The complexity of Proofs in Parameterized Resolution

Nicola Galesi
Dept of Computer Science
Università di Roma La Sapienza

Joint works with

Olaf Beyersdorff and Massimo Lauria

Parametrized complexity

Fixed Parameter Tractability:
A problem is fixed parameter tractable with parameter k if it can be solved in time $f(k)n^{O(1)}$, for some computable function f.

Idea: Classical intractable problems may have efficient solutions for small value of the parameter even if the total size of the input is large

W-Hierarchy Problems apparently no fpt classified according to a hierarchy

$$FPT \subseteq W[1] \subseteq W[2] \ldots..$$

Weighted CNF SATISFIABILITY complete for W[2]

- Input: A CNF formula F
- Param: k
- Question: is there a sat assignment for F with weight k ?

Proof systems

Proof Systems for a language $L \subseteq \{0,1\}^*$ [Cook-Reckhow]:

Polynomial time onto mapping $F: \{0,1\}^* \to L$

Example:

L =TAUT, F(x)=A interpreted as x is a proof of A.

F is a poly time algorithm verifying that x is a proof of A

Parameterized proof systems
[Dantchev,Martin,Szeider, FOCS07]

[DMS] Parametrized Proof system for a language $L \subseteq \{0,1\}^* \times N$

Is an onto mapping $F : \{0,1\}^* \times N \rightarrow L$, where F can be computed by an FPT algorithm

[BGL] Simpler and equivalent definition (wrt proof lenght)

Is an onto mapping $F: \{0,1\}^* \rightarrow L$, where F is polynomial time computable

Parameterized proof systems

Example [DMS]

Weighted CNF SAT: Set of boolean formulas satisfiable by assignment of weight (# of 1s) at most k (W[2]-complete [DMS])

Parameterized Contradiction = (F,k), s.t. F has no satisfying assignment of weight at most k .

PCon set of all parameterized contradiction (co-W[2]-complete [DMS])

L=Pcon

Example: G=(V,E) undirected graph with no vertex cover of size ≤ k

$$\left(\bigwedge_{(u,v)\in E} (p_u \vee p_v),\ \ k \right)$$

FPT-bounded proof systems

Polynomially bounded proof system F for TAUT:

For every A in TAUT there is (a proof) x in {0,1}* (hence F(x)=A) s.t $|x| \leq p(|A|)$, for some polynomial p.

Fpt-bounded proof systems P for L

If there is a function f s.t. for every (A,k) in L there is an x in{0,1}* s.t. P(x)=(A,k) (x is a proof of (A,k)) and $|x| \leq f(k)|A|^{O(1)}$

Complexity Consequences

Thm [CR]
There is a poly-bounded proof system for TAUT iff TAUT $\in$ NP

Thm [DMS]
Let C be a parameterized complexity class and let L be a co-C complete problem. If there is no fpt bounded proof system for L, then FPT $\neq$ C

Def [Flum,Grohe] para-NP
Class of all parameterized language s that can be decided by a nondeterministic Turing Machine in time $f(k)n^{O(1)}$, for some computable function f

Thm [BGL]
L a parameterized language. There exits an fpt-bounded proof system for L iff L is in para-NP.

Parameterized proof complexity

Proof Complexity [CR]:

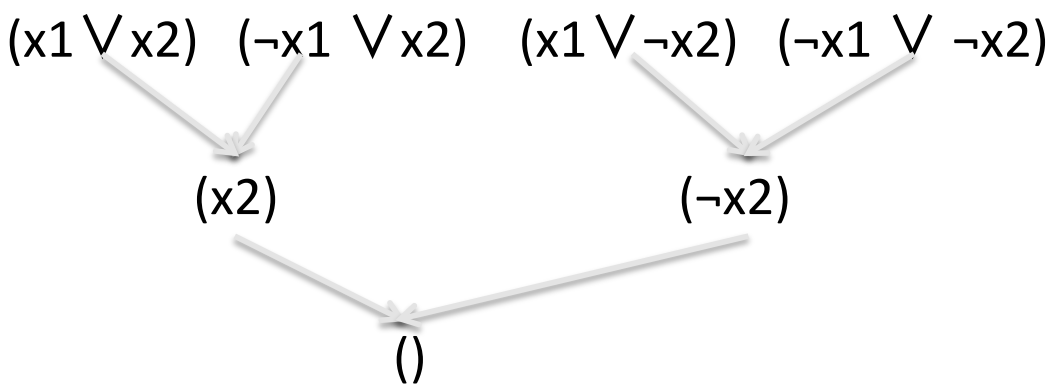
Idea: Gain evidence that NP ≠ co-NP by proving that proof systems are not polynomially bounded. I.e. prove that in any proof systems there exists a family F of tautologies requiring super-polynomially long proofs (in the size of F)

Parametrized Proof Complexity [DMS]

Prove lenght lower bounds to gain evidence that certain parametrized problems are not fixed parameter tractable

Tree-like Resolution

Refutational system for UNSAT CNF formulas



Proof size : number of clauses used in the proof, i.e. nodes in the tree.

F(x1,…,xn) hard of TLR: minimal proof size for F is superpolynomial (or, better, exponential) in |F(x1,…,xn)|

⇒ TLR is not poly bounded

Parameterized Treelike Resolution [DMS]

Refutational system for Pcon (F,k), F CNF

Axioms:

Clauses defining the (UNSAT) CNF formula F(x1,…,xn)

+

Clauses excluding assignments with more than k 1's

(**But in the size we count only those different actually used** !!!)

