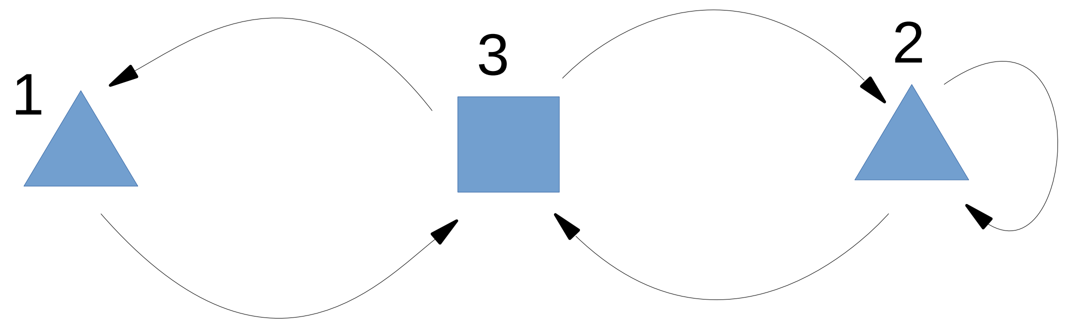
Improved Complexity Analysis of Quasi-Polynomial Algorithms Solving Parity Games

Paweł Parys, Aleksander Wiącek
University of Warsaw

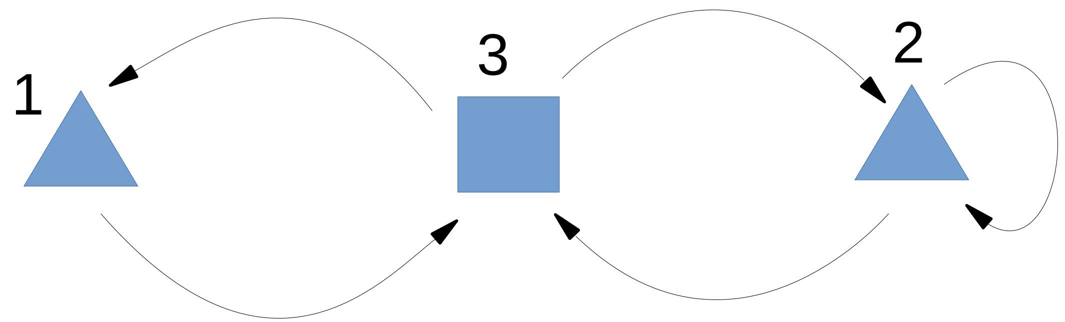
CiE 2023, Batumi

Parity games



- Priorities on vertices
- Player owning the current vertex choses the next vertex
- Player ☐ wins if the biggest priority seen infinitely often is even.

