Plan for Today

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

Name: Reachability Game
Format: $(A, \text{REACH}(R))$ with $R \subseteq V$

Winning condition: $\text{Occ}(\rho) \cap R \neq \emptyset$
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:** \((A, \text{SAFE}(S))\) with \(S \subseteq V\)

![Graph Diagram]

- **Winning condition:** \(\text{Occ}(\rho) \subseteq S\)
- **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:**
\[ \text{Inf}(\rho) \cap F \neq \emptyset \]  \[ \text{P} \]

**Solution complexity:**

- 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:** $(A, \text{COBÜCHI}(C))$ with $C \subseteq V$

Winning condition: $\text{Inf}(\rho) \subseteq C$

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: $(\mathcal{A}, \text{PARITY}(\Omega))$ with $\Omega: V \to \mathbb{N}$

Winning condition: $\min(\inf(\Omega(\rho)))$ even
Solution complexity: $\text{NP} \cap \text{co-NP}$
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: 
\[(\mathcal{A}, \text{MULLER}(\mathcal{F})) \text{ with } \mathcal{F} \subseteq 2^V\]

Winning condition: 
\[\text{Inf}(\rho) \in \mathcal{F}\]

Solution complexity: 
\[\text{P, NP} \cap \text{co-NP, PSPACE}-complete\]

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:** 
\( (\mathcal{A}, \text{PARITY}(\Omega)) \) with \( \mathcal{A} \) induced by PDS \( \mathcal{P} \)

**Winning condition:** 
\( \min(\text{Inf}(\Omega(\rho))) \) even

**Solution complexity:** 
\( \text{EXPTIME}\)-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 Game

**Format:** \((A, \text{CHREACH}(\mathcal{R}))\) with \(\mathcal{R} \subseteq 2^V\)

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

---

**Name:** Generalized Reachability Game

**Format:** 
\((A, \text{CHREACH}(\mathcal{R}))\) with \(\mathcal{R} \subseteq 2^V\)
Weak Parity

- **Name:** Weak Parity Game
- **Format:** \((A, \text{WPARITY}(\Omega))\) with \(\Omega: V \rightarrow \mathbb{N}\)

**Winning condition:** \(\min(Occ(\Omega(\rho)))\) 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:

Format:

Weak Muller Game

\((A, \text{WMULLER}(\mathcal{F}))\) with \(\mathcal{F} \subseteq 2^V\)

Winning condition:

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
**Name:** Request-Response Game

**Format:**

\[(\mathcal{A}, \text{REQRES}((Q_j, P_j)_{j \in [k]})) \text{ with } Q_j, P_j \subseteq V\]

**Winning condition:**

\[\forall j \forall n (\rho_n \in Q_j \rightarrow \exists m \geq n. \rho_m \in P_j)\]

**Solution complexity:** \(\text{EXPTIME}\)-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:** 
  \[ \langle A, \text{RABIN}((Q_j, P_j)_{j \in [k]}) \rangle \text{ with } Q_j, P_j \subseteq V \]

  ![Diagram of the Rabin Game](image)

- **Winning condition:** \( \exists j (\text{Inf}(\rho) \cap Q_j \neq \emptyset \land \text{Inf}(\rho) \cap P_j = \emptyset) \)
- **Solution complexity:** \( \text{NP-complete} \)
- **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:** \((\mathcal{A}, \text{STREETT}((Q_j, P_j)_{j \in [k]}))\) with \(Q_j, P_j \subseteq V\)

- **Winning condition:** \(\forall j (\inf(\rho) \cap Q_j \neq \emptyset \rightarrow \inf(\rho) \cap P_j \neq \emptyset)\)
- **Solution complexity:** \(\text{co-NP}-\text{complete}\)
- **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 $A$ exists $\varphi_A$ s.t. $t \in L(A) \iff t \models \varphi_A$
- For every $\varphi$ exists $A_\varphi$ s.t. $t \models \varphi \iff t \in L(A_\varphi)$

**Consequence:** Satisfiability of S2S reduces to PTA emptiness

(Parity) Games everywhere:

- Acceptance game $G(A, t)$ for complement closure of PTA
- Emptiness game $G(A)$ 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 (◊), 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 $\sigma$ and Player 1 uses strategy $\tau$. 
Both players choose their moves simultaneously.

Matching pennies: randomized strategy winning with probability 1.

The “Snowball Game”: for every $\varepsilon$, randomized strategy winning with probability $1 - \varepsilon$. 
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 \((v_0)\) can be countered by going to \(v_1\) or \(v_2\).
Higher-order Pushdown Automata

- Level-1 stack: finite sequence over $\Gamma$ (standard stack)
- Level-$(k + 1)$ stack: finite sequence of level-$k$ stacks
- Operations (various definitions possible):
  - $\text{push}_{\gamma}$ and $\text{pop}_{\gamma}$ for $\gamma \in \Gamma$: push and pop on level 1
  - $\text{copy}_k$: copy the topmost level-$k$ stack and add it to the level-$(k + 1)$ stack
  - $\text{delete}_k$: 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.*
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:

<table>
<thead>
<tr>
<th>$w$</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Sc}_{{0}}$</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\text{Acc}_{{0}}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$\text{Sc}_{{0,1,2}}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\text{Acc}_{{0,1,2}}$</td>
<td>${0}$</td>
<td>${0}$</td>
<td>${0,1}$</td>
<td>${0,1}$</td>
<td>${0,1}$</td>
<td>${0,1}$</td>
<td>${0,1}$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

**Theorem**

Player $i$ has strategy to bound the opponent’s scores by two when starting in $W_i(G)$.

**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 $\text{NP} \cap \text{co-NP}$.***
Many other variants

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