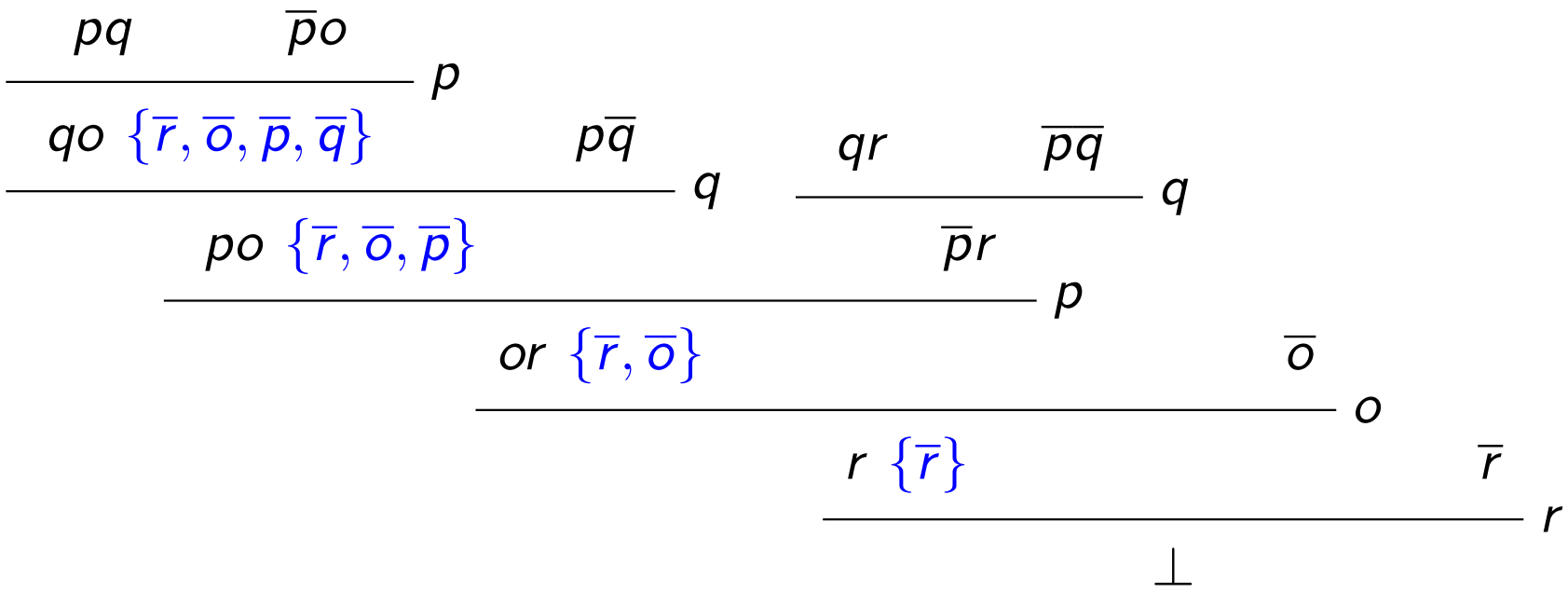
RecyclePivots

Example



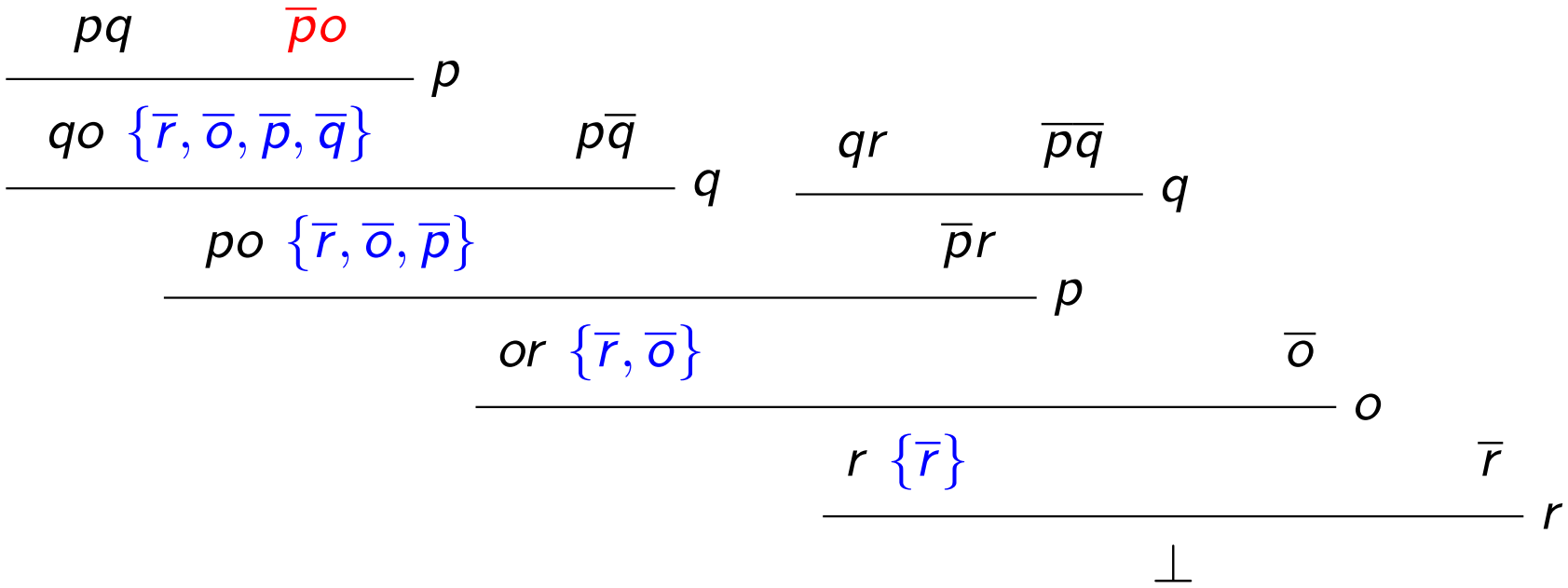
RecyclePivots

Example



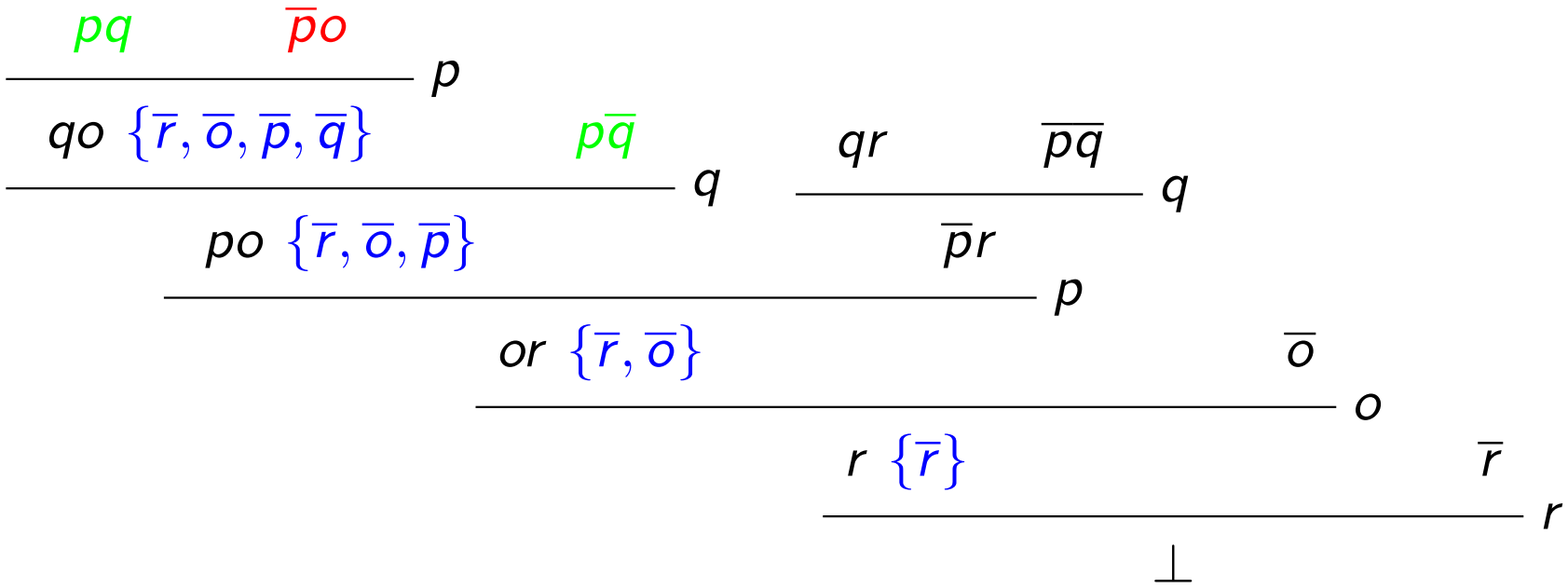
RecyclePivots

Example



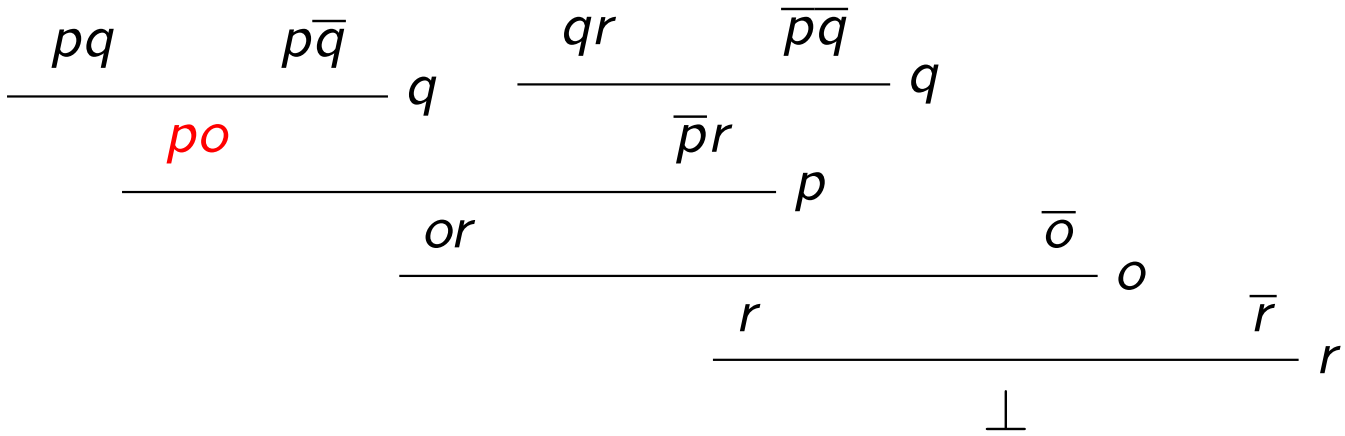
RecyclePivots

Example



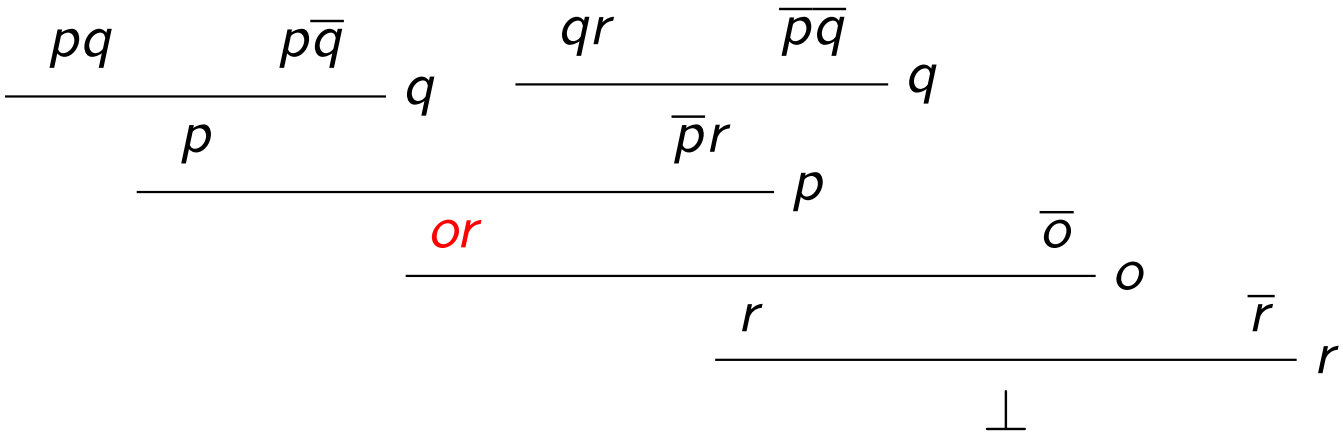
RecyclePivots

Example



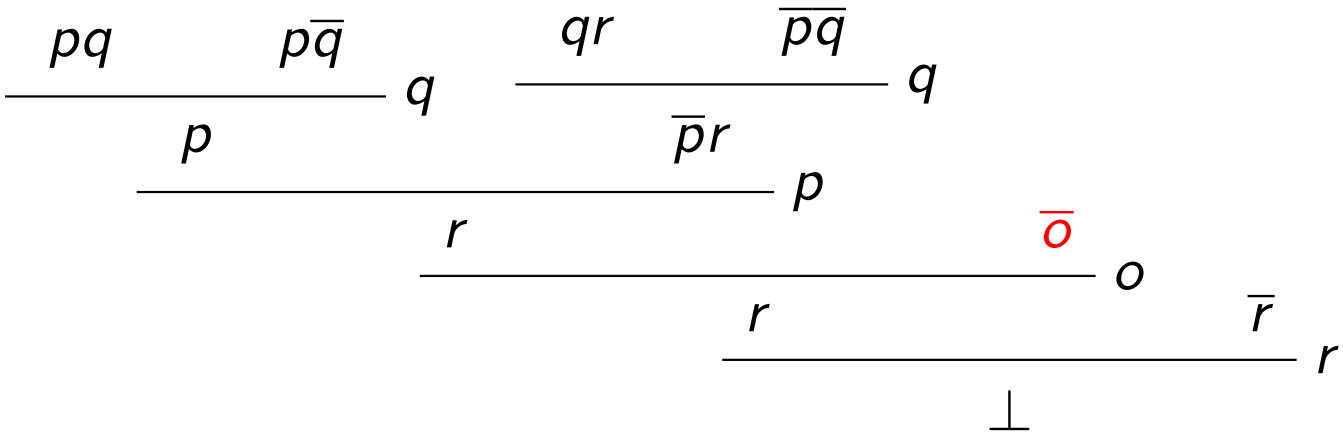
RecyclePivots

Example



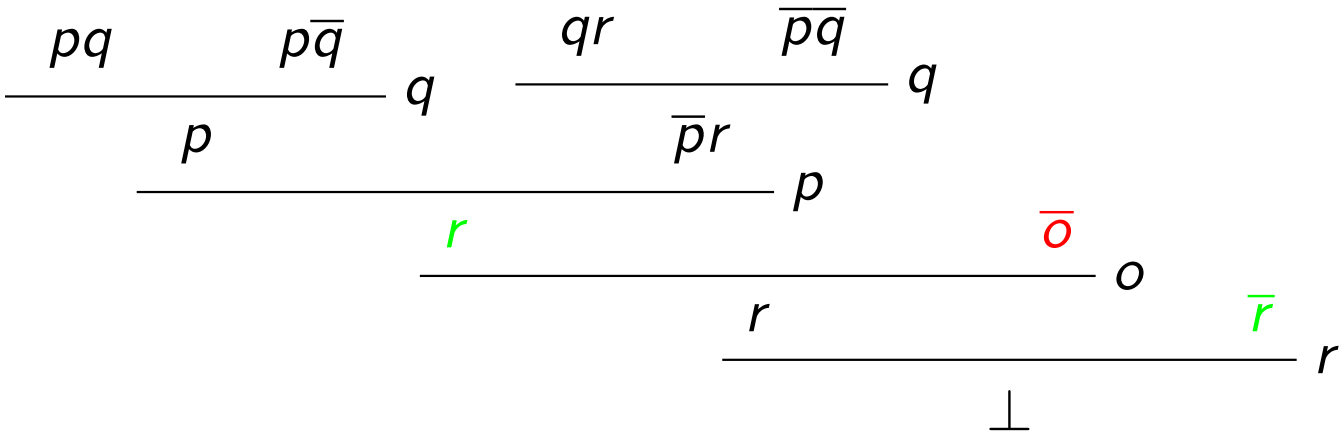
RecyclePivots

Example



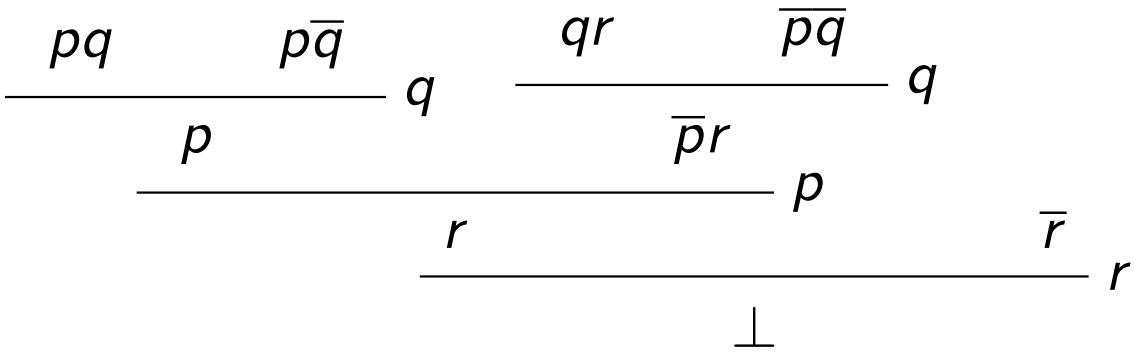
RecyclePivots

Example



RecyclePivots

Example



Outline

Outline

Transformation Framework

Features

- Local rewriting rules

Transformation Framework

Features

- Local rewriting rules
  - B reduction
  - A perturbation

Transformation Framework

Features

- Local rewriting rules
  - B reduction
  - A perturbation

- Rule context

$$\dfrac{\dfrac{pqC \qquad \overline{p}D}{qCD}\,p \qquad \overline{q}E}{CDE}\,q$$

Transformation Framework

Features

- Local rewriting rules
  - B reduction
  - A perturbation
- Rule context

$$\dfrac{\dfrac{pqC \qquad \overline{p}D}{qCD}\,p \qquad \overline{q}E}{CDE}\,q$$

