Succinct progress measures and solving parity games in quasi-polynomial time

Marcin Jurdziński   Ranko Lazić

DIMAP
Department of Computer Science
University of Warwick

RELEVANT RECENT PAPERS

- Calude, Jain, Khoussainov, Li, Stephan STOC 2017
  Solving parity games in quasipolynomial time
  Best Paper Award

- J., Lazić LICS 2017
  Succinct progress measures for solving parity games

- Gimbert, Ibsen-Jensen arXiv 1702
  A short proof of correctness of the quasi-polynomial time algorithm for parity games

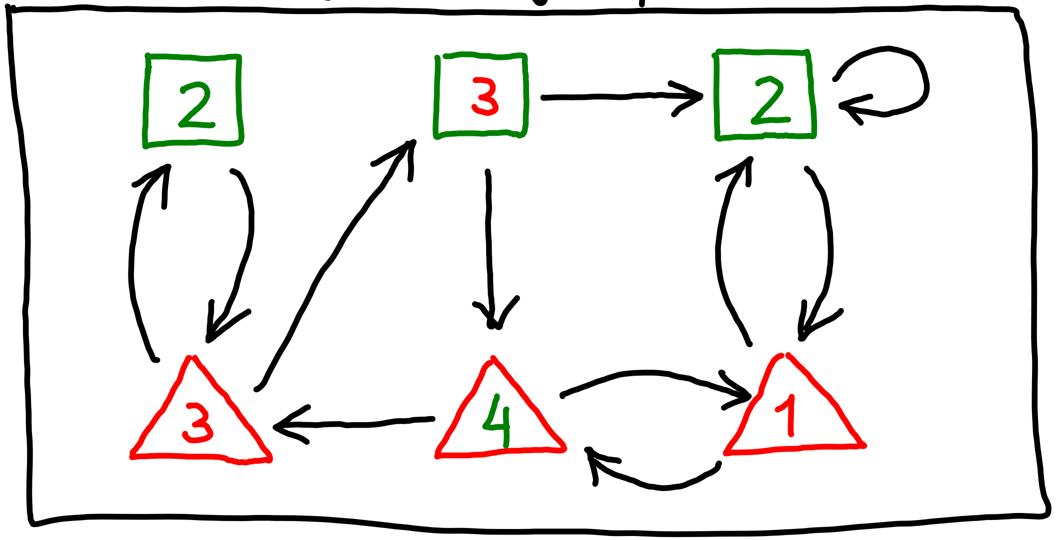
- Fearnley, Jain, Schewe, Stephan, Wojtczak SPIN 2017
  An ordered approach to solving parity games in quasipolynomial time and quasilinear space