Parity games

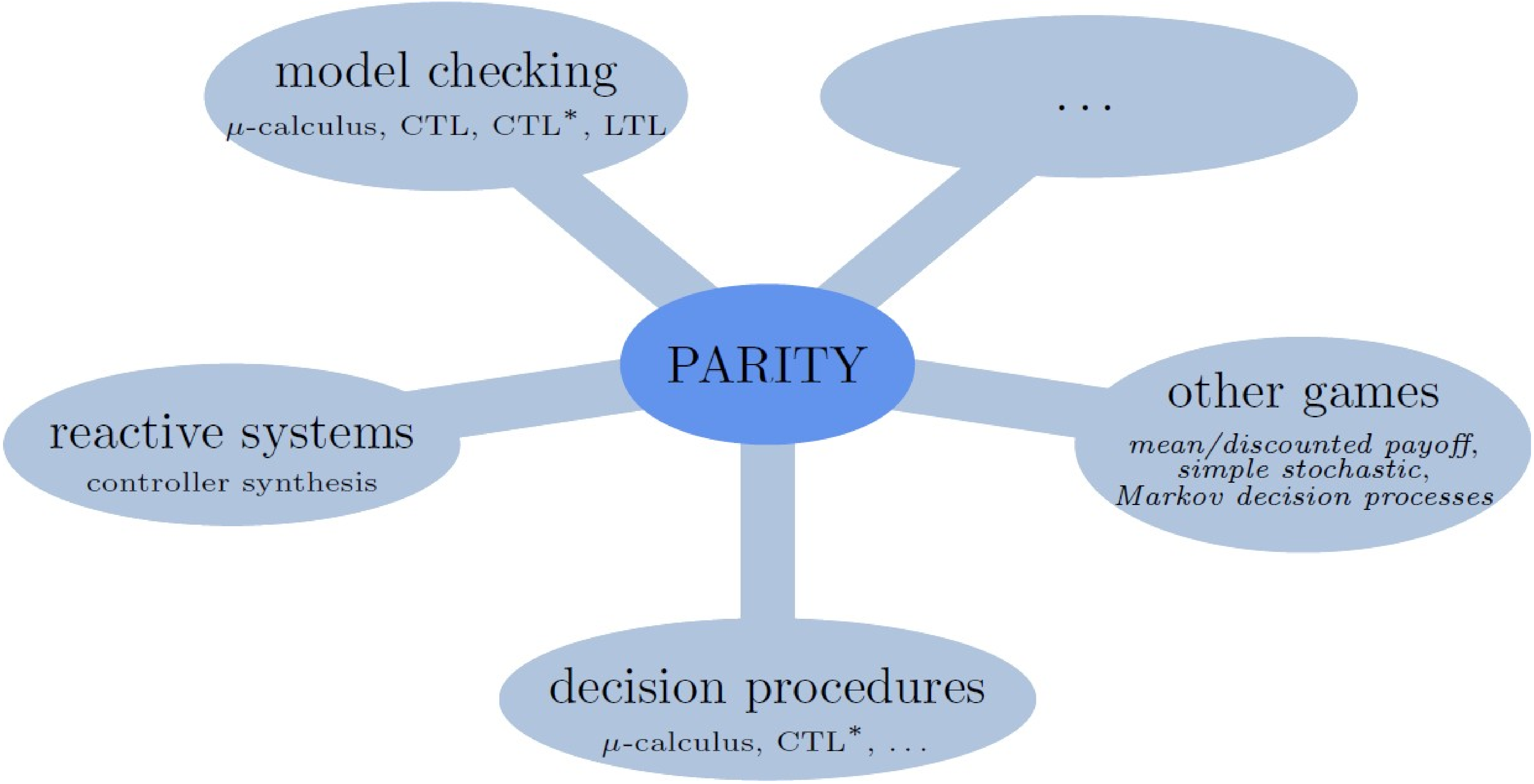


- Priorities on vertices
- Player owning the current vertex choses the next vertex
- Player ☐ wins if the biggest priority seen infinitely often is even.

Algorithmic problem:
Given a game graph, decide which player has a winning strategy.

Long standing open problem:
Can we solve parity games in PTIME?

Parity games



Recent results

Long standing open problem:
Decide in PTIME which player has a winning strategy.

Recent result:
This can be decided in quasi-polynomial time, i.e. $n^{O(\log n)}$

A few algorithms achieving this:

- play summaries - Calude, Jain, Khoussainov, Li, Stephan 2017
- antagonistic play summaries - Fearnley, Jain, Schewe, Stephan, Wojtczak 2017
- succinct progress measures - Jurdziński, Lazić 2018
- register games - Lehtinen 2018
- recursive à la Zielonka - Parys 2019
- improved recursive à la Zielonka - Lehtinen, Schewe, Wojtczak 2019
- symmetric progress measures - Jurdziński, Morvan, Ohlmann, Thejaswini 2020
- strategy iteration - Koh, Loho 2021

This paper:

Small improvement in the complexity analysis of the algorithms

Previous: $O(mdn^{\log_2 e + \log_2(d/\log_2 n)})$

New: $O(m\frac{1}{d}n^{\log_2 e + \log_2(d/\log_2 n)})$

where

n – number of nodes

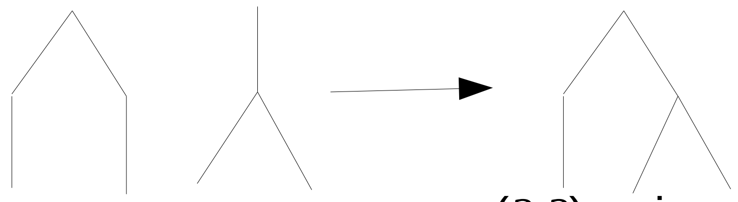
m – number of edges

d – number of priorities

(we skip polylogarithmic factors)

