Infinite Games

Recap and Outlook

Martin Zimmermann

Saarland University

July 26th, 2016

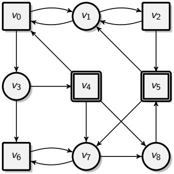
Plan for Today

- Review
- Change Log Lecture Notes
- Exam
  - Organizational Matters
  - Questions
- Outlook: Even More Games

Review

Reachability

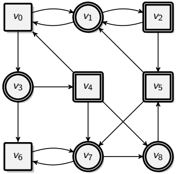
- **Name:** **Reachability Game**
- Format: $(\mathcal{A}, \textsc{Reach}(R))$ with $R \subseteq V$



- Winning condition: $\mathrm{Occ}(\rho) \cap R \neq \emptyset$
- Solution complexity: linear time in $|E|$
- Algorithm: attractor
- Memory requirements for Player 0: uniform positional
- Memory requirements for Player 1: uniform positional
- Dual game: safety

Safety

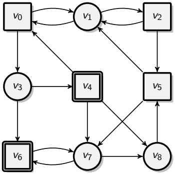
- **Name:** **Safety Game**
- Format: $(\mathcal{A}, \textsc{Safety}(S))$ with $S \subseteq V$



- Winning condition: $\mathrm{Occ}(\rho) \subseteq S$
- Solution complexity: linear time in $|E|$
- Algorithm: dualize + attractor
- Memory requirements for Player 0: uniform positional
- Memory requirements for Player 1: uniform positional
- Dual game: reachability

Büchi

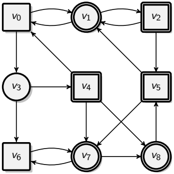
- **Name:** **Büchi Game**
- Format: $(\mathcal{A}, \text{Büchi}(F))$ with $F \subseteq V$



- Winning condition: $\text{Inf}(\rho) \cap F \neq \emptyset$
- Solution complexity: **P**
- Algorithm: iterated attractor
- Memory requirements for Player 0: uniform positional
- Memory requirements for Player 1: uniform positional
- Dual game: co-Büchi

Co-Büchi

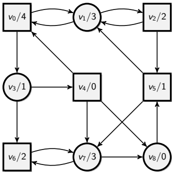
- **Name:** **Co-Büchi Game**
- Format: $(\mathcal{A}, \textsc{coBüchi}(C))$ with $C \subseteq V$



- Winning condition: $\mathrm{Inf}(\rho) \subseteq C$
- Solution complexity: **P**
- Algorithm: dualize + iterated attractor
- Memory requirements for Player 0: uniform positional
- Memory requirements for Player 1: uniform positional
- Dual game: Büchi

Parity

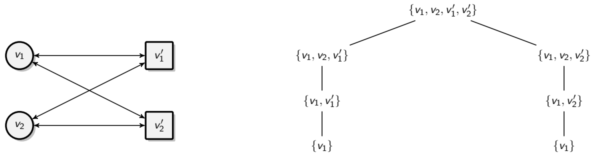
- **Name:** **Parity Game**
- Format: $(\mathcal{A}, \textsc{Parity}(\Omega))$ with $\Omega \colon V \to \mathbb{N}$



- Winning condition: $\min(\mathrm{Inf}(\Omega(\rho)))$ even
- Solution complexity: **NP ∩ co-NP**
- Algorithm: progress measures and many others
- Memory requirements for Player 0: uniform positional
- Memory requirements for Player 1: uniform positional
- Dual game: parity

Muller

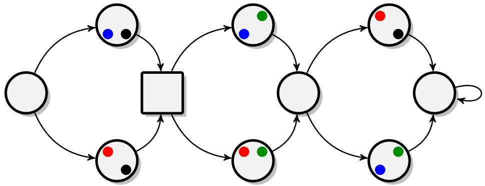
- **Name:** **Muller Game**
- Format: $(\mathcal{A}, \textsc{Muller}(\mathcal{F}))$ with $\mathcal{F} \subseteq 2^V$



- Winning condition: $\mathrm{Inf}(\rho) \in \mathcal{F}$
- Solution complexity: **P**, **NP** $\cap$ **co-NP**, **PSPACE**-complete
- Algorithm: reduction to parity and many others
- Memory requirements for Player 0: $|V|!$
- Memory requirements for Player 1: $|V|!$
- Dual game: Muller

Generalized Reachability

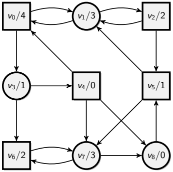
- **Name:** **Generalized Reachability Game**
- Format: $(\mathcal{A}, \textsc{GenReach}(\mathcal{R}))$ with $\mathcal{R} \subseteq 2^V$



- Winning condition: $\forall R \in \mathcal{R}.\, \mathrm{Occ}(\rho) \cap R \neq \emptyset$
- Solution complexity: **PSPACE**-complete
- Algorithm: Simulate for $|V| \cdot |\mathcal{R}|$ steps
- Memory requirements for Player 0: $2^{|\mathcal{R}|}$
- Memory requirements for Player 1: $\binom{|\mathcal{R}|}{\lfloor |\mathcal{R}|/2 \rfloor}$
- Dual game: disjunctive safety

Weak Parity

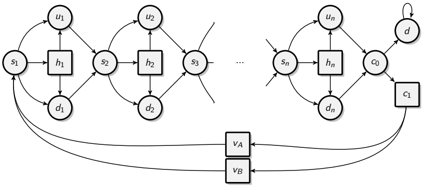
- **Name:** **Weak Parity Game**
- Format: $(\mathcal{A}, \text{wParity}(\Omega))$ with $\Omega \colon V \to \mathbb{N}$



- Winning condition: $\min(\text{Occ}(\Omega(\rho)))$ even
- Solution complexity: **P**
- Algorithm: iterated attractor
- Memory requirements for Player 0: uniform positional
- Memory requirements for Player 1: uniform positional
- Dual game: weak parity

Weak Muller

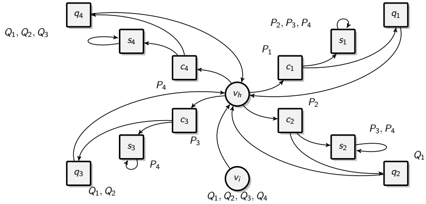
- **Name:** **Weak Muller Game**
- Format: $(\mathcal{A}, \text{wMuller}(\mathcal{F}))$ with $\mathcal{F} \subseteq 2^V$



- Winning condition: $\text{Occ}(\rho) \in \mathcal{F}$
- Solution complexity: **PSPACE**-complete
- Algorithm: reduction to weak parity or direct one
- Memory requirements for Player 0: $2^{|V|}$
- Memory requirements for Player 1: $2^{|V|}$
- Dual game: weak Muller

Request-Response

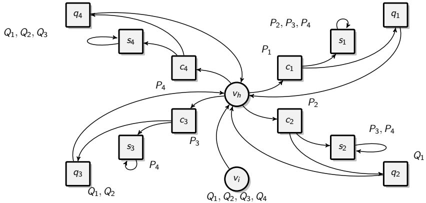
- **Name:** **Request-Response Game**
- Format: $(\mathcal{A}, \textsc{ReqRes}((Q_j, P_j)_{j\in[k]}))$ with $Q_j, P_j \subseteq V$



- Winning condition: $\forall j\, \forall n(\rho_n \in Q_j \to \exists m \geq n.\, \rho_m \in P_j)$
- Solution complexity: **EXPTIME**-complete
- Algorithm: reduction to Büchi
- Memory requirements for Player 0: $k \cdot 2^k$
- Memory requirements for Player 1: 2^k
- Dual game: n/a

Rabin

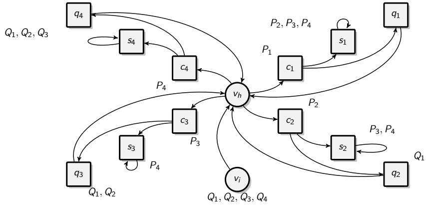
- **Name:** **Rabin Game**
- Format: $(\mathcal{A}, \mathrm{RABIN}((Q_j, P_j)_{j\in[k]}))$ with $Q_j, P_j \subseteq V$



- Winning condition: $\exists j(\mathrm{Inf}(\rho) \cap Q_j \neq \emptyset$ and $\mathrm{Inf}(\rho) \cap P_j = \emptyset)$
- Solution complexity: **NP**-complete
- Algorithm: reduction to parity or direct one
- Memory requirements for Player 0: uniform positional
- Memory requirements for Player 1: $k!$
- Dual game: Streett