- Exhaustiveness up to symmetry

Transformation Framework

Local rewriting rules

- B rules

$B1$	$\dfrac{\dfrac{pqC \qquad \overline{p}qD}{qCD}\, p \qquad p\overline{q}E}{pCDE}\, q \;\Rightarrow\; \dfrac{pqC \qquad p\overline{q}E}{pCE}\, q$

Transformation Framework

Local rewriting rules

- B rules

$B1$	$\dfrac{\dfrac{pqC \qquad \overline{p}qD}{qCD}\, p \qquad p\overline{q}E}{pCDE}\, q \;\Rightarrow\; \dfrac{pqC \qquad p\overline{q}E}{pCE}\, q$

  - Redundancy as reintroduction variable after elimination

Transformation Framework

Local rewriting rules

- B rules

$B1$	$\dfrac{\dfrac{pqC \qquad \overline{p}qD}{qCD}\, p \qquad p\overline{q}E}{pCDE}\, q \;\Rightarrow\; \dfrac{pqC \qquad p\overline{q}E}{pCE}\, q$

- Redundancy as reintroduction variable after elimination
- Subproof simplification

Transformation Framework

Local rewriting rules

- B rules

$B1$	$\dfrac{\dfrac{pqC \qquad \overline{p}qD}{qCD}\, p \qquad p\overline{q}E}{pCDE}\, q \;\Rightarrow\; \dfrac{pqC \qquad p\overline{q}E}{pCE}\, q$

- Redundancy as reintroduction variable after elimination