Universal trees

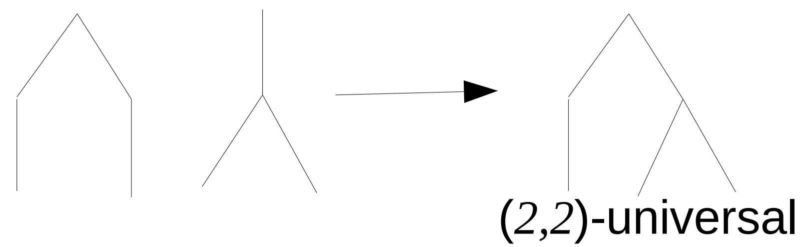
A tree U (of height h) is (*n*,*h*)-universal if every tree of height h with n leaves embeds in U.



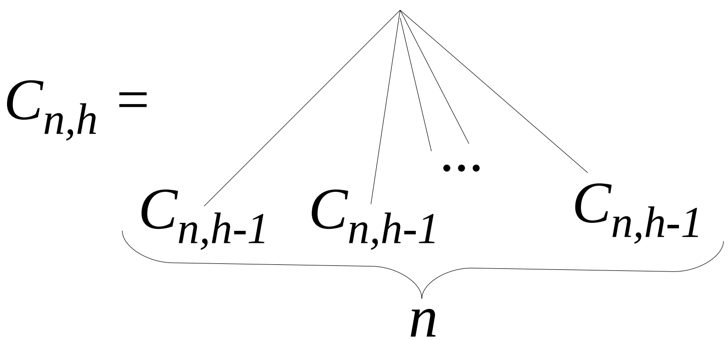
(*2*,*2*)-universal

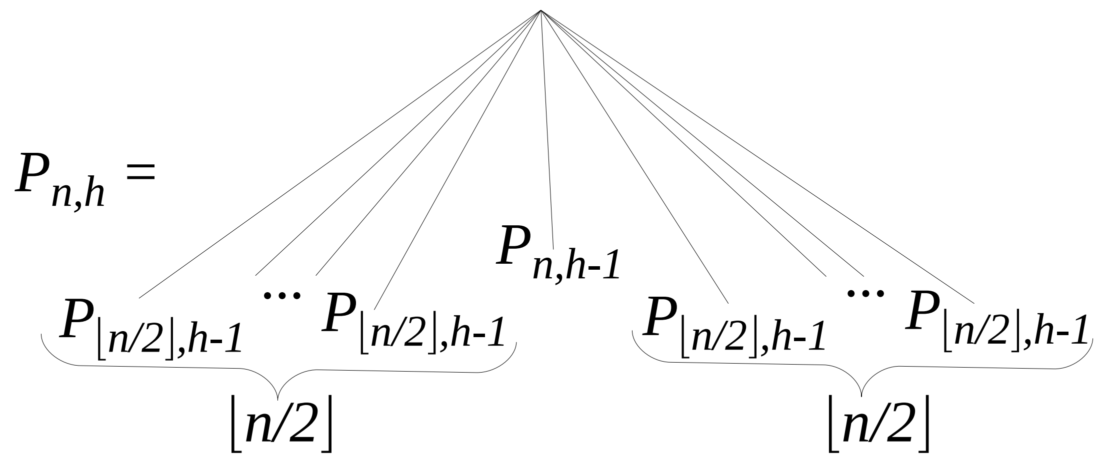
Universal trees

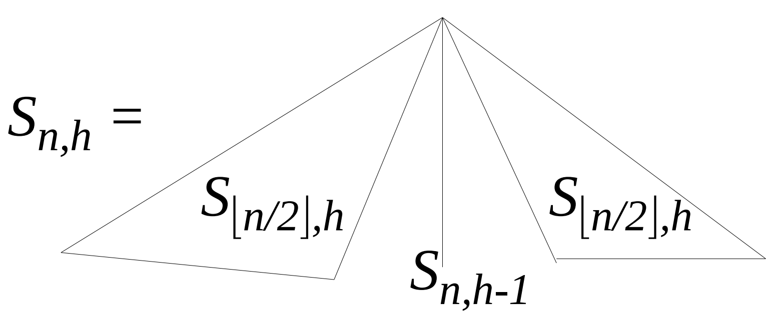
A tree U (of height h) is <u>(n,h)-universal</u> if every tree of height h with n leaves embeds in U.