PARITY GAMES

$n = |V|$

$m = |E|$

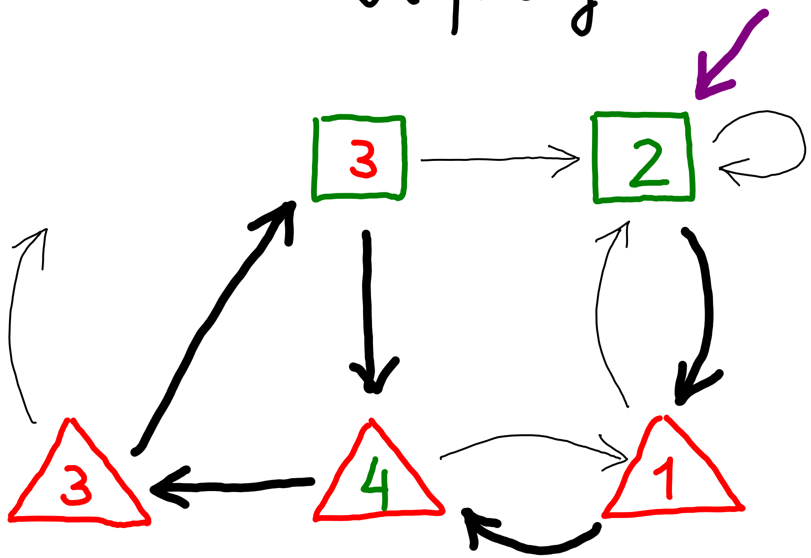
A game graph



$G = (V = V_{Even} \cup V_{Odd}, E, \pi)$

$\pi : V \to \{1, 2, 3, 4, 5, \ldots, d\}$

A play



POSITIONAL DETERMINACY

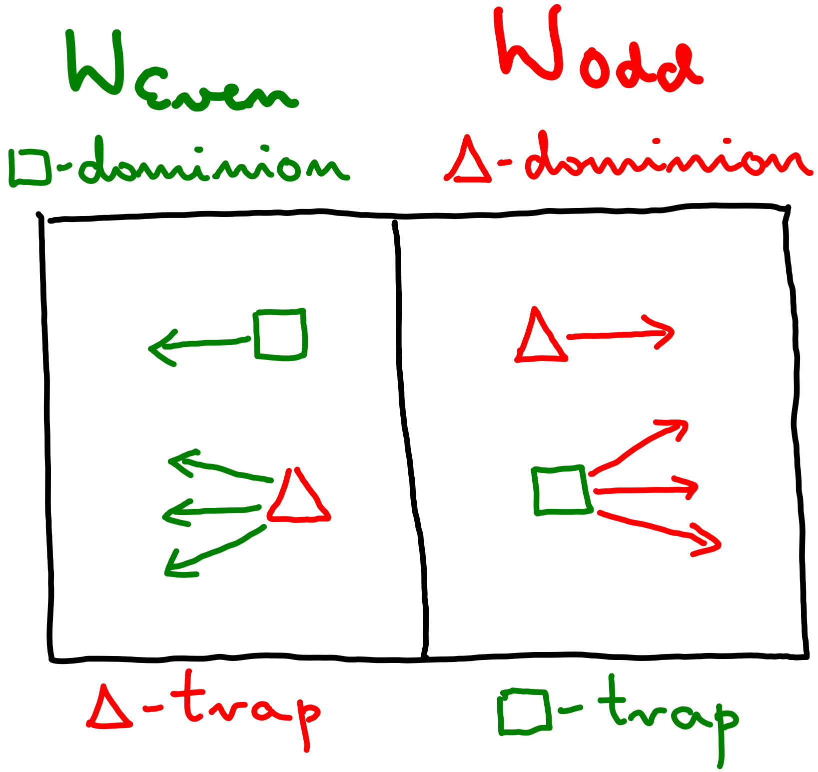
<u>Thm</u> [EJ'91, Mos'91]

