Dependency Schemes & Q-Resolution

Friedrich Slivovsky & Stefan Szeider

Vienna University of Technology

PCNF	$Q_1 X_1 \dots Q_n X_n$	$\phi(X_1, \dots, X_n)$
	$Q_i \in \{\forall, \exists\}$	in CNF
	(quantifier) prefix	matrix

PCNF	$Q_1 X_1 \dots Q_n X_n$	$\phi(X_1, \dots, X_n)$
	$Q_i \in \{\forall, \exists\}$	in CNF
	(quantifier) prefix	matrix

the prefix induces a partial order R_F on the set of variables

PCNF	$Q_1 X_1 \dots Q_n X_n$	$\phi(X_1, \dots, X_n)$
	$Q_i \in \{\forall, \exists\}$	in CNF
	(quantifier) prefix	matrix

the prefix induces a partial order R_F on the set of variables

$(x, y) \in R_F$ y is to the right of x in F‘s prefix

order R_F constrains decision procedures

QDPLL

order R_F constrains decision procedures

QDPLL

1) Unit propagation

$(y \vee x_1 \vee \neg x_2 \vee \cdots \vee x_k)$

$(x_i, y) \notin R_F$ for each i

order R_F constrains decision procedures

QDPLL

1) Unit propagation

2) Decision variables

$(y \vee x_1 \vee \neg x_2 \vee \cdots \vee x_k)$

$(x_i, y) \notin R_F$ for each i

branch on v: every w with $(w, v) \in R_F$ must be assigned

order R_F constrains decision procedures

QDPLL

1) Unit propagation

2) Decision variables

3) Clause learning

$(y \vee x_1 \vee \neg x_2 \vee \cdots \vee x_k)$

$(x_i, y) \notin R_F$ for each i

branch on v: every w with $(w, v) \in R_F$ must be assigned

$$\frac{C \vee (\neg)x}{C} \quad \forall y \in var_{\exists}(C): (x, y) \notin R_F$$

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

$(x, y) \in D_F$ "y may depend on x"

Biere & Lonsing ‘10

QDPLL(D)

1) Unit propagation

2) Decision variables

3) Clause learning

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

Biere & Lonsing ‘10

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

QDPLL(D)

1) Unit propagation

2) Decision variables

3) Clause learning

$(x_1 \vee y \vee \neg x_2 \vee \cdots \vee x_k)$

$(x_i, y) \notin D_F$ for each i

Biere & Lonsing ‘10

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

QDPLL(D)

1) Unit propagation

2) Decision variables

3) Clause learning

$(x_1 \vee y \vee \neg x_2 \vee \cdots \vee x_k)$
$(x_i, y) \notin D_F$ for each i

branch on v: every w with
$(w, v) \in D_F$ must be assigned

Biere & Lonsing ‘10

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

QDPLL(D)

1) Unit propagation

$(x_1 \vee y \vee \neg x_2 \vee \cdots \vee x_k)$
$(x_i, y) \notin D_F$ for each i

2) Decision variables

branch on v: every w with
$(w, v) \in D_F$ must be assigned

3) Clause learning

$$\frac{C \vee (\neg)x}{C} \quad \forall y \in var(C): (x, y) \notin D_F$$

Biere & Lonsing ‘10

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

QDPLL(D)

1) Unit propagation

$(x_1 \vee y \vee \neg x_2 \vee \cdots \vee x_k)$
$(x_i, y) \notin D_F$ for each i

2) Decision variables

branch on v: every w with
$(w, v) \in D_F$ must be assigned

3) Clause learning

$$\frac{C \vee (\neg)x}{C} \quad \forall y \in var(C): (x, y) \notin D_F$$

implemented in DepQBF for the Standard Dependency Scheme (D^{std})

For which dependency schemes D is QDPLL(D) sound?

For which D is QDPLL(D) sound?

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

cumulative dependency schemes (Samer & Szeider ’07)

For which D is QDPLL(D) sound?

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

cumulative dependency schemes (Samer & Szeider '07) no

For which D is QDPLL(D) sound?

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

cumulative dependency schemes (Samer & Szeider '07) no

resolution path dependency scheme (Van Gelder '11, Slivovsky & Szeider '12) is cumulative, but QDPLL(D) is unsound

For which D is QDPLL(D) sound?