<u>Examples</u>:

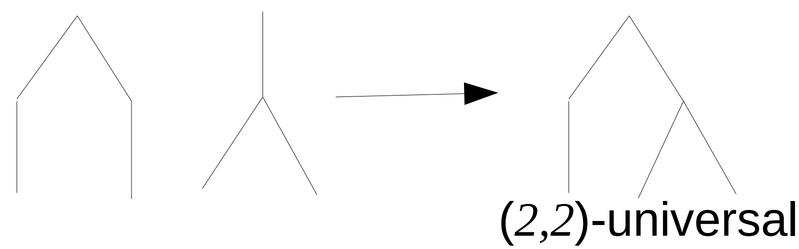




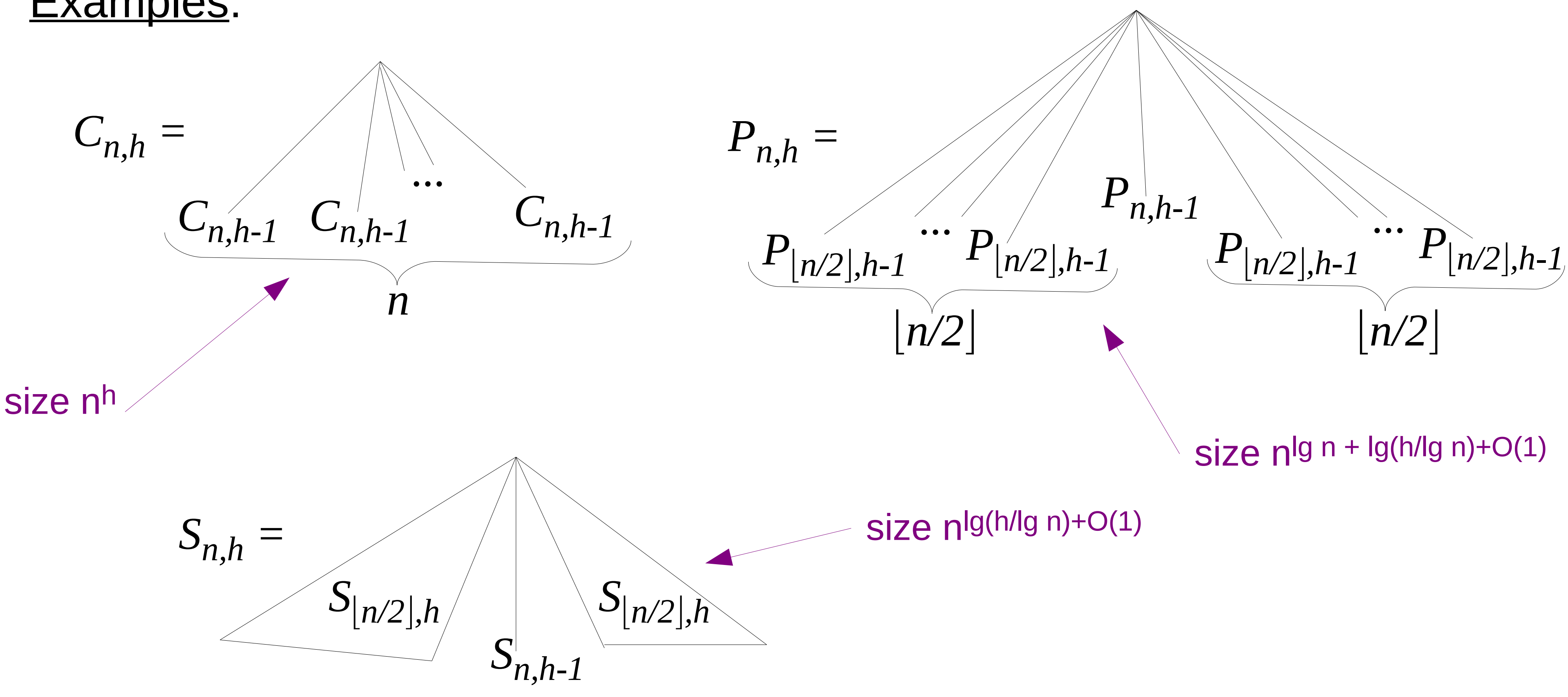


Universal trees

A tree U (of height h) is (n,h)-universal if every tree of height h with n leaves embeds in U.



