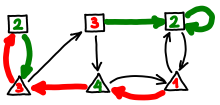
Algorithms for solving Infinite Games on Graphs

Marcin Jurdziński

University of Warwick



Why play games?

Set theory

Logic

Automata theory

Computational complexity

Game Theory

Verification

Controller synthesis

Algorithms

AI

Operations research

Economics

Why play games?

Vol B

Set theory

Logic

Automata theory

Computational complexity

Game Theory

Verification

Controller synthesis

Algorithms

AI

Operations research

Economics

Vol A

How to win infinite games?

Divide and conquer

Dynamic programming

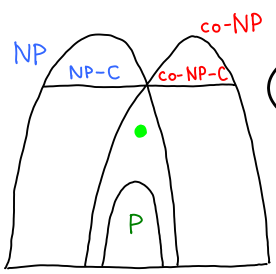
Algorithms

Mathematical programming

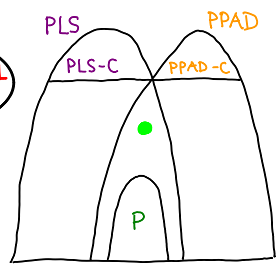
Optimization

Local search

How difficult is it to find a winning strategy?



Computational complexity



Computational complexity of finding equilibria

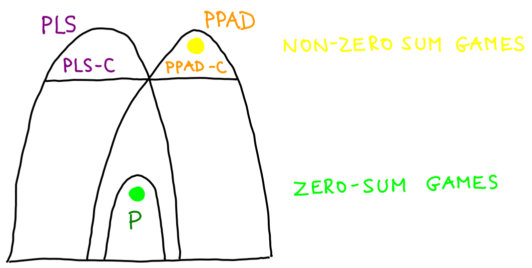
Thm [1950's]

Finding equilibria in zero-sum games (in s.f.) is poly-time equivalent to linear programming

Thm [DGP, CD 2006]

Finding equilibria in non-zero sum games (in s.f.) is poly-time equivalent to computing Brouwer fixed points

Computational complexity of finding equilibria



Plan

I Qualitative (ω-regular) games

1. Motivating example
2. Games on graphs
3. Reachability / safety games
4. Büchi / co-Büchi games
5. Parity games
6. ω-regular games
7. Two recent complexity improvements for parity games

Self-imposed limitations

- Finite graphs
- ω-Regular objectives
- Zero-sum
- Perfect-information
- Non-stochastic

Propositional formula evaluation

State s:

p	0
q	1
r	1

Property φ: $(r \vee p) \wedge (\neg p \wedge (q \vee \neg r))$

$$s \overset{?}{\models} \varphi$$

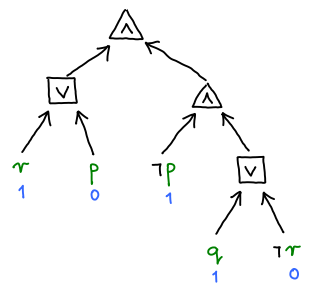
Propositional formula evaluation

State s:

p	0
q	1
r	1

Property φ: $(r \vee p) \wedge (\neg p \wedge (q \vee \neg r))$

$C_{s,\varphi}$:



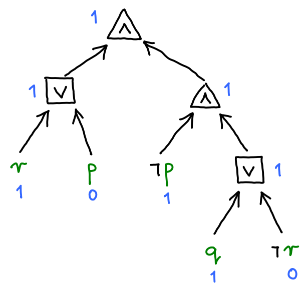
Propositional formula evaluation

State s:

p	0
q	1
r	1

Property φ: $(r \vee p) \wedge (\neg p \wedge (q \vee \neg r))$

$C_{s,\varphi}$:



Fact

$s \models \varphi$

iff

the value of circuit $C_{s,\varphi}$ is 1

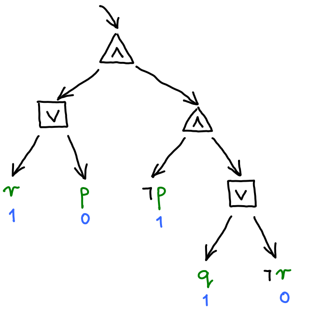
Propositional formula evaluation: Game

State s:

p	0
q	1
r	1

Property φ: $(r \vee p) \wedge (\neg p \wedge (q \vee \neg r))$

$\mathcal{G}_{s,\varphi}$:



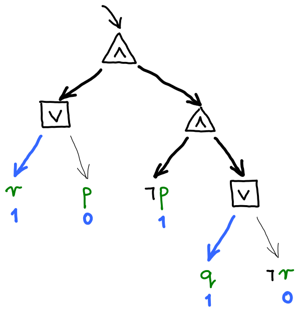
Propositional formula evaluation: Game

State s:

p	0
q	1
r	1

Property φ: $(r \vee p) \wedge (\neg p \wedge (q \vee \neg r))$

$\mathcal{G}_{s,\varphi}$:



Fact

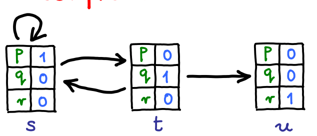
$s \models \varphi$

iff

$\boxed{\vee}$ has a strategy to reach 1 in $\mathcal{G}_{s,\varphi}$

Temporal formula evaluation

Kripke structure:



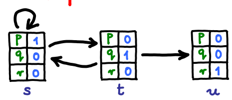
CTL property φ:

$E\big((p \vee q)\, U\, r\big)$

$s \overset{?}{\models} \varphi$

Temporal formula evaluation: Game

Kripke structure:



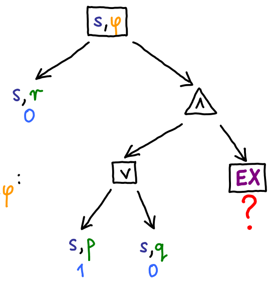
CTL property φ:

$\boxed{E\big((p \vee q)\, U\, r\big)}$

$\equiv$

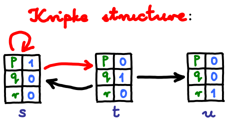
$r \vee \big((p \vee q) \wedge EX\,\varphi\big)$

$\mathcal{G}_{s,\varphi}$:



Temporal formula evaluation: Game

Kripke structure:



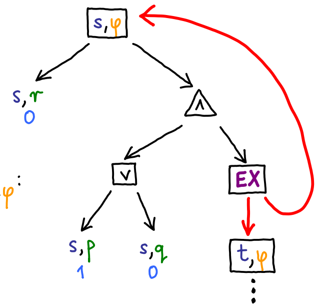
CTL property φ:

$\boxed{E\big((p \vee q)\, U\, r\big)}$

$\equiv$

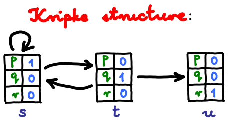
$r \vee \big((p \vee q) \wedge EX\,\varphi\big)$