There is a partition $W_{Even} \cup W_{Odd} = V$

and positional strategies

$$\sigma : V_{Even} \to V$$

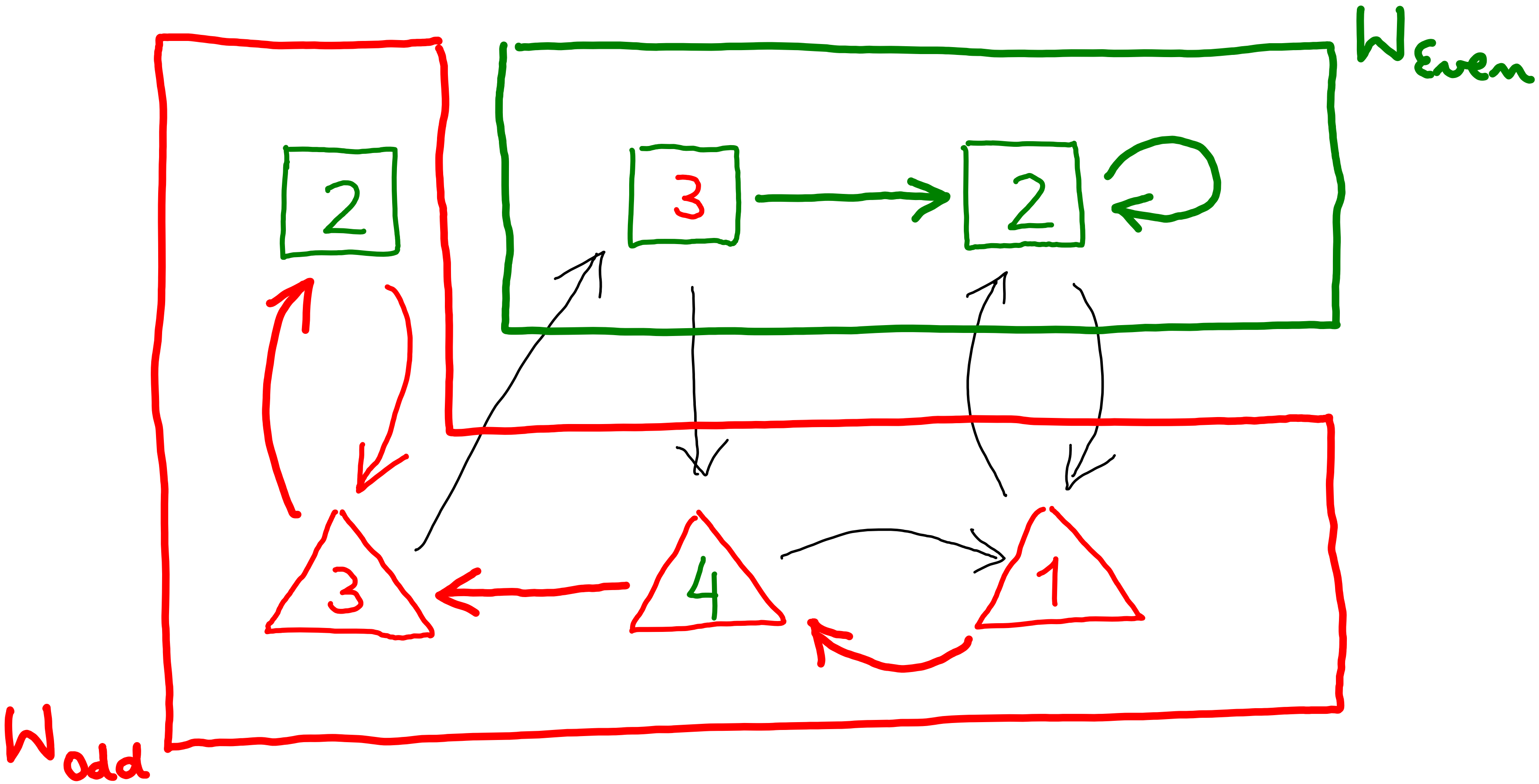
$$\tau : V_{Odd} \to V$$



such that:

- σ is a □-dominion strategy on W_{Even}
- τ is a Δ-dominion strategy on W_{Odd}

POSITIONAL DETERMINACY



<u>Corollary</u> Deciding the winner in parity games is $\boxed{\text{in NP} \cap \text{co-NP}}$

APPLICATIONS OF PARITY GAMES

- Automata theory
  - complementation
  - emptiness
  - translations
- Logic
  - satisfiability
  - fixpoint logics
- Verification
  - model checking
  - fair (bi)simulation
  - program analysis and repair
- Synthesis
- Databases and XML

- pushdown graphs
- hierarchical structures
- higher-order recursion schemes

- universal coalgebra

- stochastic systems
- timed systems
- hybrid systems

IMPACT OF PARITY GAMES

Solving Parity Games on the Playstation 3

Freark van der Berg
University of Twente, The Netherlands
f.i.vanderberg@student.utwente.nl

ABSTRACT

Parity games are a type of game in which two players 'play' on a directed graph. Solving parity games is equivalent to model checking for μ-calculus. Thus, parity game solvers can be used for model checking. This requires a lot of computational power. Many-core CPUs generally have much more computational power than other CPUs. The Playstation 3 contains an advanced, modern many-core CPU, the IBM Cell Broadband Engine Architecture (CBEA). It is a low-cost option to investigate developing efficient algorithms for many-core CPUs. However, developing efficient algorithms for The Cell remains largely uncharted territory. The Small Progress Measures par-

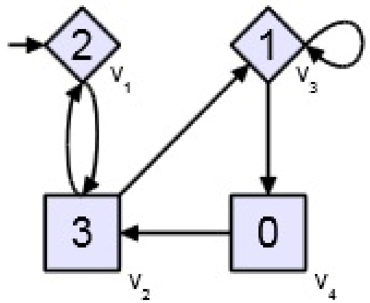


Figure 1. Parity Game

plementation for the Cell Broadband Engine Architecture,

WIDER IMPACT OF PARITY GAMES

- Structural graph theory for directed graphs

- Time complexity of Howard's policy iteration
- Time complexity of (randomized) simplex pivoting rules