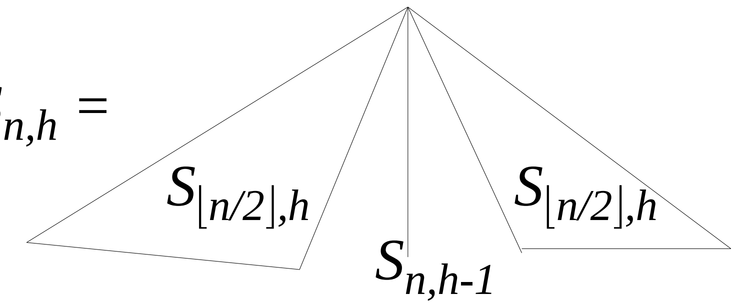
Examples:



Universal trees

A tree U (of height h) is <u>(n,h)-universal</u> if every tree of height h with n leaves embeds in U.

$S_{n,h}$ =

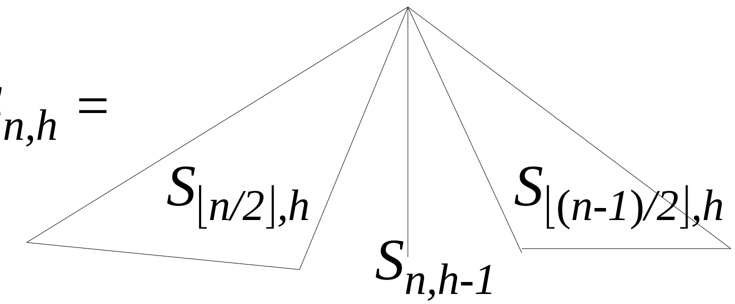


Why is it (n,h)-universal?

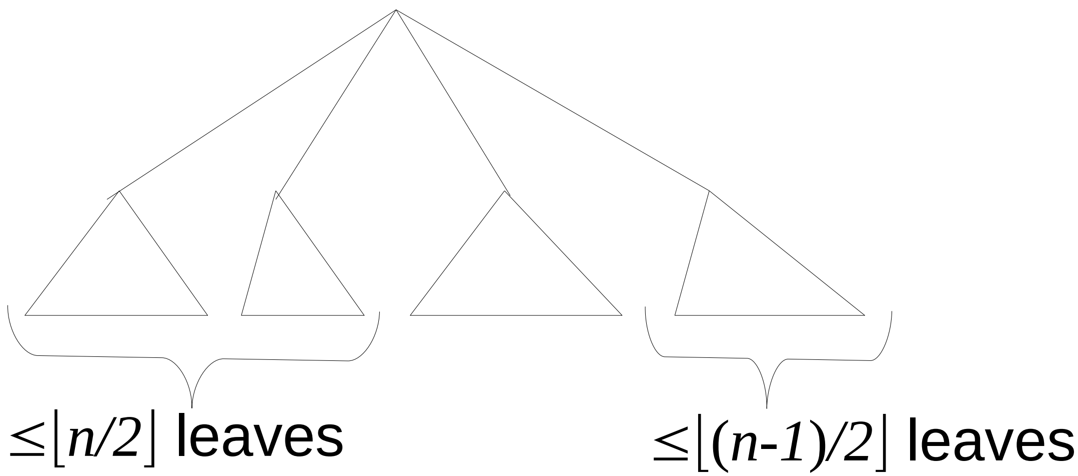
Universal trees

A tree U (of height h) is (n,h)-universal if every tree of height h with n leaves embeds in U.

$S_{n,h}$ =



Why is it (n,h)-universal?
Take any tree T of height h with n leaves.

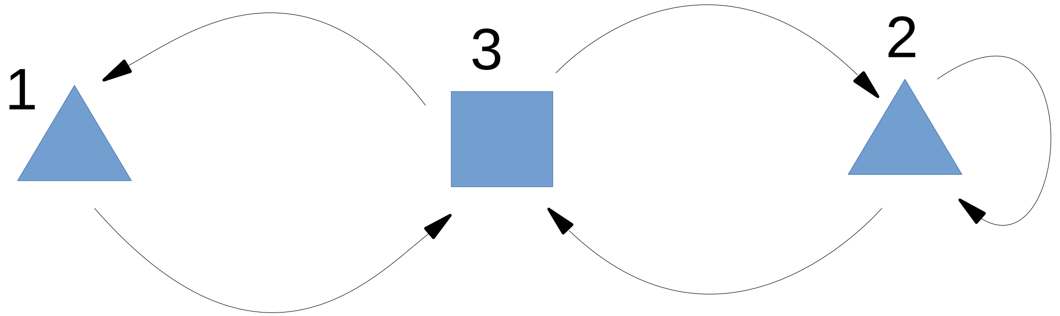


Subtree with the middle leaf goes to $S_{n,h-1}$.
Left and right part have at most $\lfloor n/2 \rfloor$ or $\lfloor (n-1)/2 \rfloor$ leaves.