$\mathcal{G}_{s,\varphi}$:



Temporal formula evaluation: Game

Kripke structure:



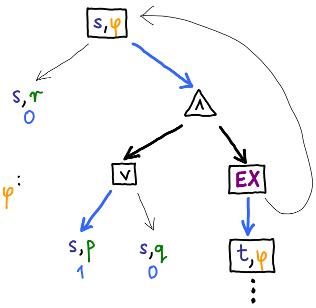
CTL property φ:

$\boxed{E\big((p \vee q)\, U\, r\big)}$

$\equiv$

$r \vee \big((p \vee q) \wedge EX\,\varphi\big)$

$\mathcal{G}_{s,\varphi}$:



Thm

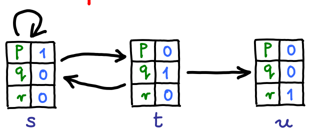
$s \models \varphi$

iff

$\boxed{\vee}$ has a strategy to reach 1 in $\mathcal{G}_{s,\varphi}$

FIXPOINT FORMULA EVALUATION

Kripke structure:



μ-calculus property φ:

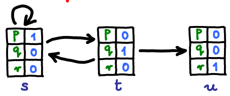
$$\boxed{\mu Z.\,(r \vee ((p \vee q) \wedge EX\, Z))}$$

$$\equiv\; r \vee ((p \vee q) \wedge EX\, \varphi)$$

$$\boxed{s \overset{?}{\models} \varphi}$$

FIXPOINT FORMULA EVALUATION

Kripke structure:

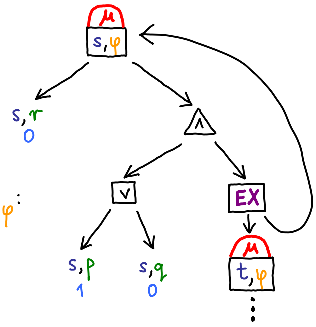


μ-calculus property φ:

$$\boxed{\mu Z.\,(r \vee ((p \vee q) \wedge EX\, Z))}$$

$$\equiv\; r \vee ((p \vee q) \wedge EX\, \varphi)$$

$\mathcal{G}_{s,\varphi}$:



THM

$s \models \varphi$

iff

$\boxed{\vee}$ has a **strategy** in $\mathcal{G}_{s,\varphi}$

- to reach 1, or
- outermost infinitely occurring fixpoint subformula is **not** a μ

PLAN

I Qualitative (ω-regular) games

1. Motivating example
2. **Games on graphs**
3. Reachability / safety games
4. Büchi / co-Büchi games
5. Parity games
6. ω - regular games
7. Two recent complexity improvements for parity games

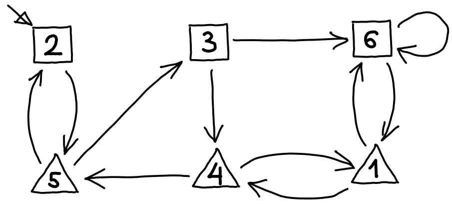
GAMES ON GRAPHS: GAME GRAPH

Players:

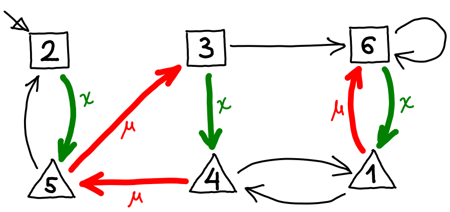
□

△

$n = |V|$ vertices, $m = |E|$ edges



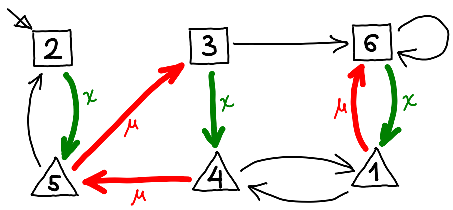
GAMES ON GRAPHS: POSITIONAL STRATEGIES



Positional **strategy** for □: $\boxed{\chi : V_\square \to V}$

Positional **strategy** for △: $\boxed{\mu : V_\triangle \to V}$

GAMES ON GRAPHS: PLAYS



Play $(2, \chi, \mu) = (2,5,3,4,5,3,4,5,\ldots) \in V^\omega$

Games on graphs: objectives

Objectives: $W_\square \subseteq V^\omega$

$W_\Delta = V^\omega \setminus W_\square$

Examples of objectives:

- **Reachability**: $W_\square = \{(v_0, v_1, v_2, \dots) : v_i \in T \text{ for some } i \in \mathbb{N}\}$
- **Safety**: $W_\square = \{(v_0, v_1, v_2, \dots) : v_i \in T \text{ for no } i \in \mathbb{N}\}$

where $T \subseteq V$

Games on graphs: winning strategies

$\chi \in \Sigma_\square$ is a **winning strategy** for $\square$ from $v \in V$,
if $\boxed{\mathrm{Play}(v, \chi, \mu) \in W_\square, \text{ for all } \mu \in \Sigma_\Delta}$

$\mu \in \Sigma_\Delta$ is a **winning strategy** for Δ from $v \in V$,
if $\boxed{\mathrm{Play}(v, \chi, \mu) \in W_\Delta, \text{ for all } \chi \in \Sigma_\square}$

Games on graphs: algorithmic problem

Given: game graph $(V = V_\square \uplus V_\Delta, E)$
objective $W_\square \subseteq V^\omega$
starting vertex $v \in V$

answer: if $\square$ has a winning strategy from v

Games on graphs: determinacy

A game $(V = V_\square \uplus V_\Delta, E; W_\square)$ is **determined**
if for every starting vertex $v \in V$

- $\square$ has a winning strategy from v, **or**
- Δ has a winning strategy from v

Martin's determinacy theorem

Thm [Martin 1975]
Every game $(V = V_\square \uplus V_\Delta, E; W_\square)$,
such that $W_\square \subseteq V^\omega$ is a Borel set,
is determined

Games on graphs: algorithmic problem

Given: game graph $(V = V_\square \uplus V_\Delta, E)$
objective $W_\square \subseteq V^\omega$

compute: $V = W_\square \uplus W_\Delta$ such that

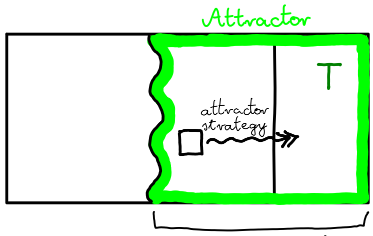
- $\square$ has a winning strategy from $W_\square$, and
- Δ has a winning strategy from W_Δ