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

cumulative dependency schemes (Samer & Szeider '07) no

resolution path dependency scheme (Van Gelder '11, Slivovsky & Szeider '12) is cumulative, but QDPLL(D) is unsound

dependency schemes for QCSP (Samer '08)?

For which D is QDPLL(D) sound?

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

cumulative dependency schemes (Samer & Szeider '07) no

resolution path dependency scheme (Van Gelder '11, Slivovsky & Szeider '12) is cumulative, but QDPLL(D) is unsound

dependency schemes for QCSP (Samer '08)? no

For which D is QDPLL(D) sound?

dependency scheme D
maps F to a set of dependencies $D_F \subseteq R_F$

cumulative dependency schemes (Samer & Szeider '07) no

resolution path dependency scheme (Van Gelder '11, Slivovsky & Szeider '12) is cumulative, but QDPLL(D) is unsound

dependency schemes for QCSP (Samer '08)? no

in this talk

1. stating sufficient conditions
2. proving soundness for D^{std}

QDPLL traces correspond to Q-resolution proofs

Giunchiglia, Narizzano, Tacchella ‘06

Q-resolution

Kleine Büning, Karpinski, Flögel ‘95

$$\frac{C \vee y \qquad \neg y \vee C'}{C \vee C'}$$

$C \vee C'$ non-tautological

$$\frac{C \vee (\neg) x}{C}$$

$\forall y \in var(C): (x, y) \notin R_F$

F is false iff the empty clause can be derived from F

QDPLL traces correspond to Q-resolution proofs

Giunchiglia, Narizzano, Tacchella ‘06

Q-resolution

Kleine Büning, Karpinski, Flögel ‘95

$$\frac{C \vee y \qquad \neg y \vee C'}{C \vee C'}$$

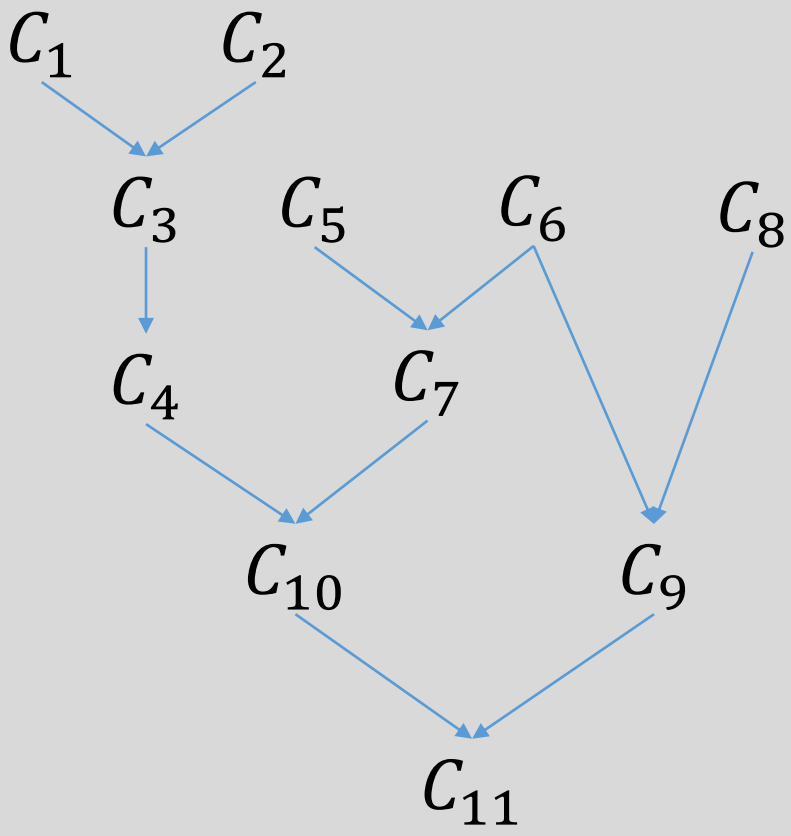
$C \vee C'$ non-tautological

$$\frac{C \vee (\neg) x}{C}$$

$\forall y \in var(C): (x, y) \notin R_F$

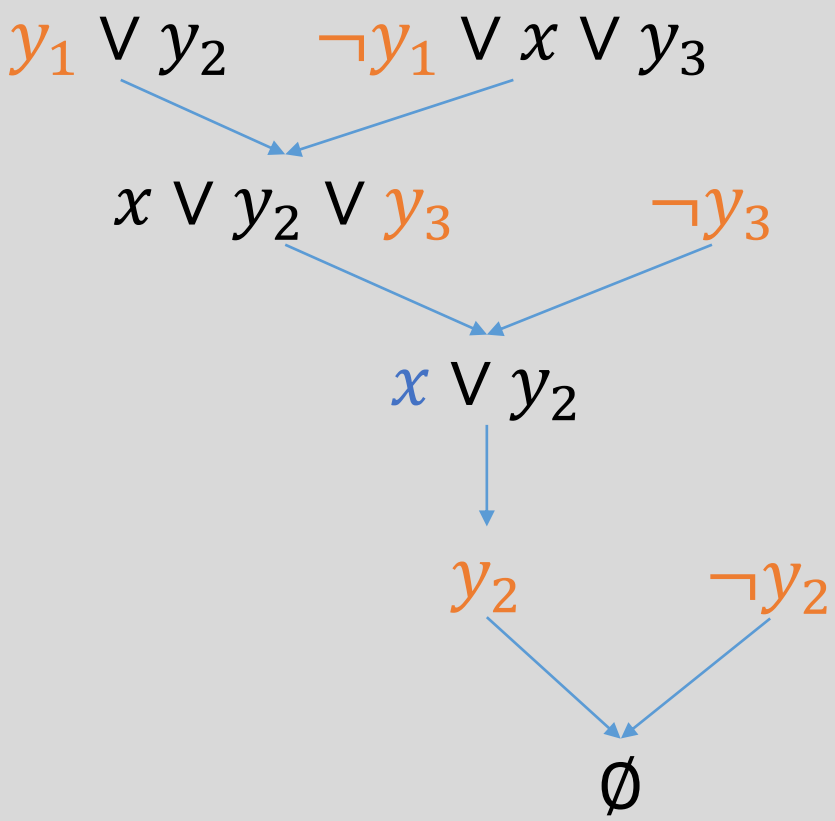
F is false iff the empty clause can be derived from F

derivations as DAGs



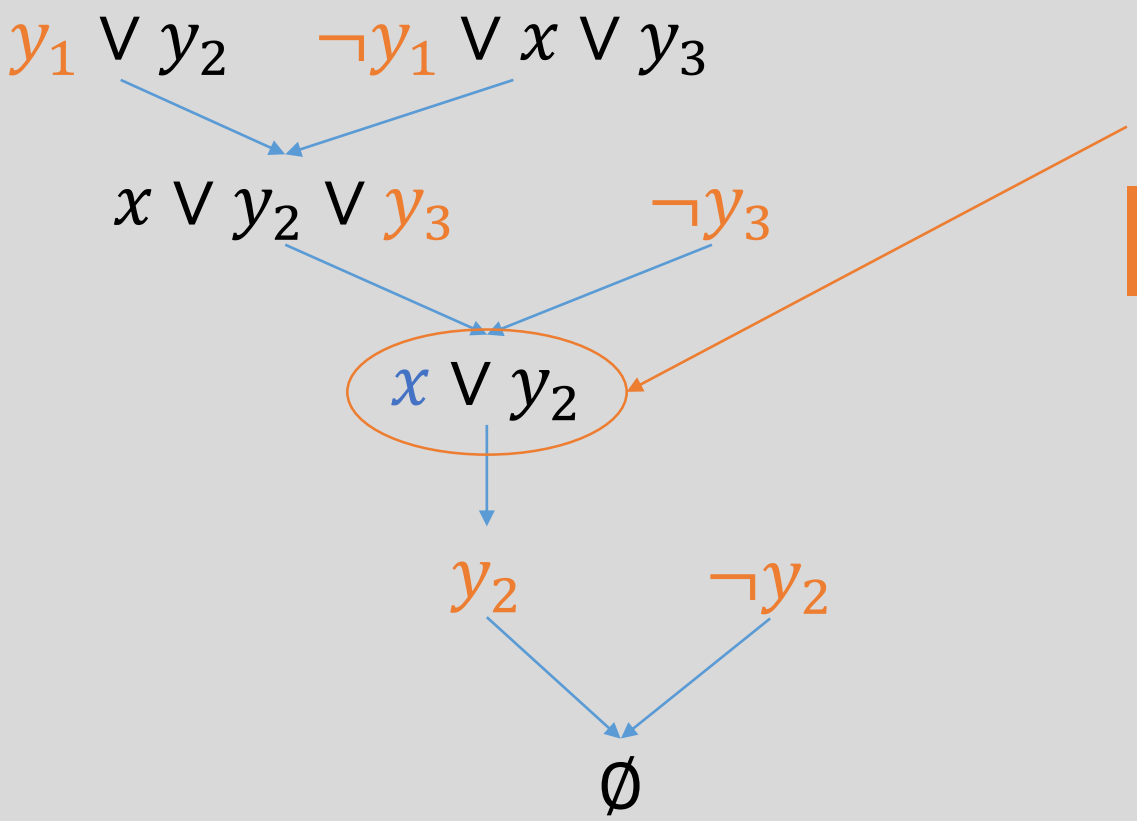
DepQBF (almost) generates Q-resolution certificates

