Understanding Cutting Planes for QBFs

Olaf Beyersdorff[1] Leroy Chew[1] Meena Mahajan[2]

Anil Shukla[2]

[1] School of Computing, University of Leeds, UK
[2] Institute of Mathematical Sciences Chennai, India

FSTTCS 2016
Chennai, India
December 15th, 2016

Outline

Outline

Resolution

- Introduced by Blake in 1937.
- Resolution is a proof system for proving that boolean formulas in a CNF form are unsatisfiable.
- The only inference rule in resolution is:

$$\frac{C \vee x \qquad D \vee \neg x}{C \vee D}$$

- CNF formula $F \in \text{UNSAT} \implies F$ has a **resolution proof** (completeness).
- A CNF formula F has a **resolution proof** $\implies F \in \text{UNSAT}$ (Soundness).

Resolution Proof

Resolution Proof

- Let $F = \{C_1, \ldots, C_k\}$ be an unsatisfiable formula over n variables.

Resolution Proof

- Let $F = \{C_1, \ldots, C_k\}$ be an unsatisfiable formula over n variables.
- A resolution proof of $F \in UNSAT$ is a sequence of clauses $\pi = \{D_1, \ldots, D_t\}$ such that
  - The last clause D_t is the empty clause $\Box$.
  - Each clause D_q is either one of the initial clauses or is derived from some clause D_m, D_n with $m, n < q$ using the resolution rule.

Resolution Proof

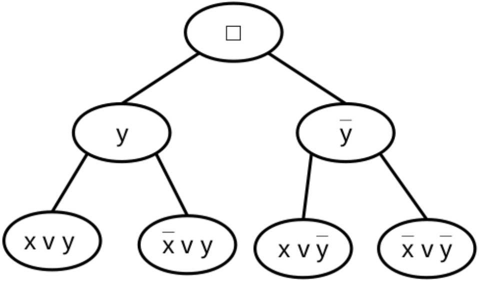
- Let $F = \{C_1, \ldots, C_k\}$ be an unsatisfiable formula over n variables.
- A resolution proof of $F \in UNSAT$ is a sequence of clauses $\pi = \{D_1, \ldots, D_t\}$ such that
  - The last clause D_t is the empty clause $\Box$.
  - Each clause D_q is either one of the initial clauses or is derived from some clause D_m, D_n with $m, n < q$ using the resolution rule.
- If we store pointers from each D_m, D_n to D_q then we actually get a DAG G_π. We call G_π, proof graph associated with π.
- If G_π is a tree then π is called a tree-like resolution proof of F.

Some Examples

- Consider the following unsatisfiable formula on two variables: $(x \vee y) \wedge (\neg x \vee y) \wedge (x \vee \neg y) \wedge (\neg x \vee \neg y)$.

Some Examples

- Consider the following unsatisfiable formula on two variables: $(x \vee y) \wedge (\neg x \vee y) \wedge (x \vee \neg y) \wedge (\neg x \vee \neg y)$.



Outline

Cutting Planes (CP) Proof System

- Introduced by Cook, Coullard, and Turán in 1987 for unsatisfiable CNF formula.
- Cutting planes deals with linear inequalities, not with clauses.
- CNF formula F is first encoded as a set of inequalities $R(F)$.

Encoding F into $R(F)$

- Clause $C = x_1 \vee \neg x_2 \vee x_3$ is encoded as $x_1 + (1 - x_2) + x_3 \geq 1$.
- Clearly any Boolean assignment α satisfies C iff α satisfies $R(C)$.
- Given, $F = C_1 \wedge \cdots \wedge C_m$.
- $R(F) = \{R(C_1), \ldots, R(C_m)\}$ and the inequalities $x \geq 1, -x \geq -1$ $\forall$ variables x, which we called Boolean axioms.
- Boolean axioms force $x \in \{0, 1\}$.

CP Proof

- Let $R(F)$ be a set of inconsistent linear inequalities .
- A CP refutation of $R(F)$ is a sequence of inequalities $\pi = I_1, I_2, \ldots, I_l$ such that:
  - The last inequality $I_l \equiv 0 \geq C$, for some positive integer C, and
  - Each inequality I_j either belongs to $R(F)$ (recall that $R(F)$ also include the Boolean axioms), or,
  - I_j is derived from some earlier inequalities in the sequence via one of the inference rules (i.e., Add, Multiply, or divide).

