Automatization and Non-Automatizability

Tobias Lieber

July 11, 2009

Outline

- Motivation
- Non-Automatizability
  - Complexity Theory for "hard" problems
  - Resolution
  - Polynomial Calculus
- Connection between Resolution and Res(k)
  - Repetition and Preliminaries

Motivation

Up to now: Lower bounds for propositional logic
If there is a short proof, then we want to find it

Definition

Definition

A proof system P is (quasi-)automatizable if there is a deterministic algorithm which returns in (quasi-)polynomial time of the shortest P-proof of a tautology τ its P-proof.

Definition

Definition

A proof system P is (quasi-)automatizable if there is a deterministic algorithm which returns in (quasi-)polynomial time of the shortest P-proof of a tautology τ its P-proof.

Definition

A proof system P is weakly automatizable if there is a proof system S that p-simulates P and is automatizable.

Approximation Algorithms

Definition

The approximation ratio ρ of an algorithm for an optimization problem is defined by

$$\rho := \max\left\{\frac{OPT(A)}{OPT}, \frac{OPT}{OPT(A)}\right\}.$$

Definition

An optimization problem has a polynomial time approximation scheme (PTAS), if there is an algorithm, which for every $\epsilon > 0$ computes, in time of at most $n^{O(\frac{1}{\epsilon})}$, an $(1 + \epsilon)$-approximation.

Definition

An optimization problem has a polynomial time approximation scheme (PTAS), if there is an algorithm, which for every $\epsilon > 0$ computes, in time of at most $n^{O(\frac{1}{\epsilon})}$, an $(1+\epsilon)$-approximation.

Definition

An optimization problem has an efficient polynomial time approximation scheme (EPTAS), if there is an algorithm, which for every $\epsilon > 0$ computes, in time of at most $f(\frac{1}{\epsilon})p(n)$, an $(1+\epsilon)$-approximation (p a polynomial, f computable).

Parametrized Complexity

Definition

$\mathcal{FPT}$ consists of all languages $L \subseteq \Sigma^* \times \mathbb{N}$ for which there exists an algorithm Φ, a constant c and a recursive function $f : \mathbb{N} \to \mathbb{N}$ such that:

- the running time of $\Phi(x, k)$ is at most $f(k)|x|^c$
- $(x, k) \in L$ iff $\Phi(x, k) = 1$

Definition

The class $\mathcal{W}[\mathcal{P}]$ contains all the problems which can be parametrized reduced to weighted circuit satisfiability:
Input: A circuit C and an positive integer k.
Question: Is there a satisfying assignment with k ones?

Definition

The problem monotone minimum circuit satisfying assignment (MMCSA) is an optimization problem with a circuit C with n variables as input as input.
Objective function: $\sigma(a)$ which returns the number of ones in an assignment $a \in \{0,1\}$ such that $C(a) = 1$.

Definition

$$\sigma(C) = \min_{a \text{ is solution of MMCSA}} \sigma(a)$$

Definition

The class $\mathcal{FPR}$ of parametrized problems consists of all languages $L \subseteq \Sigma^* \times \mathbb{N}$ for which there is a probabilistic algorithm Φ, a constant c and a recursive function $f : \mathbb{N} \to \mathbb{N}$ such that:

- $\Phi(x, k)$ runs in at most $f(k)|x|^c$
- if $(x, k) \in L$ then $Pr[\Phi(x, k) = 1] \geq \frac{1}{2}$
- if $(x, k) \notin L$ then $Pr[\Phi(x, k) = 1] = 0$

Self Improvement

Lemma

For every fixed integer $d \geq 1$ there exists a polynomial time computable function π which maps monotone circuits into monotone circuits with $\sigma(\pi(C)) = \sigma(C)^d$ for all C.

Fact

$$\mathcal{FPT} \subseteq \mathcal{FPR}$$

$$\mathcal{FPT} \subseteq \mathcal{W}[\mathcal{P}]$$

Fact

$$\mathcal{FPT} \subseteq \mathcal{FPR}$$

$$\mathcal{FPT} \subseteq \mathcal{W}[\mathcal{P}]$$

Fact

The decision version of MMCSA is $\mathcal{W}[\mathcal{P}]$-complete.

Fact

If a problem A has an EPTAS then A is in $\mathcal{FPT}$.

What do we want to show?

Goal

If Resolution or tree-like Resolution is automatizable, then $\mathcal{W}[\mathcal{P}] \subseteq \text{co-}\mathcal{FPR}$.

