The Tournament Divide and Conquer Technique

Daniel Rubery
Georgiy Platonov
Mohammad Rafayet Ali
Xiong Zhang

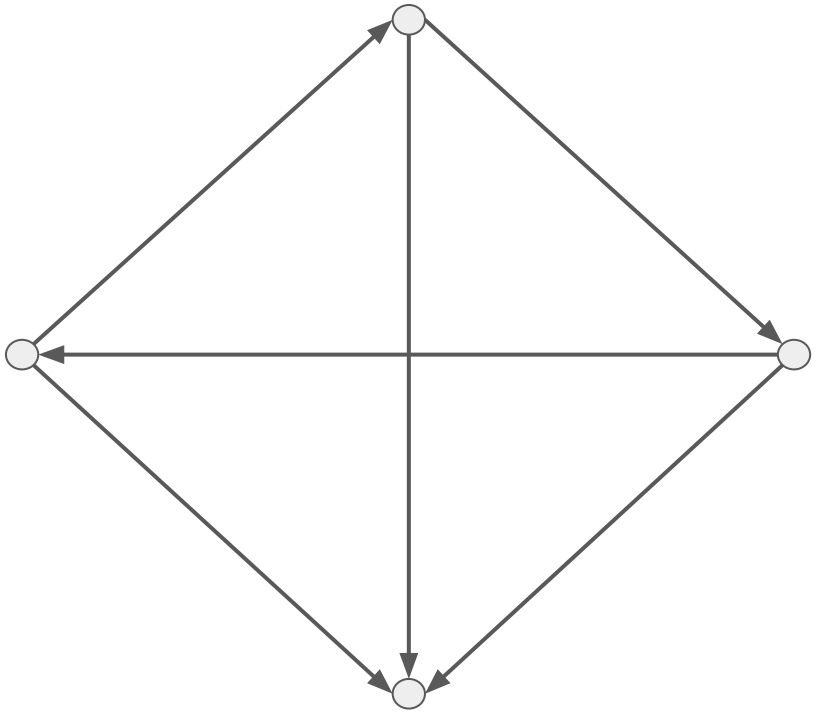
K-tournaments

Consider the set of k people playing a game with the following properties:

1) Game consists of multiple matches, with only two players in each match
2) Each player play against each other exactly once.
3) Every match has a winner

Such a setting can be represented by a directed graph.

Example: 4-tournament



Curious fact about tournaments

The interesting fact about tournaments is that we can select a very small subset of players such that each player not from that subset defeats someone from that set.

How small?

A logarithmically small!

Theorem 3.1:

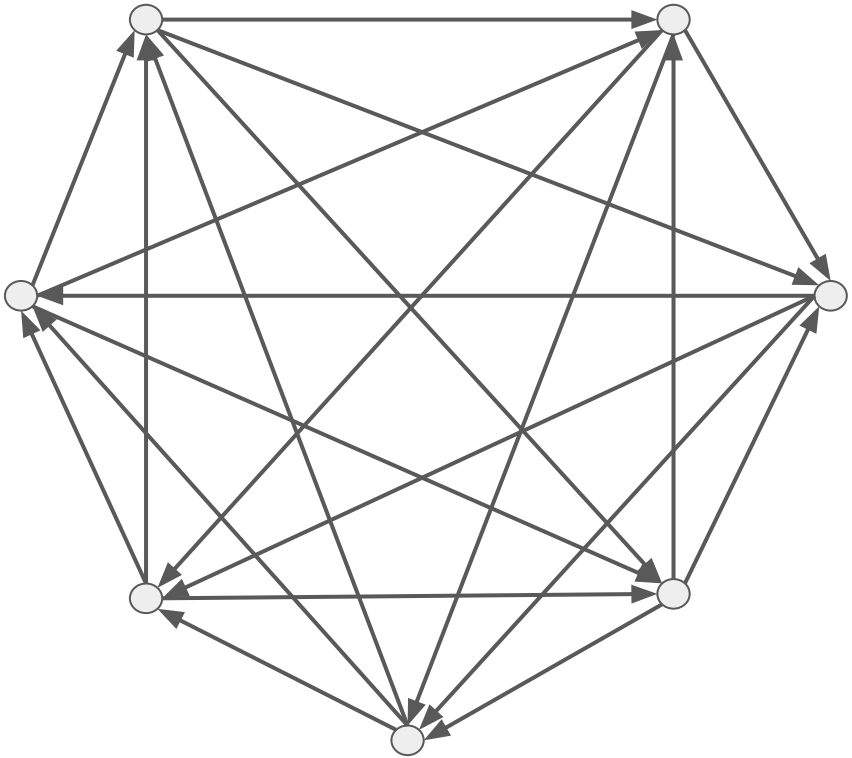
Let G = (V, E) be a k-tournament, where V = {1, …, k} is the set of nodes or players, and E is the set of edges or matches.