Why universal trees?

1) It is enough to consider positional strategies: given a node, player chooses some fixed successor, no matter what was the history of the play.
If a player can win, then he can win positionally.

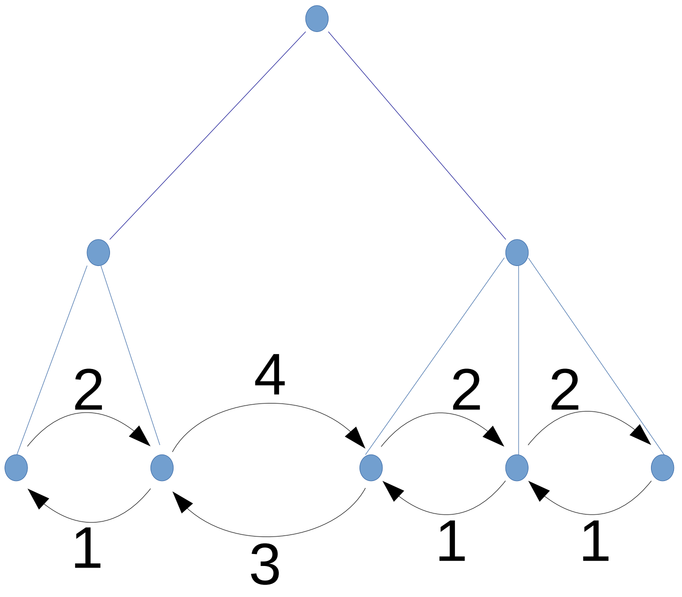
Consequence: the problem is in NP∩coNP.
In fact it is also in UP∩coUP (Jurdziński 1998)
The search variant is in PLS, PPAD, CLS (Daskalakis, Papadimitriou 2011)



Why universal trees?

1) It is enough to consider positional strategies: given a node, player choses some fixed successor, no matter what was the history of the play.
If a player can win, then he can win positionally.

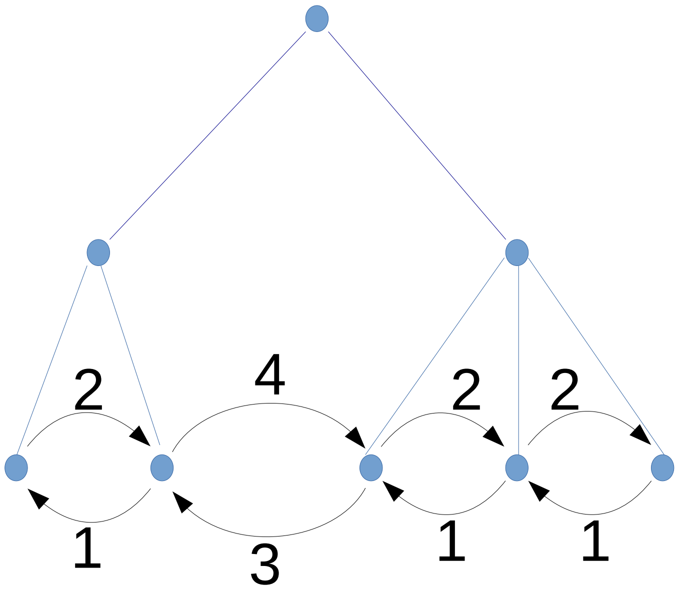
2) After fixing a positional strategy, a game graph defines a tree of height $d/2$ with n leaves (game node = tree leaf)



Why universal trees?

1) It is enough to consider positional strategies: given a node, player choses some fixed successor, no matter what was the history of the play.
If a player can win, then he can win positionally.

2) After fixing a positional strategy, a game graph defines a tree of height *d/2* with *n* leaves (game node = tree leaf)



3) Idea: checking a universal tree = checking all positional strategies

Why universal trees?

All known quasipolynomial algorithms solving parity games use (explicitly or implicitly) universal trees.