$$F = \exists y_1 \forall x \exists y_2 \exists y_3 \ (y_1 \vee y_2) \wedge (\neg y_1 \vee x \vee y_3) \wedge (\neg y_3) \wedge (\neg y_2)$$



DepQBF (almost) generates Q-resolution certificates

$$F = \exists y_1 \forall x \exists y_2 \exists y_3 \ (y_1 \vee y_2) \wedge (\neg y_1 \vee x \vee y_3) \wedge (\neg y_3) \wedge (\neg y_2)$$

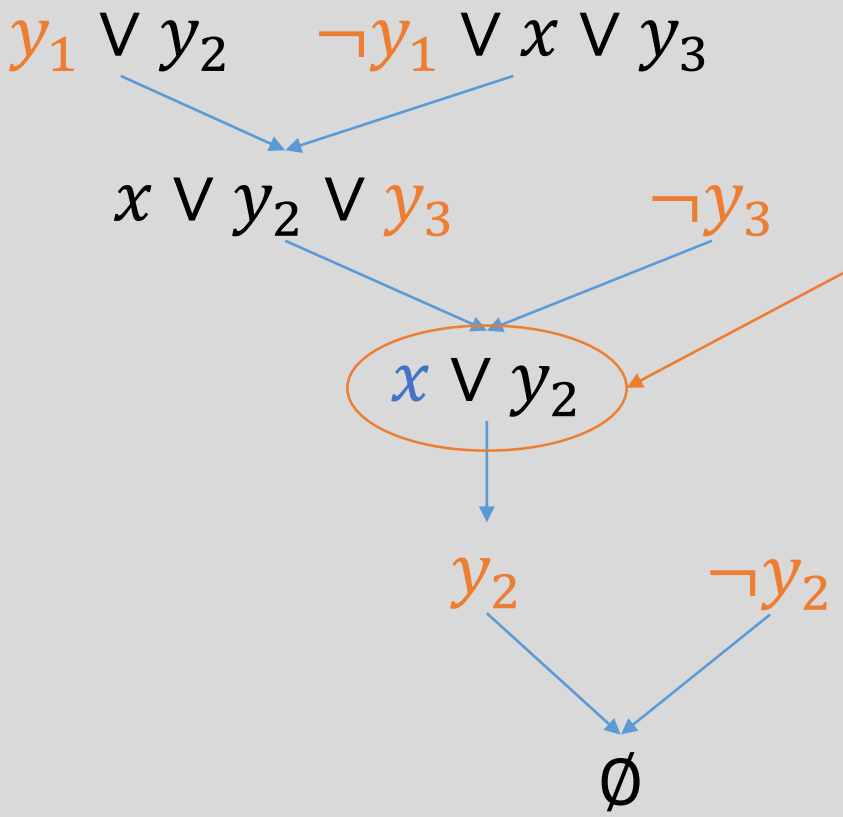


$(x, y_2) \in R_F$

not admissible in Q-resolution

DepQBF (almost) generates Q-resolution certificates

$$F = \exists y_1 \forall x \exists y_2 \exists y_3 \ (y_1 \vee y_2) \wedge (\neg y_1 \vee x \vee y_3) \wedge (\neg y_3) \wedge (\neg y_2)$$