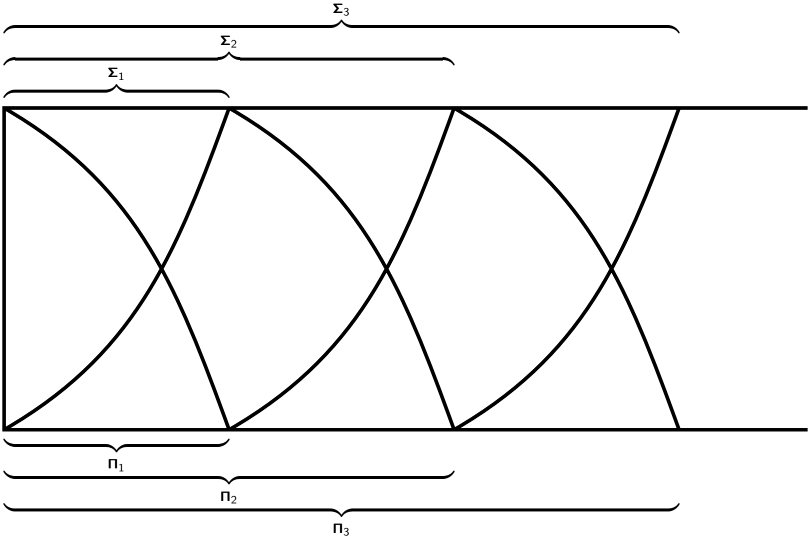
Streett

- **Name:** **Streett Game**
- Format: $(\mathcal{A}, \textsc{Streett}((Q_j, P_j)_{j \in [k]}))$ with $Q_j, P_j \subseteq V$

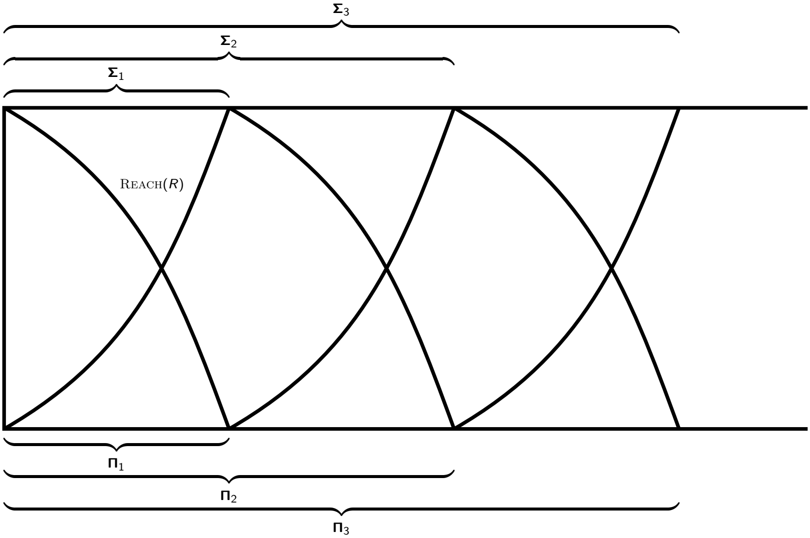


- Winning condition: $\forall j (\mathrm{Inf}(\rho) \cap Q_j \neq \emptyset \Rightarrow \mathrm{Inf}(\rho) \cap P_j \neq \emptyset)$
- Solution complexity: **co-NP**-complete
- Algorithm: reduction to parity or direct one
- Memory requirements for Player 0: $k!$
- Memory requirements for Player 1: uniform positional
- Dual game: Rabin