CP Proof

Add: from $\sum_k c_k x_k \geq C$ and $\sum_k d_k x_k \geq D$ derive $\sum_k (c_k + d_k) x_k \geq C + D$.

Multiply: from $\sum_k c_k x_k \geq C$ derive $\sum_k d c_k x_k \geq dC$, where $d \in \mathbb{Z}^+$.

Divide: from $\sum_k c_k x_k \geq C$ derive $\sum_k \frac{c_k}{d} x_k \geq \left\lceil \frac{C}{d} \right\rceil$, where $d \in \mathbb{Z}^+$ divides each c_k.

Examples

- Consider the CNF formula:
  $(x \vee y) \wedge (\neg x \vee y) \wedge x \vee \neg y) \wedge (\neg x \vee \neg y)$.
- We have the following linear inequalities:
  - $x + y \geq 1$,
  - $(1 - x) + y \geq 1$,
  - $x + (1 - y) \geq 1$, and
  - $(1 - x) + (1 - y) \geq 1$ encoding it.
  - We also have Boolean axioms.

CP Proof Example

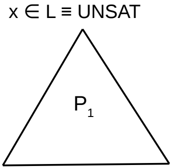
$$\begin{aligned} x + y &\geq 1 \text{ axioms} \\ (1-x) + y &\geq 1 \text{ axioms} \\ 1 + 2y &\geq 2 \text{ after addition} \\ 2y &\geq 1 \text{ after rechange} \\ y &\geq 1 \text{ after division} \end{aligned}$$

$$\begin{aligned} x + (1-y) &\geq 1 \text{ axioms} \\ (1-x) + (1-y) &\geq 1 \text{ axioms} \\ 1 + 2(1-y) &\geq 2 \text{ after addition} \\ 2(1-y) &\geq 1 \text{ after rechange} \\ (1-y) &\geq 1 \text{ after division} \end{aligned}$$

- Now add inequalities $y \geq 1$ and $(1-y) \geq 1$ to derive $0 \geq 1$.

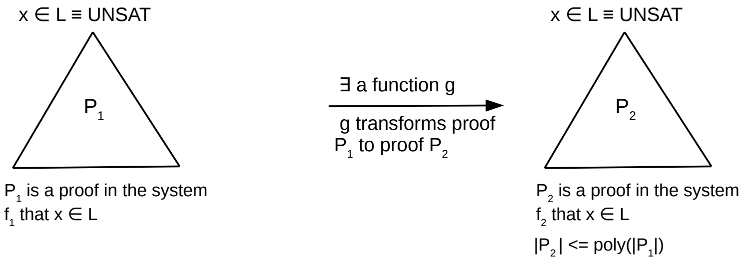
Outline

f_2 Simulates f_1

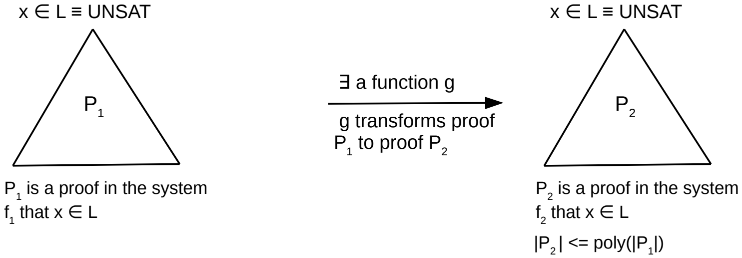


P_1 is a proof in the system f_1 that $x \in L$

f_2 Simulates f_1



f_2 p-simulates f_1



In addition, if g is poly time computable then we say that f_2 p-simulates f_1.

f_2 cannot simulate f_1

- Intuitively, f_2 cannot simulate f_1 if there exists a family of polynomial sized formulas F_n, such that,
  - F_n has short proof in f_1 but,
  - Requires exponential sized proofs in the system f_2.
- If f_1 cannot simulate f_2 and f_2 cannot simulate f_1 then the proof systems f_1 and f_2 are **incomparable**.