- Subproof simplification
- Subproof root strengthening

Transformation Framework

Local rewriting rules

- A rules

$A2$	$\dfrac{\dfrac{pqC \qquad \overline{p}D}{qCD}\,p \qquad \overline{q}E}{CDE}\,q \;\Rightarrow\; \dfrac{\dfrac{pqC \qquad \overline{q}E}{pCE}\,q \qquad \overline{p}D}{CDE}\,p$

Transformation Framework

Local rewriting rules

- A rules

$A2$	$\dfrac{\dfrac{pqC \qquad \overline{p}D}{qCD}\,p \qquad \overline{q}E}{CDE}\,q \;\Rightarrow\; \dfrac{\dfrac{pqC \qquad \overline{q}E}{pCE}\,q \qquad \overline{p}D}{CDE}\,p$

  - Pivots swapping

Transformation Framework

Local rewriting rules

- A rules

$A2$	$\dfrac{\dfrac{pqC \qquad \overline{p}D}{qCD}\,p \qquad \overline{q}E}{CDE}\,q \;\Rightarrow\; \dfrac{\dfrac{pqC \qquad \overline{q}E}{pCE}\,q \qquad \overline{p}D}{CDE}\,p$

  - Pivots swapping
  - Topology perturbation

Transformation Framework

Local rewriting rules

- A rules

$A2$	$\dfrac{\dfrac{pqC \qquad \overline{p}D}{qCD}\,p \qquad \overline{q}E}{CDE}\,q \;\Rightarrow\; \dfrac{\dfrac{pqC \qquad \overline{q}E}{pCE}\,q \qquad \overline{p}D}{CDE}\,p$