Example

$F(x1,\ldots,xn) \wedge (\neg xi_1 \vee \neg xi_2 \vee \ldots \vee \neg xi_{k+1})$

Proof size ranges to $O(n^k)$

PARA Axioms may help

Total Ordering Principle:

$$TOP_n =_{def} \begin{cases} \bigwedge_{\substack{i,j\in[n]\\ i\neq j}} (\neg x_{i,j} \vee \neg x_{j,i}) & \text{Antisymmetry} \\ \bigwedge_{\substack{i,j\in[n]\\ i\neq j}} (x_{i,j} \vee x_{j,i}) & \text{Totality} \\ \bigwedge_{\substack{i,j,k\in[n]\\ i\neq j\neq k\neq i}} (\neg x_{i,j} \vee \neg x_{j,k} \vee x_{i,k}) & \text{Transitivity} \\ \bigwedge_{i\in[n]} \bigvee_{j\in[n]} x_{j,i} & \text{Predecessor} \end{cases}$$

[BGM;DMS] (TOP, k) has $O(2^{2k+2})$ size proof in PTLR.

Kernel

Def [BGL] Kernel

A PCon CNF formula F has a kernel if there is a function f and a $F' \subseteq F$, s.t.

- F' is depends only on f(k) variables
- (F',k) is Pcon

Obs.

If F has a kernel, then there is a PTLR proof of size $O(2^{f(k)})$ for (F,k)

Cor (TOP,k) has $O(2^{2k+2})$ size proof in PTLR.

Proof. Take $A,B \subseteq [n]$, $|A|=|B|=k+1$, s.t. $A \cap B = \emptyset$.

Totality axiom $x_{i,j} \vee x_{j,i}$ for $i \in A$, $j \in B$ form a kernel over 2k+2 Vars.

Kernel

Obs

PEB(G) (2-colors Pebbling Formulas e.g. over Pyramidal graph) has a kernel over 2k variables.

Cor (PEB(G),k) has efficient PTLR refutations

Thm[BSIW03]

PEB(G) requires TLR refutations of length $2^{\Omega(\sqrt{n})}$

[DMS] Result

Let $\{F_\psi\}_n$ be a family of propositional unsat CNF formulas obtained by a translation of a FO order formula ψ which has no finite but some infinite models, then exactly one of the two cases holds:

1. $(\{F_\psi\}_n, k)$ has PTLR refutations of size $f(k)n^{O(1)}$ for some function f
2. Any PTLR refutation of $(\{F_\psi\}_n, r)$ requires size n^{k^ε} for some $\varepsilon \leq 1$.

Moreover 2 holds iff the the induced subgraph of the infinite model of ψ has no finite dominating set .

Lower bound technique tailored for formulas with property 2

Our Results

- Combinatorial characterization of proof size in PTLR based on a two-player game

⇒ General technique to prove size lower bounds in PTLR

- Applied to prove hardness of PHP and of LOP in PTLR

- bounded width UNSAT CNF have short PTLR refutations

- PHP requires size $n^{\Omega(k)}$ in Para Resolution

Prover Delayer Game for TLR

Definition [Pudlák,Impagliazzo 00].

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

Prover: queries a variables x of F

Delayer: 1. answers with a value for x or

2. leaves it unset and scores 1 point

but Prover decides the value of x

Prover wins as soon as he falsifies a clause of F

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: x1 ?
Delayer: unset
Gain: 1$
Prover x1=T

$$F[x1 = T] =_{def} (x_2 \vee x_3) \wedge (\neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_3)$$

II round

Prover: x2 ?
Delayer: unset
Gain: 2$
Prover x2=T

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

III round

Prover: x3 ?
Delayer: unset
Gain: 3$
Prover x3=F → 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 [PI00]

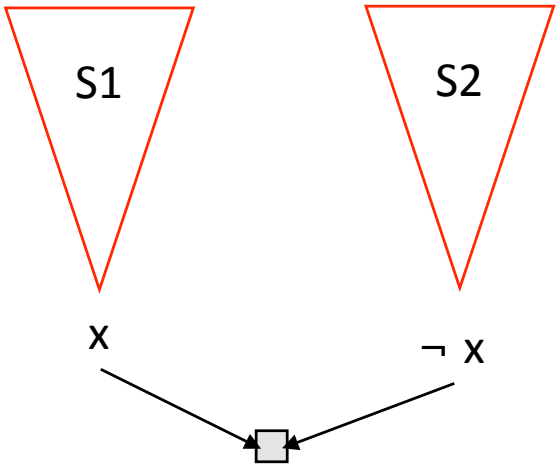
If F has TLR refutations of size S, then the Prover wins the PD game on F leaving at most O(log S) dollars to the Delayer

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



Say S1<=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's Invariant

The Prover keeps the following invariant IP:

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

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 $|P[\alpha_f]|=1$

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

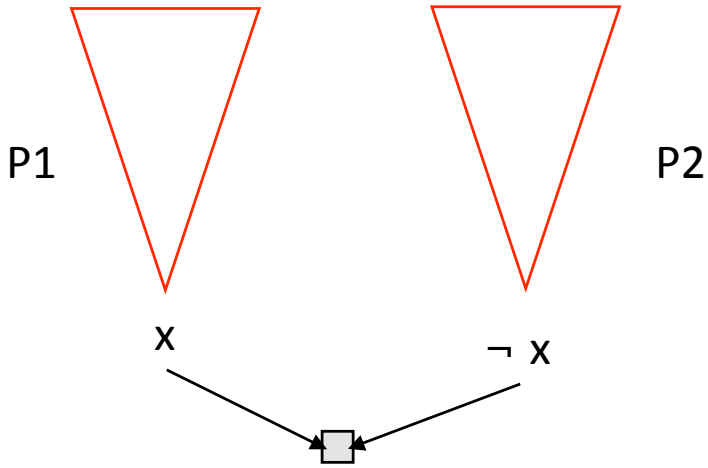
Keeping the Invariant

Base. At the beginning of the game:

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

Induction.

$P[\alpha_k] =$



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

Prover chooses to present x

Keeping the Invariant

I Case. Delayer gives a value i ={0,1} to x.

Then $p_{k+1}=p_k$

$$\left|P[\alpha_{k+1}]\right| = \left|P[\alpha_k \cup x = i]\right| \overset{Contruction}{\leq} \left|P[\alpha_k]\right| \overset{Induction}{\leq} \frac{|P|}{2^{p_k}} \overset{p_k = p_{k+1}}{=} \frac{|P|}{2^{p_{k+1}}}$$

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

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

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

Asymmetric Prover-Delayer Game

Observation on PI-game:

- Delayer gains # of points (1 or 0) according to (what we can argue about) the fraction of actual proof size Prover can avoid when he sets a variable
  - ½ if Prover chooses the "small" subtree .
  - 1 if Prover should choose (but never he do it !) the "large" subtree

In PTLR

Prover avoids fraction of proofs in both cases, so he should decide between setting x=0 or x=1 considering the fraction of proof he can avoid. The Delayer, as in PI game, should score accordingly

Asymmetric Prover-Delayer Game

Fraction of proof avoided by Prover

For α partial assignment and x a variable not in α, let $c_0(x,\alpha)$ and $c_1(x,\alpha)$ be functions s.t.

$$\frac{1}{c_0(x,\alpha)} + \frac{1}{c_1(x,\alpha)} = 1 \qquad (*)$$

IDEA

$c_0(x,\alpha)$ and $c_1(x,\alpha)$ related to the amount of information available to the Prover for his (0,1) choice on x.

Modify PD-game into (c_0,c_1)-PD game as follows:

Assume α is the assignment built so far in the game and Prover queries x. Then:

- Delayer gets $\log(c_\varepsilon(x,\alpha))$ points if he leaves x to the Prover and Prover sets x to ε.

Main theorem

Thm [BGM]

Let (F,k) $\in$ PCon and let c_0, c_1 functions verifying (*) for all partial assignments to var(F). If (F,k) has PTLR proof of size S, then Prover wins any (c_0,c_1)-PD game on F leaving at most O(log S) Points to the Delayer.

Proof

Similar to Pudlák-Impagliazzo using the strategy:

Prover sets $x=\varepsilon$ if $|P[\alpha \cup x=\varepsilon]| \leq |P[\alpha]|/c_\varepsilon(x,\alpha)$

Works using $\dfrac{1}{c_0(x,\alpha)} + \dfrac{1}{c_1(x,\alpha)} = 1$

Pigeon Hole Principle

$$\neg PHP_n^{n+1} =_{def} \begin{cases} \bigwedge_{i \in [n+1]} (p_{i,1} \vee \cdots \vee p_{i,n}) \\ \bigwedge_{i \neq i' \in [n+1]} \bigwedge_{j \in [n]} (\neg p_{i,j} \vee \neg p_{i',j}) \end{cases}$$

Assignment to PHP vars

	1	**2**		**i**			**n+1**
1	0						
2	0		0				
		0					
j	0			1			
n							

C_0 and C_1

Let α be a partial assignment and $x_{i,j}$ a variable not in α

$$z_i(\alpha) = \text{ holes not available for pigeon } i$$

$$c_1(x_{i,j},\alpha) = n - z_i(\alpha)$$

$$c_0(x_{i,j},\alpha) = \frac{n - z_i(\alpha)}{n - z_i(\alpha) - 1}$$

	1	2		i			n+1
1	0			0			
2	0		0	0			
		0		0			
j	0			?			
n							

$z_i(\alpha)$

$n-z_i(\alpha)$

IDEA: Delayer gets points according to available positions for pigeon i

Delayer's Strategy

On $x_{i,j}$ leave decision to Prover if there are at least k free positions for i and no immediate contradiction arise