PLAN

I Qualitative (ω-regular) games

1. Motivating example
2. Games on graphs
3. **Reachability / safety games**
4. Büchi / co-Büchi games
5. Parity games
6. ω-regular games
7. Two recent complexity improvements for parity games

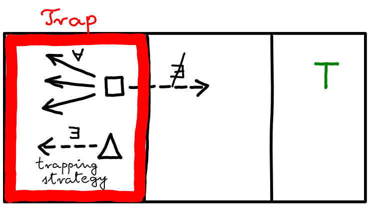
REACHABILITY GAMES



$W_\square = attr_\square(T)$

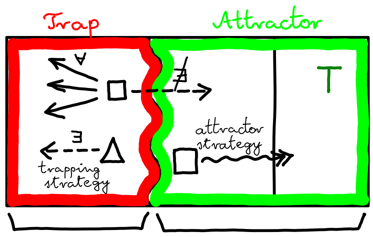
- **Reachability**: $W_\square = \{(v_0, v_1, v_2, \dots) : v_i \in T \text{ for some } i \in \mathbb{N}\}$

SAFETY GAMES



- **Reachability**: $W_\square = \{(v_0, v_1, v_2, \dots) : v_i \in T \text{ for some } i \in \mathbb{N}\}$
- **Safety**: $W_\Delta = \{(v_0, v_1, v_2, \dots) : v_i \in T \text{ for no } i \in \mathbb{N}\}$

REACHABILITY/SAFETY GAMES

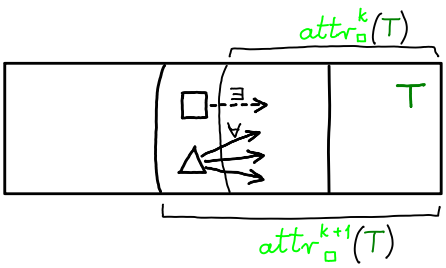


$W_\Delta = V \setminus attr_\square(T) \qquad W_\square = attr_\square(T)$

FACT

Reachability/safety games are **positionally determined**

SOLVING REACHABILITY/SAFETY GAMES: COMPUTING ATTRACTORS



FACT $attr_\square(T) = \bigcup_{k \geq 0} attr_\square^k(T)$

can be computed in $O(m)$ time

PLAN

I Qualitative (ω-regular) games

1. Motivating example
2. Games on graphs
3. Reachability / safety games
4. **Büchi / co-Büchi games**
5. Parity games
6. ω-regular games
7. Two recent complexity improvements for parity games

Games on graphs: (co-)Büchi objectives

- **Büchi**:

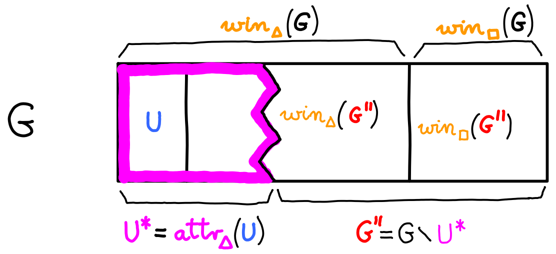
$$W_\square = \{(v_0, v_1, v_2, \ldots) : v_i \in T \text{ for } \textbf{infinitely many } i \in \mathbb{N}\}$$

- **co-Büchi**:

$$W_\Delta = \{(v_0, v_1, v_2, \ldots) : v_i \in T \text{ for } \textbf{finitely many } i \in \mathbb{N}\}$$

where $T \subseteq V$

Oracle Lemma

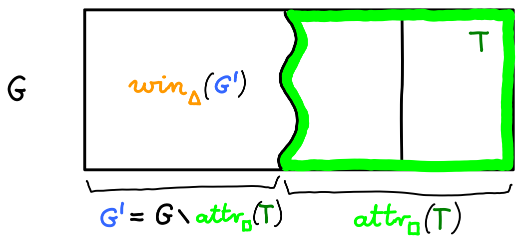


Lemma If $\boxed{U \subseteq win_\Delta(G)}$

then $win_\Delta(G) = U^* \cup win_\Delta(G'')$

$win_\square(G) = win_\square(G'')$

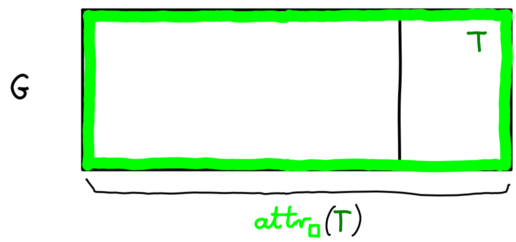
T-attractor Lemma



Lemma

1. $win_\Delta(G') = V \setminus attr_\square(T) \subseteq win_\Delta(G)$
2. If $G' = \emptyset$ then $win_\square(G) = V$

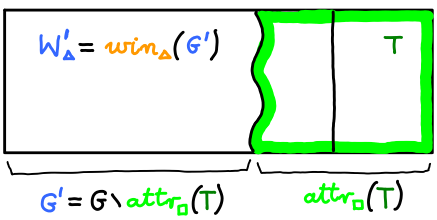
T-attractor Lemma



Lemma

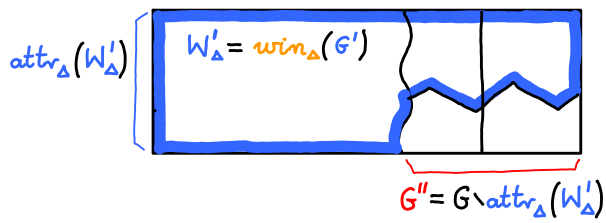
1. $win_\Delta(G') = V \setminus attr_\square(T) \subseteq win_\Delta(G)$
2. If $G' = \emptyset$ then $win_\square(G) = V$

A divide-and-conquer Algorithm



Win(G): if $W'_\Delta = \emptyset$ then $(W_\square, W_\Delta) := (V, \emptyset)$
else $(W''_\square, W''_\Delta) := \text{WIN}(G'')$
$(W_\square, W_\Delta) := (W''_\square, attr_\Delta(W'_\Delta) \cup W''_\Delta)$
return $(W_\square, W_\Delta)$

A divide-and-conquer Algorithm



Win(G): if $W'_\Delta = \emptyset$ then $(W_\square, W_\Delta) := (V, \emptyset)$
else $(W''_\square, W''_\Delta) := \text{WIN}(G'')$
$(W_\square, W_\Delta) := (W''_\square, attr_\Delta(W'_\Delta) \cup W''_\Delta)$
return $(W_\square, W_\Delta)$

Positional determinacy of (co-Büchi) games

Lemma

Büchi / co-Büchi games are **positionally determined**

(both on finite and **infinite graphs**)

