

Exponential separations: Tree- vs Dag-like Resolution

Nicola Galesi

Dipartimento di Informatica
Università degli Studi di Roma “La Sapienza”

30 Marzo – 3 Aprile
Dipartimento di Informatica “R. Capocelli”

History of Results

The Result.

A family of UNSAT formulae requiring exponential size TLR but admitting polynomial size DLR.

History

- (1) [Goerdt 92] gave a first example of UNSAT formula with polynomial size DLR but requiring quasipolynomial TLR refutations (ad hoc modification of the PHP)
- (2) [Bonet,Galesi,Esteban,Johannsen 98] Gave the first exponential separation. Use the Interpolation method together with a circuit lower bound for monotone real NC hierarchy.
- (3) [Ben-Sasson, Impagliazzo,Wigderson03] Simplify and slightly improve the result for Resolution

Plan of the day

1. First we introduce a general setting to prove lower bound for tree-like Resolution: the **Prover-Delayer Game**
2. Then we introduce **Pebbling Games** on graphs and briefly discuss an important old result of [Celoni,Paul,Tarjan 77]
3. We build a family of UNSAT formulae **Peb(G)** encoding a principle related to pebbling of directed acyclic graphs G
4. We will show there are polynomial size DLR refutations for **Peb(G)**, for any dag G.
5. Using the Prover Delayer game and the result of [CPT 77] we will prove that there exists a graph G s.t. Peb(G) requires exponential size proofs in TLR

Notions and Techniques

1. the **Prover-Delayer Game**
2. **Pebbling Games** and [CPT 77] result
3. Construction of **Peb(G)**
4. [BSIW03] proof

Prover Delayer Game

Prover Delayer Game

Definition.

In the PD game there are two players playing on a UNSAT k-CNF formula F

Prover: asks for variables x of F

Delayer: answers with a value for x or leaves it unset
and win 1\$

Prover wins as soon as he falsifies a clause of F

Objective: Maximize the gain of Delayer

Prover Delayer Game

Example.

$$F =_{def} (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge \\ (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

I round

Prover: x_1 ?

Delayer: unset

Gain: 1\$

Prover $x_1=T$

$$F[x_1=T] =_{def} (x_2 \vee x_3) \wedge (\neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_3)$$

II round

Prover: x_2 ?

Delayer: unset

Gain: 2\$

Prover $x_2=T$

$$F[x_1=T, x_2=T] =_{def} (x_3) \wedge (\neg x_3)$$

III round

Prover: x_3 ?

Delayer: unset

Gain: 3\$

Prover $x_3=F \rightarrow$ WIN

TLR proof size vs PD game

Idea.

“Good” strategies for the Delayer on a unsat CNF F , give
“good” lower bounds for TLR refutations of F

Thm [Pudlak, Impagliazzo]

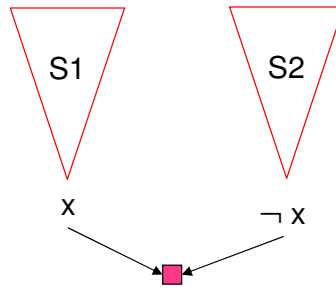
If F has TLR refutations of size S , then the Prover wins the PD game on F leaving $O(\log S)$ dollars to the Delayer

Cor

If in any PD game over F the Delayer wins at least p dollars, then the shortest TLR refutations of F are of size $2^{\Omega(p)}$.

Proof of the Theorem

Assume to have a size S TLR refutation P for F , $|P| = S$



Notice that both $S1$ and $S2$ cant be $\leq S/2$. Say $S1 \leq S/2$.

For a generic round k , denote by

α_k = the assignment to variables of F built so far

p_k = the total number of dollars scored by the Delayer so far

Prover Rule

The Prover keeps the following invariant IP:

$$|P[\alpha_k]| \leq \frac{|P|}{2^{p_k}} = \frac{S}{2^{p_k}}$$

Assume the Prover is able to keep the invariant along the game.

Let f the final round. We want to calculate p_f , knowing that at the end $|P[\alpha_f]| = 1$

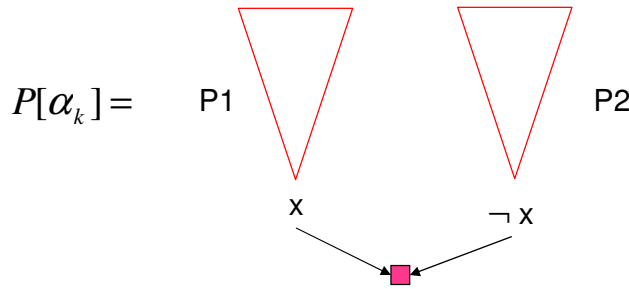
$$1 \leq \frac{S}{2^{p_f}} \Rightarrow p_f = O(\log S)$$

Keeping the Invariant - I

Base. At the beginning of the game:

$\alpha = \emptyset, p = 0$; and the IP follows

Induction.



$$\text{Wlog } |P1| \leq \frac{|P[\alpha_k]|}{2}$$

Prover chooses to present x

Keeping the Invariant- II

I Case. Delayer gives a value $i \in \{0, 1\}$ to x .

