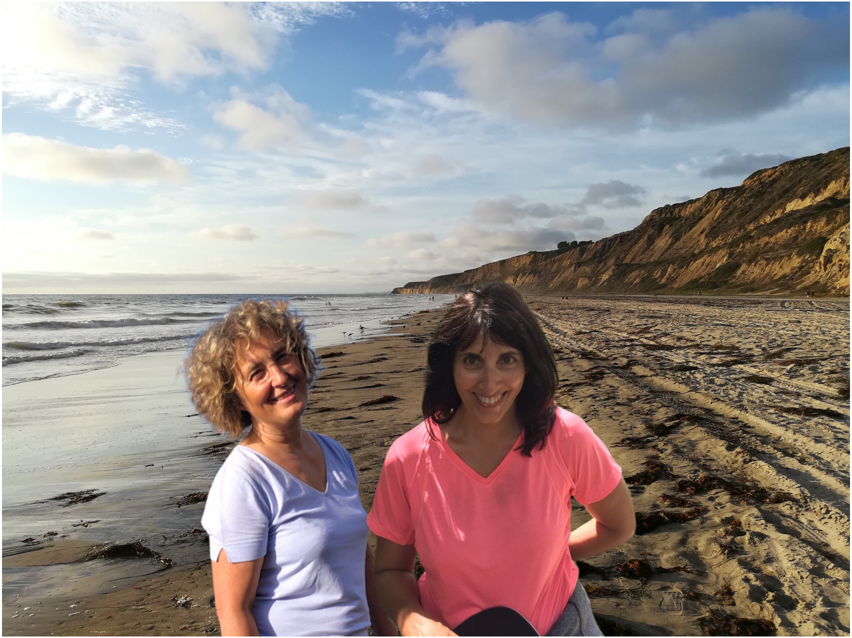
My Collaboration with Toni

Weak Automat∗ability

Maria Luisa Bonet



Propositional Proof System

Definition: A propositional proof system is a polynomial time computable function from $\{0,1\}^*$ onto TAUT.

A pps is a polynomial time proof-verification algorithm P. On input (x,F), if P accepts the pair (x,F), we say that x is a P-proof of F.

Questions:

- How big (in size) is the proof of a tautology in a given proof system?
- What is the cost (in time) of finding the (smallest) proof?

Automatizable Proof Systems

Definition [Bonet-Pitassi-Raz, 97] A proof system P is automatable if there exists an algorithm that takes as input a formula F and returns a proof p of F in the system P in poly time in the size of the shortest P-proof of F.

Variants of the Definition:

- quasy-poly time $n^{O(\log n)}$ automatizable
- A proof system P is Weakly Automatable if there is an automatable proof system that simulates P

Definition: Propositional proof system Q p-simulates P, if there is a polynomial-time function f such that Q(f(x)) = P(x) for all x.

Equivalences of Weak Automatability definitions

A pair (A,B) is a disjoint NP-pair if A,B $\in$ NP and A $\cap$ B $= \emptyset$.

Definition [Razborov] Canonical NP-pair for a propositional proof system P is:

Ref(P) $= \{(\phi, 1^m) |$ P has a refutation of ϕ of size m$\}$

SAT$=\{(\phi, 1^m) | \phi$ is satisfiable$\}$

The following are equivalent:

- The canonical NP-pair for a pps P is polynomially separable.
- A system P is Weakly Automatable if there is an automatable system that simulates P
- P is Weakly Automatable if there exists an algorithm that takes as input a formula F and returns a proof p of F in poly time in the size of the shortest P-proof of F.

Interpolation [Krajicek]

Observation: If $F(\vec{x}, \vec{y}) \wedge G(\vec{x}, \vec{z})$ is unsatisfiable, then, given any assignment $\vec{\alpha}$ for $\vec{x}$, either $F(\vec{\alpha}, \vec{y})$ is unsatisfiable or $G(\vec{\alpha}, \vec{z})$ is unsatisfiable.

Interpolation Problem:
Given an unsatisfiable formula $F(\vec{x}, \vec{y}) \wedge G(\vec{x}, \vec{z})$ and an assignment α to the x variables,

return 0 if $F(\vec{\alpha}, \vec{y})$ is unsatisfiable,
return 1 if $G(\vec{\alpha}, \vec{z})$ is unsatisfiable.

Definition: P has feasible interpolation if the Interpolation problem is solvable in polynomial time respect to the smallest P-refutation of $F \wedge G$.

Relationship between automatizability and interpolation

Theorem[Impagliazzo, Bonet-Pitassi-Raz] If P is automatizable, then P has feasible interpolation.

Proof Sketch

Let n be the size of the smallest P-refutation of $F(\vec{x}, \vec{y}) \wedge G(\vec{x}, \vec{z})$.
Let α be an assignment on the x variables.
Run the automatization algorithm on F for $p(n)$ steps.
If it succeeds return 0, otherwise return 1.