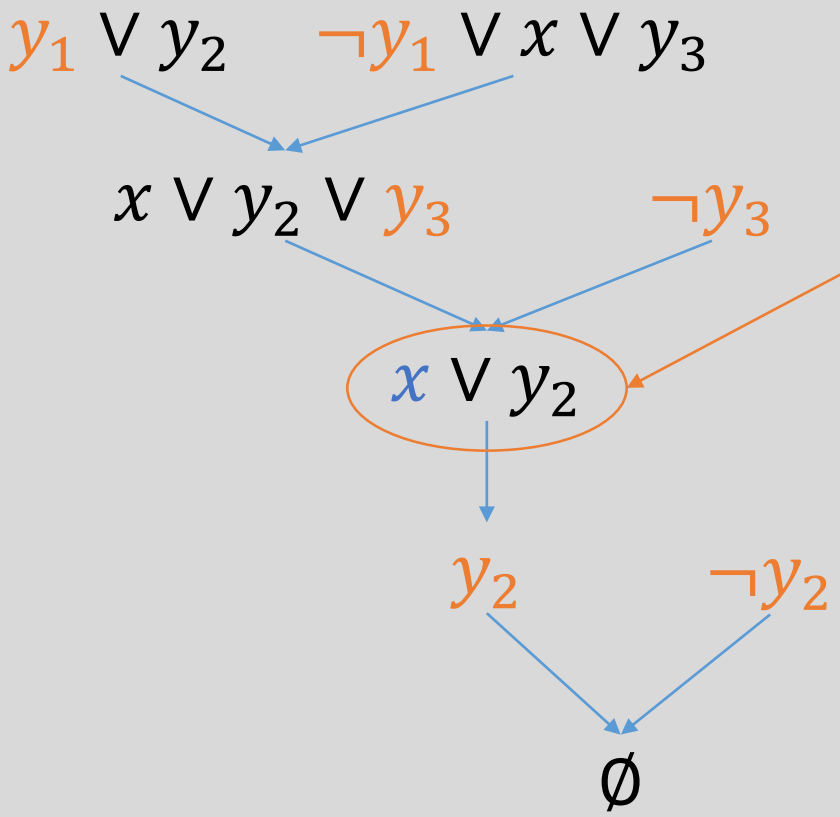
$(x, y_2) \in R_F$

not admissible in Q-resolution

DepQBF implements QDPLL(D^{std})

DepQBF (almost) generates Q-resolution certificates

$$F = \exists y_1 \forall x \exists y_2 \exists y_3 \ (y_1 \vee y_2) \wedge (\neg y_1 \vee x \vee y_3) \wedge (\neg y_3) \wedge (\neg y_2)$$



$(x, y_2) \in R_F$

not admissible in Q-resolution

DepQBF implements QDPLL(D^{std})

$$\frac{C \vee (\neg)x}{C} \quad \forall y \in var(C): (x, y) \notin D_F^{std}$$

$(x, y_2) \notin D_F^{std}$

Q(D)-resolution

D can be computed in polynomial time

$$\frac{C \vee (\neg)x}{C} \quad \forall y \in var(C): (x, y) \notin \boldsymbol{D_F}$$

Q(D)-resolution

D can be computed in polynomial time

$$\frac{C \vee (\neg)x}{C} \quad \forall y \in var(C): (x, y) \notin \boldsymbol{D_F}$$

working hypothesis

QDPLL(D) generates Q(D)-resolution proofs

Q(D)-resolution

D can be computed in polynomial time

$$\frac{C \vee (\neg)x}{C} \quad \forall y \in var(C): (x, y) \notin \boldsymbol{D_F}$$

working hypothesis

QDPLL(D) generates Q(D)-resolution proofs

If Q(D)-resolution is sound then QDPLL(D) is sound.

Q(D^{std})-resolution is sound

Theorem

Q-resolution simulates Q(D^{std})-resolution.