$x_{i,j} = 0$ if pigeon i already occupied OR hole j already occupied

$x_{i,j} = 1$ if pigeon i NOT occupied and $z_i(\alpha) \geq n-k$

OW: leaves to the Prover

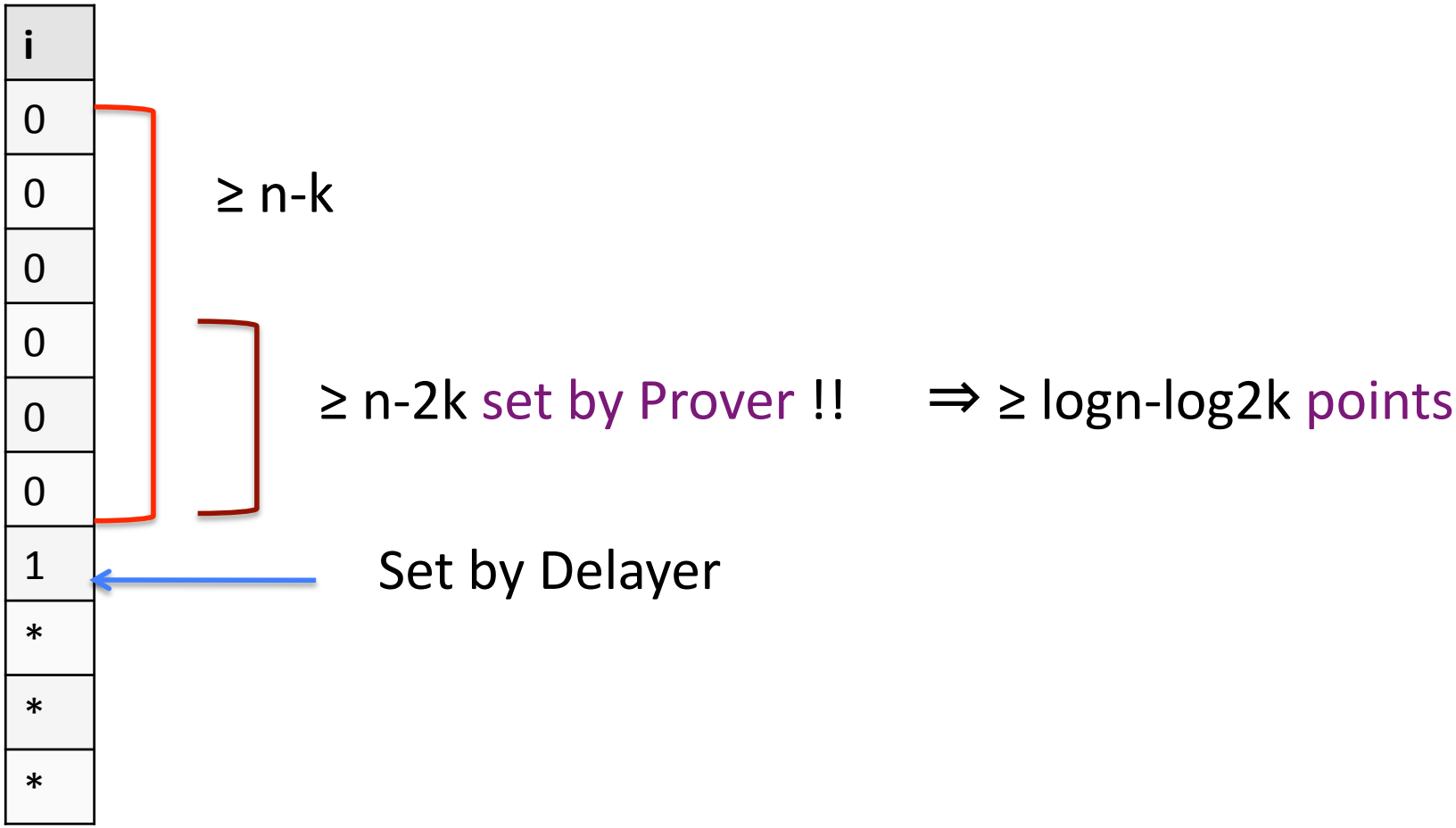
Claim

At the end of the game is falsified a parameterized axiom

Cor. There are exactly k+1 1's in the final assignment

PROOF

OVERALL IDEA: For each pigeon i set to 1, Delayer will earn "many" points since Prover has set "many" 0's for that pigeon



PROOF

i
0
0
0
0
0
0
1
*
*
*

$z_i(\alpha)$

$\geq z_i(\alpha) - k - 1$

set by Prover !!

$\Rightarrow \quad \geq \sum_{z=0}^{z_i(\alpha)-k-1} \frac{n-z}{n-z+k-1}$ points

+

Set by Prover $\Rightarrow$ $\log(n-z_i(\alpha))$ points

$\geq \log n - \log k$ points

ALL TOGETHER k+1 1's $\Rightarrow$ $\geq k(\log n - \log(2k))$ points for Delayer

Lower bounds for PTRL

Thm

There are no fpt-bounded PTLR refutations for PHP.

Thm

There are no fpt-bounded PLTR refutations for LOP.