Then there exists a subset H of V, such that:

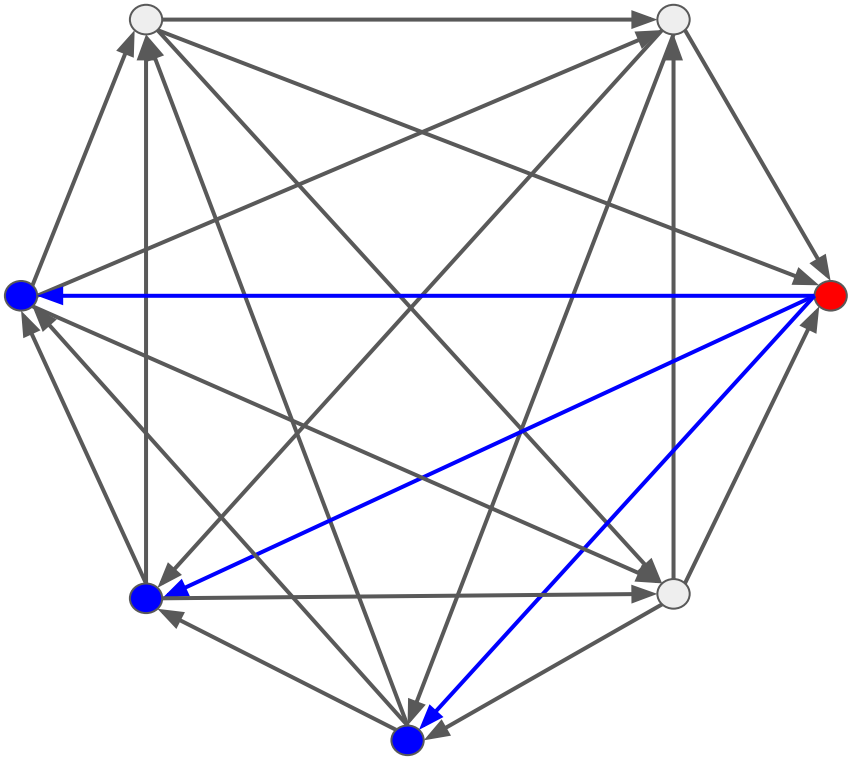
1) $|H| \leqslant \lfloor log(k+1) \rfloor$

2) $\forall v(v \in (V - H) \implies \exists g(g \in H \wedge (g, v) \in E))$

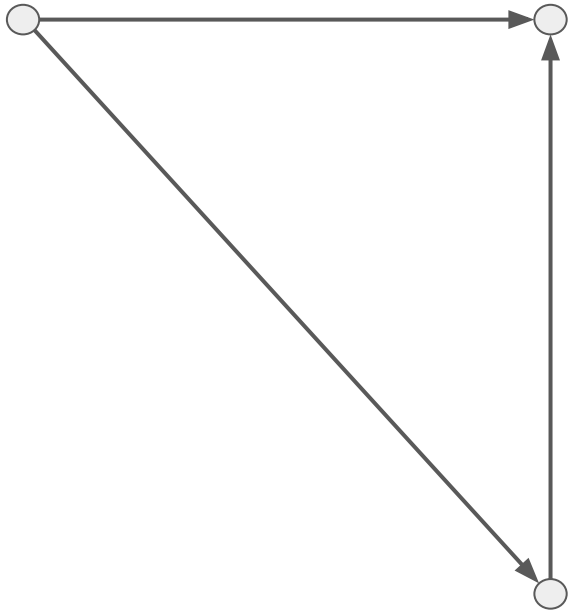
Example: 7-tournament



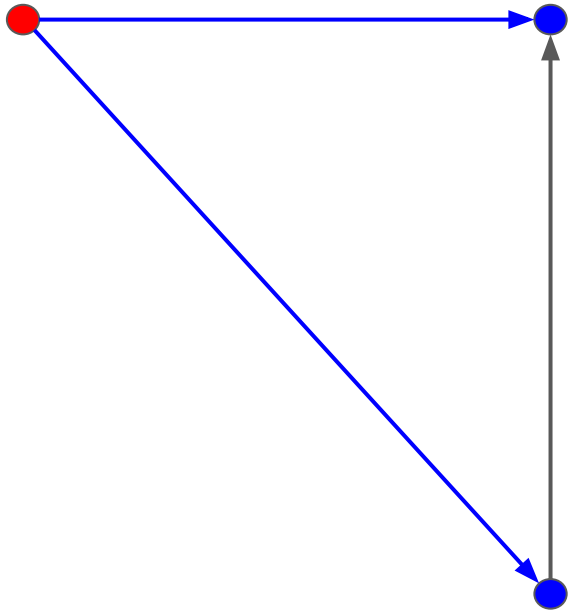
Example: 7-tournament



Example: 7-tournament



Example: 7-tournament



Proof of Theorem 3.1:

- In a k-tournament, each player plays exactly k - 1 games.
- There must be at least one player who lost to at least $\lceil (k - 1)/2 \rceil$ other players.
- Add this player to H.
- Remove the nodes corresponding to that player and to all the players who defeated him from the graph. The resulting graph has at most $\lceil k/2 \rceil - 1$ vertices.
- Apply the same procedure to the new graph.
- Since at each step we decrease the size of graph by factor of 2 (at least), we end up with at most $\lfloor \log (k + 1) \rfloor$ steps, so $|H| \leq \lfloor \log (k + 1) \rfloor$.