Idea: Show P doesn't have feasible interpolation, under assumptions?

Idea goes back to [Krajicek-Pudlak] for Extended Frege.

Frege Proof Systems

Frege

A few axioms schemes like:

$$A \wedge B \to A$$
$$A \to (B \to A \wedge B)$$
$$A \to (B \to A)$$

plus the Modus Ponens rule of inference:

$$\frac{A \qquad\qquad A \to B}{B}$$

Bounded Depth Frege or AC_0-Frege

Frege where all formulas have a constant number of $\wedge/\vee$ alternations, and connectives have unbounded degrees.

TC_0-Frege

Bounded Depth Frege + threshold and parity connectives and rules for them.

Diffie-Hellman Cryptographic Scheme

Alice and Bob want to establish secret shared key.
large prime number P, generator g of Z_p^* (public)

Alice		Bob
large secret a		large secret b
	$\longrightarrow g^a \bmod P \longrightarrow$	
	$\longleftarrow g^b \bmod P \longleftarrow$	
$g^{ab} \bmod P$		$g^{ab} \bmod P$

Note: If $P = p_1 p_2$ where p_1 and p_2 are primes, then breaking D-H is harder than factoring.

No feasible interpolation for Frege Proof Systems

Theorem [Bonet-Pitassi-Raz] Frege Systems and even TC_0-Frege Systems (refutational) do not have feasible interpolation, unless factoring is solvable in polynomial time.

Proof Sketch Let m be a number and g a generator of Z_m^*. Let

$A_0(X, Y, a, b)$ be
$X = g^a \bmod m$ and $Y = g^b \bmod m$ and last bit of $g^{ab} \bmod m$ is 0

and $A_1(X, Y, c, d)$ be
$X = g^c \bmod m$ and $Y = g^d \bmod m$ and last bit of $g^{dc} \bmod m$ is 1

$A_0 \wedge A_1$ is unsat. since
$g^{ab} = X^b = g^{cb} = g^{bc} = Y^c = g^{dc} \bmod m$.
and has small refutations in the Frege proof system.

Now, feasible interpolation would imply that Diffie-Hellman Bit-Commitment is unsecure, and this implies that factoring is easy.

Non-automatability and non weak-automatability

Under the cryptographic assumption:

- Frege or even TC_0-Frege don't have feasible interpolation
- No system that simulates Frege or TC_0-Frege has feasible interpolation
- Frege or even TC_0-Frege are not automatizable
- Frege or even TC_0-Frege are not weakly automatizable

Non-automatizability for Bounded Depth Frege

Theorem [Bonet-Domingo-Gavaldà-Maciel-Pitassi] AC_0-Frege Systems do not have feasible interpolation, unless factoring can be computed in subexponential time.

- There exist AC_0 circuits (of depth $2k$) of size polynomial in n to add $\log^k n$ bits.
- TC_0-Frege proofs of size polynomial in n in which all the threshold and parity connectives have fan-in polylog n can be simulated by AC_0-Frege proofs of size polynomial in n.
- AC_0-Frege doesn't have feasible interpolation, unless factoring can be computed by sub-exponential size circuits.
- AC_0-Frege is not automatizable or weakly automatizable, under the same assumption.

Non Weakly Automatable proof systems under assumptions

[Krajicek-Pudlak] Extended Frege

[Bonet-Pitassi-Raz] Frege, TC_0 Frege

[Bonet-Domingo-Gavalda-Maciel-Pitassi] AC_0 Frege.

Non Automatable proof systems under assumptions

[Atserias-Müller, Alekhnovich-Razborov] Resolution.

[Garlik] Res(k).

[deRezende-Göös-Nordström-Pitassi-Robere-Sokolov]
Nullstallensatz and Polynomial Calculus.

[Göös-Koroth-Mertz-Pitassi] Cutting Planes.

[Grosser-Robere?] Sherali-Adams.

Open: Sum-of-Squares

Discussion

Thanks to Albert Atserias and Pavel Pudlak

Discussion

Given a simple graph game, deciding whether a player has a winning strategy is in $NP \cap coNP$.

[Atserias-Maneva] If depth 2 Frege is weakly automatizable, mean payoff games can be decided in polynomial time.

[Pitassi-Huang] If depth 2 Frege is weakly automatizable, then simple stocastic games can be decided in polynomial time.

[Beckmann-Pudlak-Thapen] If resolution is weakly automatizable, then parity games can be decided in polynomial time.

But:

[Calude-Jain-Khoussainov-Li-Stephan] Quasi-polynomial time algorithm solving parity games.

Discussion

Dead ends in trying to show weak automatability of Resolution

- Proof systems like Polynomial Calculus, Sheraly-Adams, Sum-of-squares,... are stronger than Resolution.
- These systems are not automatable.
- They have efficient algorithms to find proofs of small degree (or small degree and polynomial coefficients).
- Could these algorithm be automatable procedures for Resolution?
- NO

Discussion

[Bonet-Galesi] The Ordering Principle requires high Resolution width, but it has small Resolution refutations.