Running time of the divide-and-conquer algorithm

$T(n)$

WIN(G):

if $W'_\triangle = \emptyset$ then $(W_\square, W_\triangle) := (V, \emptyset)$

else $(W''_\square, W''_\triangle) := \text{WIN}(G'')$

$(W_\square, W_\triangle) := \left(W''_\square, \text{attr}_\triangle(W'_\triangle) \cup W''_\triangle\right)$

return $(W_\square, W_\triangle)$

$T(n-1)$

Recurrence: $T(n) \leqslant T(n-1) + O(m)$

Solution: $T(n) = O(n \cdot m)$

Plan

I Qualitative (ω-regular) games

1. Motivating example
2. Games on graphs
3. Reachability / safety games
4. Büchi / co-Büchi games
5. **Parity games**
6. ω-regular games
7. Two recent complexity improvements for parity games

Games on graphs: parity objectives

$\text{Inf}(a_0, a_1, a_2, \dots) = \{ a : a_i = a \text{ for } \textbf{infinitely many } i \in \mathbb{N} \}$

Parity objective:

$W_\square = \{ (v_0, v_1, v_2, \dots) : \max\left(\text{Inf}(p(v_0), p(v_1), p(v_2), \dots)\right) \text{ is even} \}$

where $p : V \to \mathbb{N}$ is the priority function

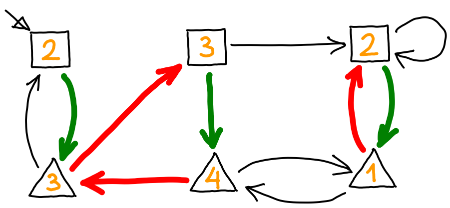
Parity Games

Players:

$\overset{6}{n}$ vertices, $\overset{11}{m}$ edges, $\overset{4}{d}$ priorities

Even = $\square$ = 0

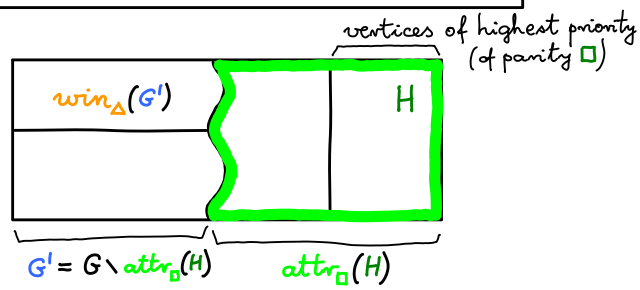
Odd = $\triangle$ = 1



Winner of an infinite play:

parity of the **highest priority** occurring **infinitely often**

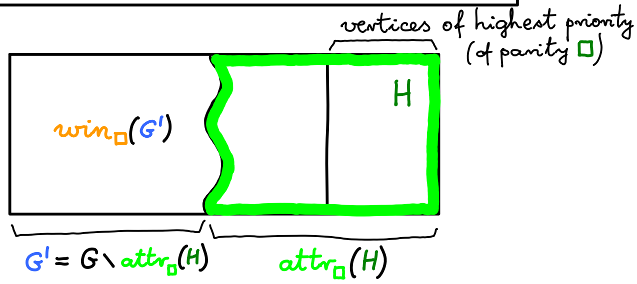
Highest-Priority Attractor Lemma



Lemma

1. If $\text{win}_\triangle(G') \neq \emptyset$ then $\text{win}_\triangle(G') \subseteq \text{win}_\triangle(G)$

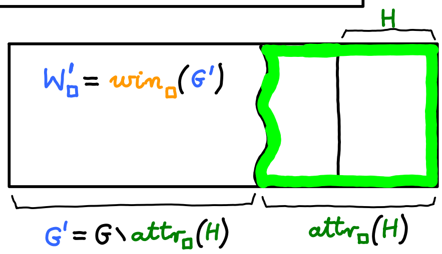
Highest-Priority Attractor Lemma



Lemma

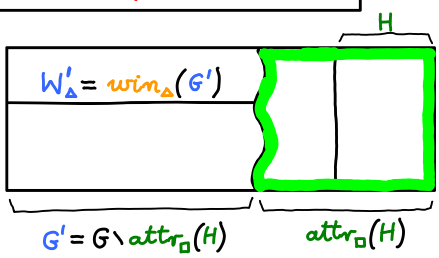
1. If $win_\triangle(G') \neq \emptyset$ then $win_\triangle(G') \subseteq win_\triangle(G)$
2. If $win_\triangle(G') = \emptyset$ then $win_\square(G) = V$

A divide-and-conquer Algorithm



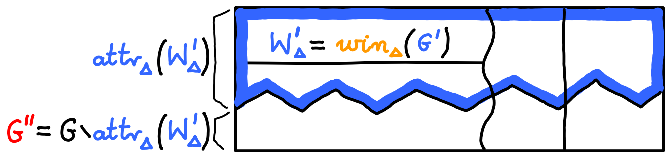
WIN(G): $(W'_\square, W'_\triangle) := \mathrm{WIN}(G')$
  if $W'_\triangle = \emptyset$ then $(W_\square, W_\triangle) := (V, \emptyset)$
    else $(W''_\square, W''_\triangle) := \mathrm{WIN}(G'')$
      $(W_\square, W_\triangle) := (W''_\square, attr_\triangle(W'_\triangle) \cup W''_\triangle)$
  return $(W_\square, W_\triangle)$

A divide-and-conquer Algorithm



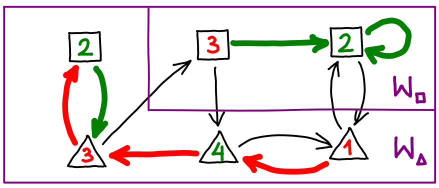
WIN(G): $(W'_\square, W'_\triangle) := \mathrm{WIN}(G')$
  if $W'_\triangle = \emptyset$ then $(W_\square, W_\triangle) := (V, \emptyset)$
    else $(W''_\square, W''_\triangle) := \mathrm{WIN}(G'')$
      $(W_\square, W_\triangle) := (W''_\square, attr_\triangle(W'_\triangle) \cup W''_\triangle)$
  return $(W_\square, W_\triangle)$

A divide-and-conquer Algorithm



WIN(G): $(W'_\square, W'_\triangle) := \mathrm{WIN}(G')$
  if $W'_\triangle = \emptyset$ then $(W_\square, W_\triangle) := (V, \emptyset)$
    else $(W''_\square, W''_\triangle) := \mathrm{WIN}(G'')$
      $(W_\square, W_\triangle) := (W''_\square, attr_\triangle(W'_\triangle) \cup W''_\triangle)$
  return $(W_\square, W_\triangle)$

Positional Determinacy



Thm [Emerson, Jutla; Mostowski 1991]

Parity games are **positionally determined**
(also on **infinite graphs**)

Corollary

(Deciding the winner in parity games is in NP ∩ co-NP)