Definitions

- P-sel
  - A language L is P-selective if there is a polynomial-time function f such that:
  - f(x,y) is either x or y
  - If $x \in L$ or $y \in L$, $f(x,y) \in L$
  - Notice that f(x,y) can do anything if x and y are not in L, so in some sense, f(x,y) selects which string is “more likely” to be in L
- Example:
  - For a fixed real number r, {<a,b> | a/b < r}

Definitions

- P/poly
  - Can be thought of as having “small circuits” that decide each length
  - Easier to think of it as polynomial amount of advice
  - More generally, for a class of languages C and class of functions F, let C/F denote the class of languages L such that:
    - There exists a language $A \in C$, and h(n) such that $|h(n)| \in F$, and $L = \{x \mid \langle x,h(|x|)\rangle \in A\}$
  - So P/poly is equivalently:
    - There is a language $A \in P$ and h(n) of polynomial length such that $x \in L$ iff $\langle x,h(|x|)\rangle \in A$

Connection Between Tournaments and P-sel

- If L is P-selective, let f be a P-selector for L
- Then for a given length n, f gives a tournament on $L^{=n}$
  - We have an edge from a to b if f(a,b) = b
  - Note: This requires f(a,b) = f(b,a)
- Then Theorem 3.1 gives a set H with at most $\log(2^n+1) \leq n+1$ strings
- Every string in $L^{=n}$ is either in H, or beats a string in H
- Furthermore, any string in H or that beats a string in H is in $L^{=n}$

P-sel ⊆ P/poly

- Recall our definitions. To show a language L is in P/poly, we need a function g and a language A∈P such that:
  - x∈L iff <x,g(|x|)>∈A
- So let g(n) encode the, at most, n+1 strings in H
  - |g(n)| is polynomial in n
- Then A is accepted by the following deterministic Turing machine:
  - On input <x,y> do the following:
    - For each h in y, accept if x=h or f(x,h)=x
    - If we fail all the above, reject
- Then x∈L iff <x,g(|x|)>∈A, so L is in P/poly
- In fact, since g(n) has at most n+1 strings of length n, it has quadratic length
- So we can strengthen this result to P-sel ⊆ P/quadratic

3.2 Optimal Advice for the Semi-feasible Sets

Theorem 3.9

If G is a k-tournament, then there is a $v \in V_G$ such that $V_G = R_{2,G}(v)$.

Required Notations

Given a directed graph G, and a node $v \in V_G$, let

$$R_{0,G}(v) = \{v\}$$

And for each i > 0, let

$$R_{i,G}(v) = R_{i-1,G}(v) \cup \{z \in V_G \mid \left(\exists w \in R_{i-1,G}(v)\right)[(w,z) \in E_G]\}.$$

For any i, G, and $S \subseteq V_G$, define

$$R_{i,G}(S) = \{w \in V_G \mid (\exists v \in S)\left[w \in R_{i,G}(v)\right]\}.$$

If G is a k-tournament, then there is a $v \in V_G$ such that $V_G = R_{2,G}(v)$.

Proof: By Induction

1-tournament and 2-tournament holds trivially.

Assume: k'-tournament holds in G' graph.

Consider: k'+1 –tournament in G graph where a is the new vertex.

If G is a k-tournament, then there is a $v \in V_G$ such that $V_G = R_{2,G}(v)$.

There is a node b in G' such that $R_{2,G'}(b) = V_{G'}$

Case 1: If the edge between a and b points to a.

Case 2: If the edge between a and b points to b.

If G is a k-tournament, then there is a $v \in V_G$ such that $V_G = R_{2,G}(v)$.

- **Case 1:** If the edge between a and b points to a.

We are done. As $R_{2,G}(b) = V_G$

If G is a k-tournament, then there is a $v \in V_G$ such that $V_G = R_{2,G}(v)$.

- **Case 2:** If the edge between a and b points to b.
  - If $a \in R_{2,G}(b)$ then we are done.
  - If $a \notin R_{2,G}(b)$ then
    - For each node c $\in R_{1,G}(b)$ edge between a and c points to c.
    - So $R_{2,G}(a) = V_G$

Theorem 3.10 $\text{P-sel} \subseteq \text{NP/linear}$

What does it mean?

Theorem 3.10 $\text{P-sel} \subseteq \text{NP/linear}$

What does it mean?

If L is a P-sel set, then L is also a NP/linear set

L is a P-sel set $\Rightarrow$ there exists a selector function f for L (which selects the one that is "more likely" in L)

L is a NP/linear set $\Rightarrow$ there is an advice function g and set A such that if $x \in L$ then $\langle x, g(|x|)\rangle \in A$

How to prove this?

Hint Hint: We'll use Theorem 3.9 (if G is a k-tournament, we can find a "core" node which has a relatively short distance (<=2) to any other nodes)

Theorem 3.10 $\text{P-sel} \subseteq \text{NP/linear}$

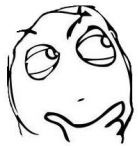
What we have: a selector function f for set L

What we want: a linear advice function and an “interpreter” set A

Consider this advice function g:

$$g(n) = \begin{cases} 1^{n+1} & if L^{=n} = \emptyset \\ 0\omega_n & otherwise \end{cases}$$

ω_n is a length n string in $L^{=n}$ such that, by Theorem 3.9, each node in the tournament induced on $L^{=n}$ by f can be reached from ω_n via paths of length at most two. **(Basically ω_n is the core of the tournament graph!)**



This is great, now we have a linear advice (=n+1) and it seems to be useful, but how to construct the set A (the interpreter) based on the advice function?

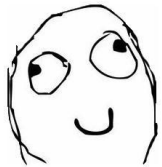
Theorem 3.10 $\text{P-sel} \subseteq \text{NP/linear}$

The interpreter is defined as:

$$A = \{\langle x, 0\omega\rangle | \text{there is a path of length at most two, in the tournament induced on } L^{=n} \text{ by } f, \text{ from } \omega \text{ to } x\}$$

So if x is in L, by construction of the advice function we know <x, g(|x|)> is in A

What if x is not in L?