Is this necessary?
Papers
Czerwiński, Daviaud, Fijalkow, Jurdziński, Lazić, Parys 2019
Arnold, Niwiński, Parys 2021
define two general approaches such that

- all known quasipolynomial algorithms follow these approaches
- every algorithm following this approach has to use a universal tree

Why universal trees?

All known quasipolynomial algorithms solving parity games use (explicitly or implicitly) universal trees.

Is this necessary?
Papers
Czerwiński, Daviaud, Fijalkow, Jurdziński, Lazić, Parys 2019
Arnold, Niwiński, Parys 2021
define two general approaches such that

- all known quasipolynomial algorithms follow these approaches
- every algorithm following this approach has to use a universal tree

Complexity of the (best) algorithms?

$$O(m \cdot |S_{n,d/2}|)$$

Improvement 1: this can be changed to

$$O(m \cdot |S_{n/2,d/2}|)$$

i.e., we can use universal trees for $n/2$ leaves

(not really new – already observed in some older papers,
but not present in papers with the best complexity)

Why universal trees?

Complexity of the (best) algorithms?

$O(m \cdot |S_{n,d/2}|)$

Improvement 1: this can be changed to

$O(m \cdot |S_{n/2,d/2}|)$

i.e., we can use universal trees for $n/2$ leaves

(not really new – already observed in some older papers,
but not present in papers with the best complexity)

Idea: map only nodes of odd priority (or only nodes of even priority)
to leaves of the universal tree.
There are at most $n/2$ of them.

Why universal trees?

Complexity of the (best) algorithms?

$O(m \cdot |S_{n,d/2}|)$

Improvement 1: this can be changed to

$O(m \cdot |S_{n/2,d/2}|)$

i.e., we can use universal trees for $n/2$ leaves

(not really new – already observed in some older papers,
but not present in papers with the best complexity)

Idea: map only nodes of odd priority (or only nodes of even priority)
to leaves of the universal tree.
There are at most $n/2$ of them.

Anyway: it is essential to bound the size of universal trees.

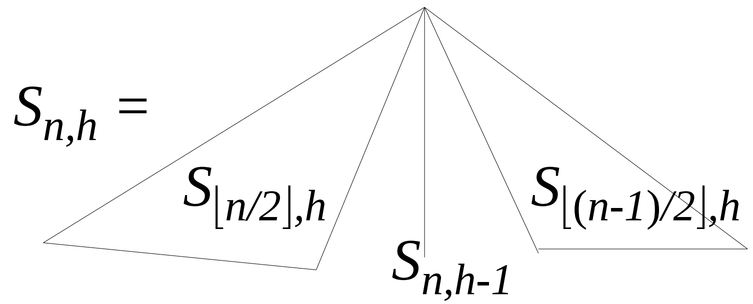
What is the size?

Recursive formula:

$|S_{0,h}|=0$

$|S_{n,0}|=1$

$|S_{n,h}|=|S_{n,h-1}|+|S_{\lfloor n/2\rfloor,h}|+|S_{\lfloor (n-1)/2\rfloor,h}|$



Theorem

$$|S_{n,h}|\leq n\cdot\binom{h-1+\lfloor\log_2 n\rfloor}{\lfloor\log_2 n\rfloor}\leq n^{1+\log_2 e+\log_2(1+h/\log_2 n)}$$

(we did better analysis – previous bound was greater *h* times)

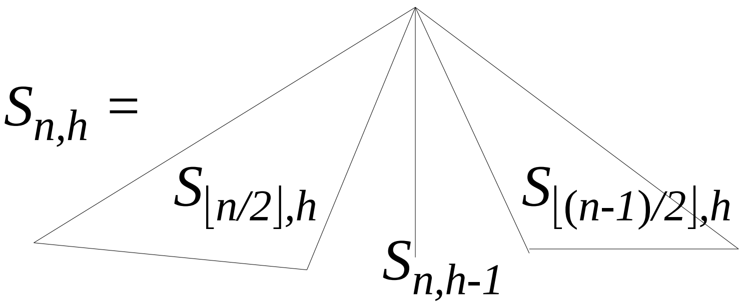
What is the size?