Running time of the divide-and-conquer algorithm

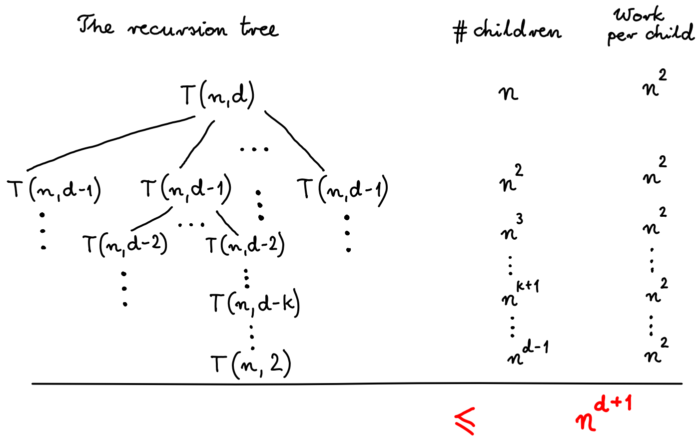
$T(n)$

WIN(G): $(W'_\square, W'_\triangle) := \mathrm{WIN}(G')$ $T(n-1)$
  if $W'_\triangle = \emptyset$ then $(W_\square, W_\triangle) := (V, \emptyset)$
    else $(W''_\square, W''_\triangle) := \mathrm{WIN}(G'')$ $T(n-1)$
      $(W_\square, W_\triangle) := (W''_\square, attr_\triangle(W'_\triangle) \cup W''_\triangle)$
  return $(W_\square, W_\triangle)$

Recurrence: $T(n) \leqslant 2 \cdot T(n-1) + O(n^2)$

Solution: $T(n) = O(2^n)$

Running time of the divide-and-conquer algorithm



Plan

I Qualitative (ω-regular) games

1. Motivating example
2. Games on graphs
3. Reachability / safety games
4. Büchi / co-Büchi games
5. Parity games
6. ω-regular games
7. Two recent complexity improvements for parity games

Games on graphs: ω-regular objectives

- **Muller** objectives:
  $W_\square = \{(v_0,v_1,v_2,\dots) : \mathrm{Inf}(\bar v) \in \mathcal{F}\}$ where $\mathcal{F} \subseteq 2^V$

- **Rabin** objectives:
  $W_\square = \{(v_0,v_1,v_2,\dots) :$ for **some** $j = 0,1,\dots,k-1$, we have $\mathrm{Inf}(\bar v) \cap G_j \neq \emptyset$ **and** $\mathrm{Inf}(\bar v) \cap R_j = \emptyset\}$

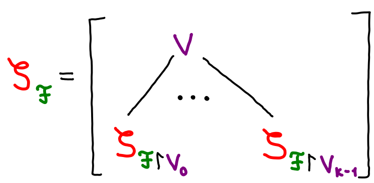
- **Streett** objectives:
  $W_\square = \{(v_0,v_1,v_2,\dots) :$ for **all** $j = 0,1,\dots,k-1$, we have $\mathrm{Inf}(\bar v) \cap G_j \neq \emptyset$ **implies** $\mathrm{Inf}(\bar v) \cap R_j \neq \emptyset\}$

where $G_0, R_0, G_1, R_1, \dots, G_{k-1}, R_{k-1} \subseteq V$

Split tree of a Muller objective

- If $\mathcal{F} = \emptyset$ or $\mathcal{F} = 2^V$ then $\mathcal{S}_\mathcal{F} = [V]$
- If $\emptyset \subsetneq \mathcal{F} \subsetneq 2^V$ and
  if $V_0, V_1, \dots, V_{k-1}$ are all **maximal** subsets of V, such that $V_j \in \mathcal{F}$ iff $V \notin \mathcal{F}$

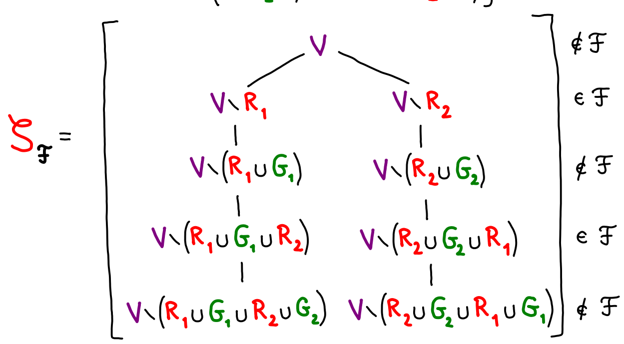
then



Split tree of a Rabin objective

Two **Rabin** pairs: $(G_1,R_1), (G_2,R_2)$

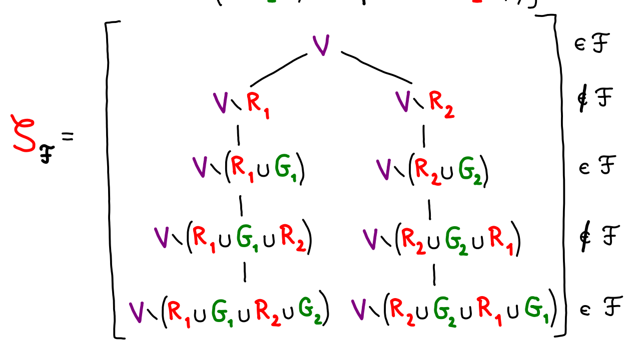
$\mathcal{F} = \{U \subseteq V : (U \cap G_1 \neq \emptyset$ **and** $U \cap R_1 = \emptyset)$ **or** $(U \cap G_2 \neq \emptyset$ **and** $U \cap R_2 = \emptyset)\}$



Split tree of a Streett objective

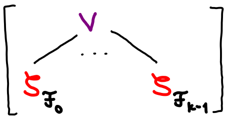
Two **Streett** pairs: $(G_1,R_1), (G_2,R_2)$

$\mathcal{F} = \{U \subseteq V : (U \cap G_1 \neq \emptyset$ **implies** $U \cap R_1 \neq \emptyset)$ **and** $(U \cap G_2 \neq \emptyset$ **implies** $U \cap R_2 \neq \emptyset)\}$



Memory number $m_{\mathcal{F}}$

For $\mathcal{Z}_{\mathcal{F}} =$



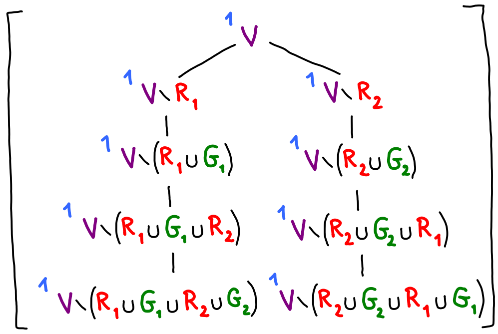
define:

$$m_{\mathcal{F}} = \begin{cases} 1 & \text{if } \mathcal{F} = \emptyset \text{ or } \mathcal{F} = 2^V \\ \max_{i=0}^{k-1} \{ m_{\mathcal{F}_i} \} & \text{if } V \notin \mathcal{F} \\ \sum_{i=0}^{k-1} m_{\mathcal{F}_i} & \text{if } V \in \mathcal{F} \end{cases}$$