- Computational complexity of search problems
- Computational complexity of path-following algorithms

COMPLEXITY OF DIVIDE-AND-CONQUER ALGORITHMS

- Plain vanilla [McN'93, Zie'98]: $n^{d+O(1)}$

$$\boxed{T(n,d) \leq n \cdot T(n,d-1) + O(nm)}$$

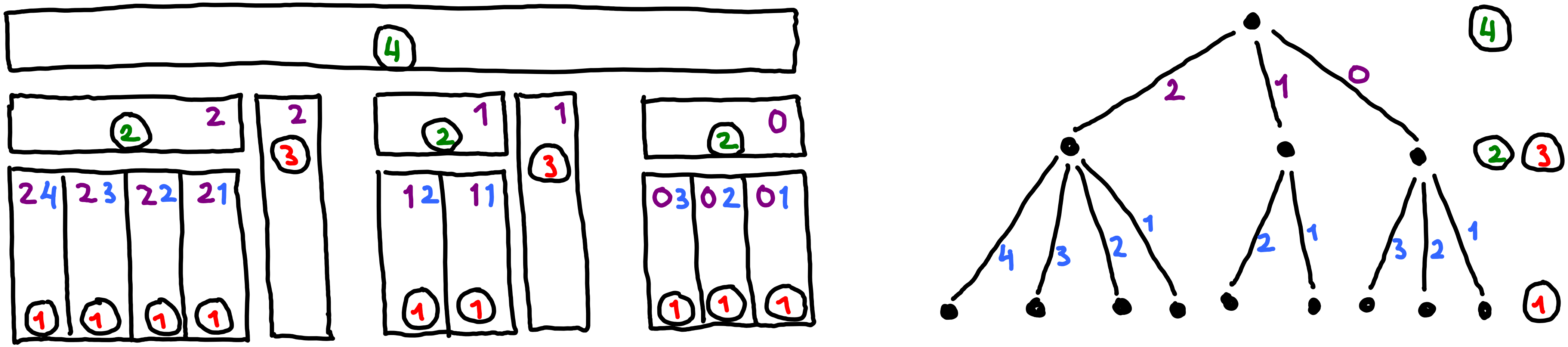
- Dominion preprocessing by brute force [JPZ'06'08]: $n^{O(\sqrt{n})}$

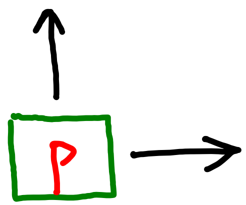
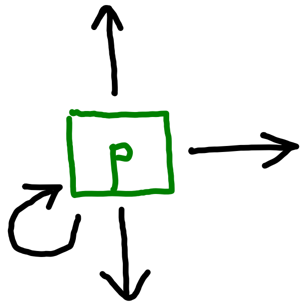
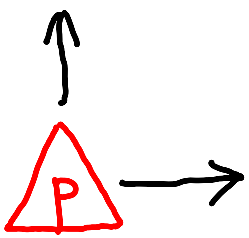
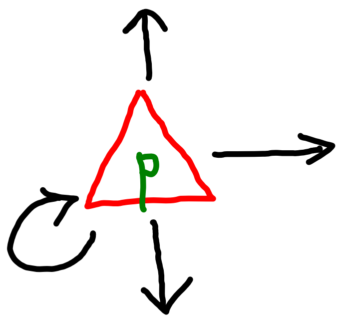
$$\boxed{T(n) \leq T(n-1) + T(n-\sqrt{n}) + n^{O(\sqrt{n})}}$$