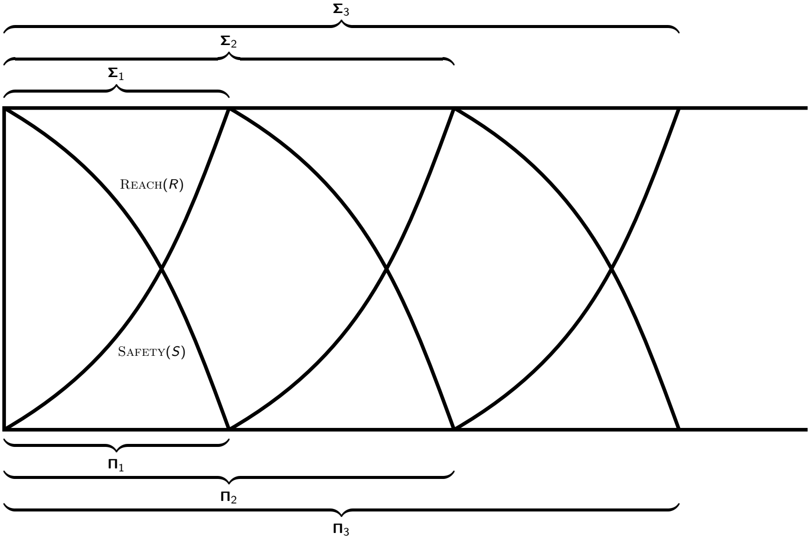
Reducibility



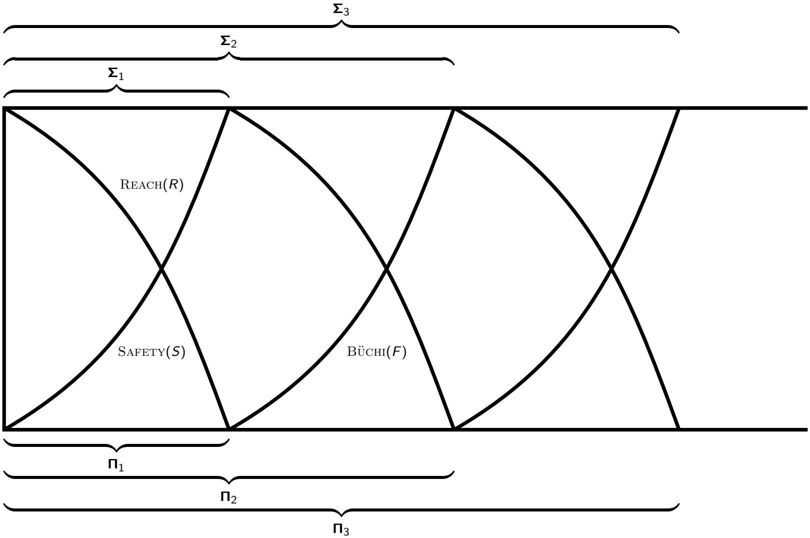
Reducibility



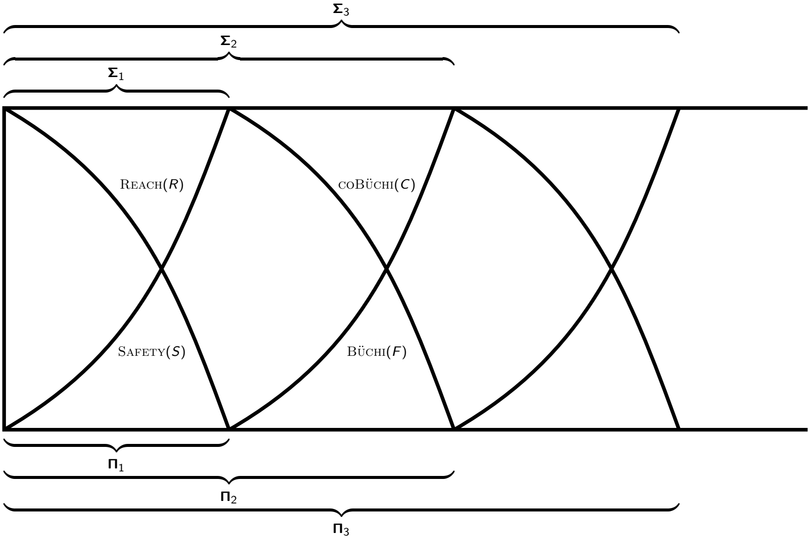
Reducibility



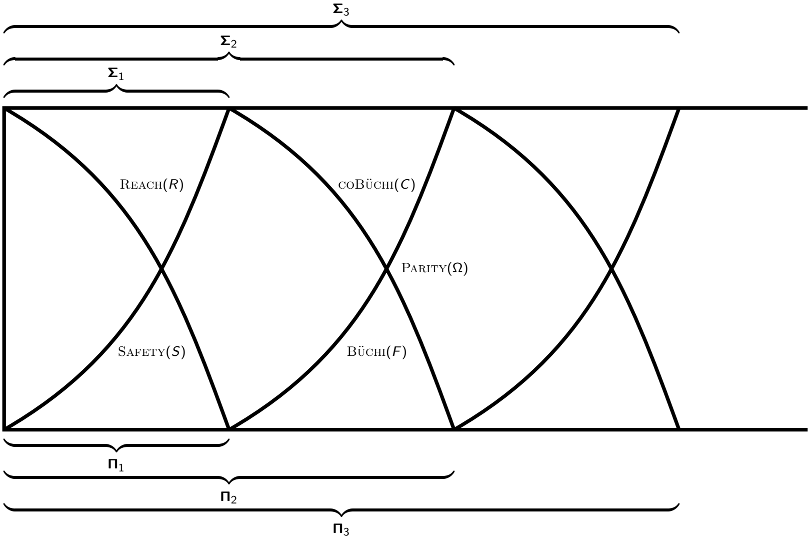
Reducibility



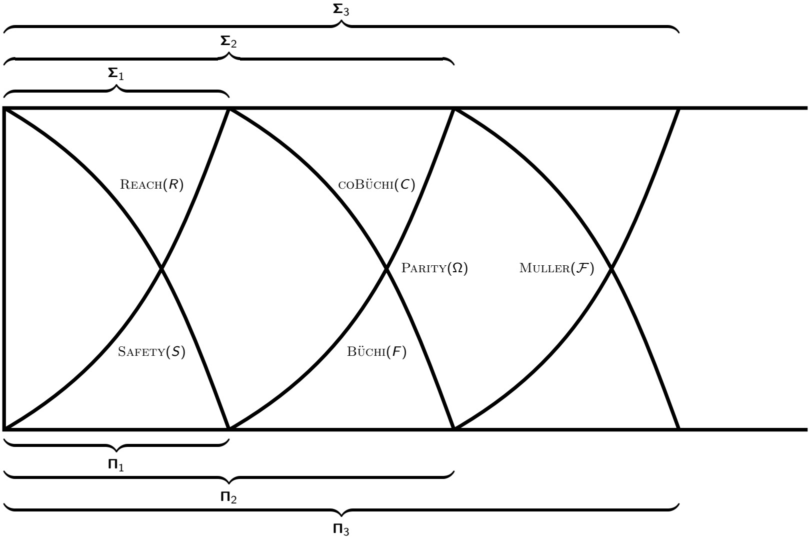
Reducibility



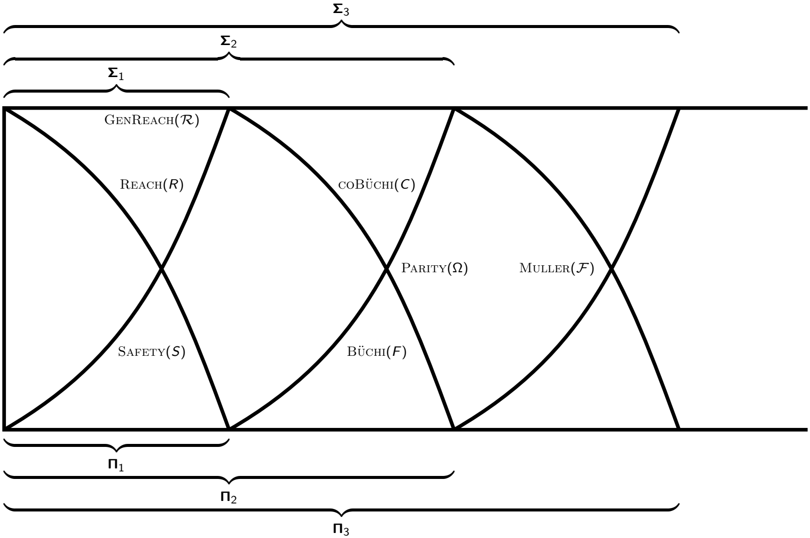
Reducibility



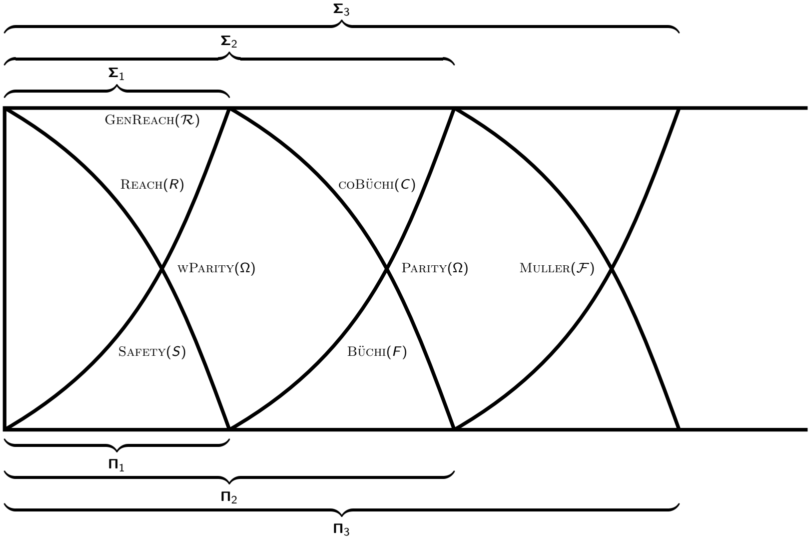
Reducibility



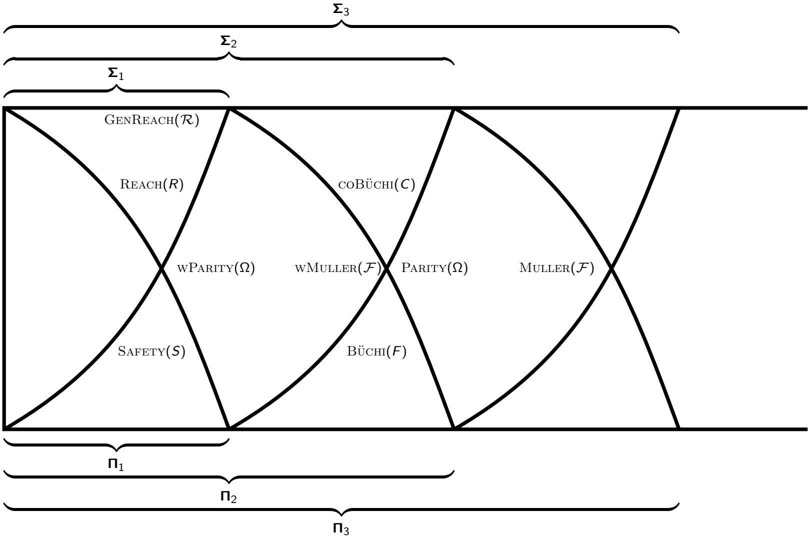
Reducibility



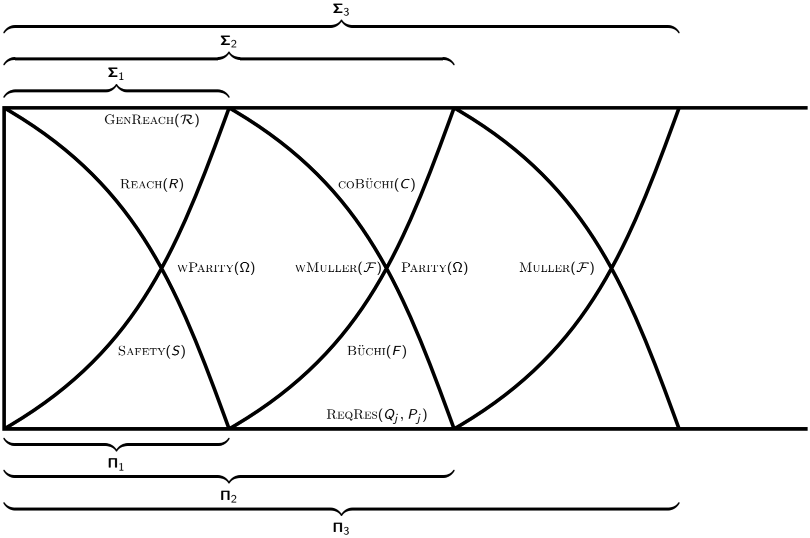
Reducibility



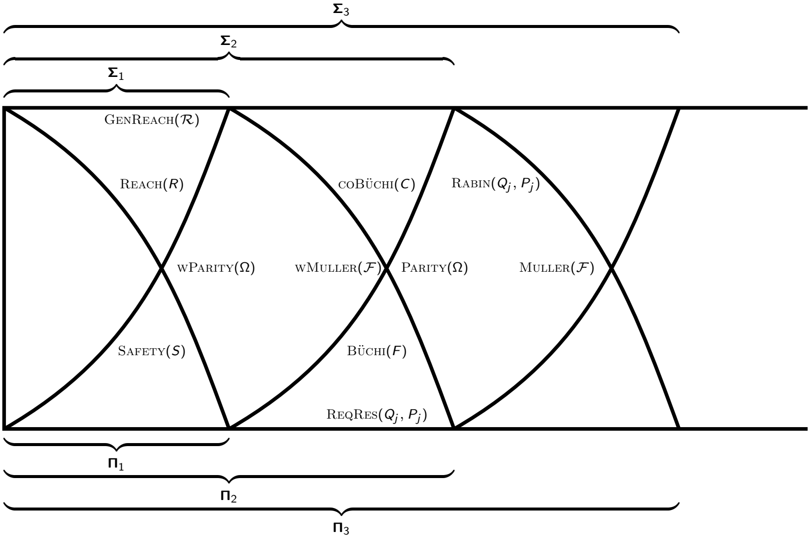
Reducibility



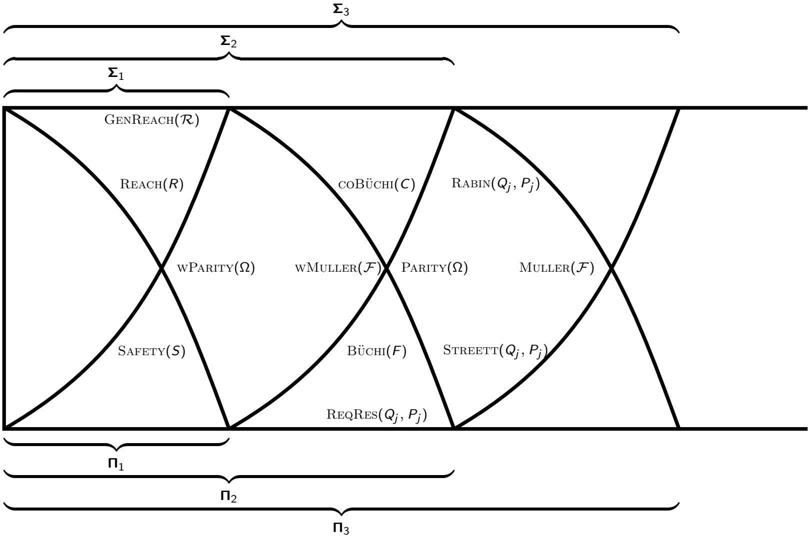
Reducibility



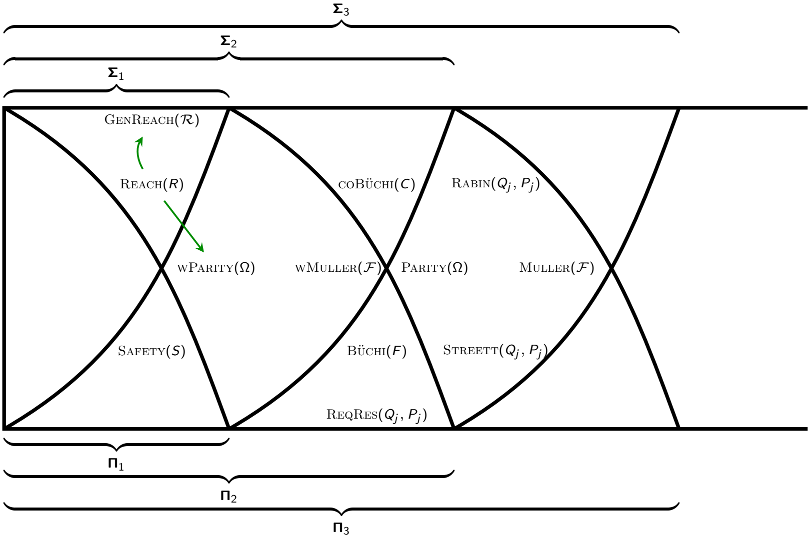
Reducibility



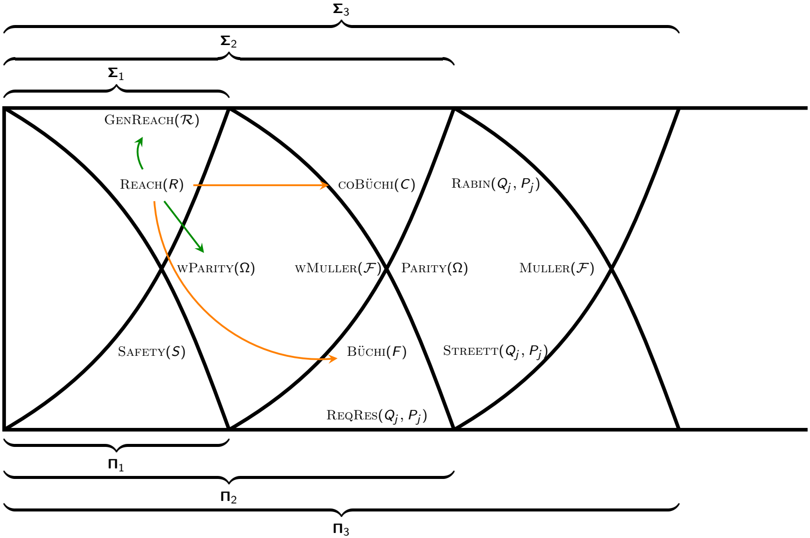
Reducibility



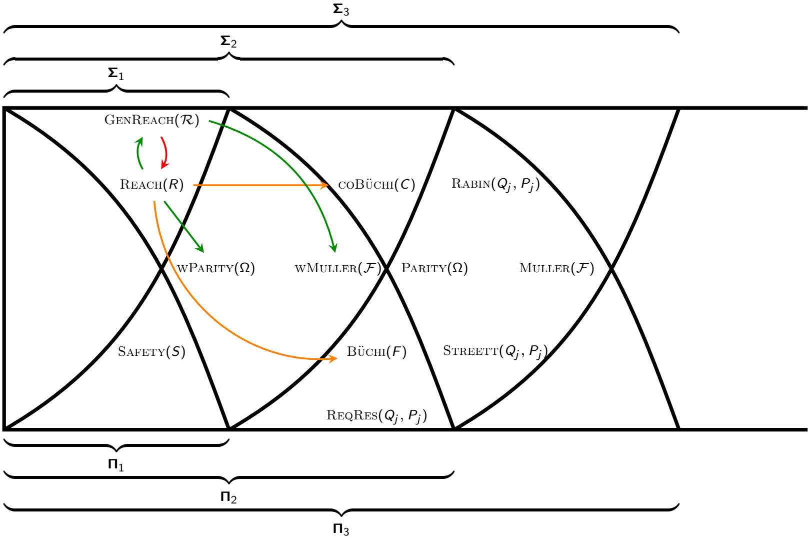
Reducibility



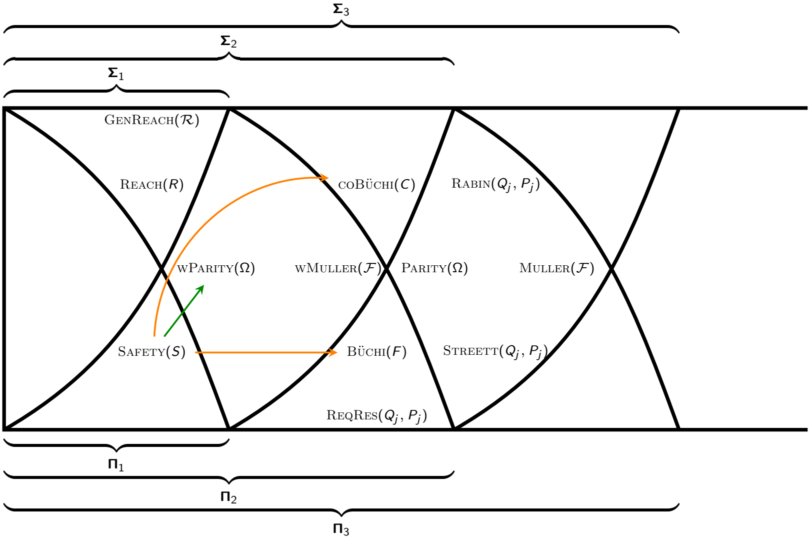
Reducibility



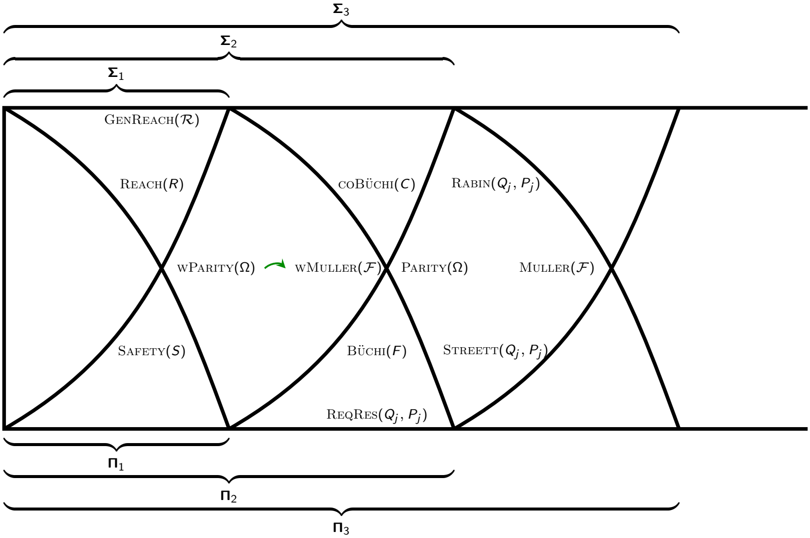
Reducibility



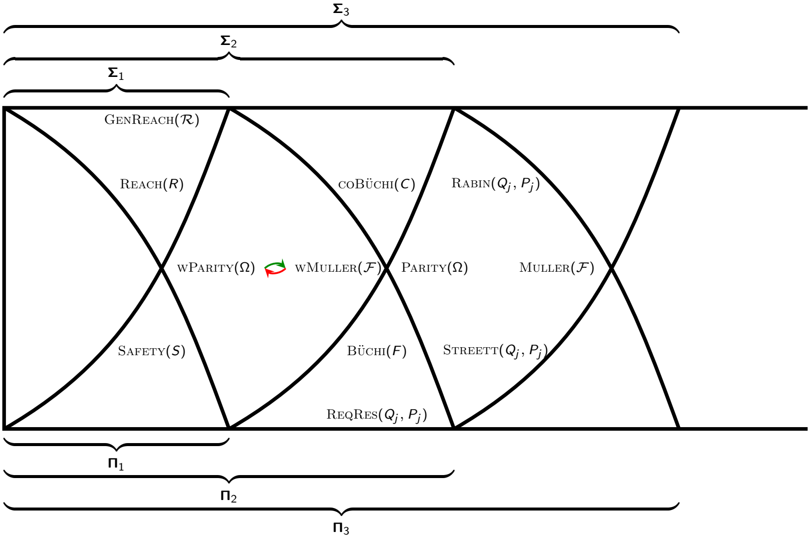
Reducibility



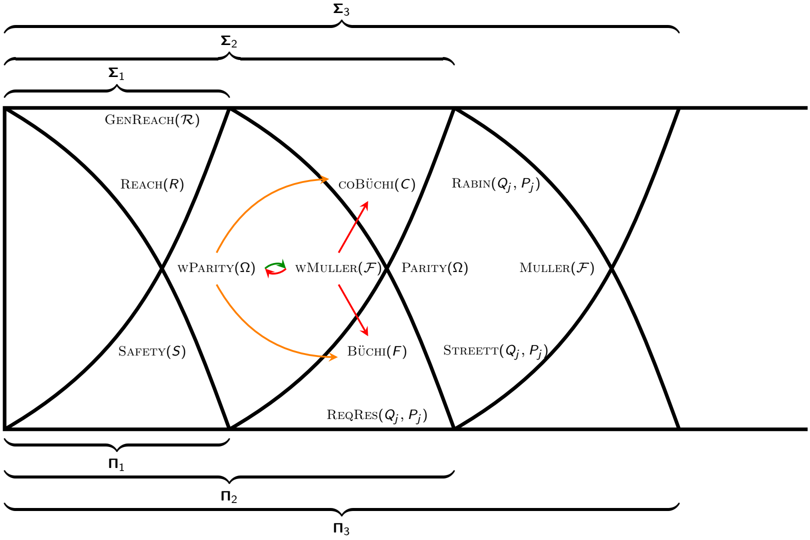
Reducibility



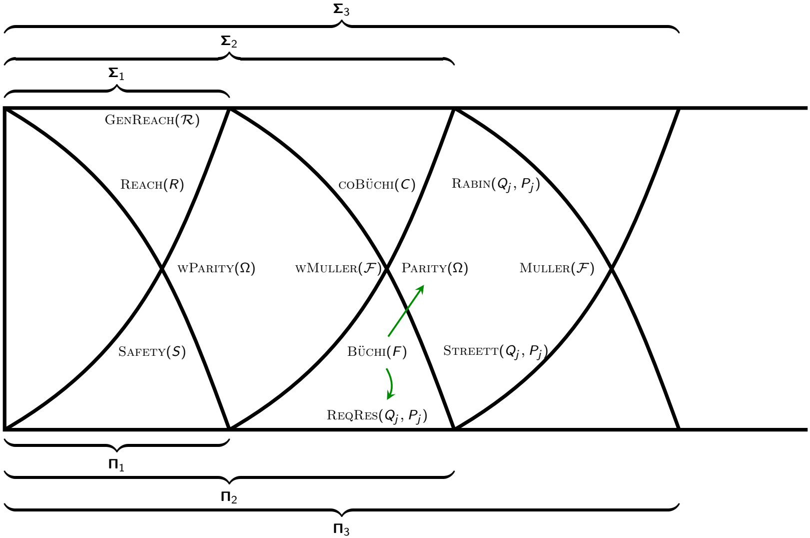
Reducibility



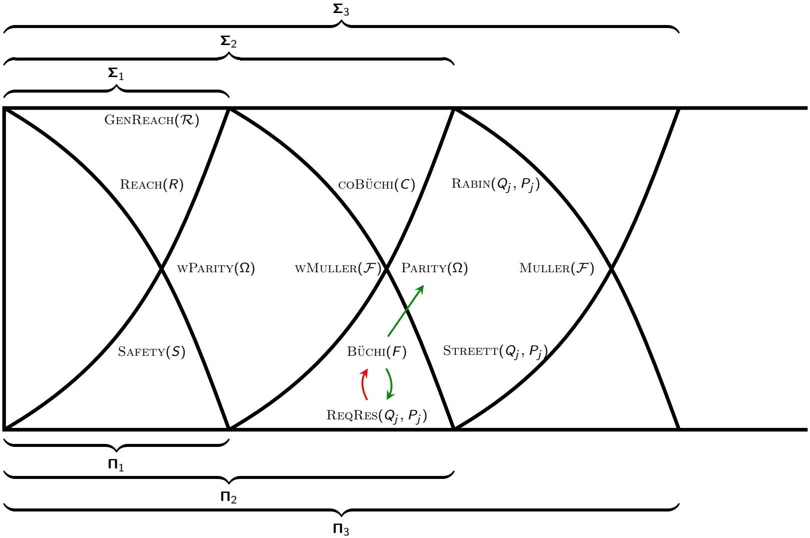
Reducibility



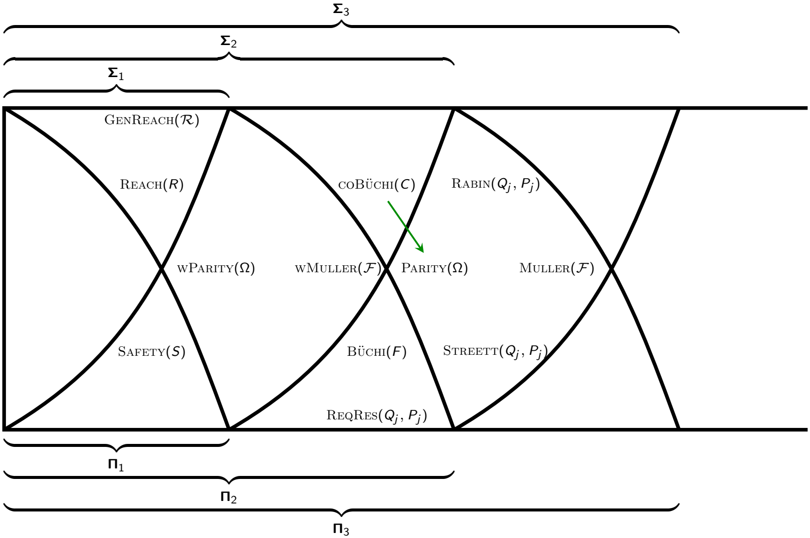