If $x \notin L$, then $\langle x, g(|x|)\rangle \notin A$, as if $\langle x, g(|x|)\rangle \in A$, then we have a directed path from ω to x within 2 steps (0, 1, 2):

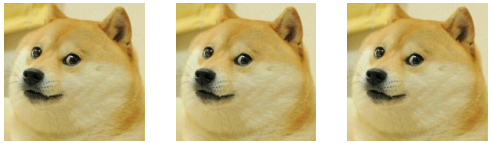
$$\omega \to \ldots \to x$$

Remember an edge from $a \to b$ means that $f(a, b) = b$, or "b beats a in the tournament". So b is "more likely" in L than a. If a is in already in L, then b must be in L.

In our this case a is ω, b is x. So based on our assumption ($\langle x, g(|x|)\rangle \in A$), x turns out to be in L!!! We get a contradiction, so our assumption is not correct.

So now we know if $x \notin L$, then $\langle x, g(|x|)\rangle \notin A$! And thus by our construction of g and A, L is a NP/linear set.

This is the end of lecture 1.

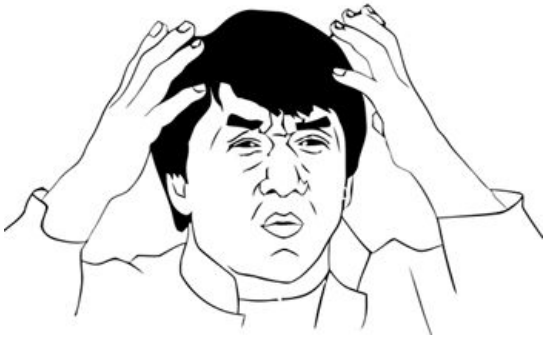


The Tournament Divide and Conquer Technique

Daniel Rubery
Georgiy Platonov
Mohammad Rafayet Ali
Xiong Zhang

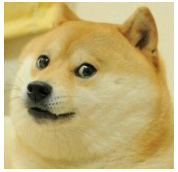
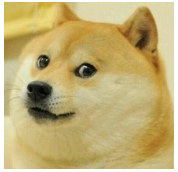
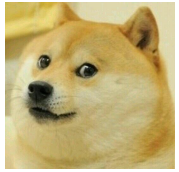
Attention

- We'll have an in-class quiz at the end of this lecture.
- In the quiz you need to show some understanding to the material covered today.
- During the quiz you **can't** use today's slides, but you **can** take a sheet of note during the lecture (if you didn't do it before) and use it in the quiz.



Be prepared!

Recap

At the end of last lecture we proved Theorem 3.10:

$$\text{P-sel} \subseteq \text{NP/linear}$$

L is a P-sel set $\Rightarrow$ there exists a selector function f for L (which selects the one that is "more likely" in L)

L is a NP/linear set $\Rightarrow$ there is an advice function g (linear!) and NP set A such that $x \in L \iff \langle x, g(|x|)\rangle \in A$

And remember we constructed an elegant advice function:

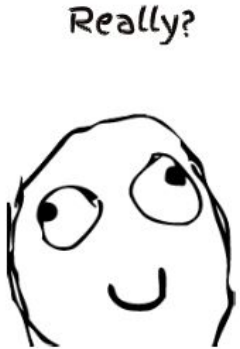
For each $n \geq 0$

$$g(n) = \begin{cases} 1^{n+1} & if L^{=n} = \emptyset \\ 0\omega_n & otherwise \end{cases}$$

ω_n is the core of the tournament graph!

The length of the advice string here is n + 1, but is it optimal? Or can we have even shorter advice strings, like in the length n ?

No, not a chance!



Theorem 3.13 $\text{P-sel} \not\subseteq \text{NP}/n.$

We'll show that n bits simply cannot hold enough information to disambiguate a certain family of semi-feasible sets.

How? 