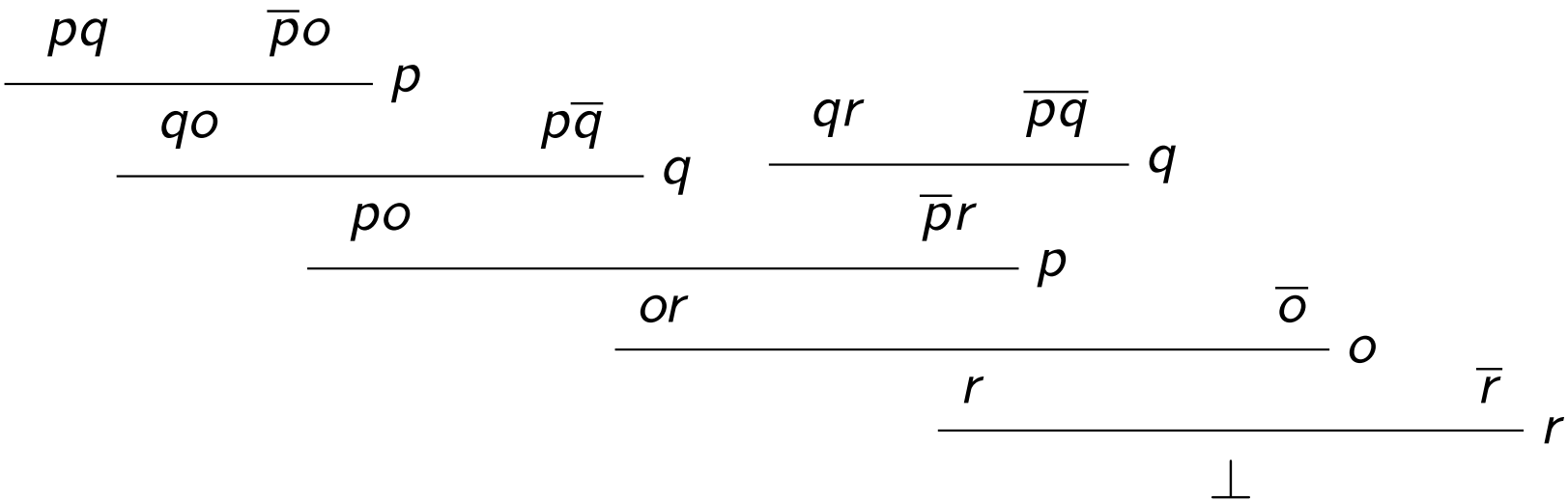
- Pivots swapping
- Topology perturbation
- Redundancies exposure

Local rewriting rules

$A1$	$\dfrac{\dfrac{pqC \quad \overline{p}qD}{qCD}\,p \quad \overline{q}E}{CDE}\,q \Rightarrow \dfrac{pqC \quad \overline{q}E}{pCE}\,q \quad \dfrac{\overline{q}E \quad \overline{p}qD}{\overline{p}DE}\,q$
$A2$	$\dfrac{\dfrac{pqC \quad \overline{p}D}{qCD}\,p \quad \overline{q}E}{CDE}\,q \Rightarrow \dfrac{pqC \quad \overline{q}E}{pCE}\,q \quad \overline{p}D$
$B1$	$\dfrac{\dfrac{pqC \quad \overline{p}qD}{qCD}\,p \quad p\overline{q}E}{pCDE}\,q \Rightarrow \dfrac{pqC \quad p\overline{q}E}{pCE}\,q$
$B2$	$\dfrac{\dfrac{pqC \quad \overline{p}D}{qDC}\,p \quad p\overline{q}E}{pCDE}\,q \Rightarrow \dfrac{pqC \quad p\overline{q}E}{pCE}\,q \quad \overline{p}D$
$B2'$	$\dfrac{\dfrac{pqC \quad \overline{p}D}{qDC}\,p \quad p\overline{q}E}{pCDE}\,q \Rightarrow \dfrac{pqC \quad p\overline{q}E}{pCE}\,q$
$B3$	$\dfrac{\dfrac{pqC \quad \overline{p}D}{qCD}\,p \quad \overline{p}\overline{q}E}{\overline{p}CDE}\,q \Rightarrow \overline{p}D$