Reducibility



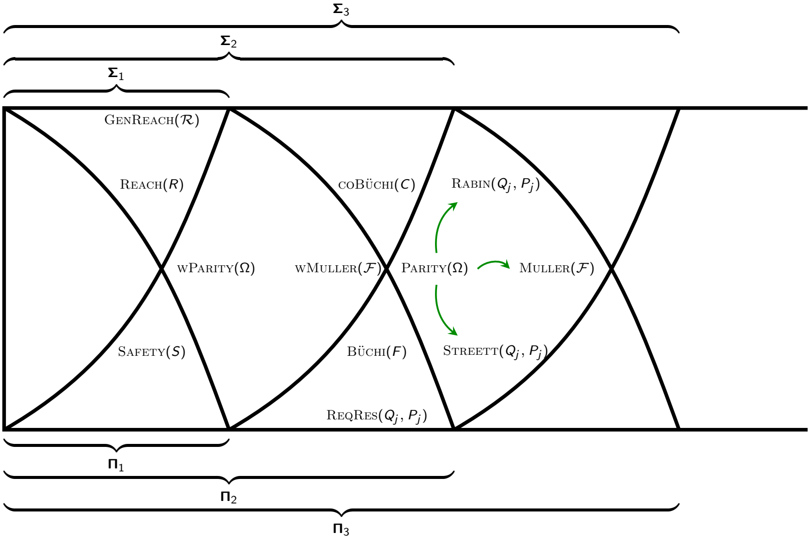
Reducibility



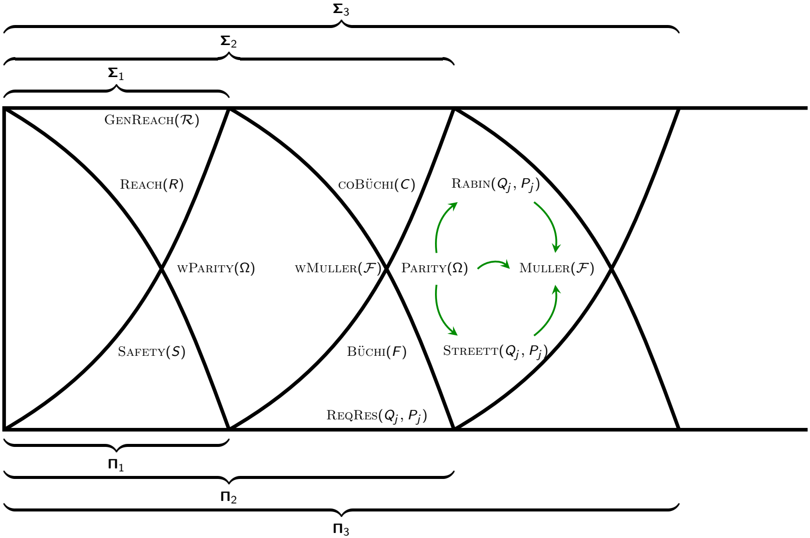
Reducibility



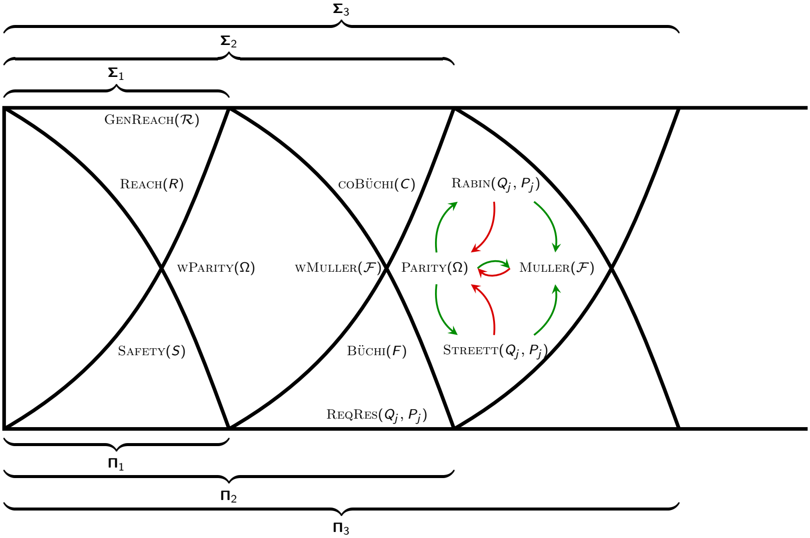
Reducibility



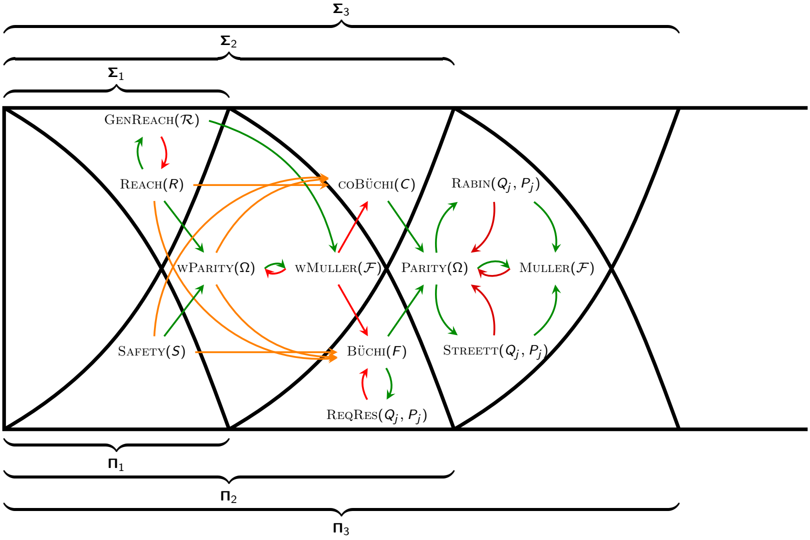
Reducibility



Reducibility



Reducibility



Wadge Games

(Wadge) reductions are (Wadge) games!

- A winning strategy for II in the Wadge game $W(L, L')$ is a witness for the existence of a Wadge reduction $L \leq L'$.
- A winning strategy for I in the Wadge game $W(L, L')$ is a witness for the non-existence of a Wadge reduction $L \leq L'$.

S2S and Parity Tree Automata

- S2S: Monadic second-order logic over two successors
- PTA: Parity tree automata

Both formalisms are equivalent:

- For every $\mathscr{A}$ exists $\varphi_{\mathscr{A}}$ s.t. $t \in \mathcal{L}(\mathscr{A}) \Leftrightarrow t \models \varphi_{\mathscr{A}}$
- For every φ exists $\mathscr{A}_\varphi$ s.t. $t \models \varphi \Leftrightarrow t \in \mathcal{L}(\mathscr{A}_\varphi)$

Consequence: Satisfiability of S2S reduces to PTA emptiness

(Parity) games everywhere:

- Acceptance game $\mathcal{G}(\mathscr{A}, t)$ for complement closure of PTA
