Infinite Games

Lecture 15

Martin Zimmermann

Universität des Saarlandes

February 6th, 2014

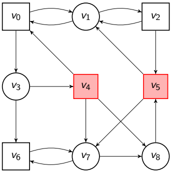
Plan for Today

- Review
- Exam
  - Organizational matters
  - Questions
- Outlook: even more games

Review

Reachability

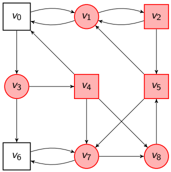
- **Name:** **Reachability Game**
- Format: $(\mathcal{A}, \textsc{Reach}(R))$ with $R \subseteq V$



- 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

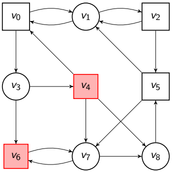
- **Name:** **Safety Game**
- Format: $(\mathcal{A}, \textsc{safe}(S))$ with $S \subseteq V$



- 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

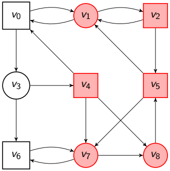
- **Name:** **Büchi Game**
- Format: $(\mathcal{A}, \text{BÜCHI}(F))$ with $F \subseteq V$



- Winning condition: $\mathrm{Inf}(\rho) \cap F \neq \emptyset$
- 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

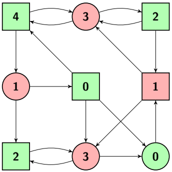
- **Name:** **Co-Büchi Game**
- Format: $(\mathcal{A}, \textsc{CoBüchi}(C))$ with $C \subseteq V$



- 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

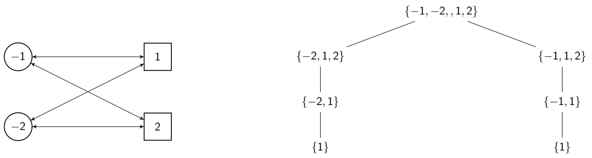
- **Name:** **Parity Game**
- Format: $(\mathcal{A}, \textsc{parity}(\Omega))$ with $\Omega \colon V \to \mathbb{N}$



- Winning condition: $\min(\mathrm{Inf}(\Omega(\rho)))$ even
- Solution complexity: **NP $\cap$ 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

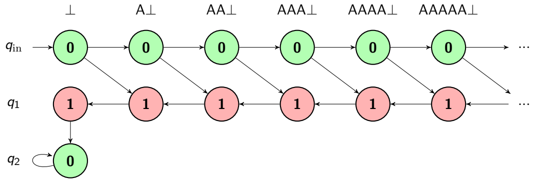
- **Name:** **Muller Game**
- Format: $(\mathcal{A}, \textsc{Muller}(\mathcal{F}))$ with $\mathcal{F} \subseteq 2^V$



- Winning condition: $\mathrm{Inf}(\rho) \in \mathcal{F}$
- Solution complexity: **P**, **NP** ∩ **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

Pushdown Parity

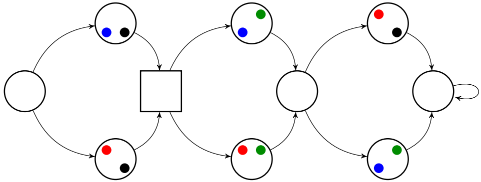
- **Name:** **Pushdown Parity Game**
- Format: $(\mathcal{A}, \textsc{parity}(\Omega))$ with $\mathcal{A}$ induced by PDS $\mathcal{P}$



- Winning condition: $\min(\mathrm{Inf}(\Omega(\rho)))$ even
- Solution complexity: **EXPTIME**-complete
- Algorithm: reduction to parity games
- Memory requirements for Player 0: infinite (pd. transducer)
- Memory requirements for Player 1: infinite (pd. transducer)
- Dual game: pushdown parity

Generalized Reachability

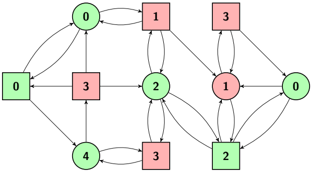
- **Name:** **Generalized Reachability Game**
- Format: $(\mathcal{A}, \textsc{chReach}(\mathcal{R}))$ with $\mathcal{R} \subseteq 2^V$