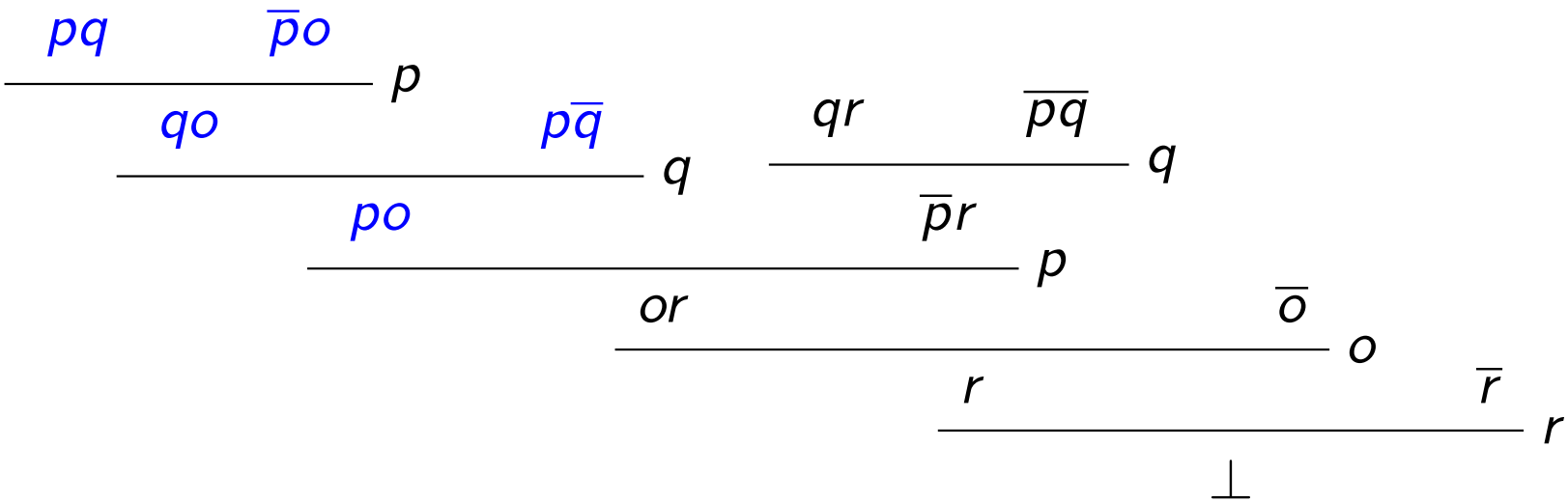
Rule-based Approach

Example



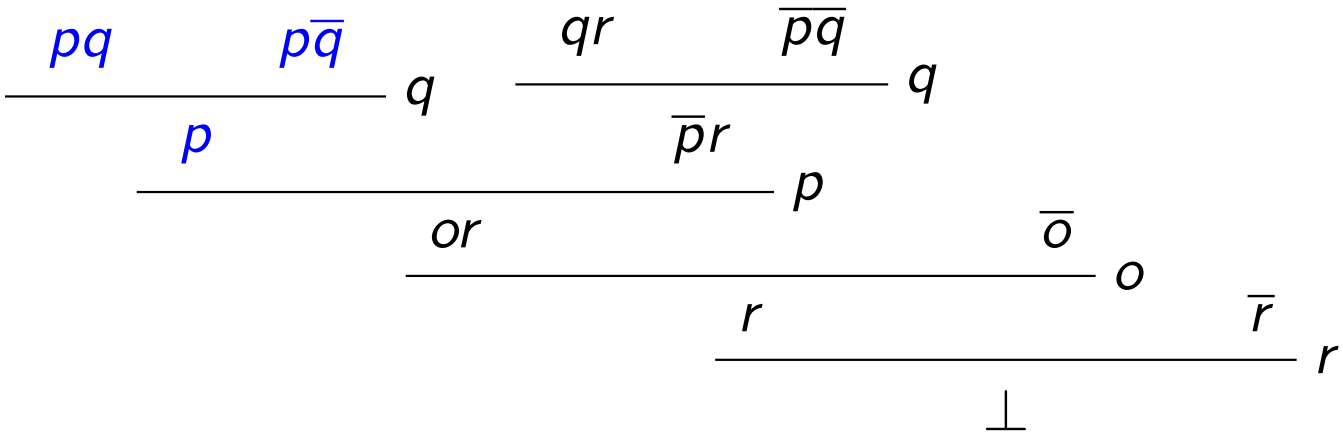
Rule-based Approach

Example



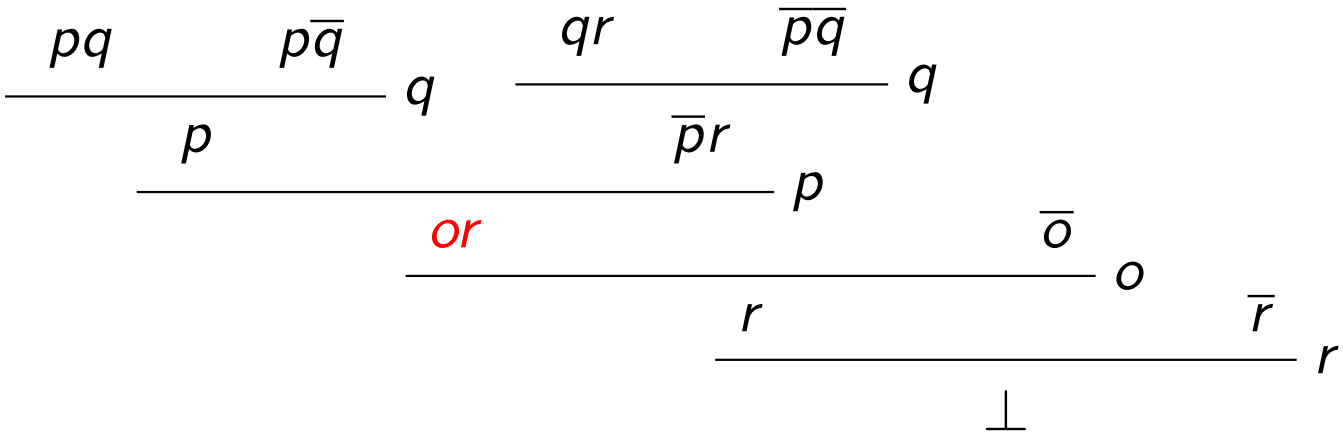
Rule-based Approach

Example



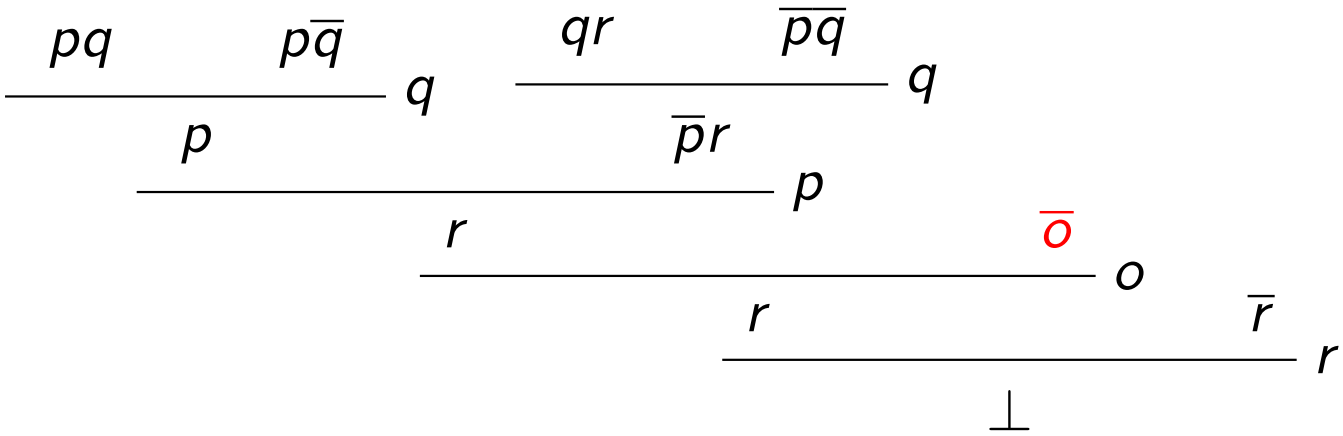
Rule-based Approach

Example



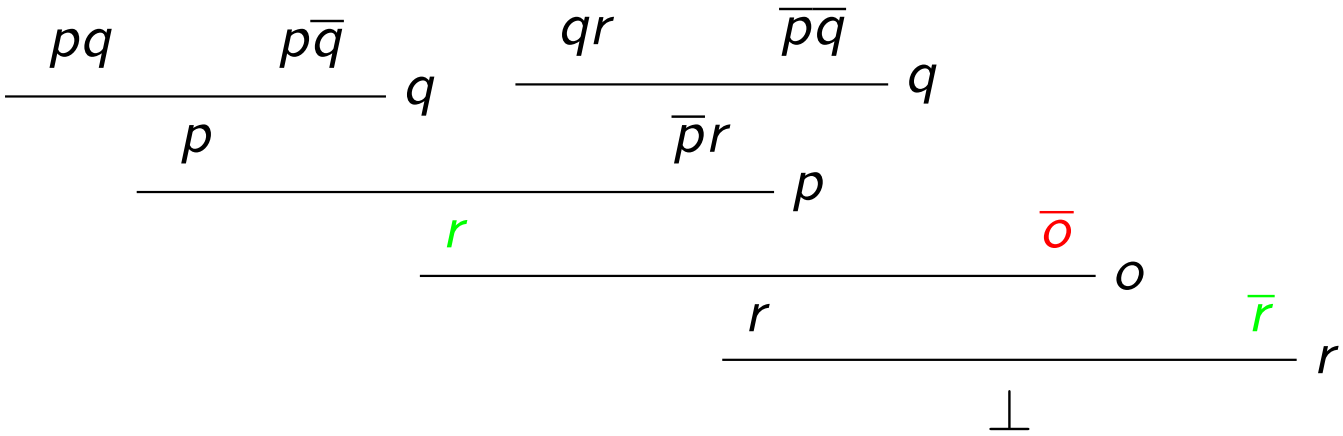
Rule-based Approach

Example



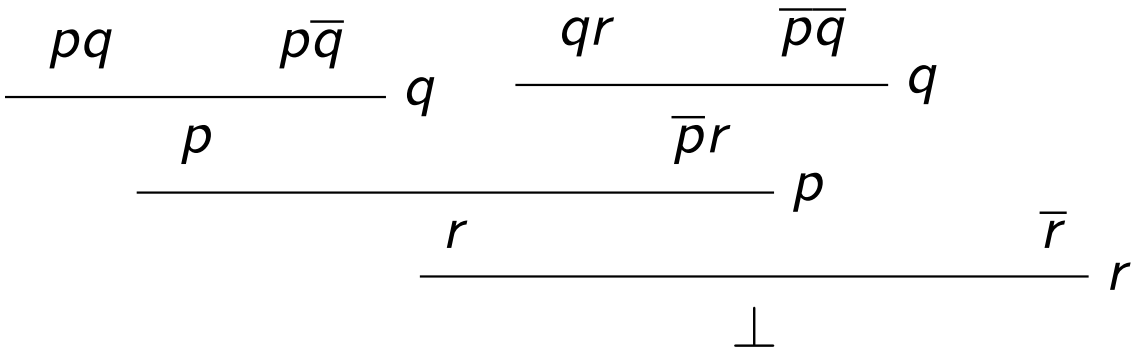
Rule-based Approach

Example



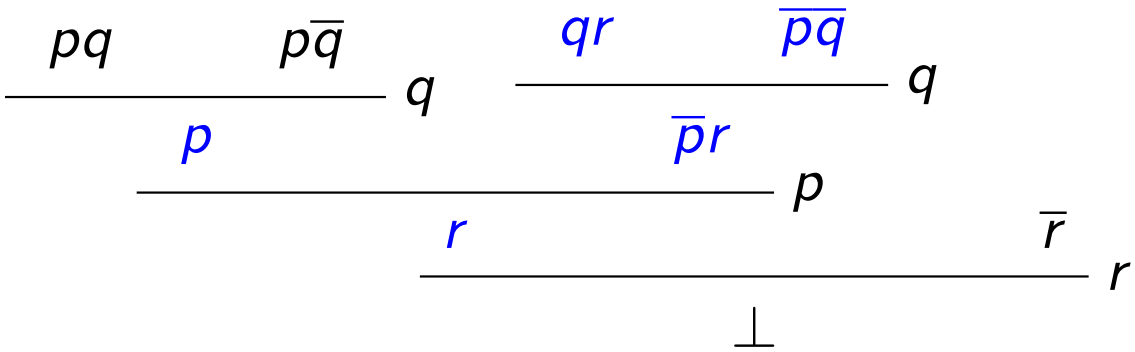
Rule-based Approach

Example



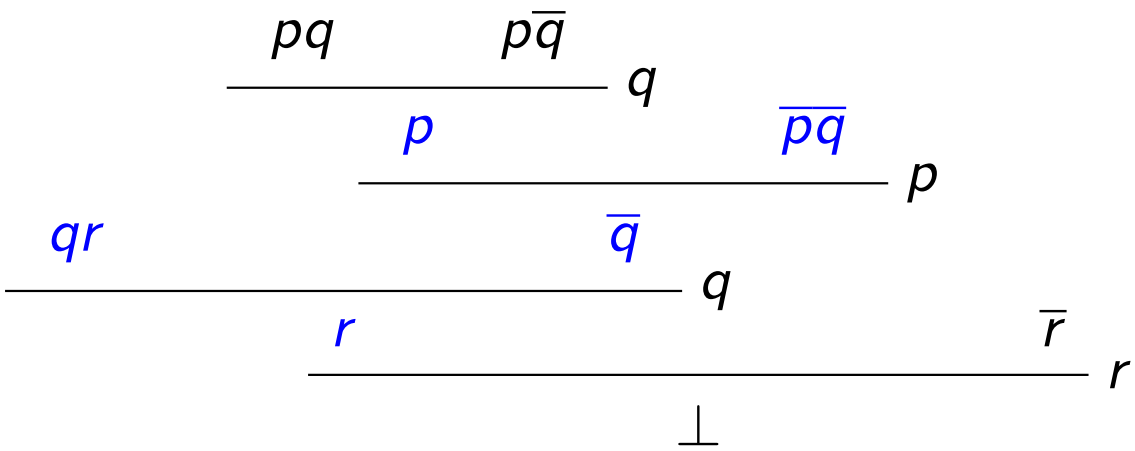