Resolution vs Cutting Planes

- Cutting Planes p-simulates Resolution ((Cook, Coullard, and Turán 1987).
- Resolution cannot simulate Cutting Planes (witness family PHP_n: based on pigeonhole principle).

Outline

Quantified Boolean Formulas (QBFs)

- Consider a false QBF formula

$$\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_i x_i \ldots \mathcal{Q}_j x_j \ldots \mathcal{Q}_n x_n.\ F,$$

  where F is a quantifier free CNF formula over variables $x_1, \ldots, x_n$, each $\mathcal{Q} \in \{\exists, \forall\}$.
- We say x_i is on left of x_j or x_i is before x_j.
- x_n is the innermost variable (rightmost variable).
- Several Resolution based proof system have been developed for false QBFs. For example Q-Res, QU-Res and so on.

Q-Res: Definition

- Q-Res = resolution + ∀-reduction [Kleine Büning, Karpinski, and Flögel; 1995].
- Q-Res proof system proofs the falseness of QBF formulas.
- Q-Res has two inference rules:
  - **Resolution rule**: $\frac{C \vee x \quad D \vee \neg x}{C \vee D}$, where x is existential literal and $C \vee D$ is not a tautology.
  - ∀-**reduction**: $\frac{C \vee x}{C}$, where x is universal variable, and all existential variable in C are before x in the prenex of the given QBF formula.
- If the resolution rule is also permitted on universal variables, then we get QU-Res proof systems (Allen Van Gelder; 2012).

Expansion Based QBF Resolution Proof System

- There are two main paradigms in QBF solving: Expansion based solving and CDCL solving.
- An example of CDCL based QBF proof system is Q-Res (which we have seen).
- An example of expansion based QBF proof system is $\forall$Exp+Res [Janota and Marques-Silva; 2013].

Outline

CP+∀red Proof System

- We introduced a new proof systems for false QBFs based on Cutting Planes.
- CP+∀red = Cutting Planes + ∀-Red Rules.
- Like Cutting Planes, CP+∀red works with linear inequalities.
- Given a false QBF $\mathcal{F} \equiv \mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ F$, where $F = C_1 \wedge \cdots \wedge C_m$.
- Encode it as $\phi \equiv \mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ \phi_F$, where $\phi_F = \{R(C_1), \ldots, R(C_m)\} \cup B$, B is the set of Boolean axioms.
- Clearly $\mathcal{F}$ is false iff ϕ is false.

CP+∀red Refutations

- A CP+∀red proof π of ϕ (and therefore of $\mathcal{F}$) is a quantified sequence of inequalities, that is
- $\pi \equiv \mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ [I_1, \ldots, I_l]$ where, the last inequality $I_l \equiv 0 \geq C$, for some positive constant C. For every $j \in \{1, \ldots, l\}$,
  - $I_j \in \phi_F$ (recall that ϕ_F also includes the Boolean axioms), or
  - I_j is derived from the earlier inequalities in the sequence via Add, Multiply, Divide (same as in Cutting Planes proof system), or ∀-Red rule.

CP+∀red Refutations

- ∀-Red rule: From $\sum_{k\in[n]\setminus\{i\}} c_k x_k + h x_i \geq C$ derive

$$\begin{cases} \sum_{k\in[n]\setminus\{i\}} c_k x_k \geq C & \text{if } h > 0; \\ \sum_{k\in[n]\setminus\{i\}} c_k x_k \geq C - h & \text{if } h < 0. \end{cases}$$

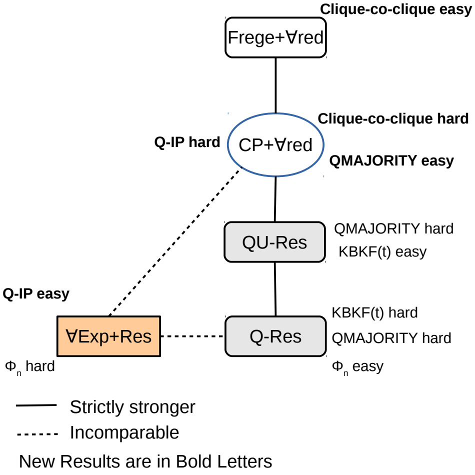
- This rule can be used provided variable x_i is universal, and provided all existential variables y with nonzero coefficients in the hypothesis should come before x_i. (That is, if x_j is existential and $c_j \neq 0$, then $j < i$.
- Observe that when $h > 0$, we are replacing x_i by 0, and when $h < 0$, we are replacing x_i by 1. We say that the universal variable x_i has been reduced.