A direct way is to find a set L that is

1. semi-feasible (i.e. P-sel set)
2. but is not NP/n

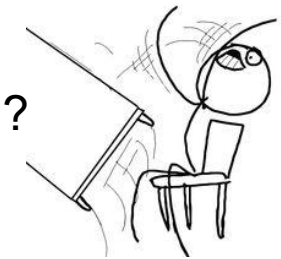
Construction Strategy

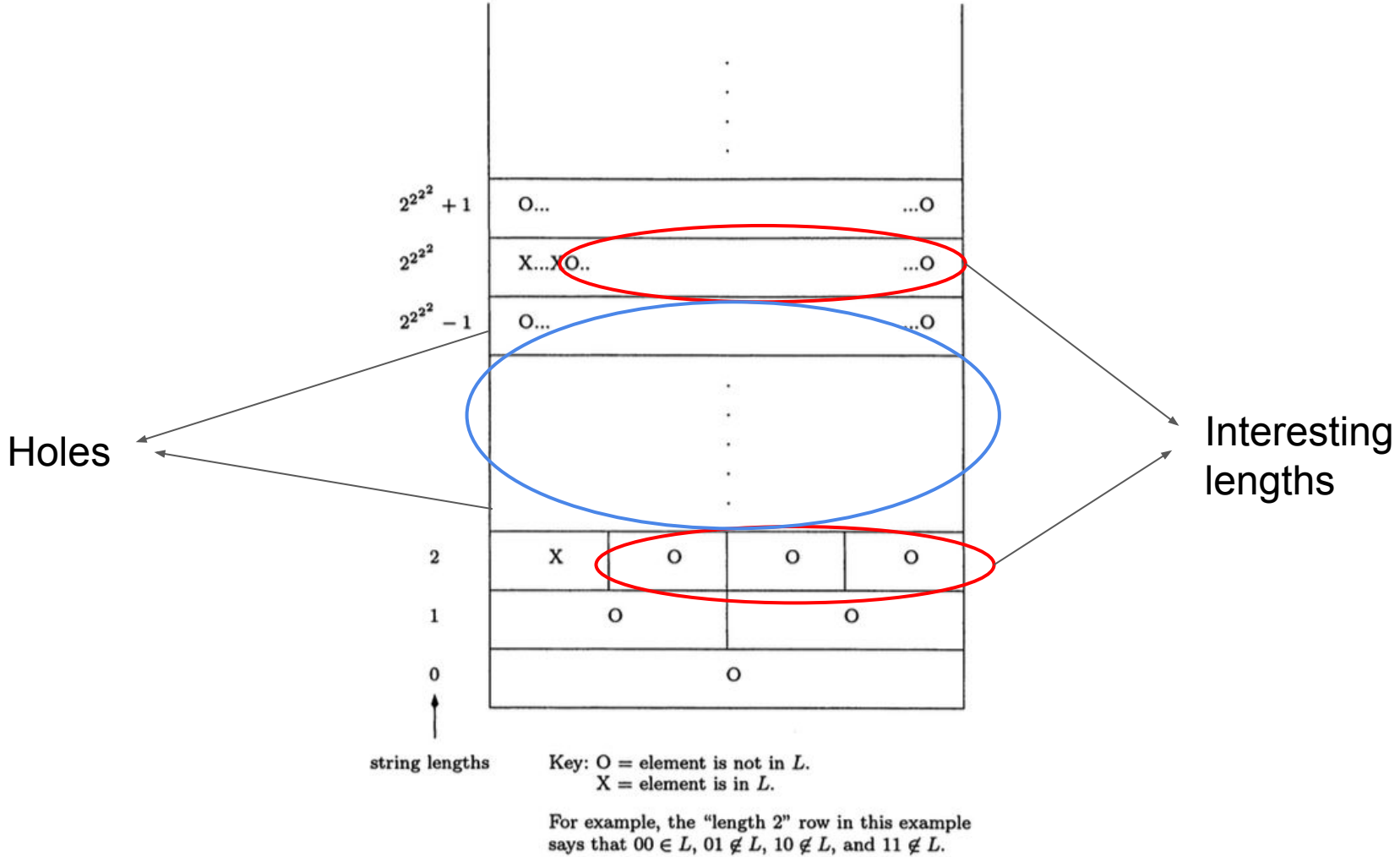
We want to construct a set L, consisting mostly of holes.

Holes?

That means at widely spaced lengths, our set will include exactly some (possibly empty) left cut of strings of that length, and at other lengths it will be empty.

What does it mean?




Holes
Interesting lengths
$2^{2^{2^2}}+1$
$2^{2^{2^2}}$
$2^{2^{2^2}}-1$
O...
...O
X...XO..
...O
O...
...O
2
X
O
O
O
1
O
O
0
O
string lengths
Key: O = element is not in L.
X = element is in L.
For example, the "length 2" row in this example says that $00 \in L$, $01 \notin L$, $10 \notin L$, and $11 \notin L$.

Construction Strategy

We want to construct a set L, consisting mostly of holes.

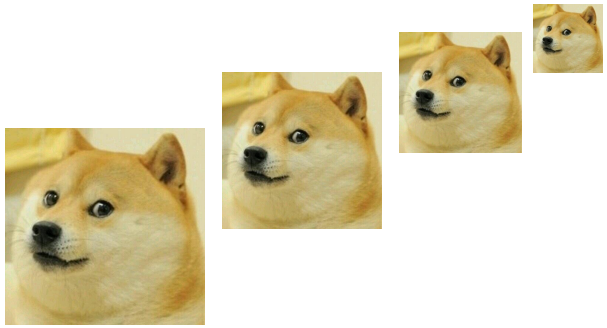
Yet we will ensure that the set is of limited complexity, so we can conduct a brute-force search for strings.

So P(olynomial) selector function exists!!

Weird Notation for Length of Strings

Let $l_0 = 2$, and for each $i \geq 1$, let $l_i = 2^{2^{2^{l_{i-1}}}}$
Let $Q = \{l_0, l_1, l_2, ...\}$

Example for l: $l_1 = 2^{2^{2^2}}, l_2 = 2^{2^{2^{2^{2^{2^2}}}}}, ...$



3 Conditions for the Construction

$L \subseteq \Sigma^{\ell_0} \bigcup \Sigma^{\ell_1} \bigcup \Sigma^{\ell_2} \bigcup \cdots$. That is, all strings in L have lengths from the set Q. (3.5)

For each x and y, if $|x| = |y|$ and $x \leq_{lex} y$ and $y \in L$, then $x \in L$. That is, at each length L is a (perhaps empty) left cut of the strings at that length. (3.6)

$L \in \mathrm{DTIME}[2^{2^n}]$. (3.7)

2 Claims

Claim 3.14 *Any set L satisfying equations 3.5, 3.6, and 3.7 is semi-feasible.*

Claim 3.15 *There is a set $L \notin \mathrm{NP}/n$ satisfying equations 3.5, 3.6, and 3.7.*

Translation?

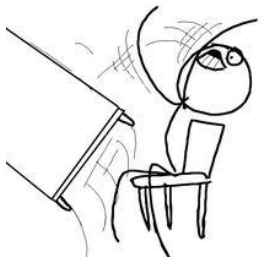
There is a semi-feasible set (satisfying the 3 requirements) which is not NP/n .

Hint: It’s not surprising if we have questions for both of the claims later!