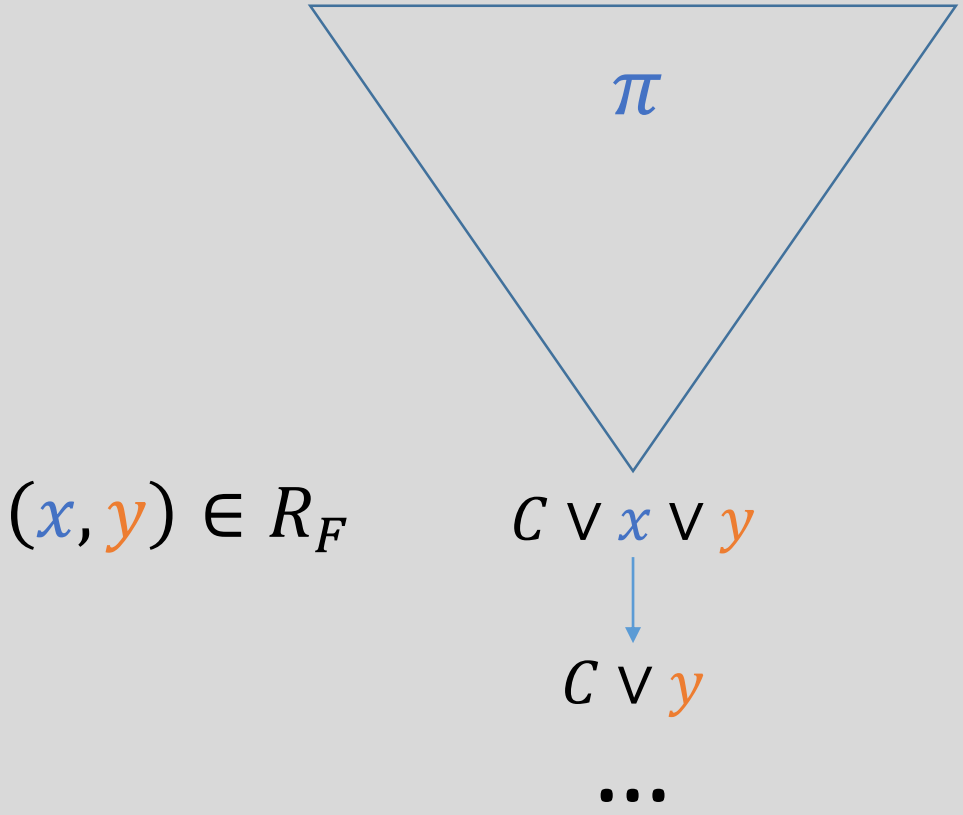
Q(D^{std})-resolution is sound

Theorem

Q-resolution simulates Q(D^{std})-resolution.

There is an $O(3^n)$ time algorithm that, given an arbitrary Q(D)-resolution refutation π of a formula F, computes a Q-resolution refutation of F or concludes that π is not a Q(D^{std})-refutation.

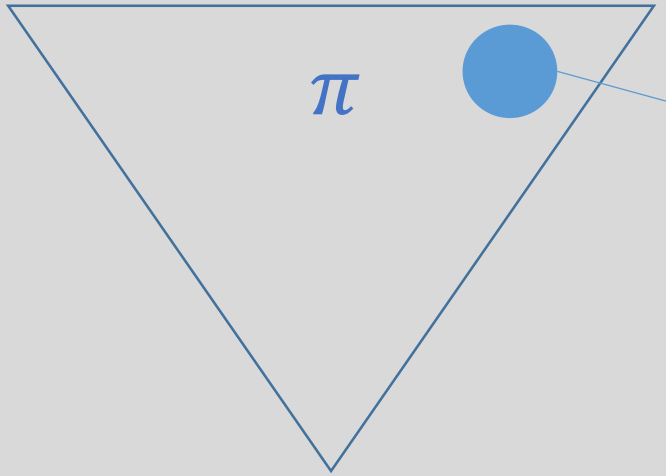
Q(D^{std})-resolution is sound



Lemma

π contains a resolution step on z with $(z, x) \in R_F$, or $(x, y) \in D_F^{std}$.