Recursive formula:

$$|S_{0,h}| = 0$$

$$|S_{n,0}| = 1$$

$$|S_{n,h}| = |S_{n,h-1}| + |S_{\lfloor n/2 \rfloor,h}| + |S_{\lfloor (n-1)/2 \rfloor,h}|$$



Theorem

$$|S_{n,h}| \leq n \cdot \binom{h-1+\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor} \leq n^{1+\log_2 e+\log_2(1+h/\log_2 n)}$$

(we did better analysis – previous bound was greater h times)

Lower bound?

Every (n,h)-universal tree satisfies

$$|U_{n,h}| \geq \binom{h+\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor} \geq \left(\frac{n}{2}\right)^{\log_2(1+h/\log_2 n)}$$

(Czerwiński, Daviaud, Fijalkow, Jurdziński, Lazić, Parys 2019 + our improvements)

What is the size?

Upper bound:

$$|S_{n,h}| \le n \cdot \binom{h-1+\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor} \le n^{1+\log_2 e + \log_2(1+h/\log_2 n)}$$

Lower bound:

$$|U_{n,h}| \ge \binom{h+\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor} \ge \left(\frac{n}{2}\right)^{\log_2(1+h/\log_2 n)}$$

$$\frac{\text{upper bound}}{\text{lower bound}} \le n$$

What is the size?

Upper bound:

$$|S_{n,h}| \le n \cdot \binom{h-1+\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor} \le n^{1+\log_2 e + \log_2(1+h/\log_2 n)}$$

Lower bound:

$$|U_{n,h}| \ge \binom{h+\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor} \ge \left(\frac{n}{2}\right)^{\log_2(1+h/\log_2 n)}$$

$$\frac{\text{upper bound}}{\text{lower bound}} \le n$$

Open questions:

- Can this be improved?
- Is there any universal tree smaller than $S_{n,h}$?

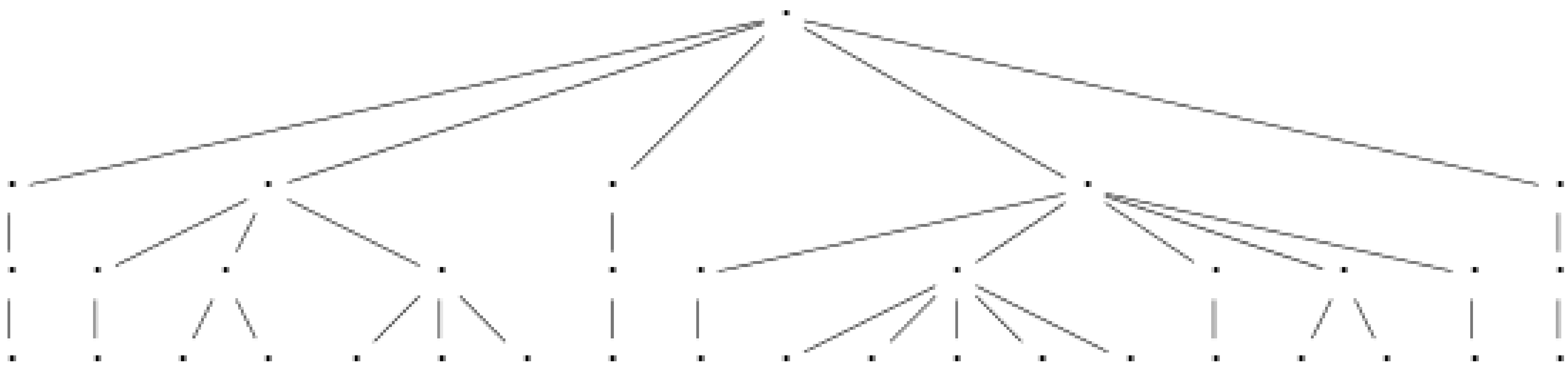
What is the size?

Open questions:

- Can the bounds be improved?
- Is there any universal tree smaller than $S_{n,h}$?

Partial answers:

- For $h=2$ the tree $S_{n,2}$ is optimal.
- There is exists a “strange” $(5,3)$-universal tree of the same size as $S_{5,3}$



Summary

Small improvement in the complexity of solving parity games:

Previous: $O(mdn^{\log_2 e + \log_2(d/\log_2 n)})$ New: $O(m\frac{1}{d}n^{\log_2 e + \log_2(d/\log_2 n)})$

Small improvement in bounds for size of (n,h)-universal tree:

$\frac{\text{upper bound}}{\text{lower bound}} \leq n$ (previously: nh)

Thank you!