- Emptiness game $\mathcal{G}(\mathscr{A})$ for emptiness check of PTA

"The mother of all decidability results"

Change Log
Lecture Notes

Change Log Lecture Notes 1/2

Old definition:

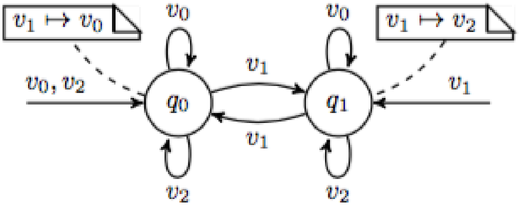
Definition 2.7 (Game). *A game* $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ *consists of an arena* $\mathcal{A}$ *and a set of winning plays* $\mathrm{Win} \subseteq \mathrm{Plays}(\mathcal{A})$. *We call a play* ρ *winning for Player* 0 *if, and only if,* $\rho \in \mathrm{Win}$ *and winning for Player* 1 *otherwise.*

New definition:

Definition 2.7 (Game). *A game* $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ *consists of an arena* $\mathcal{A}$ *and a set of winning plays* $\mathrm{Win} \subseteq V^\omega$. *We call a play* ρ *winning for Player* 0 *if, and only if,* $\rho \in \mathrm{Win}$ *and winning for Player* 1 *otherwise.*

Change Log Lecture Notes 2/2

Graphical notation for finite-state strategies:



We represent the initialization function as labeled initial arrows.

Exam

Organizational Matters

End-of-term exam

- When: August 1st, 2016, 10:15 - 12:15
- Where: HS 003, Building E1 3