CP+$\forall$red is Complete and Sound for false QBFs

- $\mathcal{F}$ is false QBF $\implies$ $\mathcal{F}$ (its encoding) has a CP+$\forall$red refutation.
  - Proved by showing that CP+$\forall$red p-simulates QU-Res which is known to be complete for false QBFs.
- There is a CP+$\forall$red refutation of $\mathcal{F}$ (its encoding) $\implies$ $\mathcal{F}$ is a false QBF.
  - Because the inference rules are sound.

Outline

CP+∀red is above QU-Res and below Frege+∀red but Incomparable with expansion-based calculi



Outline

Strategy Extraction

- $\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ F$ can be seen as a game between universal ($\forall$) and existential ($\exists$) players.
- A strategy for any universal variable u is a function from all the variables before u to $\{0, 1\}$.
- A QBF $\mathcal{F}$ is false iff there exists a winning strategy for the universal player.
- A QBF proof system has a strategy extraction property for a particular circuit size $\mathcal{C}$ whenever we can efficiently extract from every refutation π of a QBF formula $\mathcal{F}$ a winning strategy for the universal player in the circuit class $\mathcal{C}$.

Strategy Extraction for CP+∀red

- We have shown that from CP+∀red proof of length l (number of inequalities), we can extract a winning strategy for the universal player as an LTF-decision list of length l. Using it we showed exponential lower bound for CP+∀red.

Decision lists (Rivest 1987)

- A *decision list* is a list D of pairs

$$(t_1, v_1), \ldots, (t_r, v_r)$$

  where each t_i is a term (conjunction, $\wedge$, of literals), and
- v_i is a value in $\{0, 1\}$, and
- The last term t_r is the constant term **true** (i.e., the empty term). The length of D is r.

Decision lists (Rivest 1987)

- A decision list D defines a Boolean function as follows:
  - For any assignment α, $D(\alpha)$ is defined to be equal to v_j where j is the least index such that $t_j|_\alpha = 1$.
  - Such an item always exists, since the last term always evaluates to 1.

LTF-decision lists (Marchand and Golea 1993)

- In LTF-decision lists, instead of terms one uses linear threshold functions.
- Linear threshold functions are of the form:

$$\sum a_i x_i \geq t,$$

where a_i and t are integers (real number also allowed, but we do not need this.)

Inner Product Function and LTF-decision Lists lower bound

- Inner product function computes Inner product $\pmod 2$ of two Boolean vectors. That is,

$$\forall x, y \in \{0,1\}^n, \quad \mathsf{IP}(x,y) = \begin{cases} 1 & \text{if } \sum_i x_i y_i \equiv 1 \pmod 2 \\ 0 & \text{otherwise} \end{cases}$$

Theorem (Turán and Vatan 1997)

Every LTF-decision lists computing Inner Product $\pmod 2$ function has length greater than $2^{n/2} - 1$.

Lower Bounds via Strategy Extraction

- Consider the formula based on IP:
  $Q\text{-}IP \equiv \exists\vec{x}\forall z.\ [IP(\vec{x}) \neq z]$
- Clearly the only winning strategy for the universal variable z is $(z \leftarrow IP(\vec{x}))$.
- We can easily encode the above formula as a short QBF.
- If the formula has a CP+$\forall$red proof of length l (number of inequalities) then by strategy extraction we can extract LTF-decision list of length l, which is a winning strategy for z, and hence computing $IP(\vec{x})$. It follows that l must be exponential.