- Dominion preprocessing by progress measure lifting [Sch'07'17]: $n^{\frac{d}{3}+O(1)}$

$$\boxed{T(n,d) \leq \sqrt[3]{n} \cdot T(n,d-1) + n^{\frac{d}{3}+O(1)}}$$

THE PARITY KEYBOARD DECOMPOSITION



	p is odd	p is even
at least one out edge		
all out edges		

A PROGRESS MEASURE

$$\boxed{\mu : V \rightarrow \mathbb{N}^{d/2}}$$

- Edge (v,u) is progressive if
  - $\mu(v)|_{\pi(v)} \geq_{lex} \mu(u)|_{\pi(v)}$ and $\pi(v)$ is even
  - $\mu(v)|_{\pi(v)} >_{lex} \mu(u)|_{\pi(v)}$ and $\pi(v)$ is odd

- Vertex v is progressive if
  - $v \in V_{Even}$ and at least one outedge is progressive
  - $v \in V_{Odd}$ and all outedges are progressive

μ is a progress measure if all vertices are progressive

THE SMALL PROGRESS MEASURE THEOREM

<u>Thm</u> TFAE

(1) There is a parity keyboard decomposition of V

(2) There is a small progress measure

$$\mu : V \to \underbrace{\boxed{\{0,1,2,\dots,n\}^{d/2}}}_{J_{n,d/2}}$$

(3) There is a positional □-dominion strategy on V

THE LIFTING ALGORITHM

1. **Let** $\mu(v) := (0, 0, \ldots, 0) \in \mathcal{J}^{\top}_{n, d/2}$ **for all** $v \in V$

2. **While there is** a vertex v that is **not progressive**

   do let $\mu := \mathrm{MinLift}_v(\mu)$

3. **Return**

   (a) the progressive vertices (winning positions)

   (b) the progressive edges (positional winning strategy)

COMPLEXITY OF THE LIFTING ALGORITHM

- Time: $O\left(\sum_{v\in V} d\cdot \deg(v)\cdot |J_{n,d/2}|\right) = O(dm\cdot |J_{n,d/2}|)$

$$= n^{\frac{d}{2}+O(1)}$$

- Space: $O(dn)$

SUCCINCT PROGRESS MEASURE SEARCH SPACE ?

$$\mathcal{L} = [V \to L_{n,d/2}]$$

Goal: $\boxed{|L_{n,d/2}| \leqslant n^{\log d + O(1)}}$

SUCCINT ADAPTIVE MULTI-COUNTERS

$$\mathbb{B} = \{\mathbb{0}, \mathbb{1}\}$$

$$L_{n,d/2} = \left\{ (s_{d-1}, s_{d-3}, \dots, s_1) : s_i \in \mathbb{B}^* \text{ and } \sum_{i=1}^{d/2} |s_{2i-1}| \leqslant \lceil \lg n \rceil \right\}$$

<u>Fact</u> $\quad |L_{n,d/2}| \leqslant 2^{\lceil \lg n \rceil \cdot \left(1 + \lceil \lg \frac{d}{2} \rceil\right)} = n^{\lg d + O(1)}$

AN "ADAPTIVE" ORDER ON $\mathbb{B}^*$

For all $b \in \mathbb{B}$ and $\bar{s}, \bar{t} \in \mathbb{B}^*$:

- $\mathbb{0}\bar{s} \prec \bar{\varepsilon}$
- $\bar{\varepsilon} \prec \mathbb{1}\bar{s}$
- $b\bar{s} \prec b\bar{t}$ iff $\bar{s} \prec \bar{t}$

$$\boxed{\mathbb{0} \prec \varepsilon \prec \mathbb{1}}$$

where $\bar{\varepsilon} \in \mathbb{B}^*$ is the empty string

A succinct PROGRESS MEASURE

$$\boxed{\mu : V \to \cancel{\mathbb{N}^{d/2}}\ L_{n,d/2}}$$

- Edge (v,u) is progressive if
  - $\mu(v)|_{\pi(v)} \ \cancel{\geqslant_{lex}}\ \succeq_{lex}\ \mu(u)|_{\pi(v)}$ and $\pi(v)$ is even
  - $\mu(v)|_{\pi(v)} \ \cancel{>_{lex}}\ \succ_{lex}\ \mu(u)|_{\pi(v)}$ and $\pi(v)$ is odd

- Vertex v is progressive if
  - $v \in V_{Even}$ and at least one outedge is progressive
  - $v \in V_{Odd}$ and all outedges are progressive