- 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

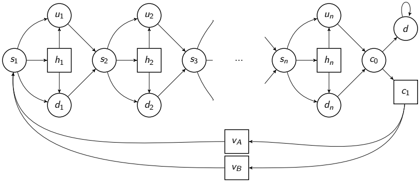
- **Name:** **Weak Parity Game**
- Format: $(\mathcal{A}, \textsc{wparity}(\Omega))$ with $\Omega\colon V \to \mathbb{N}$



- Winning condition: $\min(\mathrm{Occ}(\Omega(\rho)))$ 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

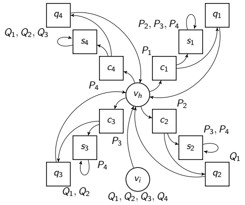
- **Name:** **Weak Muller Game**
- Format: $(\mathcal{A}, \textsc{wMuller}(\mathcal{F}))$ with $\mathcal{F} \subseteq 2^V$



- Winning condition: $\mathrm{Occ}(\rho) \in \mathcal{F}$
- 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

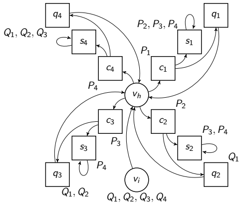
- **Name:** **Request-Response Game**
- Format: $(\mathcal{A}, \textsc{ReqRes}((Q_j, P_j)_{j\in[k]}))$ with $Q_j, P_j \subseteq V$



- Winning condition: $\forall j\, \forall n(\rho_n \in Q_j \rightarrow \exists m \geq n.\, \rho_m \in P_j)$
- 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

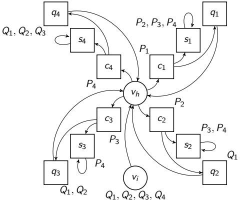
- **Name:** **Rabin Game**
- Format: $(\mathcal{A}, \textsc{rabin}((Q_j, P_j)_{j \in [k]}))$ with $Q_j, P_j \subseteq V$



- Winning condition: $\exists j (\mathrm{Inf}(\rho) \cap Q_j \neq \emptyset \wedge \mathrm{Inf}(\rho) \cap P_j = \emptyset)$
- 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

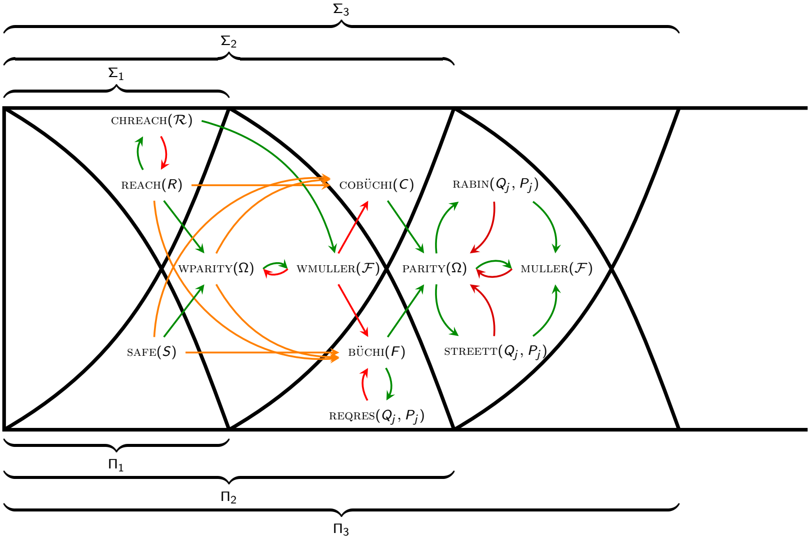
Streett

- **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



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”

Exam

Organizational Matters

End-of-term exam

- When: February 13th, 2014, 09:30 - 11:30
- Where: HS 003, Building E1.3
- Mode: Open-book
- What to bring: Student ID
- Exam inspection: Feb. 14th, 2014, 15:00 - 16:00 (Room 328?)