FPT algorithm for VERTEX COVER as DPLL

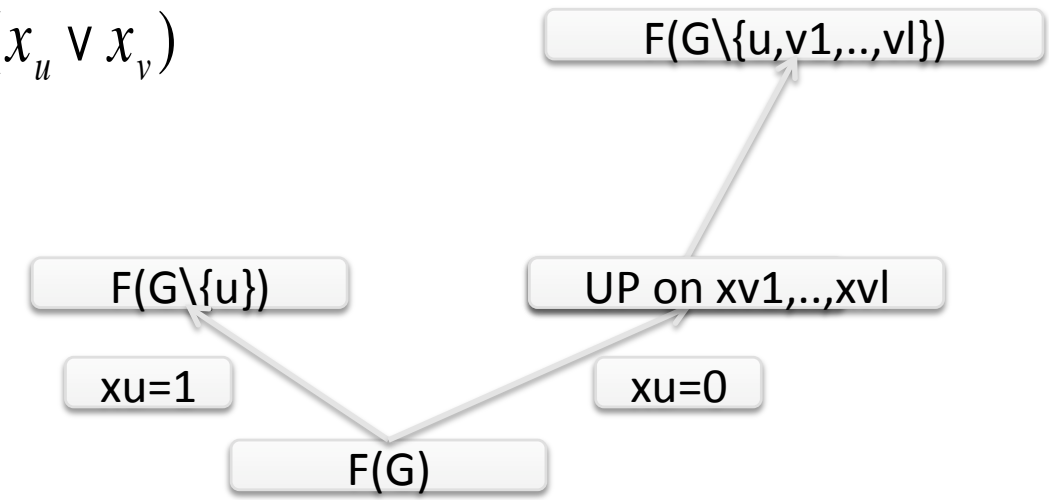
Thm [Folk]. (Vertex Cover,k) is FPT

G=(V,E), C is Vertex Cover: if (u,v) $\in$ E either u $\in$C or v$\in$C

Obs: if u $\notin$ C, then all its neighbours are in C

Algo: if neither G\{u} has a (VC,k-1) nor G\{u,v1,..vl} has (VC,k-l), then G has no (VC,k)

DPLL on $F(G) = \bigwedge_{(u,v)\in E} (x_u \vee x_v)$

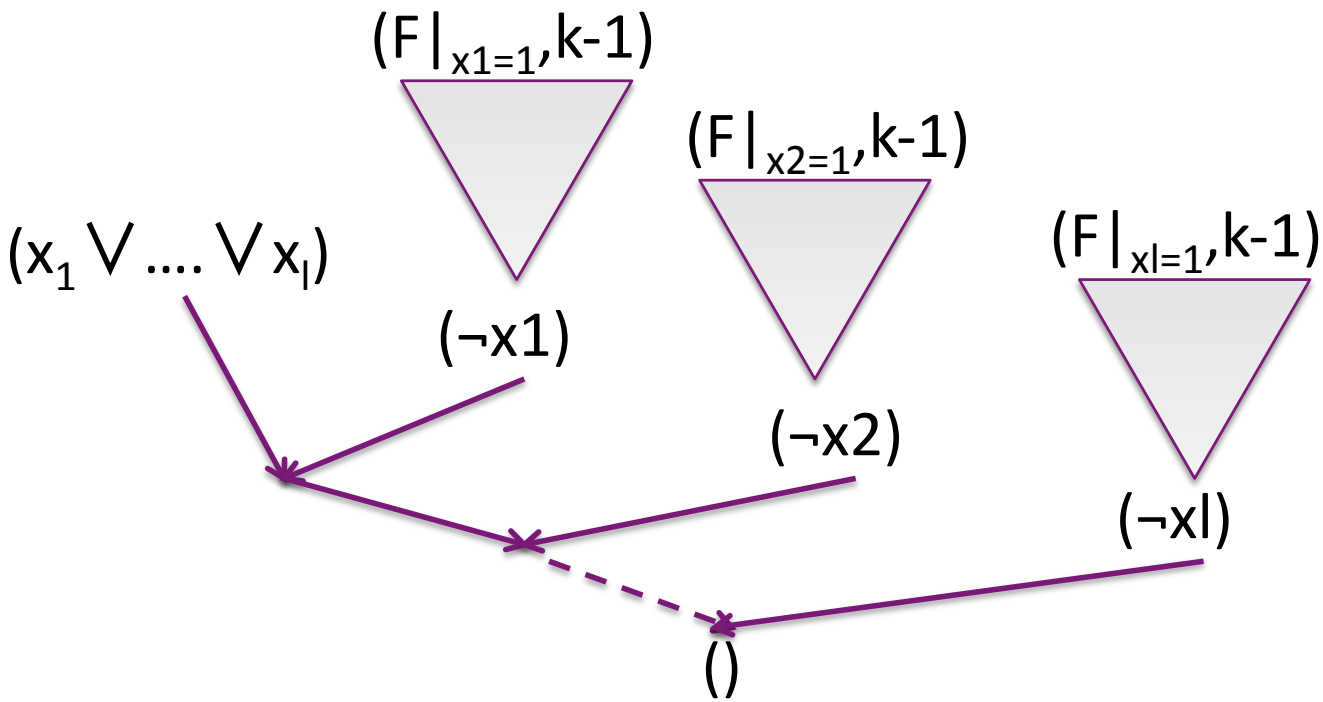


Bounded Width CNF Formulas in PTLR

Thm. Let (F,k) be Pcon with F w-CNF. Then (F,k) has a PTLR proof of size $O(w^{k+1})$.

Proof. By induction.

F has an all-positive clause $(x_1 \vee \ldots \vee x_l)$, $l \leq w$



Bounded width WEIGHTED SAT

Cor. If (F,k) $\notin$ Pcon with F d -CNF. Then previous algo returns an assignment of size $\leq$ k in time $O(|F|k\, d^{k+1})$.