Q(D^{std})-resolution is sound



resolution on z with $(z, x) \in R_F$

idea lower this resolution step past the universal reduction step

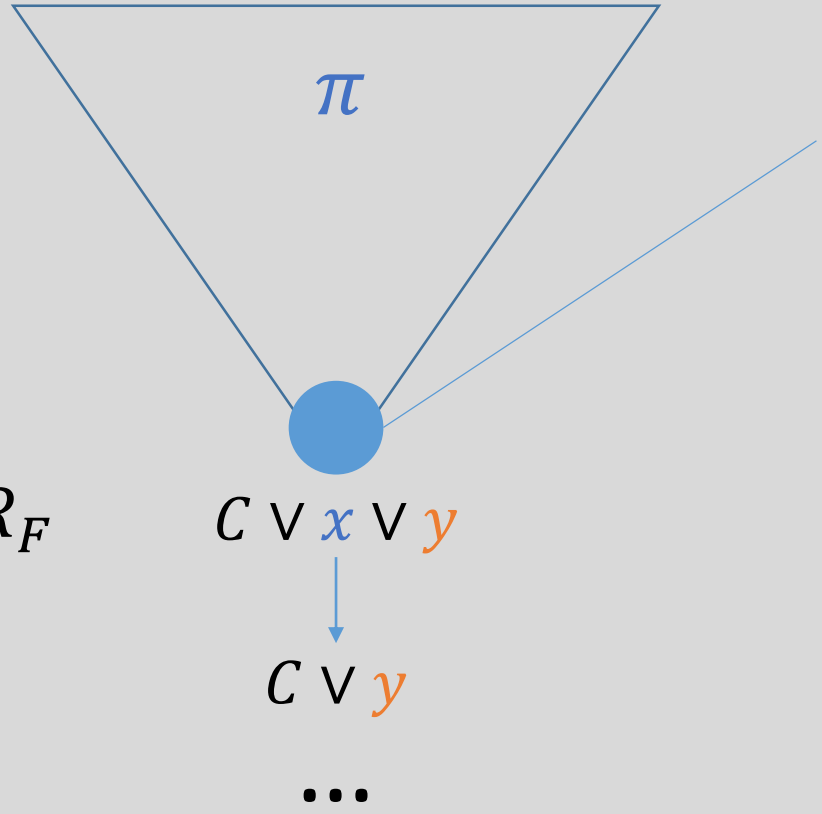
$(x, y) \in R_F$

$C \vee x \vee y$

$C \vee y$

...