μ is a succinct progress measure if all vertices are progressive

THE SUCCINCT PROGRESS MEASURE THEOREM

<u>Thm</u> TFAE

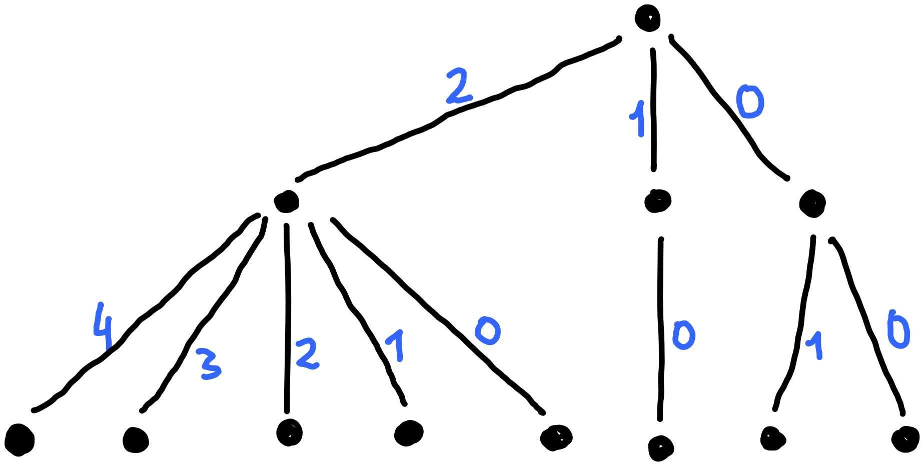
(1) There is a (small) progress measure

$$\mu : V \to \mathbb{J}_{n,d/2}$$

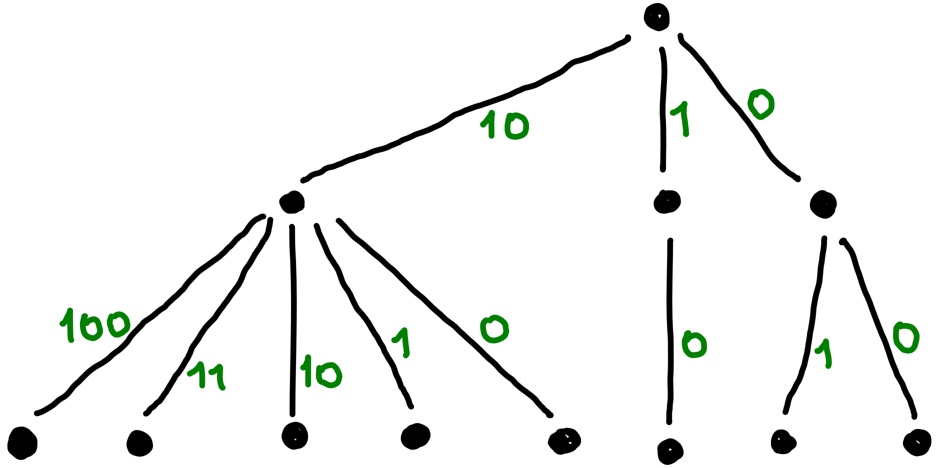
(2) There is a succinct progress measure