Cor [DF99,CF08]

WEIGHTED d-SAT (=k) is W[1]-complete

WEIGHTED d-SAT($\leq$k) is FPT

WEIGHTED SAT (=k) is W[2]-complete

WEIGHTED SAT($\leq$k) is W[2]-complete

Easy formulas for PTLR

Question:

Are there UNSAT CNF with ftp- bounded proof in PTLR that are neither Bounded width nor have a bounded kernel ?

Answer: YES !!!

$$LOP_n^* =_{def} \begin{cases} x_{i,j} \quad \text{only for } i < j \\ \bigwedge_{i<j<k\in[n]} (x_{i,j} \vee x_{j,k} \vee \neg x_{i,k}) \quad \text{Transitivity} \\ \bigwedge_{i<j<k\in[n]} (\neg x_{i,j} \vee \neg x_{j,k} \vee x_{i,k}) \quad \text{Transitivity} \\ \bigwedge_{i\in[n]} (\bigvee_{j<i} \neg x_{j,i} \vee \bigvee_{j>i} x_{j,i}) \quad \text{Predecessor} \end{cases}$$

Daglike Parametrized Resolution

Parameterized Daglike Resolution: PHP ?

Pudlak's Games in Resolution

Intepret a refutation as a Prover –Adversary(Delayer) process

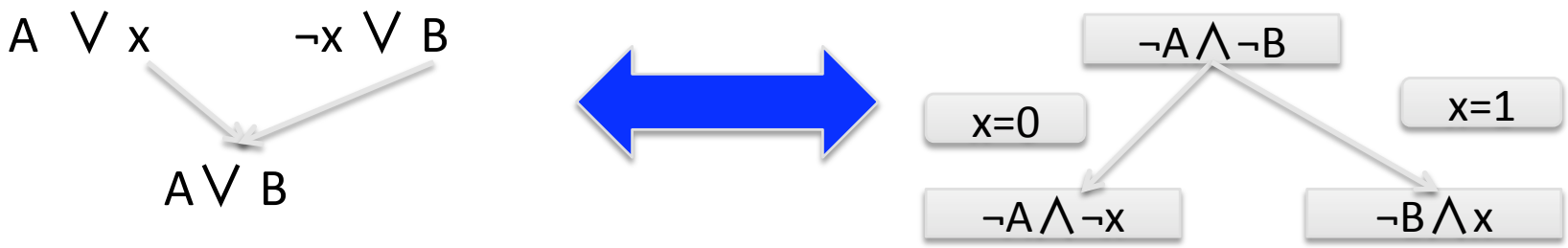
- Delayer claims to have an assignment with at most k ones which satisfies F
- Prover wants to expose the Delayer lie by
  * querying variables to Delayer
  * saving such values into a memory (RECORD)
  * at will deleting informations from the RECORD

STRATEGY for the PROVER:

- a dag with a single source node marked by the empty clause
- define at each round the variable to be queried
- what values are kept in the RECORD at each stage

COMPLEXITY of STRATEGY S: number of different RECORDS that can appear in all possible games where Prover plays according to S

RECORDS and CLAUSES



PROOFS to STRATEGIES:

Look at the record which falsifies the current clause in the proof. Queries for the unique variablere resolved to obtain the clause and add the Delayer's answer (negated) to the RECORD

STRATEGIES to PROOFS:

- RECORDS corresponds to CLAUSES
- DELETION corresponds to WEAKENING
- QUERIES corresponds to RESOLUTION

Delayer (Super)strategies: lower bounds

Delayer's Aim: Force the Player to use "many" ($n^{\Omega(k)}$) different records

Think of Delayer's strategy as a collection of single strategies (superstragies)
Which will force the Prover to generate many records in all games.

We use Randomized Delayer strategy.

FACT. Let P be a Prover stragegy with R records. D be a randomized Delayer strategy. Then if for each record $r \in R$

$$\Pr_D[\text{ Prover wins on record } r] \leq 1/S$$

Then $|R| \geq S$.

Proof. Probabilty that P wins against D is $\leq |R|/S$ (by union bound) and since Prover always wins, then $|R| \geq S$.

Idea of Delayer's strategy

DELAYER chooses some random fairly big partial assignment α.

DELAYER wants to force the PROVER to write it (or most part of it) down in the RECORD.