- Mode: Open-book
- What to bring: Student ID
- Exam inspection: TBA

Organizational Matters

End-of-term exam

- When: August 1st, 2016, 10:15 - 12:15
- Where: HS 003, Building E1 3
- Mode: Open-book
- What to bring: Student ID
- Exam inspection: TBA

End-of-semester exam: September 20th, 2016 (more information after first exam)

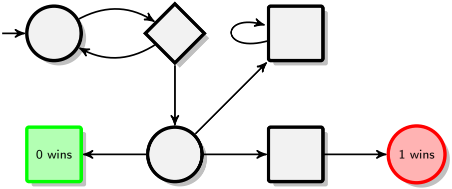
Questions

Challenge us before we challenge you in the exam.

Outlook

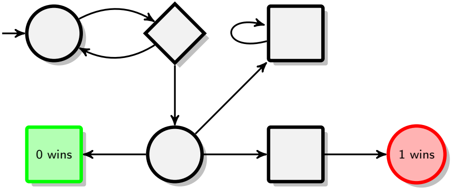
(Simple) Stochastic Games

- Enter a new player (◇), it flips a coin to pick a successor.



(Simple) Stochastic Games

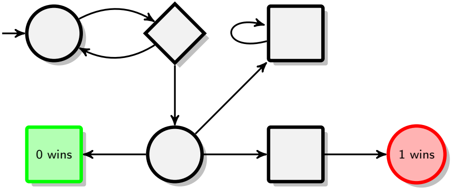
- Enter a new player (◇), it flips a coin to pick a successor.



- No (sure) winning strategy...
- ...but one with probability 1.

(Simple) Stochastic Games

- Enter a new player (◇), it flips a coin to pick a successor.



- No (sure) winning strategy...
- ...but one with probability 1.

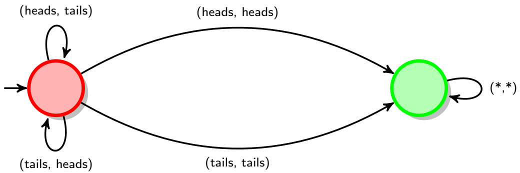
  Value of the game for Player 0: $\max_{\sigma} \min_{\tau} p_{\sigma,\tau}$

where $p_{\sigma,\tau}$ is the probability that Player 0 wins when using strategy σ and Player 1 uses strategy τ.