Rule-based Approach

Example



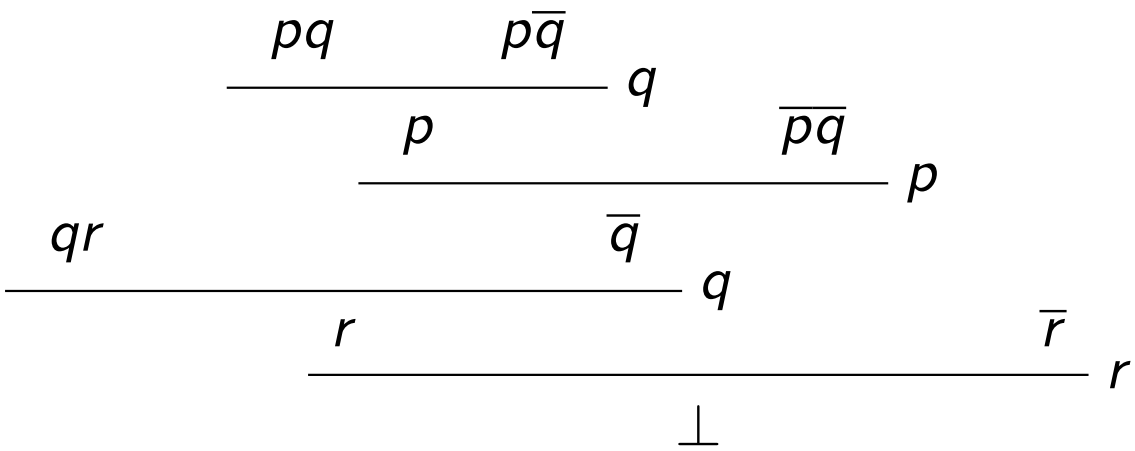
Rule-based Approach

Example



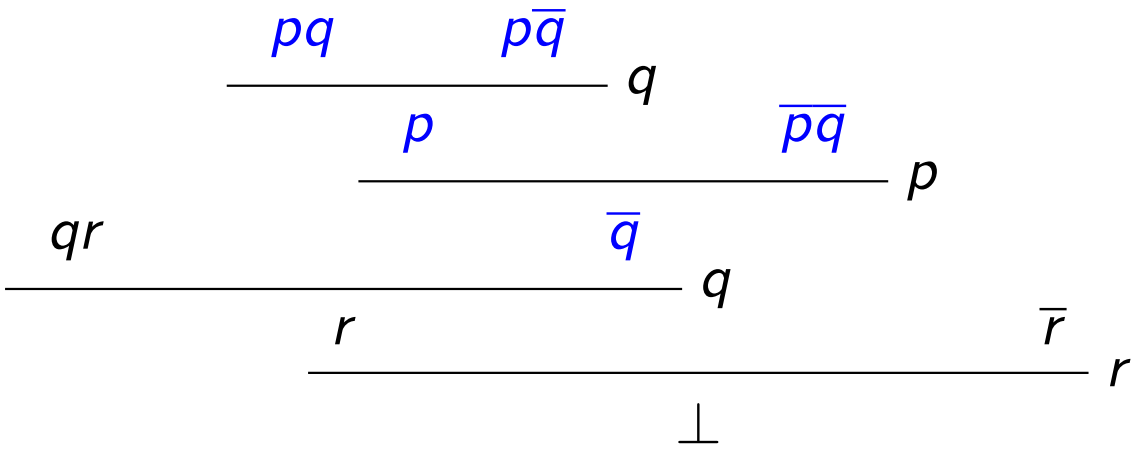
Rule-based Approach

Example



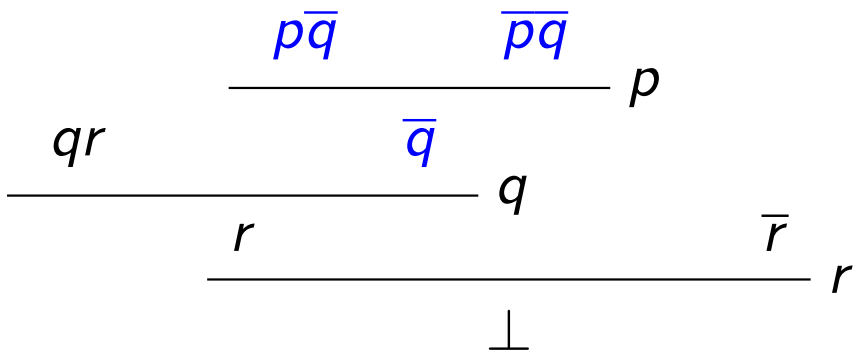
Rule-based Approach

Example



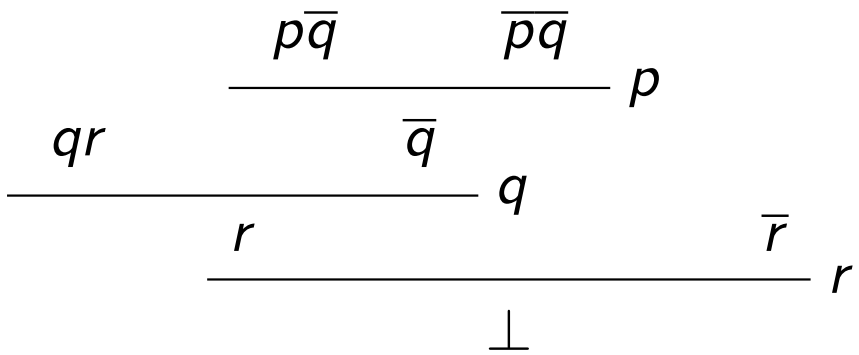
Rule-based Approach

Example



Rule-based Approach

Example



Comparison

- RecyclePivots

Comparison

- RecyclePivots
  - **Pros**
    Global information
    Fast and effective
  - **Cons**
    Cannot expose redundancies