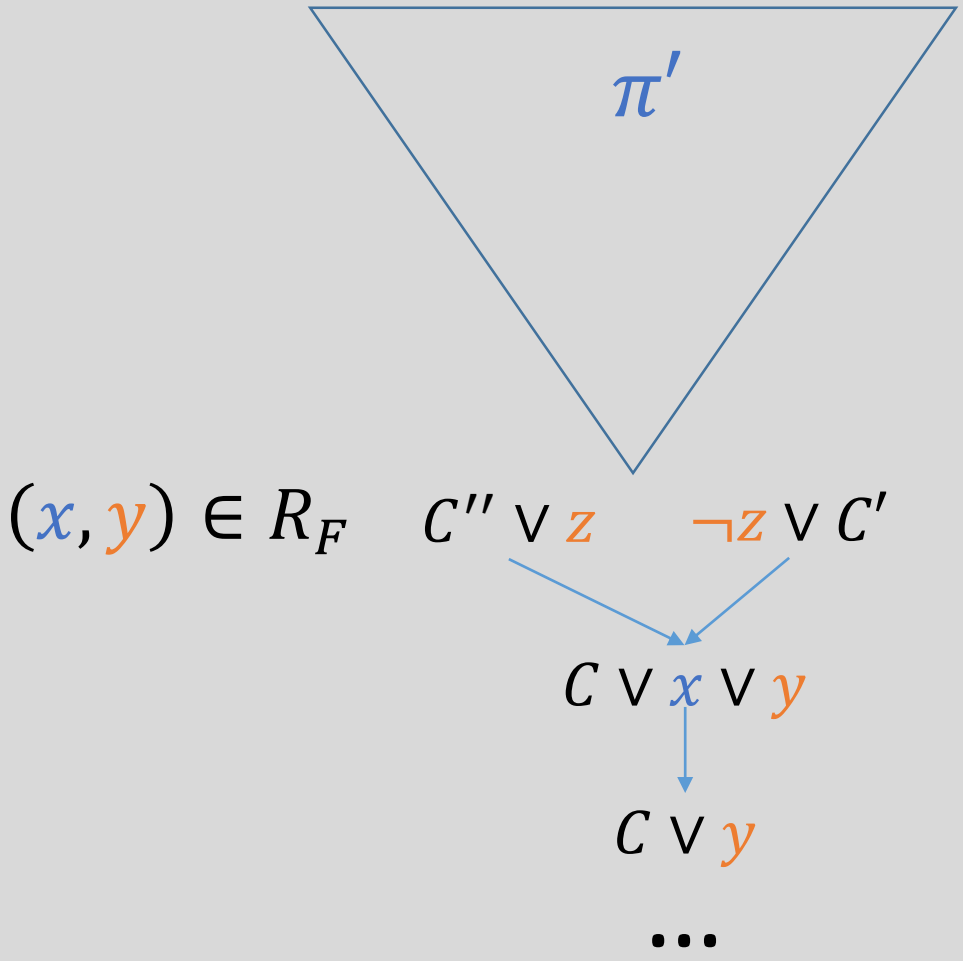
Q(D^{std})-resolution is sound



resolution on z with $(z, x) \in R_F$

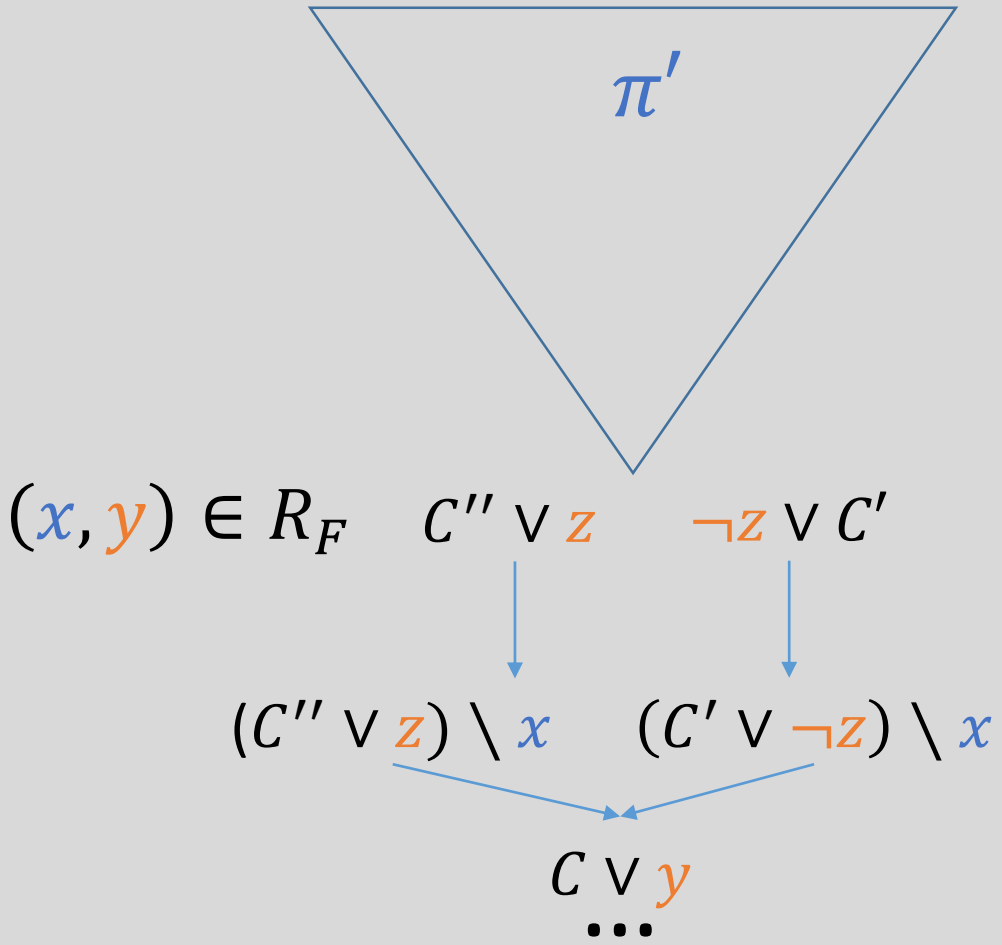
idea lower this resolution step past the universal reduction step

Q(D^{std})-resolution is sound



Because z is to the left of x, it does not interfere with forall reduction.

Q(D^{std})-resolution is sound



Because z is to the left of x, it does not interfere with forall reduction.

So we can raise forall reduction.

repeat for subderivations

Future Work

- Proving soundness of Q(D^{std})-term-resolution
- Proving soundness of Q(D^{res*})-resolution
- Can we extract models without rewriting?
- Proof complexity