Concurrent Games

- Both players choose their moves simultaneously

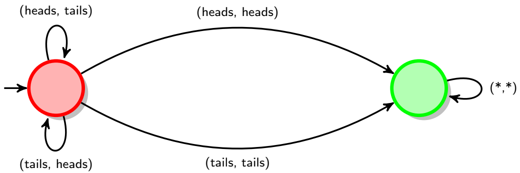
Matching pennies:



Concurrent Games

- Both players choose their moves simultaneously

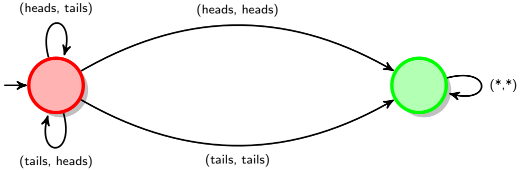
Matching pennies: randomized strategy winning with probability 1.



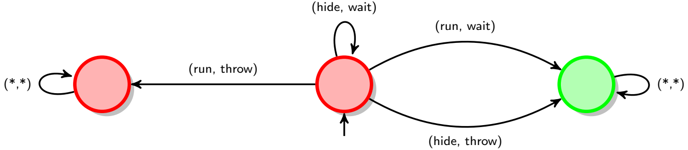
Concurrent Games

- Both players choose their moves simultaneously

Matching pennies: randomized strategy winning with probability 1.



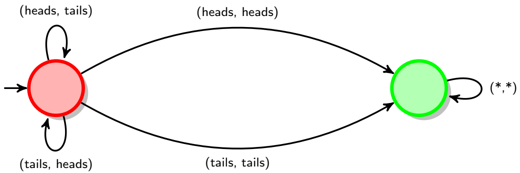
The “Snowball Game”:



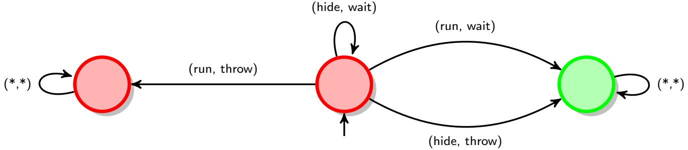
Concurrent Games

- Both players choose their moves simultaneously

Matching pennies: randomized strategy winning with probability 1.

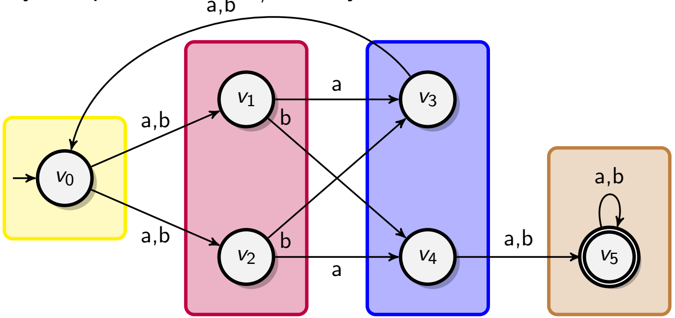


The "Snowball Game": for every ε, randomized strategy winning with probability $1 - \varepsilon$.



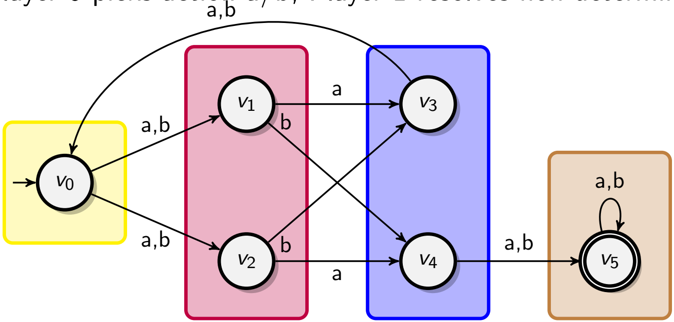
Games of Imperfect Information

- Players do not observe sequence of states, but sequence of non-unique observations (yellow, purple, blue, brown).
- Player 0 picks action a/b, Player 1 resolves non-determinism.



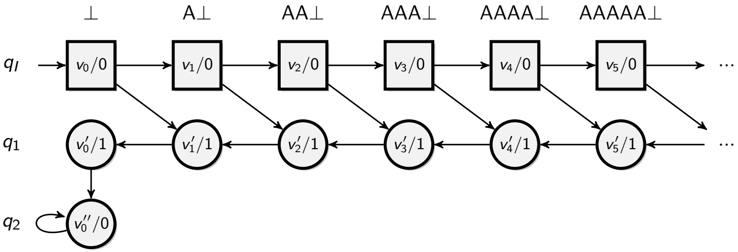
Games of Imperfect Information

- Players do not observe sequence of states, but sequence of non-unique observations (yellow ◯, purple ◯, blue ◯, brown ◯).
- Player 0 picks action a/b, Player 1 resolves non-determinism.

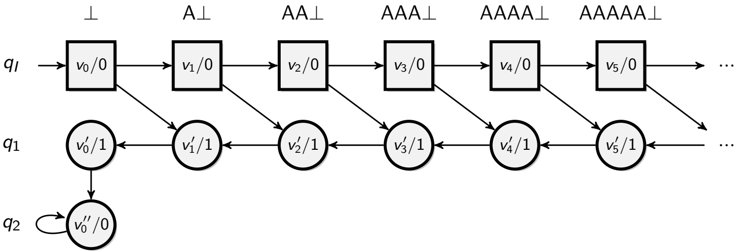


No winning strategy for Player 0: every fixed choice of actions to pick at (◯ ◯ ◯)*(◯ ◯) can be countered by going to v_1 or v_2.

Pushdown Games



Pushdown Games



- Pushdown Parity Games can be reduced to parity games in exponentially sized arenas ⇒ EXPTIME-complete.
- Both players have positional winning strategies (but these are now infinite objects!).
- Finite representation of winning strategies: pushdown automata with output.

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy $\Rightarrow$ winning regions preserved

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$								
$\mathrm{Acc}_{\{0\}}$								
$\mathrm{Sc}_{\{0,1,2\}}$								
$\mathrm{Acc}_{\{0,1,2\}}$								

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1							
$\mathrm{Acc}_{\{0\}}$	$\emptyset$							
$\mathrm{Sc}_{\{0,1,2\}}$								
$\mathrm{Acc}_{\{0,1,2\}}$								

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2						
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$						
$\mathrm{Sc}_{\{0,1,2\}}$								
$\mathrm{Acc}_{\{0,1,2\}}$								

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0					
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$					
$\mathrm{Sc}_{\{0,1,2\}}$								
$\mathrm{Acc}_{\{0,1,2\}}$								

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy $\Rightarrow$ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0				
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$				
$\mathrm{Sc}_{\{0,1,2\}}$								
$\mathrm{Acc}_{\{0,1,2\}}$								

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1			
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$			
$\mathrm{Sc}_{\{0,1,2\}}$								
$\mathrm{Acc}_{\{0,1,2\}}$								

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2		
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$		
$\mathrm{Sc}_{\{0,1,2\}}$								
$\mathrm{Acc}_{\{0,1,2\}}$								

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy $\Rightarrow$ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
$\mathrm{Sc}_{\{0,1,2\}}$								
$\mathrm{Acc}_{\{0,1,2\}}$								

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	0
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathrm{Sc}_{\{0,1,2\}}$								
$\mathrm{Acc}_{\{0,1,2\}}$								

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	0
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathrm{Sc}_{\{0,1,2\}}$	0							
$\mathrm{Acc}_{\{0,1,2\}}$	$\{0\}$							

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	0
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathrm{Sc}_{\{0,1,2\}}$	0	0						
$\mathrm{Acc}_{\{0,1,2\}}$	$\{0\}$	$\{0\}$						

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	0
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathrm{Sc}_{\{0,1,2\}}$	0	0	0					
$\mathrm{Acc}_{\{0,1,2\}}$	$\{0\}$	$\{0\}$	$\{0,1\}$					

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy $\Rightarrow$ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	0
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathrm{Sc}_{\{0,1,2\}}$	0	0	0	0				
$\mathrm{Acc}_{\{0,1,2\}}$	$\{0\}$	$\{0\}$	$\{0,1\}$	$\{0,1\}$				

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	0
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathrm{Sc}_{\{0,1,2\}}$	0	0	0	0	0			
$\mathrm{Acc}_{\{0,1,2\}}$	$\{0\}$	$\{0\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$			

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	0
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathrm{Sc}_{\{0,1,2\}}$	0	0	0	0	0	0		
$\mathrm{Acc}_{\{0,1,2\}}$	$\{0\}$	$\{0\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$		

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy $\Rightarrow$ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	0
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathrm{Sc}_{\{0,1,2\}}$	0	0	0	0	0	0	0	
$\mathrm{Acc}_{\{0,1,2\}}$	$\{0\}$	$\{0\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	0
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathrm{Sc}_{\{0,1,2\}}$	0	0	0	0	0	0	0	1
$\mathrm{Acc}_{\{0,1,2\}}$	$\{0\}$	$\{0\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\emptyset$

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy $\Rightarrow$ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	0
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathrm{Sc}_{\{0,1,2\}}$	0	0	0	0	0	0	0	1
$\mathrm{Acc}_{\{0,1,2\}}$	$\{0\}$	$\{0\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\emptyset$

Theorem

Player i has strategy to bound the opponent's scores by two when starting in $W_i(\mathcal{G})$.