End-of-semester exam: March 20th, 2014 (more information after first exam)

Questions

Challenge us before we challenge you in the exam.

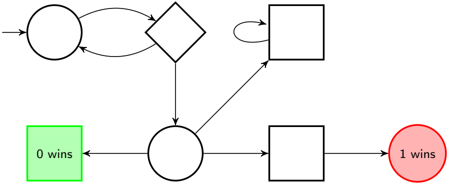
There will also be a tutorial where you can ask further questions!

- When: March 11th, 2014, 16:00 - 18:00
- Where: SR U.11, Building E2.5

Outlook

(Simple) Stochastic Games

- Enter a new player ($\Diamond$), it flips a coin to pick a successor.



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

More formally: Value of the game

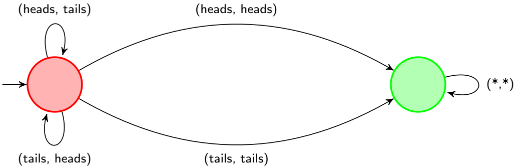
$$\max_{\sigma} \min_{\tau} p_{\sigma,\tau}$$

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

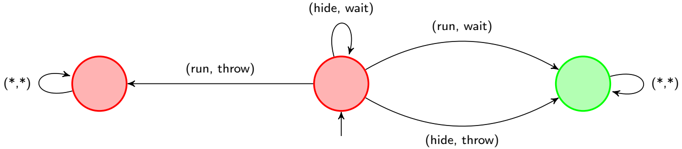
Concurrent Games

- Both players choose their moves simultaneously

Matching pennies: randomized strategy winning with probability 1.

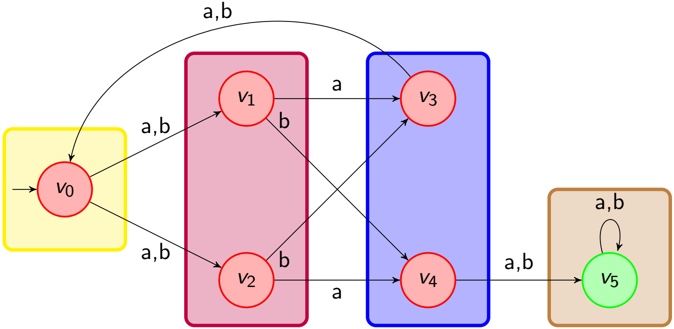


The "Snowball Game": for every ε, randomized strategy winning with probability $1 - \varepsilon$.



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.



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.

Higher-order Pushdown Automata

- Level-1 stack: finite sequence over Γ (standard stack)
- Level-$(k+1)$ stack: finite sequence of level-k stacks
- Operations (various definitions possible):
  - push_γ and pop_γ for $\gamma \in \Gamma$: push and pop on level 1
  - copy_k: copy the topmost level-k stack and add it to the level-$(k+1)$ stack
  - delete_k: delete the topmost level-k stack

Example: on the blackboard

Theorem

Parity games on configuration graphs of higher-order pushdown automata can be solved algorithmically.

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

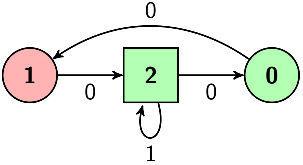
Theorem

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 two “answer” 1 by 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 $\mathbf{NP} \cap \textbf{co-NP}$.

Many other variants

- More winning conditions: various quantitative conditions (parity with costs, waiting times for RR games, and many more)
- 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

- Even pushdown games can be played in finite time. What about higher-order pushdown games?
- How to compute optimal strategies for parity games with costs?
- Games with delay: how much lookahead is necessary for different winning conditions? What effect has lookahead on the memory requirements?
- ...
- Your own idea?
- Generalized reachability games with sets of size two: **P**, **NP**, or **PSPACE**?
- Exact complexity of parity games.

Thank You

&

Good luck for the exam