Proof of Claim 3.14

Claim 3.14 *Any set L satisfying equations 3.5, 3.6, and 3.7 is semi-feasible.*

Let L satisfy the 3 requirements, to prove L is semi-feasible, we need to construct a P-selector function f for L



$$f(x,y) = \begin{cases} x & \text{if } |y| \notin Q, \\ y & \text{if } |x| \notin Q \wedge |y| \in Q, \\ \min\{x,y\} & \text{if } |x| \in Q \wedge |y| \in Q \wedge |x| = |y|, \\ \min\{x,y\} & \text{if } |x| \in Q \wedge |y| \in Q \wedge |x| \neq |y| \wedge \min\{x,y\} \in L, \\ \max\{x,y\} & \text{if } |x| \in Q \wedge |y| \in Q \wedge |x| \neq |y| \wedge \min\{x,y\} \notin L. \end{cases}$$

Proof of Claim 3.14

Let’s go through all the cases one by one:

$$f(x,y) = \begin{cases} x & \text{if } |y| \notin Q, \\ y & \text{if } |x| \notin Q \wedge |y| \in Q, \\ \min\{x,y\} & \text{if } |x| \in Q \wedge |y| \in Q \wedge |x| = |y|, \\ \min\{x,y\} & \text{if } |x| \in Q \wedge |y| \in Q \wedge |x| \neq |y| \wedge \min\{x,y\} \in L, \\ \max\{x,y\} & \text{if } |x| \in Q \wedge |y| \in Q \wedge |x| \neq |y| \wedge \min\{x,y\} \notin L. \end{cases}$$

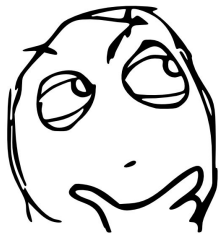
Can you see that by this construction, if one of (x, y) is in L, then the output of f(x,y) is in L!

$$f(x,y) = \begin{cases} x & \text{if } |y| \notin Q, \\ y & \text{if } |x| \notin Q \wedge |y| \in Q, \\ \min\{x,y\} & \text{if } |x| \in Q \wedge |y| \in Q \wedge |x| = |y|, \\ \min\{x,y\} & \text{if } |x| \in Q \wedge |y| \in Q \wedge |x| \neq |y| \wedge \min\{x,y\} \in L, \\ \max\{x,y\} & \text{if } |x| \in Q \wedge |y| \in Q \wedge |x| \neq |y| \wedge \min\{x,y\} \notin L. \end{cases}$$

But is this function P time computable?

For the first 3 cases: clearly yes, since we only need to check the length of x and y

For the last 2 cases, the trickiest part is in computing whether or not min{x, y} is in L. Is this part P time computable?



Recall that Q is of the form: $Q = \{2, 2^{2^{2^2}}, 2^{2^{2^{2^{2^{2^2}}}}}, \ldots\}$

In the last two cases, if $|x| \in Q, |y| \in Q, \text{and } |x| \neq |y|$

Then the following must hold:

$$\max\{|x|, |y|\} \geq 2^{2^{2^{\min\{|x|,|y|\}}}}$$

Why?

For example $|x| = l_i$ and $|y| = l_j$, and $j > i$

if $j - i = 1$ then $max\{|x|, |y|\} = 2^{2^{2^{min\{|x|,|y|\}}}}$

if $j - i > 1$ then $max\{|x|, |y|\} > 2^{2^{2^{min\{|x|,|y|\}}}}$

L is in 2^{2^n} time on input of length n. (Condition 3.7)

So we'll have a machine testing whether $min\{x, y\} \in L$, and that machine will run in $2^{2^{min\{|x|,|y|\}}}$, which time amount is polynomial in $max(|x|, |y|)$, since $max\{|x|, |y|\} \geq 2^{2^{2^{min\{|x|,|y|\}}}}$

Thus we can easily compute whether $min\{x, y\}$ is in L in time polynomial in $|x| + |y|$.

Thus f is a P-selector for L, i.e. L is semi-feasible.

Proof of Claim 3.15

There is a set $L \notin NP/n$ satisfying Equation 3.5, 3.6, and 3.7

Definitions

- N_1, N_2, N_3, … standard NDPT
- $NP = \{B | (\exists i)[L(N_i) = B]\}$
- $<.,.>$ a 2-ary pairing function
- $\widehat{N_1}, \widehat{N_2}, \widehat{N_3}$, … be such that, for each i and k $\widehat{N}_{\langle i,k \rangle}$ is simply N_i

Proof by Diagonalization

Stage l: We define the contents of $L^{=l}$

If $l \notin Q$ then

$L^{=l} = \phi$

Move to stage $l + 1$

If $l \in Q$ then do the following for at most 2^{2^l} steps (total).

If number of steps reached but we couldn't complete then

$L^{=l} = \phi$

Proof by Diagonalization

Say l is $< i, k > th$ element of Q.

$$\langle i, k \rangle = \left| |\{j | j \in Q \ \wedge j \leq l\}| \right|$$

Consider $\widehat{N}_{<i,k>}$, for each 2^l potential advice string y of length l do:

For each 2^l strings x of length l run $\widehat{N}_{\langle i,k \rangle}(\langle x, y \rangle)$

If none of the 2^l runs accept then

$$rightmost_y = 1^{l-1}$$

Otherwise,

$$rightmost_y = \max\{x | \, |x| = l \wedge \widehat{N}_{\langle i,k \rangle}(\langle x, y \rangle) accepts\}$$