Then $p_{k+1} = p_k$

$$|P[\alpha_{k+1}]| = |P[\alpha_k \cup x = i]| \stackrel{\text{Contraction}}{\leq} |P[\alpha_k]| \stackrel{\text{Induction}}{\leq} \frac{|P|}{2^{p_k}} \stackrel{p_k = p_{k+1}}{=} \frac{|P|}{2^{p_{k+1}}}$$

II Case. Delayer gets 1 \$ but leaves Prover to choose. Prover gives to x the value to proceed into P1 (the smaller).

$$p_{k+1} = 1 + p_k, \text{ but } |P1| \leq \frac{|P[\alpha_k]|}{2}$$

$$|P[\alpha_{k+1}]| = |P[\alpha_k \cup x = 0]| \leq \frac{|P[\alpha_k]|}{2} \stackrel{\text{Induction}}{\leq} \frac{|P|}{2 * 2^{p_k}} \stackrel{1 + p_k = p_{k+1}}{=} \frac{|P|}{2^{p_{k+1}}}$$

Prover Delayer Game: Conclusions

Why PD Game ?

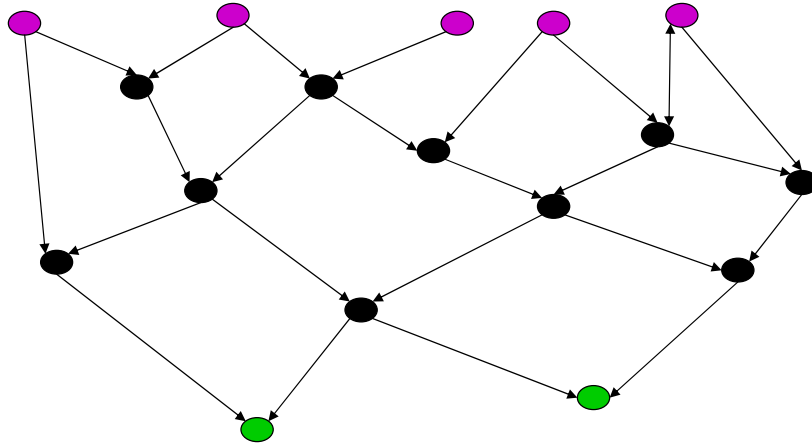
To prove an exponential lower bounds for TLR refutations of some UNSAT k -CNF F over n variables, is sufficient to find a strategy for the Delayer in the PD game over F that allow her to win $\Omega(n)$ dollars against any strategy of the Prover.

Not an easy task - after all

Since this means that to prove unsatisfiability of F the Prover needs to ask simultaneously almost all variables.

Pebbling Games

Pebbling Games on DAG



- Source Nodes
- Internal Nodes
- Target nodes
- Pebbles

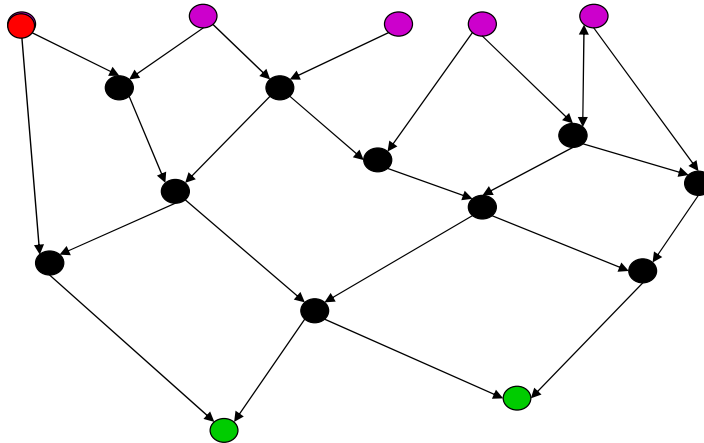
Pebbling Games: Rules

Game:

Place pebbles ● on dag's nodes according to the rules

Rule 1

Sources nodes can be pebbled freely



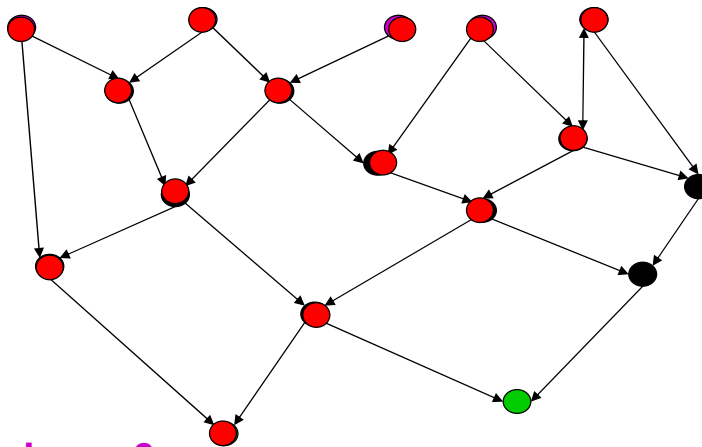
Pebbling Games: Aim & Complexity

Aim

Put a Pebble on some target node

Complexity Measure: Pebbling number

Maximal number of pebbles placed **simultaneously** on the graph.



Pebbling Number = 6

Definitions

Pebbling number of a strategy to pebble G

Let S be strategy to pebble a dag G.

$$P_n(G,S) = \max \# \text{ of simultaneous pebbles on } G \text{ following } S$$

Pebbling number of G

$$P_n(G) = \min \{ p_n(G,S) \mid S \text{ strategy to pebble } G \}$$

Hardness Results: prove that a graph G requires high $p_n(G)$

Original Motivations: Prove space lower bounds for Turing Machine: pebbling game models space in TM

An Important Old Result

[Celoni, Paul, Tarjan 77]

Found (constructively) a directed acyclic graph G over n nodes with in-degree ≤ 2 such that

$$pn(G) \geq \Omega(n / \log n)$$

Pebbling Formulas

Modelling PG as UNSAT Formulas

Let G be a dag with indegree ≤ 2 . We encode the principle that the pebbling game must terminate successfully pebbling a target node. Put in an UNSAT formula we say that

1. Sources nodes can be always pebbled
2. Internal nodes can be pebbled whenever its parents are
3. But no target node is ever pebbled

Modelling PG as UNSAT Formulas

$G=(V,E)$

$v \in V$, x_v iff "node v has been pebbled"

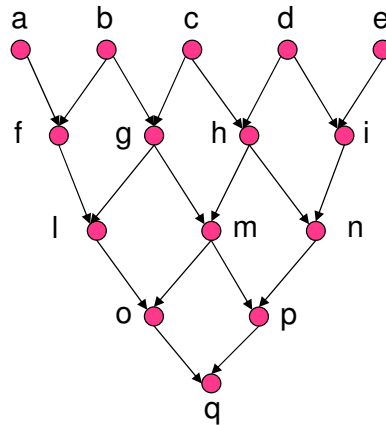
$\text{Peb}^0(G)$ x_v for any source node $v \in V$
 $x_v \wedge x_w \rightarrow x_z$ for any $(v,z),(w,z) \in E$
 $\neg x_v$ for any target $v \in V$

[Exercise 8] Prove that $\text{Peb}^0(G)$ is unsatisfiable

Peb⁰(G) is not Sufficient

Thm. There are polynomial size in $n=|V|$ TLR refutations of Peb⁰(G).

Proof Idea. Visit the graph G passing once from any node in a depth first fashion, starting from the bottom. Apply to the following pyramidal graph and then generalize to any graph



Adding Complexity to Peb⁰(G)

Idea. Use pebbles of two colors ● ● and let us consider a node v pebbled if it is pebbled by one of two colours

Why.

- 1) Similar to the formulas used in the previous exponential separations [BEGJ98] (pyramidal)
- 2) From the previous proof for Peb⁰(G). Not possible to carry on that proofs on such a modification.

The New Principle

1. Sources nodes can be always pebbled by any of the two colours
2. Internal nodes can be pebbled of any colours whenever its parents are pebbled by any color
3. No target node is ever pebbled with any colour

Modelling 2-color PG as UNSAT Formula

$G=(V,E)$, dag with indegree ≤ 2 .

$v \in V$, there are two variables for each node

$x_{(v,R)}$ iff “node v has been pebbled RED”

$x_{(v,B)}$ iff “node v has been pebbled BLUE”

$$Peb(G) = \begin{cases} x_{(v,R)} \vee x_{(v,B)} & \forall v \in V, v \text{ source} \\ x_{(v,a)} \wedge x_{(w,b)} \rightarrow x_{(z,R)} \vee x_{(z,B)} & \forall (v,z), (w,z) \in E, \forall a,b \in \{R,B\} \\ \neg x_{(v,R)}, \neg x_{(v,B)} & \forall v \in V, v \text{ target} \end{cases}$$

[Exercise 8] Prove that for all dag G , $Peb(G)$ is unsatisfiable and give polynomial size DLR refutations.