Comparison

- RecyclePivots
  - **Pros**
    Global information
    Fast and effective
  - **Cons**
    Cannot expose redundancies
- Rule-based approach

Comparison

- RecyclePivots
  - **Pros**
    Global information
    Fast and effective
  - **Cons**
    Cannot expose redundancies
- Rule-based approach
  - **Pros**
    Flexibility in rules application
    Flexibility in amount of transformation
    Can expose redundancies
  - **Cons**
    Local information

Outline

Implementation

A Simple Algorithm

- Based on a sequence of proof traversals (e.g. topological order)

Implementation

A Simple Algorithm

- Based on a sequence of proof traversals (e.g. topological order)
- Parameterized in number of traversals and time limit

Implementation

A Simple Algorithm

- Based on a sequence of proof traversals (e.g. topological order)
- Parameterized in number of traversals and time limit
- Examination non-leaf clauses

Implementation

A Simple Algorithm

- Based on a sequence of proof traversals (e.g. topological order)
- Parameterized in number of traversals and time limit
- Examination non-leaf clauses
  - Pivot in both antecedents $\longrightarrow$ update, match context, apply rule

$$\frac{qC'D' \qquad \overline{q}E'}{CDE}\,q \quad \Rightarrow \quad \frac{qC'D' \qquad \overline{q}E'}{C'D'E'}\,q \quad \Rightarrow \quad \frac{\dfrac{pqC' \qquad \overline{p}D'}{qC'D'}\,p \qquad \overline{q}E'}{C'D'E'}\,q$$

Implementation

A Simple Algorithm

- Based on a sequence of proof traversals (e.g. topological order)
- Parameterized in number of traversals and time limit
- Examination non-leaf clauses
  - Pivot in both antecedents $\longrightarrow$ update, match context, apply rule

$$\frac{qC'D' \qquad \overline{q}E'}{CDE}\,q \quad \Rightarrow \quad \frac{qC'D' \qquad \overline{q}E'}{C'D'E'}\,q \quad \Rightarrow \quad \frac{\dfrac{pqC' \qquad \overline{p}D'}{qC'D'}\,p \qquad \overline{q}E'}{C'D'E'}\,q$$

  - Pivot not in both antecedents $\longrightarrow$ remove resolution step

$$\frac{C'D' \qquad \overline{q}E'}{CDE}\,q \quad \Rightarrow \quad C'D'$$

Implementation

A Simple Algorithm

- Based on a sequence of proof traversals (e.g. topological order)
- Parameterized in number of traversals and time limit
- Examination non-leaf clauses
  - Pivot in both antecedents $\longrightarrow$ update, match context, apply rule

$$\frac{qC'D' \qquad \overline{q}E'}{CDE}\,q \quad \Rightarrow \quad \frac{qC'D' \qquad \overline{q}E'}{C'D'E'}\,q \quad \Rightarrow \quad \frac{\dfrac{pqC' \qquad \overline{p}D'}{qC'D'}\,p \qquad \overline{q}E'}{C'D'E'}\,q$$

  - Pivot not in both antecedents $\longrightarrow$ remove resolution step

$$\frac{C'D' \qquad \overline{q}E'}{CDE}\,q \quad \Rightarrow \quad C'D'$$

- Easy combination with RecyclePivots

Evaluation

Framework and Benchmarks

- Implemented in C++ and integrated with OpenSMT
- Available at **www.inf.usi.ch/phd/rollini/hvc.html**

Evaluation

Framework and Benchmarks

- Implemented in C++ and integrated with OpenSMT
- Available at **www.inf.usi.ch/phd/rollini/hvc.html**

- Benchmarks

Evaluation

Framework and Benchmarks

- Implemented in C++ and integrated with OpenSMT
- Available at **www.inf.usi.ch/phd/rollini/hvc.html**

- Benchmarks
  - SMT: SMT-LIB library
  - SAT: SAT competition
  - Academic and industrial problems

Combined Approach Evaluation

Experimental results over SMT: QF_UF, QF_IDL, QF_LRA, QF_RDL

	#	Avg_{nodes}	Avg_{edges}	Avg_{core}	$T(s)$	Max_{nodes}	Max_{edges}	Max_{core}
RP	1370	6.7%	7.5%	1.3%	1.7	65.1%	68.9%	39.1%
Ratio								
0.01	1366	8.9%	10.7%	1.4%	3.4	66.3%	70.2%	45.7%
0.025	1366	9.8%	11.9%	1.5%	3.6	77.2%	79.9%	45.7%
0.05	1366	10.7%	13.0%	1.6%	4.1	78.5%	81.2%	45.7%
0.075	1366	11.4%	13.8%	1.7%	4.5	78.5%	81.2%	45.7%
0.1	1364	11.8%	14.4%	1.7%	5.0	78.8%	83.6%	45.7%
0.25	1359	13.6%	16.6%	1.9%	7.6	79.6%	84.4%	45.7%
0.5	1348	15.0%	18.4%	2.0%	11.5	79.1%	85.2%	45.7%
0.75	1341	16.0%	19.5%	2.1%	15.1	79.9%	86.1%	45.7%
1	1337	16.7%	20.4%	2.2%	18.8	79.9%	86.1%	45.7%

- *Ratio* — time threshold as fraction of solving time
- # — number of benchmarks solved
- Avg_{nodes}, Avg_{edges}, Avg_{core} — average reduction in proof size
- $T(s)$ — average transformation time in seconds
- Max_{nodes}, Max_{edges}, Max_{core} — max reduction in proof size

Combined Approach Evaluation

Experimental results over SMT: QF_UF, QF_IDL, QF_LRA, QF_RDL

	#	Avg_{nodes}	Avg_{edges}	Avg_{core}	$T(s)$	Max_{nodes}	Max_{edges}	Max_{core}
RP	1370	6.7%	7.5%	1.3%	1.7	65.1%	68.9%	39.1%
Ratio								
0.01	1366	8.9%	10.7%	1.4%	3.4	66.3%	70.2%	45.7%
0.025	1366	9.8%	11.9%	1.5%	3.6	77.2%	79.9%	45.7%
0.05	1366	10.7%	13.0%	1.6%	4.1	78.5%	81.2%	45.7%
0.075	1366	11.4%	13.8%	1.7%	4.5	78.5%	81.2%	45.7%
0.1	1364	11.8%	14.4%	1.7%	5.0	78.8%	83.6%	45.7%
0.25	1359	13.6%	16.6%	1.9%	7.6	79.6%	84.4%	45.7%
0.5	1348	15.0%	18.4%	2.0%	11.5	79.1%	85.2%	45.7%
0.75	1341	16.0%	19.5%	2.1%	15.1	79.9%	86.1%	45.7%
1	1337	16.7%	20.4%	2.2%	18.8	79.9%	86.1%	45.7%

- *Ratio* — time threshold as fraction of solving time
- # — number of benchmarks solved
- Avg_{nodes}, Avg_{edges}, Avg_{core} — average reduction in proof size
- $T(s)$ — average transformation time in seconds
- Max_{nodes}, Max_{edges}, Max_{core} — max reduction in proof size