Memory number of a Rabin objective

$\mathcal{Z}_{\mathcal{F}} =$

$m_{\mathcal{F}\restriction U}$



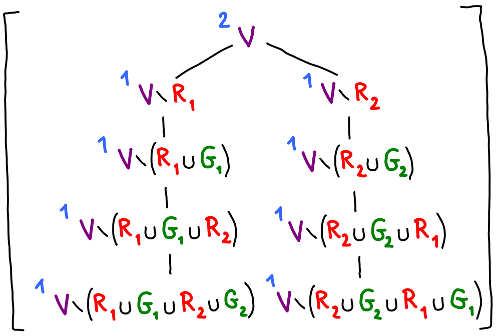
Proposition

If $\mathcal{F} \subseteq 2^V$ is Rabin then $\boxed{m_{\mathcal{F}} = 1}$

Memory number of a Streett objective

$\mathcal{Z}_{\mathcal{F}} =$

$m_{\mathcal{F}\restriction U}$



Proposition

If $\mathcal{F} \subseteq 2^V$ is Streett with k pairs then $\boxed{m_{\mathcal{F}} \leqslant k!}$

Strategies with memory

- Positional strategies:

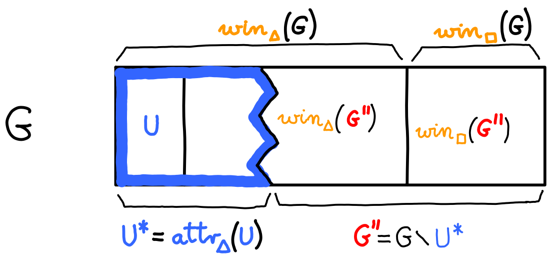
$$\chi : V_\square \longrightarrow V$$

- Strategies with **memory**:

$$\chi : M \times V_\square \longrightarrow M \times V$$

automaton states alphabet output

Oracle Lemma

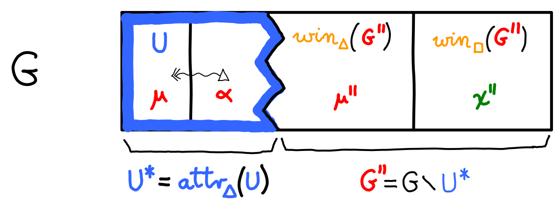


Lemma If $\boxed{U \subseteq win_\Delta(G)}$

then $win_\Delta(G) = U^* \cup win_\Delta(G'')$

$win_\square(G) = win_\square(G'')$

Oracle lemma with memory



Lemma

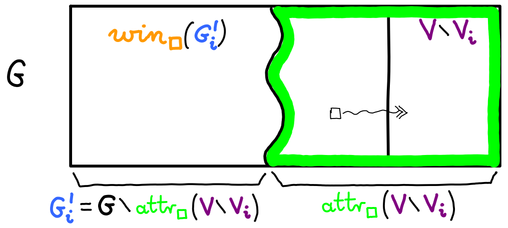
If μ is winning for Δ from U

μ'' is winning for Δ from $win_\Delta(G'')$

then $(\mu \cup \alpha \cup \mu'')$ is winning for Δ from $(U^* \cup win_\Delta(G''))$

$$|Memory(\mu \cup \alpha \cup \mu'')| \leqslant \max\{Memory(\mu), Memory(\mu'')\}$$

Muller games: $V\setminus V_i$ attractors lemma

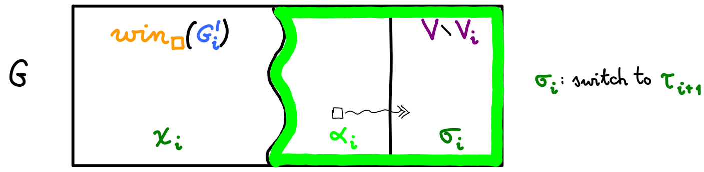


Lemma

If $win_\triangle(G_i^!) = \emptyset$ for all $i = 0, 1, \dots, k-1$

then $win_\square(G) = V$

Muller games: $V\setminus V_i$ attractors lemma



Lemma

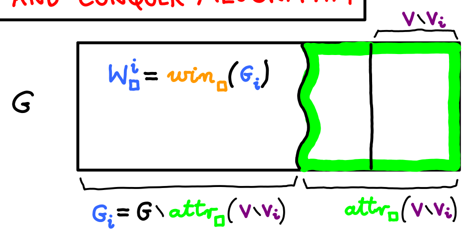
If χ_i is winning for $\square$ in $G_i^!$ for all $i = 0, 1, \dots, k-1$

$\tau_i = \chi_i \cup \alpha_i \cup \sigma_i$

then $\boxed{\tau = \tau_0 \oplus \tau_1 \oplus \cdots \oplus \tau_{k-1}}$ is winning for $\square$ from V

$$|Memory(\tau)| = \sum_{i=0}^{k-1} |Memory(\tau_i)| \leq \sum_{i=0}^{k-1} m_{\mathcal{F}_i} = m_{\mathcal{F}}$$

A divide-and-conquer Algorithm



$\underline{\text{Win}}(G)$: $(W_\square^i, W_\triangle^i) := \underline{\text{Win}}(G_i)$ for all $i = 0, 1, \dots, k-1$

if $W_\triangle^i = \emptyset$ for all $i = 0, 1, \dots, k-1$

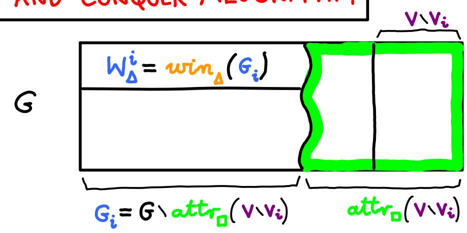
then $(W_\square, W_\triangle) := (V, \emptyset)$

else $(W_\square^{"}, W_\triangle^{"}) := \text{Win}(G_i^{"})$ s.t. $W_\triangle^i \neq \emptyset$