$$\kappa : V \to \mathbb{L}_{n,d/2}$$

ORDERED TREE CODING

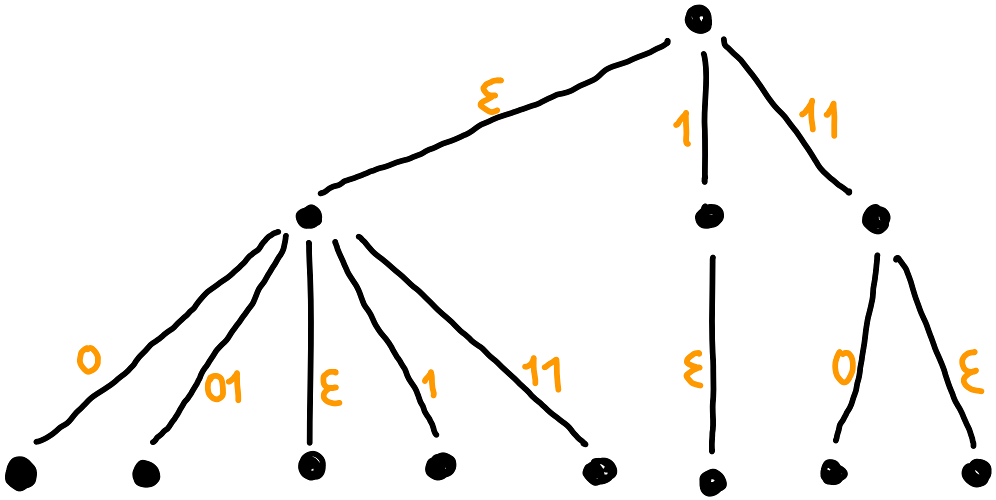


$(\mathbb{N}^2, \leqslant_{lex})$



$((\mathbb{B}^*)^2, \leqslant_{lex})$

binary numbers



$((\mathbb{B}^*)^2, \preceq_{lex})$

binary strings

THE succinct LIFTING ALGORITHM

$\bot \in L_{n,d/2}$

1. **Let** $\mu(v) := \cancel{(0,0,\ldots,0) \in \mathbb{J}}^{\top}_{n,d/2}$ **for all** $v \in V$

2. **While** **there is** a vertex v that is **not progressive**

   do **let** $\mu := \cancel{\mathrm{MinLift}}_v(\mu)$

   MinLift in $\mathcal{L}$

3. **Return**

   (a) the progressive vertices (winning positions)

   (b) the progressive edges (positional winning strategy)

COMPLEXITY OF THE SUCCINCT LIFTING ALGORITHM

- Time: $O\left(\sum_{v\in V} \overset{\log d\cdot\log n}{\cancel{d}}\cdot \deg(v)\cdot \left|\overset{L_{n,d/2}}{J_{n,d/2}}\right|\right) = \tilde{O}\left(m\cdot |L_{n,d/2}|\right)$

$$= \boxed{n^{\log d + O(1)}}$$

- Space: $O\left(\overset{\log d\cdot\log n}{\cancel{d}}\, n\right) = \boxed{\tilde{O}(n)}$

THE SIZE OF $L_{n,d/2}$

- $\left|L_{n,d/2}\right| \leqslant 2^{\lceil \lg n \rceil} \cdot \binom{\lceil \lg n \rceil + \frac{d}{2}}{\frac{d}{2}}$

- $$\left|L_{n,d/2}\right| = \begin{cases} O\left(n \cdot \lg^{d/2} n\right) & \text{if } d = O(1) \\ O\left(n^{1+o(1)}\right) & \text{if } d = o(\log n) \\ \tilde{\Theta}\left(n^{\lg(\delta+1) + \lg(e_\delta) + 1}\right) & \text{if } d = 2\lceil \delta \cdot \lg n \rceil \\ O\left(d \cdot n^{\lg\left(\frac{d}{\lg n}\right) + 1.45}\right) & \text{if } d = \omega(\log n) \end{cases}$$

where $e_\delta = \left(1 + \frac{1}{\delta}\right)^\delta$

RUNNING TIMES

d	CJKLS'17	JL'17	GI-J'17	FJSSW'17
$O(1)$		$O(m\cdot n\cdot \lg^{\frac{d}{2}+1} n)$		$O(m\cdot n\cdot \lg^{d-1} n)$
$o(\log n)$		$O(m\cdot n^{1+o(1)})$		
$2\lceil \delta\cdot \lg n\rceil$		$\tilde{O}(m\cdot n^{\lg(\delta+1)+\lg(e_\delta)+1})$		
$\lceil \lg n\rceil$	$O(n^5)$	$O(m\cdot n^{2.38})$	$O(m\cdot n^{2.55})$	
$\omega(\log n)$	$O(n^{\lg d+6})$	$O(d\cdot m\cdot n^{\lg(\frac{d}{\lg n})+1.45})$		
$\Omega(\log^2 n)$			$O(d\cdot m\cdot n^{\lg(\frac{d}{\lg n})+1.45})$	$O(d\cdot m\cdot n^{\lg(\frac{d}{\lg n})+1.45})$