What do we want to show?

Goal

If Resolution or tree-like Resolution is automatizable, then $\mathcal{W}[\mathcal{P}] \subseteq \text{co-}\mathcal{FPR}$.

Roadmap

1. Create a PTAS
2. Get rid of the exponent

Lemma

There exists a polynomial time computable function τ *which maps any pair* $(C, 1^m)$*, with a monotone circuit* C *and an integer* m*, to an unsatisfiable CNF* $\tau(C, m)$ *such that:*

$$S_T(\tau(C, m)) \leq |C| m^{O(\min\{\sigma(C), \log m\})}$$

and

$$S(\tau(C, m)) \geq m^{O(\min\{\sigma(C), \log m\})}.$$

Lemma

If Resolution or tree-like Resolution is automatizable then there exists an constant $h > 1$ *and an algorithm* Φ *working on pairs* (C, k)*, where* C *is a monotone circuit and* k *is an integer such that:*

- *the running time of* $\Phi(C, k)$ *is at most* $\exp(O(k^2))|C|^{O(1)}$
- *if* $\sigma(C) \leq k$ *then* $\Phi(C, k) = 1$
- *if* $\sigma(C) \geq hk$ *then* $\Phi(C, k) = 0$*.*

Lemma

If Resolution or tree-like Resolution is automatizable then there exists an constant $h > 1$ and an algorithm Φ working on pairs (C, k), where C is a monotone circuit and k is an integer such that:

- *the running time of $\Phi(C, k)$ is at most $\exp(O(k^2))|C|^{O(1)}$*
- *if $\sigma(C) \leq k$ then $\Phi(C, k) = 1$*
- *if $\sigma(C) \geq hk$ then $\Phi(C, k) = 0$.*

Proof.

$r := 2^{h \max\{k, \frac{\log |C|}{k}\}}$
$S(C, r)$: build CNF, simulate refutation, stop after $(r^k|C|)^{h_0}$ steps
if $S(C, r) \geq (|C|r^k)^{h_1}$ return 1 otherwise 0 □

Theorem

If Resolution or tree-like Resolution is automatizable then for any fixed $\epsilon > 0$ *there exists an algorithm* Φ *receiving as input a monotone circuit* C *which runs in time* $\exp(\sigma(C)^{O(1)})|C|^{O(1)}$ *and approximates* $\sigma(C)$ *within a factor* $1 + \epsilon$*.*

Theorem

If Resolution or tree-like Resolution is automatizable then for any fixed $\epsilon > 0$ there exists an algorithm Φ receiving as input a monotone circuit C which runs in time $\exp(\sigma(C)^{O(1)})|C|^{O(1)}$ and approximates $\sigma(C)$ within a factor $1+\epsilon$.

Proof.

From the last lemma we can construct an approximation algorithm with approximation ratio h:
Compute $\Phi(C,1)\dots\Phi(C,l)$ while $\Phi(C,l)\neq 0$ and return l if $\Phi(C,l)=0$ □

Theorem

If Resolution or tree-like Resolution is automatizable then $\mathcal{W}[\mathcal{P}] \subseteq co\text{-}\mathcal{FPR}$.

Theorem

If Resolution or tree-like Resolution is automatizable then $\mathcal{W}[\mathcal{P}] \subseteq$ *co-*$\mathcal{FPR}$.

Proof.

Construct a (randomized) circuit $\beta(C, k)$ and $\alpha(k)$ in polynomial time:

$$\sigma(C) \leq k \Rightarrow Pr[\sigma(\beta(C,k)) \leq \alpha(k)] = 1$$
$$\sigma(C) \geq k+1 \Rightarrow Pr[\sigma(\beta(C,k)) \geq 2\alpha(k)] \geq \frac{1}{2}$$

□

Fact

P[A set of s circuits has less or equal than $sn - a$ input circuits] $\leq N^k \left(\frac{4s^2n^2}{N}\right)^a$

$$P[\beta(C,N,d) \text{ is bad}] \leq \sum_{i=1}^{d-1} N^{k_{i+1}} \left(\frac{4k_{i+1}^2 n^2}{N}\right)^{k_{i+1}\sqrt{k}}$$

$$= \sum_{i=1}^{d-1} \left(\frac{4k_{i+1}^2}{n^{1-3/\sqrt{k}}}\right)^{k_{i+1}\sqrt{k}} \leq \sum_{i=1}^{d-1} \left(\frac{1}{3}\right)^{k_{i+1}\sqrt{k}} \leq \frac{1}{2}$$

Polynomial Calculus

- There is an algorithm which works in cubic time of the size of the dense representation.
- Shown results hold for PC, too.