$(W_\square, W_\triangle) := (W_\square^{"}, attr_\triangle(W_\triangle^i) \cup W_\triangle^{"})$

return $(W_\square, W_\triangle)$

A divide-and-conquer Algorithm



$\underline{\text{Win}}(G)$: $(W_\square^i, W_\triangle^i) := \underline{\text{Win}}(G_i)$ for all $i = 0, 1, \dots, k-1$

if $W_\triangle^i = \emptyset$ for all $i = 0, 1, \dots, k-1$

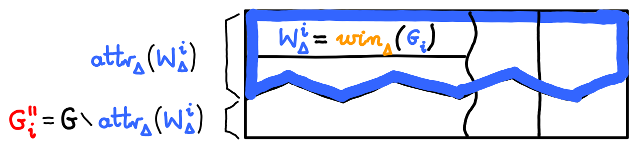
then $(W_\square, W_\triangle) := (V, \emptyset)$

else $(W_\square^{"}, W_\triangle^{"}) := \underline{\text{Win}}(G_i^{"})$ s.t. $W_\triangle^i \neq \emptyset$

$(W_\square, W_\triangle) := (W_\square^{"}, attr_\triangle(W_\triangle^i) \cup W_\triangle^{"})$

return $(W_\square, W_\triangle)$

A divide-and-conquer Algorithm



$\underline{\text{Win}}(G)$: $(W_\square^i, W_\triangle^i) := \underline{\text{Win}}(G_i)$ for all $i = 0, 1, \dots, k-1$

if $W_\triangle^i = \emptyset$ for all $i = 0, 1, \dots, k-1$

then $(W_\square, W_\triangle) := (V, \emptyset)$

else $(W_\square^{"}, W_\triangle^{"}) := \underline{\text{Win}}(G_i^{"})$ s.t. $W_\triangle^i \neq \emptyset$

$(W_\square, W_\triangle) := (W_\square^{"}, attr_\triangle(W_\triangle^i) \cup W_\triangle^{"})$

return $(W_\square, W_\triangle)$

Forgetful determinacy of ω-regular games

Thm [Gurevich, Harrington 1982; Zielonka 1998]

Muller games are **forgetfully determined**:

- $\square$ has a winning strategy with **memory** $m_{\mathcal{F}}$
- $\triangle$ has a winning strategy with **memory** $m_{\overline{\mathcal{F}}}$

Corollary [Klarlund 1991]

Player $\square$ has a **positional** winning strategy in **Rabin games**

PLAN

I Qualitative (ω-regular) games

1. Motivating example
2. Games on graphs
3. Reachability / safety games
4. Büchi / co-Büchi games
5. Parity games
6. ω-regular games
7. Two recent complexity improvements for parity games

TWO RECENT IMPROVEMENTS

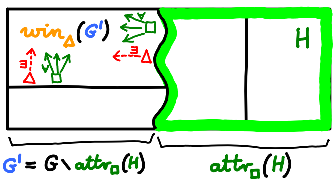
Many priorities $d = \Omega(n^{\frac{1}{2}+\varepsilon})$		Few priorities $d = O(n^{1/2})$	
Randomized	Deterministic		
Expected $n^{O(\sqrt{n})}$	$n^{O(\sqrt{n})}$	$O(n^{d/2})$	$O(n^{d/3})$
old	new	old	new

TWO RECENT IMPROVEMENTS

Many priorities $d = \Omega(n^{\frac{1}{2}+\varepsilon})$		Few priorities $d = O(n^{1/2})$	
Randomized	Deterministic		
Expected $n^{O(\sqrt{n})}$	$n^{O(\sqrt{n})}$	$O(n^{d/2})$	$O(n^{d/3})$
old	new	old	new

DOMINIONS

DEF $D \subseteq V$ is an **dominion** for Δ if Δ has a **trapping** strategy in D that is **winning** for her



FACT $win_\Delta(G')$ is a dominion for Δ in G

FACT A dominion of size $\leq \ell$ can be found in $n^{O(\ell)}$ time

A SUBEXPONENTIAL ALGORITHM

$T(n)$ | WIN(G): If there is a dominion $D \neq \emptyset$ of size $\leq \ell$ — $n^{O(\ell)}$
then return WIN$(G \setminus D) \oplus D$ — $T(n-1)$
else
$(W'_\square, W'_\Delta) :=$ WIN(G') — $T(n-1)$
if $W'_\Delta = \emptyset$ then $(W_\square, W_\Delta) := (V, \emptyset)$
else $(W''_\square, W''_\Delta) :=$ WIN(G'') — $T(n-\ell)$
$(W_\square, W_\Delta) := (W''_\square, attr_\Delta(W'_\Delta) \cup W''_\Delta)$
return $(W_\square, W_\Delta)$

Recurrence: $T(n) \leq n^{O(\ell)} + T(n-1) + T(n-\ell)$

Solution: $\boxed{T(n) = n^{O(\sqrt{n})}}$ if $\ell = \sqrt{n}$

TWO RECENT IMPROVEMENTS

Many priorities $d = \Omega(n^{\frac{1}{2}+\varepsilon})$		Few priorities $d = O(n^{1/2})$	
Randomized	Deterministic		
Expected $n^{O(\sqrt{n})}$	$n^{O(\sqrt{n})}$	$O(n^{d/2})$	$O(n^{d/3})$
old	new	old	new

A SUFFICIENT CONDITION FOR WINNING PLAYS

$$\overset{v_1}{\boxed{1}} \to \overset{v_2}{\boxed{2}} \to \overset{v_3}{\boxed{3}} \to \overset{v_4}{\boxed{2}} \to \overset{v_5}{\boxed{1}} \to \overset{v_6}{\boxed{1}} \to \overset{v_7}{\boxed{2}} \to \overset{v_8}{\boxed{3}} \to \overset{v_9}{\boxed{4}} \cdots$$

μ_3: $2 \geq 2 \geq 2 > 1 \geq 1 \geq 1 \geq 1 \geq 1 > 0$

μ_1: $1 > 0$ $\quad\quad\quad 2 > 1 > 0$

FACT

$v_1, v_2, v_3, \dots$ is winning for $\square$

if

there are $\mu_1, \mu_3, \dots, \mu_{d-1} : V \to \mathbb{N}$, s.t.