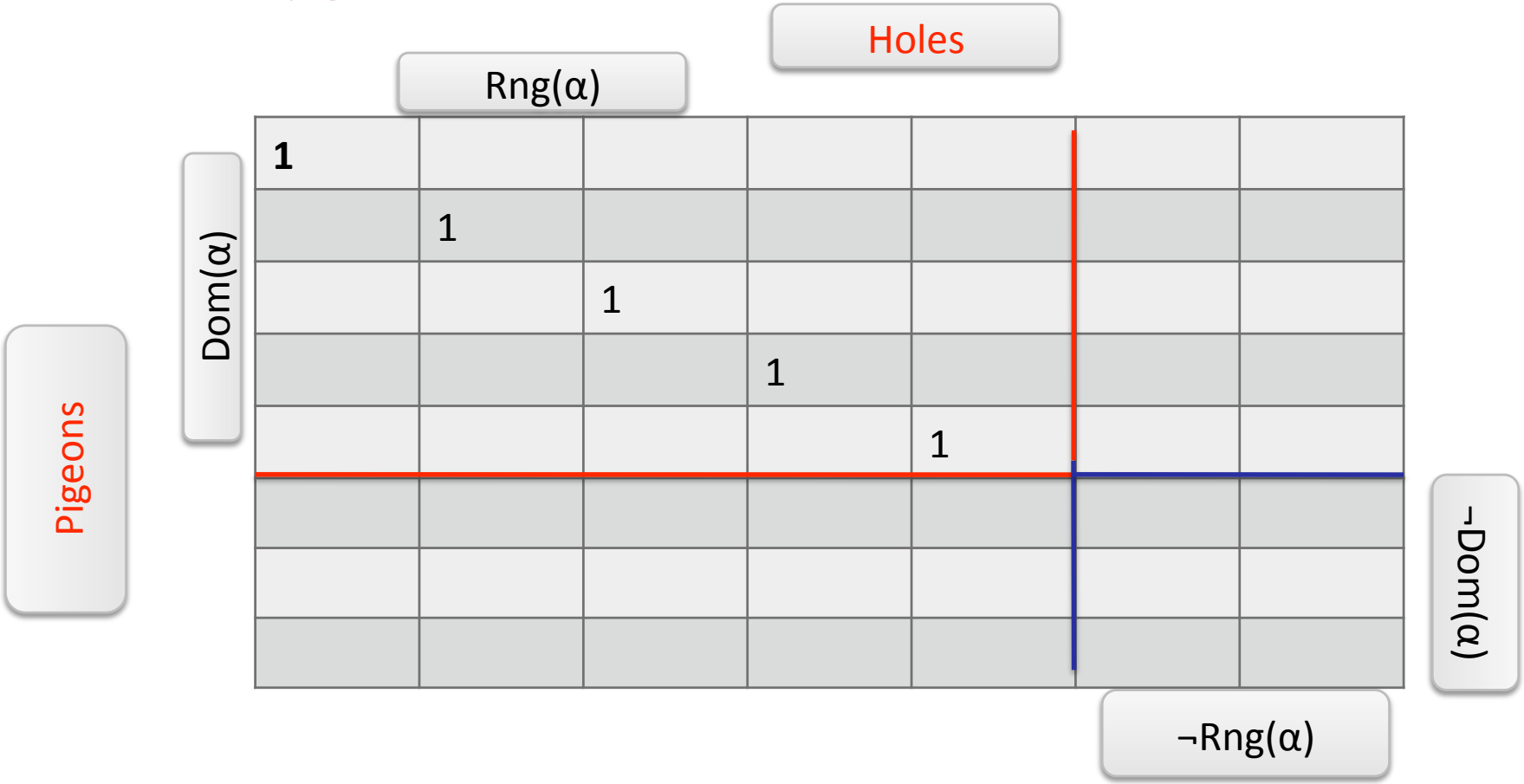
DELAYER STRATEGY must then be able to force the PROVER to RECORD a lot of INFO about α.

PROOF. It happens in "most" cases

Delayers informal strategy

First she chooses a random partial matching

α of n-2k pigeons to n-2k holes



Delayer plays answering consisently with α on (dom(α) and rng(α)) until she is able to extend the RECORD with a " sufficiently good" MATCHING on (¬dom(α) x ¬rng(α)).

Prover's Record (P)

P = PARTIAL ASSIGNMENT to PHP

Holes

Pigeons

1						
0		0			0	
0		1				0
	0					
					1	
						0

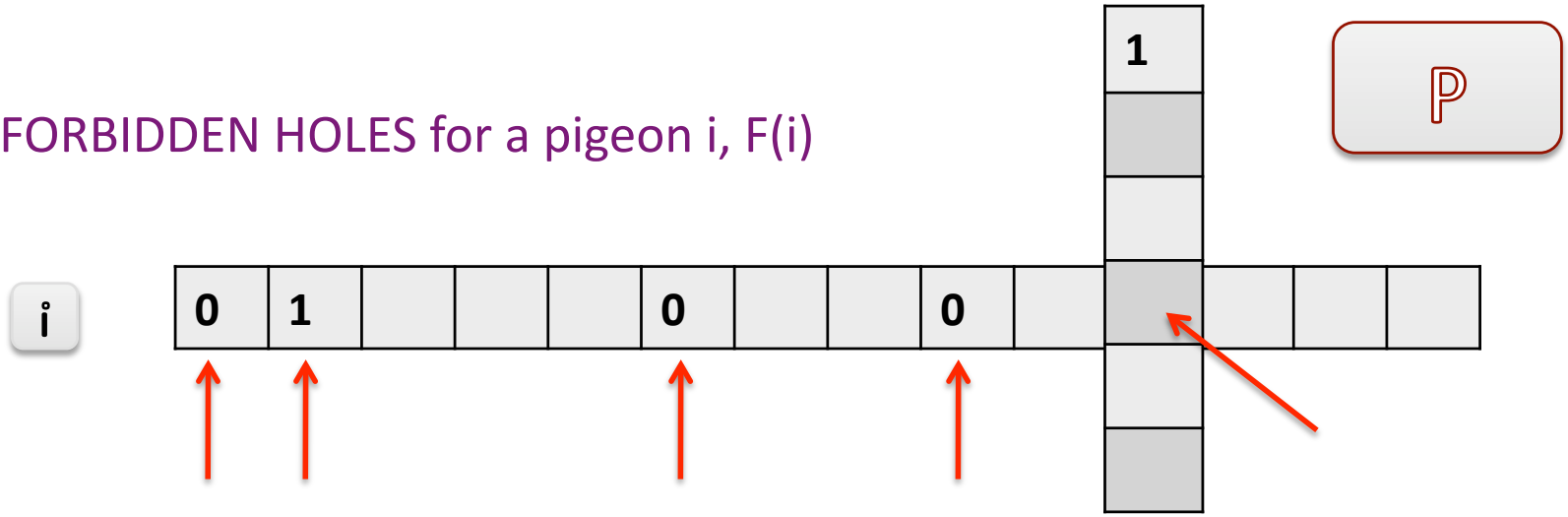
P'

