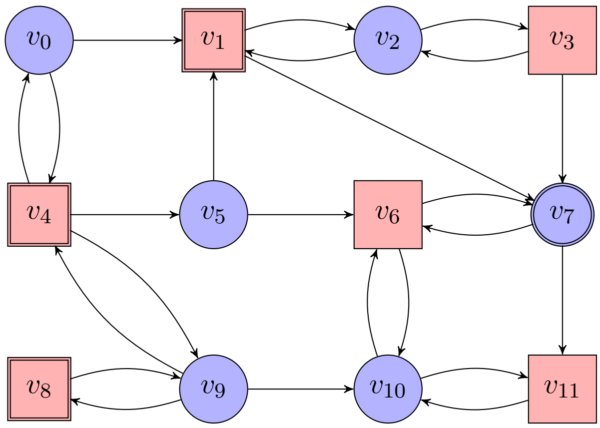
Prof. Bernd Finkbeiner, Ph.D.
Felix Klein, M.Sc.

Problem Set 12
Discussions: July 15th, 2015

Automata, Games, and Verification

Please send a mail to `agv15@react.uni-saarland.de` if you can't make it to the discussion session.

1. **Büchi Games** (presented by Group 07)



Consider the Büchi game $\mathcal{G}$ depicted above. Compute the winning regions and corresponding uniform memoryless winning strategies for Player 0 and Player 1.

2. **Arena-preserving game conversions** (presented by Group 02)

Let an arena $\mathcal{A} = (V, V_0, V_1, E)$ and a position $v_I \in V_0$ be given. Consider all combinations $(\mathit{Win}, \mathit{Win}')$ of winning conditions from the list below. For which of the combinations does it hold, that if you are given the winning condition Win, you can always convert it to the winning condition Win' such that the winning strategies for both players from v_I in $\mathcal{G} = (\mathcal{A}, \mathit{Win})$ are the same as from v_I in $\mathcal{G}' = (\mathcal{A}, \mathit{Win}')$?

   a) Reachability games
   b) Büchi games
   c) Parity games
   d) Rabin games
   e) Muller games

3. **Fair Simulation** (presented by Group 10)

Simulation is often used as an efficient method to establish language containment between automata. *Fair simulation*, the commonly used type of simulation for Büchi automata, can be described as a game:

Given two Büchi automata $\mathcal{A} = (\Sigma, Q, \{q_I\}, T, \text{BÜCHI}(F))$ and $\mathcal{A}' = (\Sigma, Q', \{q'_I\}, T', \text{BÜCHI}(F'))$, the automaton $\mathcal{A}'$ *simulates* the automaton $\mathcal{A}$ if player "Duplicator" wins the following game between Duplicator and a second player "Spoiler." The game is played in rounds, as follows: At the start, round 0, two pebbles, a and b, are placed on q_I and q'_I, respectively. Assume that, at the beginning of round n, pebble a is on state $q_n \in Q$ and pebble b is on state $q'_n \in Q'$. Then:

   a) Spoiler chooses a transition $(q_n, \sigma, q_{n+1}) \in T$ for some letter $\sigma \in \Sigma$.

b) Duplicator, responding, must chose a transition $(q'_n, \sigma, q'_{n+1}) \in T'$. If no σ-transition exists from q'_n, Spoiler wins the game at this point.

If a play is infinite, then the winner is determined according to the following rule: Duplicator wins if there are infinitely many n such that $q'_n \in F'$ or there are only finitely many n such that $q_n \in F$.

a) For two given Büchi automata $\mathcal{A}$ and $\mathcal{A}'$, reformulate this game as a parity game such that $\mathcal{A}'$ simulates $\mathcal{A}$ iff Player 1 wins the parity game (from some designated starting position).

b) Show that fair simulation is a conservative test for language containment. I.e., show that if $\mathcal{A}'$ simulates $\mathcal{A}$, then $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}')$, and give an example where $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}')$, but $\mathcal{A}'$ does not simulate $\mathcal{A}$.

4. **Update Networks** (presented by Group 15)

In an *update game* $\mathcal{G} = (\mathcal{A}, \mathit{Win})$ with $\mathcal{A} = (V, V_0, V_1, E)$ and

$$\mathit{Win} = \{\rho \in V^\omega \mid \mathrm{Inf}(\rho) = V\}$$

the players take turns, i.e, $E \subseteq (V_0 \times V_1) \cup (V_1 \times V_0)$, and Player 0 wins iff every position in V is visited infinitely often. An update game is an *update network* if Player 0 wins from every position. Update networks are for example of interest in the design of distributed networks (where each node needs to be updated with current information).

We say that Player 1 is *forced* from a position $v \in V_1$ to move to a position $v \in V_0$, if (v, v') is the only edge in E from v. For $v_0 \in V_0$, we define:

$$\mathit{Forced}(v_0) = \{v_1 \in V_1 \mid \text{ Player 1 is forced to move from } v_1 \text{ to } v_0\}.$$

A *forced cycle* is a sequence of positions $v_0^k v_1^k \dots v_0^2 v_1^2 v_0^1 v_1^1$ such that $v_0^i \in \mathit{Forced}(v_1^i)$ and $(v_1^{i+1}, v_0^i) \in E$ for all $1 \leq i \leq k$, and $(v_1^1, v_0^k) \in E$. Prove the following:

a) If $\mathcal{G}$ is an update network, then for every position $v_0 \in V_0$ there is a node $v_1 \in V_1$ from which Player 1 is forced to move to v_0.

b) If $\mathcal{G}$ is an update network with $|V_0| > 1$, then for every $v_0 \in V_0$, there is a $v'_0 \in V_0$ such that $v_0 \neq v'_0$ and there is a node $v_1 \in \mathit{Forced}(v_0)$ such that $(v'_0, v_1) \in E$ and $(v_1, v_0) \in E$.

c) If $\mathcal{G}$ is an update network with $|V_0| > 1$, then $\mathcal{G}$ has a forced cycle of length at least 4, i.e., $k \geq 2$.

5. **Mean Payoff Games (Challenge)**

A *mean payoff game* is a game $\mathcal{G} = (\mathcal{A}, \textsc{MeanPayoff}(\nu, d, w))$ with winning condition

$$\textsc{MeanPayoff}(\nu, d, w) = \Big\{\rho \in V^\omega \mid \liminf_{t \to \infty} \frac{1}{t} \sum_{i=1}^{t} w(\rho(i-1), \rho(i)) \ \geq \ \nu\Big\},$$

where ν and d are natural numbers, and $w \colon E \to \{-d, \dots, d\}$ assigns an integer value to each edge. For a given parity game, define a mean payoff game with the same game positions, such that winning memoryless strategies of the parity game are winning memoryless strategies of the mean payoff game and vice versa.