Lower bounds for Pebbling Formulas in TLR

Main Idea

Main Thm

Let G be a dag with indegree ≤ 2 . Delayer can always win $pn(G)-3$ dollars in any PD game played on $peb(G)$.

Cor [by Thm PI96;CPT77]

Take as G the dag of [CPT]. By Main Thm and Thm of [PI] we Have that $Peb(G)$ requires exponential size TLR refutations.

Informal proof - I

Tools.

Let $G=(V,E)$ our dag. Let S and T the sets of sources and target nodes in C . Denote $pn(G)$ as $pn(G,S,T)$

Informal Strategy Kept by the Delayer

- Delayer keeps sets of actual sources and target nodes in G S_i and T_i initially set resp. as $S_0=S$ and $T_0=T$.
- Assume that Prover proposes the variable $x(v,\cdot)$ talking of node v . The Delayer will let $x(v,\cdot)$ unassigned only if adding v to the target T_i the pebbling number of G is decreasing

Informal proof - II

Prop1. [Invariant Property]

After round i , if the Delayer has scored p_i points, then

$$p_i \geq \text{pn}(G, S, T) - \text{pn}(G, S_i, T_i)$$

Read: the Delayer scores as many points as the pebbling number of G

Prop2

After the last round f , the pebbling number of G is

$$\text{pn}(G, S_f, T_f) \leq 3$$

Cor

$p_f \geq \text{pn}(G) - 3$. Then when G is the [CPT] graph, the theorem follows

Delayer's Strategy - I

Let $x_{(v, \cdot)}$ the variable queried by the Prover after round i .

Case 1. $v \in S_i$, the Delayer sets $x_{(v, \cdot)} = 1$

Read: Delayer must satisfy a actual source axiom, otherwise could immediately loose

Case 2. $v \in T_i$, the Delayer sets $x_{(v, \cdot)} = 0$

Read: Delayer must satisfy a actual target axiom otherwise could immediately loose

Delayer's Strategy - II

Case 3. $v \notin S_i \cup T_i$, and $pn(G, S_i, T_i) = pn(G, S_i, T_i \cup \{v\})$.

Then the Delayer sets $x_{(v,\cdot)} = 0$ and add v to T_i

Read: Since the pebbling number of G does not decrease adding v to T_i , then Delayer can safely set the variable to the value most advantageous for her

Case 4. $v \notin S_i \cup T_i$, and $pn(G, S_i, T_i) > pn(G, S_i, T_i \cup \{v\})$.

Then the Delayer leaves $x_{(v,\cdot)}$ unset and add v to S_i .

Read: Since the pebbling number of G decreases adding v to T_i , to keep the invariant the Delayer gain 1\$ but leaves the decision on the variable to the Prover. He can then safely add v to the sources, to be meant as a "pebbled" node

Keeping the Invariant - I

Reduction Lemma.

For any $v \in V$ and any set S and T .

$$pn(G, S, T) \leq \max\{pn(G, S, T \cup \{v\}), pn(G, S \cup \{v\}, T) + 1\}$$

Proof.

To pebble T from S , first pebble $T \cup \{v\}$ from S . If this ends with a node of T pebbled we have done (in this case $pn(G, S, T) \leq pn(G, S, T \cup \{v\})$). Otherwise v to be pebbled. Then keep v pebbled and pebble T from $S \cup \{v\}$ (in this case $pn(G, S, T) \leq pn(G, S \cup \{v\}, T) + 1$)

Keeping the Invariant - I

Prop 1.

After round i , if the Delayer has scored p_i points, then

$$p_i \geq pn(G, S, T) - pn(G, S_i, T_i)$$

Proof. At the beginning $p=0$, $S_i=S$, $T_i=T$.

At any round $pn(G, S_i, T_i)$ is changing only in case 4. We lose 1 from $pn(G, S_i, T_i)$ but Delayer scores one point and since she adds v to S_i the property follows by reduction lemma

Terminating the game

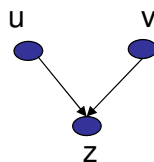
Prop2

After the last round f , the pebbling number of G is

$$pn(G, S_f, T_f) \leq 3$$

Proof.

By the strategy of the Delayer neither (actual) axiom clauses nor (actual) target clauses will ever be violated (Prove this ! [Exercise 9]). Then, since the Prover always wins, at the end will be violated an internal (actual) clause axiom.



Prover wins on this by using at most three pebbles