[Galesi-Lauria] The graph ordering principle requires high degree for PC.

[Potechin]The ordering principle requires high degree to refute in SOS.

The $Res(k)$ Resolution System

Clauses are disjunctions of conjunctions of up to k literals:

$$(l_1^1 \wedge \cdots \wedge l_{s_1}^1) \vee \cdots \vee (l_1^r \wedge \cdots \wedge l_{s_r}^r) \qquad s_1, \ldots, s_r \leq k$$

Rules of inference:

$$\frac{A}{A \vee B} \qquad \text{Weakening}$$

$$\frac{A \vee l_1 \qquad B \vee (l_2 \wedge \cdots \wedge l_s)}{A \vee B \vee (l_1 \wedge l_2 \wedge \cdots \wedge l_s)} \qquad \wedge\text{-Introduction}$$

$$\frac{A \vee (l_1 \wedge \cdots \wedge l_s) \qquad B \vee \neg l_1 \vee \cdots \vee \neg l_s}{A \vee B} \qquad \text{Cut}$$

Discussion

Reflexion Principle: $SAT^n_m(x,z) \wedge REF^n_{m,s}(x,y)$

[Pudlak] If the reflection principle of f has polynomial-size refutations in a proof system that has feasible interpolation, then f is weakly automatizable.

[Atserias-Bonet] Res(2) proves the reflexion principle of Resolution.

[Atserias-Bonet] If F has a Res(k) refutation of size S, then F(k) has a Resolution refutation of size O(kS).

[Atserias-Bonet] For constatn $k > 1$, quivalence between:
(i) Resolution is weakly automatizable
(ii) Res(k) is weakly automatizable
(iii) Res(k) has feasible interpolation.

Discussion

Does Res(2) have feasible interpolation?

[Esteban-Galesi-Messner] tree-like Res(2) has monotone feasible interpolation.

Res(2) does not have monotone feasible interpolation.

[Garlik] Res(k) doesn't have the feasible disjunction property.

Discussion

What about proof systems that have feasible interpolation? Could they prove the reflexion principle of Resolution?

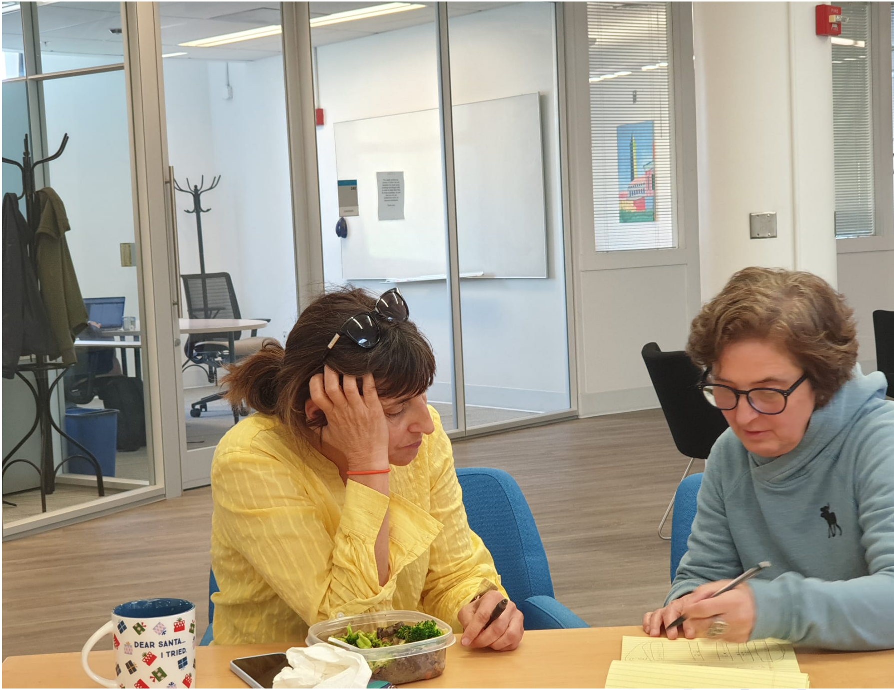
[Bonet-Pitassi-Raz, Pudlak, Krajkcek] Cutting Planes has monotone feasible interpolation. But, CP requires exponential size refutations of the reflexion principle for Resolution [Pudlak]

[Fleming-Göös-Grosser-Robere] Sheraly-Adams has monotone feasible interpolation.

[Pudlak-Sgall, Hakoniemi] Polynomial Calculus has monotone feasible interpolation.

[Hakoniemi] Sum-of-Squares has feasible interpolation.

[M. Oliveira-Pudlak] Lovász-Schrijver monotone feasible interpolation.


DEAR SANTA...
I TRIED.