Combined Approach Evaluation

Experimental results over SMT: QF_UF, QF_IDL, QF_LRA, QF_RDL

	#	Avg_{nodes}	Avg_{edges}	Avg_{core}	$T(s)$	Max_{nodes}	Max_{edges}	Max_{core}
RP	1370	6.7%	7.5%	1.3%	1.7	65.1%	68.9%	39.1%
Ratio								
0.01	1366	8.9%	10.7%	1.4%	3.4	66.3%	70.2%	45.7%
0.025	1366	9.8%	11.9%	1.5%	3.6	77.2%	79.9%	45.7%
0.05	1366	10.7%	13.0%	1.6%	4.1	78.5%	81.2%	45.7%
0.075	1366	11.4%	13.8%	1.7%	4.5	78.5%	81.2%	45.7%
0.1	1364	11.8%	14.4%	1.7%	5.0	78.8%	83.6%	45.7%
0.25	1359	13.6%	16.6%	1.9%	7.6	79.6%	84.4%	45.7%
0.5	1348	15.0%	18.4%	2.0%	11.5	79.1%	85.2%	45.7%
0.75	1341	16.0%	19.5%	2.1%	15.1	79.9%	86.1%	45.7%
1	1337	16.7%	20.4%	2.2%	18.8	79.9%	86.1%	45.7%

- *Ratio* — time threshold as fraction of solving time
- # — number of benchmarks solved
- Avg_{nodes}, Avg_{edges}, Avg_{core} — average reduction in proof size
- $T(s)$ — average transformation time in seconds
- Max_{nodes}, Max_{edges}, Max_{core} — max reduction in proof size

Combined Approach Evaluation

Experimental results over SAT

	#	Avg_{nodes}	Avg_{edges}	Avg_{core}	$T(s)$	Max_{nodes}	Max_{edges}	Max_{core}
RP	25	5.9%	6.5%	1.7%	10.8	33.1%	33.4%	30.3%
Ratio								
0.01	25	6.8%	7.9%	1.7%	32.3	34.0%	34.4%	30.5%
0.025	25	6.8%	7.9%	1.7%	32.3	34.0%	34.4%	30.5%
0.05	25	7.0%	8.2%	1.8%	40.0	34.0%	34.4%	30.5%
0.075	25	7.2%	8.4%	1.8%	49.3	34.7%	35.1%	30.5%
0.1	25	7.3%	8.4%	1.8%	60.2	34.7%	35.1%	30.5%
0.25	25	7.6%	8.8%	1.9%	125.3	39.8%	40.6%	31.7%
0.5	25	7.8%	9.1%	1.9%	243.5	41.0%	41.9%	32.1%
0.75	25	7.9%	9.3%	1.9%	360.0	41.6%	42.6%	32.1%
1	23	8.4%	9.9%	2.1%	175.6	33.1%	33.4%	30.6%

- *Ratio* — time threshold as fraction of solving time
- # — number of benchmarks solved
- Avg_{nodes}, Avg_{edges}, Avg_{core} — average reduction in proof size
- $T(s)$ — average transformation time in seconds
- Max_{nodes}, Max_{edges}, Max_{core} — max reduction in proof size

Combined Approach Evaluation

Experimental results over SAT

	#	Avg_{nodes}	Avg_{edges}	Avg_{core}	$T(s)$	Max_{nodes}	Max_{edges}	Max_{core}
RP	25	5.9%	6.5%	1.7%	10.8	33.1%	33.4%	30.3%
Ratio								
0.01	25	6.8%	7.9%	1.7%	32.3	34.0%	34.4%	30.5%
0.025	25	6.8%	7.9%	1.7%	32.3	34.0%	34.4%	30.5%
0.05	25	7.0%	8.2%	1.8%	40.0	34.0%	34.4%	30.5%
0.075	25	7.2%	8.4%	1.8%	49.3	34.7%	35.1%	30.5%
0.1	25	7.3%	8.4%	1.8%	60.2	34.7%	35.1%	30.5%
0.25	25	7.6%	8.8%	1.9%	125.3	39.8%	40.6%	31.7%
0.5	25	7.8%	9.1%	1.9%	243.5	41.0%	41.9%	32.1%
0.75	25	7.9%	9.3%	1.9%	360.0	41.6%	42.6%	32.1%
1	23	8.4%	9.9%	2.1%	175.6	33.1%	33.4%	30.6%

- *Ratio* — time threshold as fraction of solving time
- # — number of benchmarks solved
- Avg_{nodes}, Avg_{edges}, Avg_{core} — average reduction in proof size
- $T(s)$ — average transformation time in seconds
- Max_{nodes}, Max_{edges}, Max_{core} — max reduction in proof size

Combined Approach Evaluation

Experimental results over SAT

	#	Avg_{nodes}	Avg_{edges}	Avg_{core}	$T(s)$	Max_{nodes}	Max_{edges}	Max_{core}
RP	25	5.9%	6.5%	1.7%	10.8	33.1%	33.4%	30.3%
Ratio								
0.01	25	6.8%	7.9%	1.7%	32.3	34.0%	34.4%	30.5%
0.025	25	6.8%	7.9%	1.7%	32.3	34.0%	34.4%	30.5%
0.05	25	7.0%	8.2%	1.8%	40.0	34.0%	34.4%	30.5%
0.075	25	7.2%	8.4%	1.8%	49.3	34.7%	35.1%	30.5%
0.1	25	7.3%	8.4%	1.8%	60.2	34.7%	35.1%	30.5%
0.25	25	7.6%	8.8%	1.9%	125.3	39.8%	40.6%	31.7%
0.5	25	7.8%	9.1%	1.9%	243.5	41.0%	41.9%	32.1%
0.75	25	7.9%	9.3%	1.9%	360.0	41.6%	42.6%	32.1%
1	23	8.4%	9.9%	2.1%	175.6	33.1%	33.4%	30.6%

- *Ratio* — time threshold as fraction of solving time
- # — number of benchmarks solved
- Avg_{nodes}, Avg_{edges}, Avg_{core} — average reduction in proof size
- $T(s)$ — average transformation time in seconds
- Max_{nodes}, Max_{edges}, Max_{core} — max reduction in proof size

Outline

Summary and Future Work

- Summary

Summary and Future Work

- Summary
  - Rule-based proof reduction framework

Summary and Future Work

- Summary
  - Rule-based proof reduction framework
  - Pivots redundancies

Summary and Future Work

- Summary
  - Rule-based proof reduction framework
  - Pivots redundancies
  - Comparison and evaluation

Summary and Future Work

- Summary
  - Rule-based proof reduction framework
  - Pivots redundancies
  - Comparison and evaluation
- Future Work

Summary and Future Work

- Summary
  - Rule-based proof reduction framework
  - Pivots redundancies
  - Comparison and evaluation
- Future Work
  - Exploitation of DPLL proof structure

Summary and Future Work

- Summary
  - Rule-based proof reduction framework
  - Pivots redundancies
  - Comparison and evaluation
- Future Work
  - Exploitation of DPLL proof structure
  - Evaluation on concrete applications (e.g. interpolation)

Summary and Future Work

- Summary
  - Rule-based proof reduction framework
  - Pivots redundancies
  - Comparison and evaluation

- Future Work
  - Exploitation of DPLL proof structure
  - Evaluation on concrete applications (e.g. interpolation)
  - Rule-based control of interpolants' strength

Publications

- Proof reduction

S.F. Rollini, R. Bruttomesso and N. Sharygina
An Efficient and Flexible Approach to Resolution Proof Reduction.
HVC 2010.

- Proof manipulation for interpolation

R. Bruttomesso, S.F. Rollini, N. Sharygina and A. Tsitovich
Flexible Interpolation with Local Proof Transformations.
ICCAD 2010

Thank you for your attention!