Outline

Hard Formula: clique-co-clique formula

- We show that the clique-co-clique formula (Beyersdorff, Chew, Mahajan, and S.; 2015) is hard for CP+∀red. The formula encodes that the given graph on n vertices both has and does not have a k clique.
- Consider the formula (not in prenex form).

$$\exists \vec{p}\Big[\exists \vec{q}.\ \underbrace{A(\vec{p},\vec{q})}_{\text{Encodes that the graph given by } \vec{p} \text{ has a clique of size } k} \bigwedge$$

$$\forall \vec{r}\exists \vec{t}.\ \underbrace{B(\vec{p},\vec{r},\vec{t})}_{\text{Encodes that the nodes specified by } \vec{r} \text{ fail to form a } k \text{ clique in the graph } \vec{p}}$$

$$\Big]$$

Hard Formula: clique-co-clique formula

$$\exists \vec{p}\Big[\underbrace{\exists \vec{q}.\ A(\vec{p}, \vec{q})}_{\text{Is true if the graph given by } \vec{p} \text{ has a clique of size } k} \bigwedge \underbrace{\forall \vec{r} \exists \vec{t}.\ B(\vec{p}, \vec{r}, \vec{t})}_{\text{Is true if the graph given by } \vec{p} \text{ has no } k \text{ clique}} \Big]$$

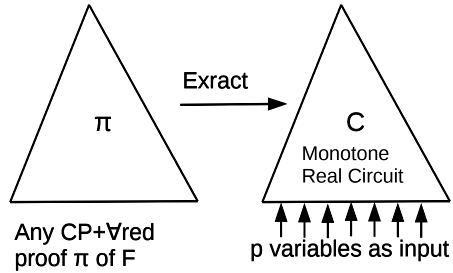
- Here variables $\vec{p}, \vec{q}, \vec{r}$, and $\vec{t}$ are disjoint.
- So we have the following QBF in closed prenex form.

$$\exists \vec{p} \exists \vec{q} \forall \vec{r} \exists \vec{t}.\ \left[A(\vec{p}, \vec{q}) \wedge B(\vec{p}, \vec{r}, \vec{t})\right]$$

Proof Idea

F = ∃ p ∃q ∀r∃t. [A(p,q) ∧ B(p, r, t)]

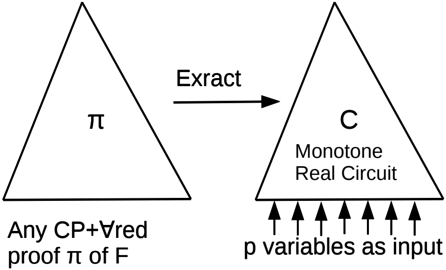
If p occurs positively in A(p,q) part then



Proof Idea

$F = \exists p\, \exists q\, \forall r \exists t.\ [A(p,q) \wedge B(p, r, t)]$

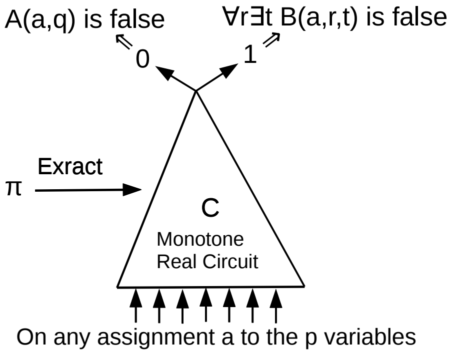
If p occurs positively in A(p,q) part then



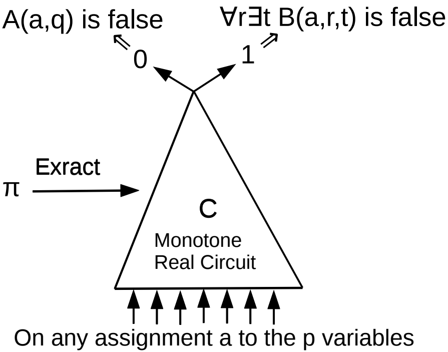
Such that:
Size of C is polynomial in the length (number of linear inqualities) of π,

Proof Idea

$F = \exists p\, \exists q\, \forall r \exists t.\ [A(p,q) \wedge B(p, r, t)]$



Proof Idea



Clearly, C is solving the k-clique problem for the given graph.
So for some appropriate k, the circuit C and therefore the proof π must be of exponential length (Pavel Pudlák; 1997).

Thank you.