Proof by Diagonalization

- For each 2^l advice strings we have at most one $rightmost$ string
- Total number of $rightmost$ string is at most 2^l.
- $J_l = \left(\Sigma^l \cup \{1^{l-1}\}\right) - \{z | (\exists y \in \Sigma^l)[rightmost_y = z]\}$ is not empty.
- Let j_l be lexicographically smallest element of J_l.
- Set $L^{=l} = \{x |\ l = |x| \wedge x \leq_{lex} j_l\}$
- Equation 3.5, 3.6, and 3.7 holds.
- Remains to show $L \notin NP/n$

Show $\boldsymbol{L \notin NP/n}$

- Let $L \in NP/n$ via NP language L'. $L(N_{i'}) = L'$
- For each k', $\widehat{N}_{\langle i',k' \rangle} = N_{i'}$
- The stage l that satisfies $\langle i', k' \rangle = ||\{j | j \in Q \wedge j \leq l\}||$ requires about $2^l 2^l \left(2^{bl^c}\right)^2$ steps
- For sufficiently large l, this is less than 2^{2^l}
- Let k'' be one k' for which completion occur.
- Let l''denote $\langle i', k'' \rangle th$ element of Q.

Show $L \notin NP/n$

- Notice- none of the $2^{l''}$ advice string of length l'', when given to $\widehat{N}_{\langle i',k' \rangle}$ yields at length l'', $L^{=l''}$
- We have two claims:

1. If $|j_{l''}| = l''$, then for each l'' length advice string y,
   - Either $\widehat{N}_{i'}(\langle j_{l''}, y \rangle)$ rejects (yet $j_{l''} \in L$)
   - Or for some length l'' string $z >_{lex} j_{l''}$ it holds that $N_{i'}(\langle z, y \rangle)$ accepts (yet $z \notin L$)
2. If $|j_{l''}| = l'' - 1$, then $L^{=l''}$ is empty but for each advice string $y \in \Sigma^{l''}$ there is an x $\in \Sigma^{l''}$ such that $N_{i'}(\langle x, y \rangle)$ accepts.

Show $\boldsymbol{L \notin NP/n}$

- Thus $L \notin L'/n$. This contradicts our assumption that $L \in NP/n$ via NP language L'.
- So $\boldsymbol{L \notin NP/n}$
- P-sel $\not\subseteq NP/n$

This is the end of the lecture 2.

The Tournament Divide and Conquer Technique

Daniel Rubery
Georgiy Platonov
Mohammad Rafayet Ali
Xiong Zhang

Lecture 3

Unique solutions collapse the Polynomial Hierarchy

Motivation

Looking at functions that give a satisfying assignment of clauses. We want some kind of function such that for any F in SAT, f(F) is a satisfying assignment of F

If there is a deterministic polynomial time function f, then P=NP

There definitely exists an FP function with NP oracle.

We will define a weaker function, between FP and FP^{NP}, whose existence implies PH = NP^{NP}

Definitions

1) Let f be a multivalued function. set-f(x) denotes the set of all values that are an output of f(x). If f(x) has no output then set-f (x) is the empty set.
2) We consider any given nondeterministic polynomial-time machine N to implicitly compute a (potentially partial) multivalued function, namely, the function f_N defined by the set set-$f_N(x)$ = { y | some computation path of N(x) outputs y }. NPMV denotes the class of functions computed in this sense by nondeterministic polynomial-time machines.

More definitions

3) A (potentially partial) multivalued function f is said to be single-valued if $\forall x(||\text{set-f}(x)|| \leq 1)$. NPSV denotes the class of all single-valued NPMV functions.
4) Given a multivalued functions f and g, we say that g is a refinement of f if
   a. $\forall x(\text{set-g}(x) \subseteq \text{set-f}(x))$, and
   b. $\forall x(\text{set-g}(x) = \varnothing \Rightarrow \text{set-f}(x) = \varnothing)$.

Yet more definitions

5) Let F be any (possibly partial, possibly multivalued) function class. We say a set L is F-selective if there is a multivalued function $f \in F$ such that
   a. $\forall x, y(\text{set-f}(x, y) \subseteq \{x, y\})$, and
   b. $\forall x, y((x \in L \vee y \in L) \Rightarrow (\text{set-f}(x,y) \subseteq L \wedge \text{set-f}(x,y) \neq \varnothing))$.

Theorem 3.21

If all NPMV functions have NPSV refinements, then $PH = NP^{NP}$.

We will use the following two lemmas:

1) NPSV-sel $\cap$ NP $\subseteq$ (NP $\cap$ coNP)/poly. (Lemma 3.25)

2) NP $\subseteq$ (NP $\cap$ coNP)/poly $\Rightarrow$ PH = NP^{NP}. (Lemma 3.26)

Proof of Lemma 3.25: NPSV-sel ∩ NP ⊆ (NP ∩ coNP)/poly

Let L be a language in NPSV-sel∩NP. Let N_L be a NPTM accepting L, and f an NPSV selector.

WLOG, for any x,y we have set-f(x,y) = set-f(y,x).

Now we construct an NP∩coNP interpreter A and an advice function g.

Let A be all strings of the form $\langle x, \langle\langle a_1,a_2,\ldots,a_z\rangle, \langle w_1,\ldots,w_{z'}\rangle\rangle\rangle$ such that:

$z = z'$

$\forall i$ (w_i is an accepting path of N_L on a_i)

$\exists i$ ($x \in$ set-f(x,a_i))

Proof of Lemma 3.25: NPSV-sel ∩ NP ⊆ (NP ∩ coNP)/poly

Clearly, A is in NP.