PROVER WINS if

- P falsifies a PHP clause
- there are k ones in P (falsfies a PARA AXIOMS)
- DELAYERS GIVE's UP

Forbidden holes and Good Pigeons

FORBIDDEN HOLES for a pigeon i, F(i)



GOOD PIGEONS (FOR DELAYER)

Pigeons not assigned by α for which the Prover has already excluded more than k of the possible 2k holes (in P') left free by α.

$$G = \{i \in \overline{dom(\alpha)} : \ |F(i) \cap \overline{rng(\alpha)}| \ \geq k\}$$

Delayers Strategy: Private info

Delayer's doesn't want the Prover to easily reach a contradiction on the pigeons from G. Then he SECRETELY assigns (as long as she can) a hole to each pigeon in G.

DELAYERS keeps a PRIVATE MATRIX D, of size (2k+1)x2k (same as P')

- D (if possible) is re-computed at each round
- D contains the info sufficient to give "NEXT ANSWER" to the PROVER

Delayers Strategy: Computing D

D is a **COMPLETION** of P'

- D(i,j) =P'(i,j) for each (i,j) s.t. P'(i,j) ≠ *
- Exactly the pigeons in G are MATCHED
- D is a MATCHING

NOTICE: D might not exist

Delayers Strategy:

PROVER queries $x_{i,j}$.

DELAYER answers

- If $i \in dom(\alpha)$ then DELAYER answers 1 if $\alpha(i)=j$ and 0 if $\alpha(i)\neq j$
- If $i \notin dom(\alpha)$ and holes $j \in rng(\alpha)$, then DELAYER answers 0
- If $i \notin dom(\alpha)$ and holes $j \notin rng(\alpha)$, then DELAYER answers $D(i,j)$

DELAYER re-computes D from P'.

- If D does not exits, GIVES UP.

Final (Winning) positions of the Game

CLAIM

When PROVER wins, then

- either P contains k ones or
- G contains at least k pigeons

Proof Idea.

- Never falsified a PHP clause
- If $|G| \leq k$, then it is alway possible to compute D.

  Analysis of G_{old} and D_{old} at the previous step

Analysis of Winning Positions: Intuition

We show that for a random α the probabilty that a CLAUSE/RECORD gives raise to a FINAL POSITION in the game is $1/n^{\Omega(k)}$

IDEA

- If a RECORD contains a "LARGE" amount of INFO about α, then there is little chance for a random α to be compatible with this RECORD

  Case (1) P contains k ONES

  Case (2) P contains a "HUGE" amount of ZEROS

- **(case 3)** The RECORD contains < k ONES and "SMALL" amount of INFO . But then this INFO is CONCENTRATED on P' and this for random α and random positions of P' wrt P happens with smll prob.

  Prover is "FORCED" to LEARN at least ¬dom(α) x ¬rng(α), which happens with small probability

Case I and II

CASE I -- P has k ONES

$rng(\alpha)$

$dom(\alpha)$

1	**0**	**0**				
	1	0			0	
	0	**1**		0	0	0
	0				0	
			0	0		
					1	
						1

$\overline{dom(\alpha)}$

$\overline{rng(\alpha)}$

Case I and II

CASE I -- P has k ONES

CASE II – P CONTAINS a SET S with at least $n^{2/3} + 2k + 1$ PIGEONS with $kn^{1/3} \log n$ FORBIDDEN HOLES

Analysis not much different form Pudlak-Haken

Case III

- P DOES NOT contain k ONES and
- P contains at MOST $n^{2/3} + 2k + 1$ PIGEONS which have more than $kn^{1/3} \log n$ FORBIDDEN HOLES

Let S be the set of such PIGEONS. We look at $S \cap \overline{dom(\alpha)}$

- $|S \cap \overline{dom(\alpha)}| \geq k$,
  S contains FEW PIGEONS, then PROVER MUST have discolosed $\overline{dom(\alpha)} \times \overline{rng(\alpha)}$

- $|S \cap \overline{dom(\alpha)}| < k$,
  for P to be WINNING there must exits a PIGEON i s.t. $|F(i) \cap \overline{rng(\alpha)}| \geq k$

FUTURE DIRECTIONS

- Unconditional non automatizability results (AR) ?

- Asymmetric Prover Delyer game vs new DPLLs ?

LATEST NEWS

- Razborov simplified our proofs for Para RES with a very nice reduction to PHP in Resolution (we are writing a joint paper right now)

- Asymmetric Prover Delyer to get $2^{\Omega(n\log n)}$ lower bounds for PHP in treelike Resolution [With Olaf and Massimo]