Want to show: Resolution is weakly automatizable iff Res(2) has feasible interpolation.

Definition

The variable $z_{l_1,\ldots,l_s}$ of variables $l_1, \ldots, l_s$ is constituted by its defining clauses:

$$\neg z_{l_1,\ldots,l_s} \vee l_i \quad \forall i \in [s]$$
$$z_{l_1,\ldots,l_s} \vee \neg l_1 \vee \cdots \wedge \neg l_s$$

It can be interpreted as $l_1 \wedge \cdots \wedge l_s$.

Want to show: Resolution is weakly automatizable iff Res(2) has feasible interpolation.

Definition

The variable $z_{l_1,\dots,l_s}$ of variables $l_1,\dots,l_s$ is constituted by its defining clauses:

$$\neg z_{l_1,\dots,l_s} \vee l_i \quad \forall i \in [s]$$
$$z_{l_1,\dots,l_s} \vee \neg l_1 \vee \dots \wedge \neg l_s$$

It can be interpreted as $l_1 \wedge \dots \wedge l_s$.

Definition

The set C_k of a set of clauses C is the union of C with all the defining clauses for the variables $z_{l_1,\dots,l_s}$.

Lemma

If the set of clauses C has a $Res(k)$ refutation of size S, then C_k has a Resolution refutation of size $O(kS)$. If the $Res(k)$ refutation is tree-like, then the Resolution refutation is also tree-like.

Definitions

The set $REF(S)$ is the set of pairs (C, m) with an CNF formula C that has an S-refutation with size m.
The set SAT^* contains the pairs (C, m) such that C is a satisfiable CNF formula.
$(REF(S), SAT^*)$ is called the canonical pair of S.
A canonical pair is separable if there is an algorithm running in polynomial time and returns *false* on every input from $REF(S)$ and *true* if (C, m) is in SAT^*.

Definitions

The set $REF(S)$ is the set of pairs (C, m) with an CNF formula C that has an S-refutation with size m.
The set SAT^* contains the pairs (C, m) such that C is a satisfiable CNF formula.
$(REF(S), SAT^*)$ is called the canonical pair of S.
A canonical pair is separable if there is an algorithm running in polynomial time and returns *false* on every input from $REF(S)$ and *true* if (C, m) is in SAT^*.

Reflection Principle

Definition

A CNF formula which is true iff

- z encodes a truth assignment of a CNF x
- x is of size r and uses n variables

is called $SAT_n^r(x, z)$.
Let us call a CNF $REF_{r,m}^n(x, y)$ if it evaluates to true iff

- y encodes an S-refutation of a CNF x
- the size of the refutation is m
- x is of size r and uses n variables

The collection of the CNFs $REF_{r,m}^n(y, z) \wedge SAT_r^n(x, z)$ is the Reflection Principle of S.

Definition

A proof system S has the interpolation property in time $T = T(m)$ if there is an algorithm which runs in time T and decides for an contradictory CNF $B := A_0(x, y_0) \wedge A_1(x, y_1)$ (x, y_0, y_1 are disjoint sets) if $A_0(x, y_0)$ or $A_1(x, y_1)$ is contradictory where m is the minimal size of an refutation of B.
If $T(m)$ is polynomial in m then S has feasible interpolation.

Theorem (Pudlak)

If the reflection principle of S has polynomial sized refutations in a proof system that has feasible interpolation, then the canonical pair for S is separable in polynomial time.

Theorem

The Reflection Principle for Resolution $SAT_r^n(x,z) \wedge REF_{r,m}^n(x,y)$ *has* $Res(2)$ *refutations of size* $(nr+nm)^{O(1)}$.

Theorem

The Reflection Principle for Resolution $SAT_r^n(x,z) \wedge REF_{r,m}^n(x,y)$ *has* $Res(2)$ *refutations of size* $(nr+nm)^{O(1)}$.

Lemma

If $Res(2)$ *has feasible interpolation, then Resolution is weakly automatizable.*

Corollary (Pudlak)

The canonical pair of a proof system S is separable in polynomial time iff S is weakly automatizable.

Corollary (Pudlak)

The canonical pair of a proof system S is separable in polynomial time iff S is weakly automatizable.

Theorem

If Resolution is weakly automatizable, then $Res(2)$ has feasible interpolation.

End

Thank you for your attention.