- $\mu_k(v_i) \geq \mu_k(v_{i+1})$ for $k \in [d-1, p(v_i))$
- $\mu_k(v_i) > \mu_k(v_{i+1})$ for $k = p(v_i)$

A SUFFICIENT CONDITION FOR WINNING PLAYS

$$\overset{v_1}{\boxed{1}} \to \overset{v_2}{\boxed{2}} \to \overset{v_3}{\boxed{3}} \to \overset{v_4}{\boxed{2}} \to \overset{v_5}{\boxed{1}} \to \overset{v_6}{\boxed{1}} \to \overset{v_7}{\boxed{2}} \to \overset{v_8}{\boxed{3}} \to \overset{v_9}{\boxed{4}} \cdots$$

$$\begin{pmatrix}\mu_3\\ \mu_1\end{pmatrix}: \begin{pmatrix}2\\1\end{pmatrix} >_{lex} \begin{pmatrix}2\\0\end{pmatrix} \geq_{lex} (2) >_{lex} (1) \geq_{lex} \begin{pmatrix}1\\2\end{pmatrix} >_{lex} \begin{pmatrix}1\\1\end{pmatrix} >_{lex} \begin{pmatrix}1\\0\end{pmatrix} \geq_{lex} (1) >_{lex} (0)$$

FACT

$v_1, v_2, v_3, \dots$ is winning for $\square$

if

there are $\mu_1, \mu_3, \dots, \mu_{d-1} : V \to \mathbb{N}$, s.t.

$$\boxed{\begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(v_i) \geq^{lex} \begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(v_{i+1})} \quad \boxed{>^{lex} \text{ if } p(v_i) \text{ is odd!}}$$

CHARACTERIZING EXISTENCE OF WINNING PLAYS

LEMMA

There **is** a winning play (from every vertex) for $\square$

if

there are $\mu_1, \mu_3, \dots, \mu_{d-1} : V \to \mathbb{N}$, s.t.

$$\boxed{\begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(v) \geq^{lex} \min_{(v,w)\in E} \begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(w)} \quad \boxed{>^{lex} \text{ if } p(v) \text{ is odd!}}$$

CHARACTERIZING UNIVERSALITY OF WINNING PLAYS

LEMMA

ALL plays (from every vertex) are winning for $\square$

if

there are $\mu_1, \mu_3, \dots, \mu_{d-1} : V \to \mathbb{N}$, s.t.

$$\boxed{\begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(v) \geq^{lex} \max_{(v,w)\in E} \begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(w)} \quad \boxed{>^{lex} \text{ if } p(v_i) \text{ is odd!}}$$

CHARACTERIZING EXISTENCE OF WINNING STRATEGIES

THEOREM

There is a winning strategy for $\square$ (from every vertex)

iff

there are $\mu_1, \mu_3, \dots, \mu_{d-1} : V \to \mathbb{N}$, s.t.

- $\boxed{\begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(v) \geq^{lex} \min_{(v,w)\in E} \begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(w)} \quad \boxed{>^{lex} \text{ if } p(v) \text{ is odd!}}$ for $v \in V_\square$
- $\boxed{\begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(v) \geq^{lex} \max_{(v,w)\in E} \begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(w)} \quad \boxed{>^{lex} \text{ if } p(v) \text{ is odd!}}$ for $v \in V_\triangle$

SMALL PROGRESS MEASURES

THEOREM

There is a winning strategy for $\square$ (from every vertex)

iff

there are $\mu_1, \mu_3, \dots, \mu_{d-1} : V \to \mathbb{N}$, s.t.

- $\boxed{\begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(v) \geq^{lex} \min_{(v,w)\in E} \begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(w)} \quad \boxed{>^{lex} \text{ if } p(v) \text{ is odd!}}$ for $v \in V_\square$
- $\boxed{\begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(v) \geq^{lex} \max_{(v,w)\in E} \begin{pmatrix}\mu_{d-1}\\ \vdots \\ \mu_{p(v_i)}\end{pmatrix}(w)} \quad \boxed{>^{lex} \text{ if } p(v) \text{ is odd!}}$ for $v \in V_\triangle$

and $\boxed{\mu_k : V \to \{0, 1, 2, \dots, n_k\}}$ where $n_k = |p^{-1}(k)|$

Small progress measures

Lemma

The number of tuples

$$\begin{pmatrix} \mu_{d-1} \\ \vdots \\ \mu_1 \end{pmatrix} \in \mathbb{N}^{d/2}$$

s.t. $\mu_k \in \{0,1,2,\ldots,n_k\}$ where $n_k = |p^{-1}(k)|$

is $\boxed{\leqslant \left(\frac{n}{d/2}\right)^{d/2}}$

Smaller Progress Measures for Dominions

Thm If $D \subseteq V$ is a dominion of size $\leqslant \ell$, then there is a progress measure $\mu : V \to M_G^\infty(\ell)$, s.t. $\mu(D) \subseteq M_G(\ell)$,

where $\boxed{M_G(\ell) = \left\{ \mu \in \mathbb{N}^{d/2} : \sum_{k=1}^{d} \mu_k \leqslant \ell \right\}}$

Cor

1. All dominions of size $\leqslant \ell$ can be found in time $\binom{\ell + \frac{d}{2}}{\frac{d}{2}}$

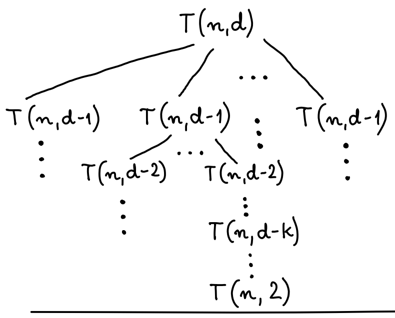
2. All dominions of size $\leqslant n^{2/3}$ can be found in time $n^{d/3}$

Progress Measure Lifting Algorithm

1. Start with $\mu : V \ni v \mapsto (0,0,\ldots,0)$
2. While μ violates the progress inequality at some $v \in V$, lift $\mu(v)$ (minimally) so that it doesn't

Running time: $\leqslant n \cdot \left(\frac{n}{d/2}\right)^{d/2}$

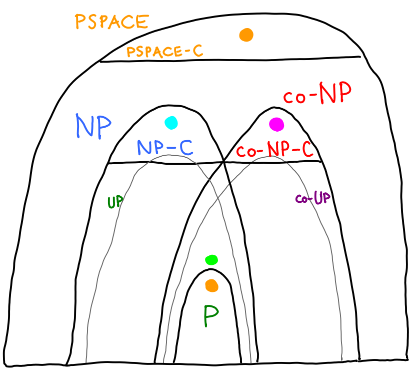
The Recursion Tree for Schewe's Algorithm



	# children	Dominion preprocessing
$T(n,d)$	$\leqslant n^{1/3}$	$n^{d/3}$
$T(n,d-1)$	$\leqslant n^{2/3}$	$n^{(d-1)/3}$
$T(n,d-2)$	$\leqslant n^{3/3}$	$n^{(d-2)/3}$
$\vdots$	$\vdots$	$\vdots$
$T(n,d-k)$	$\leqslant n^{(k+1)/3}$	$n^{(d-k)/3}$
$\vdots$	$\vdots$	$\vdots$
$T(n,2)$	$\leqslant n^{(d-1)/3}$	$n^{2/3}$

$\leqslant n \cdot n^{(d+1)/3}$

Computational complexity of solving infinite games



MULLER GAMES

STREETT GAMES

RABIN GAMES

PARITY GAMES

EXPLICIT MULLER GAMES