Playing Infinite Games in a Hurry

- Parity games in finite time: play until first loop is closed, minimal color in loop determines winner.
- Positional determinacy ⇒ winning regions preserved

No longer works for Muller games. Need scoring functions:

w	0	0	1	1	0	0	1	2
$\mathrm{Sc}_{\{0\}}$	1	2	0	0	1	2	0	0
$\mathrm{Acc}_{\{0\}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathrm{Sc}_{\{0,1,2\}}$	0	0	0	0	0	0	0	1
$\mathrm{Acc}_{\{0,1,2\}}$	$\{0\}$	$\{0\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$	$\emptyset$

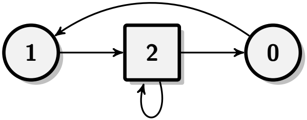
Theorem

Player i has strategy to bound the opponent's scores by two when starting in $W_i(\mathcal{G})$.

Corollary: Stopping play after first score reaches value three preserves winning regions (at most exponential play length)

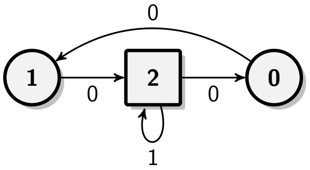
Games with Costs

- Parity game: Player 0 wins from everywhere, but it takes arbitrarily long to "answer" 1 with 0.



Games with Costs

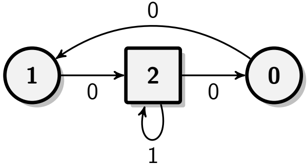
- Parity game: Player 0 wins from everywhere, but it takes arbitrarily long to "answer" 1 with 0.



- Add edge-costs: Player 0 wins if there is a bound b and a position n such that every odd color after n is followed by a smaller even color with cost $\leq b$ in between $\Rightarrow$ Player 1 wins example from everywhere (stay longer and longer in 2).

Games with Costs

- Parity game: Player 0 wins from everywhere, but it takes arbitrarily long to "answer" 1 with 0.



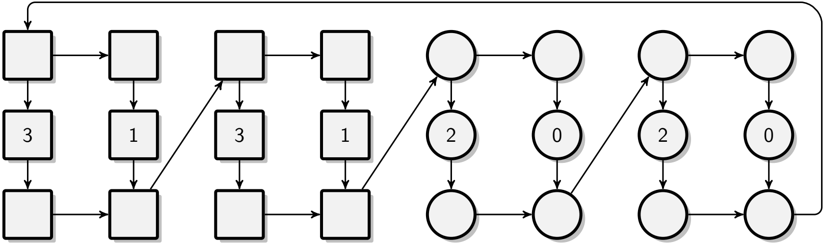
- Add edge-costs: Player 0 wins if there is a bound b and a position n such that every odd color after n is followed by a smaller even color with cost $\leq b$ in between $\Rightarrow$ Player 1 wins example from everywhere (stay longer and longer in 2).

Theorem

Parity games with costs are determined, Player 0 has positional winning strategies, and they can be solved in $\mathbf{NP} \cap \mathbf{co\text{-}NP}$.

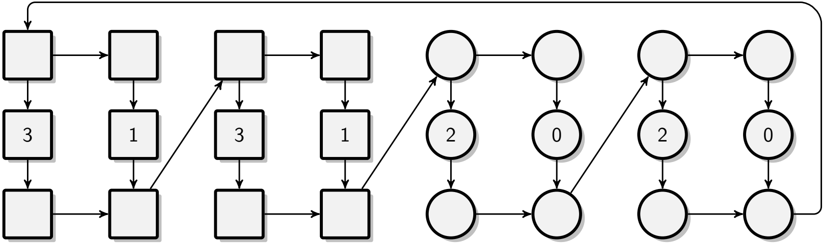
Tradeoffs

Every edge has cost 1



Tradeoffs

Every edge has cost 1

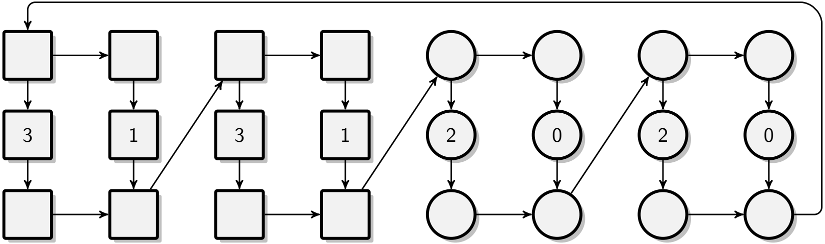


Player 0 has:

- Positional winning strategy with bound 9.
- Finite-state strategy of size 2 with bound 8.

Tradeoffs

Every edge has cost 1



Player 0 has:

- Positional winning strategy with bound 9.
- Finite-state strategy of size 2 with bound 8.

With d odd colors and d gadgets for each player: Player 0 has:

- Positional winning strategy with bound $d^2 + 3d - 1$.
- Finite-state strategy of size $2^d - 2$ with bound $d^2 + 2d$.

Many other variants

- More winning conditions: various quantitative conditions

Many other variants

- More winning conditions: various quantitative conditions
- Games on timed automata $\Rightarrow$ uncountable arenas

Many other variants

- More winning conditions: various quantitative conditions
- Games on timed automata $\Rightarrow$ uncountable arenas
- Play even longer: games of ordinal length

Many other variants

- More winning conditions: various quantitative conditions
- Games on timed automata $\Rightarrow$ uncountable arenas
- Play even longer: games of ordinal length
- Games with delay: Player 0 is allowed to skip some moves to obtain lookahead on Player 1's moves. Basic question: what kind of lookahead is necessary to win.

Many other variants

- More winning conditions: various quantitative conditions
- Games on timed automata $\Rightarrow$ uncountable arenas
- Play even longer: games of ordinal length
- Games with delay: Player 0 is allowed to skip some moves to obtain lookahead on Player 1's moves. Basic question: what kind of lookahead is necessary to win.
- More than two players $\Rightarrow$ no longer zero-sum games. Requires whole new theory (equilibria).

Many other variants

- More winning conditions: various quantitative conditions
- Games on timed automata $\Rightarrow$ uncountable arenas
- Play even longer: games of ordinal length
- Games with delay: Player 0 is allowed to skip some moves to obtain lookahead on Player 1's moves. Basic question: what kind of lookahead is necessary to win.
- More than two players $\Rightarrow$ no longer zero-sum games. Requires whole new theory (equilibria).

And: any combination of extensions discussed above.

Thesis Topics

DFG project **TriCS**: Tradeoffs in Controller Synthesis.

- How to compute optimal strategies?

Thesis Topics

DFG project **TriCS**: Tradeoffs in Controller Synthesis.

- How to compute optimal strategies?
- Games with delay: how much lookahead is necessary for different winning conditions? Tradeoffs between lookahead and memory?
- Temporal logics for the specification of reactive systems.
- ...

Thesis Topics

DFG project **TriCS**: Tradeoffs in Controller Synthesis.

- How to compute optimal strategies?
- Games with delay: how much lookahead is necessary for different winning conditions? Tradeoffs between lookahead and memory?
- Temporal logics for the specification of reactive systems.
- ...
- Your own idea?

Thesis Topics

DFG project **TriCS**: Tradeoffs in Controller Synthesis.

- How to compute optimal strategies?
- Games with delay: how much lookahead is necessary for different winning conditions? Tradeoffs between lookahead and memory?
- Temporal logics for the specification of reactive systems.
- ...
- Your own idea?
- Or what about some open problems:
  - Generalized reachability games with sets of size two: **P**, **NP**, or **PSPACE**?
  - Exact complexity of parity games.

Thesis Topics

DFG project **TriCS**: Tradeoffs in Controller Synthesis.

- How to compute optimal strategies?
- Games with delay: how much lookahead is necessary for different winning conditions? Tradeoffs between lookahead and memory?
- Temporal logics for the specification of reactive systems.
- ...
- Your own idea?
- Or what about some open problems:
  - Generalized reachability games with sets of size two: **P**, **NP**, or **PSPACE**?
  - Exact complexity of parity games.

If you are interested in working on current research topics, contact us!

Thank You

&

Good luck for the exam