$\overline{A}$ is also in NP, as follows:

Accept any syntactically ill-formed input, or if $z \neq z'$
Deterministically check that each w_i is an accepting path of a_i
Reject if, for some i, $x = a_i$
Since we now know each a_i is in L, set-f(a_i,x) must have exactly one value
For each i, non-deterministically choose a path for f(a_i,x)
    If it outputs nothing, reject
    If it outputs x, reject
    If it outputs a_i, continue
Accept if each path gives f(a_i,x) = a_i.

Proof of Lemma 3.25: NPSV-sel ∩ NP ⊆ (NP ∩ coNP)/poly

Create a tournament on $L^{=n}$. If $a,b \in L^{=n}$, then there is an edge from a to b if set-f (a,b) = {b}.

Now use Theorem 3.1 to get a set H with at most n+1 strings. For every $x \in L^{=n}$, there is an $h \in H$, such that x=h or set-f(x,h) = {x}.

Then our advice function is $g(n) = \langle\langle h_1,h_2,...,h_n\rangle,\langle w_1,w_2,...,w_n\rangle\rangle$, where each w_i is an accepting path of h_i. This is polynomial length in n.

$x \in L$ iff $\langle x, g(|x|)\rangle \in A$

So $L \in (NP \cap coNP)/poly$, and since L was arbitrary in NPSV-sel∩NP, we get the Lemma.

Proof of Lemma 3.26: $NP \subseteq (NP \cap coNP)/poly \Rightarrow PH = NP^{NP}$

By theorem 1.16 (Karp-Lipton)

$$NP \subseteq P/poly \Rightarrow PH = NP^{NP}.$$

This result relativizes, i.e.

$$\forall A(NP^A \subseteq P^A/poly \Rightarrow PH^A = NP^{NP^A}).$$

Proof of Lemma 3.26: $NP \subseteq (NP \cap coNP)/poly \Rightarrow PH = NP^{NP}$

Assume $NP \subseteq (NP \cap coNP)/poly$. Then $SAT \in (NP \cap coNP)/poly$ via some $B \in NP \cap coNP$, and some advice g in poly.

From

$$\forall A(NP^A \subseteq P^A/poly \Rightarrow PH^A = NP^{NP\hat{}A})$$

we have

$$NP^B \subseteq P^B/poly \Rightarrow PH^B = NP^{NP\hat{}B}.$$

Proof of Lemma 3.26: $NP \subseteq (NP \cap coNP)/poly \Rightarrow PH = NP^{NP}$

Having

$$NP^B \subseteq P^B/poly \Rightarrow PH^B = NP^{NP^B},$$

we are going to prove that

$$NP \subseteq (NP \cap coNP)/poly \Rightarrow NP^B \subseteq P^B/poly$$

and

$$PH^B = NP^{NP^B} \Rightarrow PH = NP^{NP}.$$

Proof of Lemma 3.26: $NP \subseteq (NP \cap coNP)/poly \Rightarrow PH = NP^{NP}$

It is easy to prove that

$$NP \subseteq NP^{B} \subseteq NP^{NP \cap coNP} = NP$$

So $NP = NP^{B}$

Proof of Lemma 3.26: $NP \subseteq (NP \cap coNP)/poly \Rightarrow PH = NP^{NP}$

For any $L \in NP$, let h be a reduction from L to SAT.

Let $pad_n(x)$ add dummy clauses to a SAT formula so that it is length n, and let l(n) be the maximum length of h(x) for x in $\{0,1\}^n$.

$x \in L$ iff $h(x) \in SAT$ iff $pad_{l(n)}(h(x)) \in SAT$ iff $\{<pad_{l(n)}(h(x)),g(l(n))>\} \in B$

So L can be decided with advice $g'(n) = <l(n),g(l(n))>$

By the advice interpreter:

$\{<x,<len,w>> \mid <pad_{len}(h(x)), w> \in B\}$, which is P^B

So $NP=NP^B \subseteq P^B/poly$

Proof of Lemma 3.26: $NP \subseteq (NP \cap coNP)/poly \Rightarrow PH = NP^{NP}$

By Karp-Lipton, $PH^B = NP^{NP^{\wedge}B}$

Finally, note that

$$B \in NP \cap coNP,\ PH^B = PH \text{ and } NP^{NP^{\wedge}B} = NP^{NP}.$$

So

$$NP \subseteq (NP \cap coNP)/poly \Rightarrow PH = NP^{NP}.$$

Theorem 3.21

Now, using these two lemmas, we can prove Theorem 3.20, as follows.

Construct a NPMV-selector for SAT by set-f(x,y) = {x,y}∩SAT.

This is NPMV by guessing an argument and assignment of variables, then outputting the argument if the assignment is valid.

By the hypothesis of the Theorem, f has a NPSV refinement g.

Then g is a NPSV selector for SAT.

Then SAT∈NPSV-sel.

Theorem 3.21

SAT$\in$NPSV-sel. Will show that this gives NP$\subseteq$NPSV-sel.

For any NP language L, let h polynomial many-one reduce L to SAT.

Then define g’(x,y) =

{x} if g(h(x),h(y)) = h(x)

{y} if g(h(x),h(y)) = h(y)

$\varnothing$ otherwise

Then g’(x,y) is an NPSV-selector for L, so L$\in$NPSV-sel, and NP$\subseteq$NPSV-sel.

Theorem 3.21

So we have that, if every NPMV function has an NPSV refinement, then $NP \subseteq NPSV\text{-}sel$.

Lemma 3.25 says $NPSV\text{-}sel \cap NP \subseteq (NP \cap coNP)/poly$

So $NP \subseteq (NP \cap coNP)/poly$

This is the hypothesis of Lemma 3.26, so $PH = NP^{NP}$

Thank you!