Games with perfect information
Exercise sheet 10

Sebastian Muskalla | TU Braunschweig
Prof. Dr. Roland Meyer | Summer term 2017

Out: June 15, **Updated: June 21** Due: June 23

Submit your solutions until Friday, June 23, 14:00, in the box next to office 343.

Exercise 1: Applying Rabin's tree theorem

In this exercise, we want to apply Rabin's tree theorem to the automaton A_1 from Part a) of Exercise 1 from the last exercise sheet.

$$A_1 = \Big(\underbrace{\{a_{/2}, b_{/2}\}}_{\Sigma}, \underbrace{\{q_0, q_1\}}_{Q}, q_0, \rightarrow, \Big(q_0 \overset{\Omega}{\mapsto} 0, q_1 \overset{\Omega}{\mapsto} 0 \Big) \Big) \quad \text{with}$$

$$\rightarrow_a = \{ (q_0, (q_1, q_1)) \}, \qquad \rightarrow_b = \{ (q_1, (q_0, q_0)) \}.$$

a) Construct the set $S = Q^{\leq n} \rightarrow D$.

   Hint: To avoid the following construction becoming excessively large, restrict the domain to vectors of states that can actually occur.

b) Construct the parity word automaton $A_{branches}$.

c) Make $A_{branches}$ deterministic by adding an error-state and the corresponding transitions. (For each symbol a, s, d, and each state q, there needs to be exactly one transition $(q, q') \in \rightarrow_{a,s,d}$.) Complement $A_{branches}$ to obtain the automaton B with $\mathcal{L}(B) = \overline{\mathcal{L}(A_{branches})}$.

d) Construct the parity tree automaton C for $\mathcal{L}'$ that simulates B on all branches of a tree.

e) Project C to Σ to obtain the automaton $\overline{A_1}$. Check that $\mathcal{L}(\overline{A_1}) = \overline{\mathcal{L}(A_1)}$ indeed holds by describing the language of $\overline{A_1}$.

Remark: I know that doing this exercise is cumbersome, but I think it will help in understanding the details of the proof of Rabin's tree theorem.

Exercise 2: Emptiness games for PTAs

a) Let A be a PTA, and assume that refuter wins the parity game $\mathcal{G}(A)$ from the initial position q_0.

   Explain how a winning strategy for refuter can be used to define a tree in $\mathcal{T} \in \mathcal{L}(A)$. Make this formal by explaining the construction of the set of nodes $\mathcal{T}$ and its labeling function $label_{\mathcal{T}}$.

b) Consider automaton A_2 from Part b) of Exercise 1 from the last exercise sheet.

$$A_2 = \Big(\underbrace{\{a_{/2}, b_{/2}\}}_{\Sigma}, \underbrace{\{q_+, q_-\}}_{Q}, q_+, \rightarrow, \Big(q_+ \overset{\Omega}{\mapsto} 1, q_- \overset{\Omega}{\mapsto} 0 \Big) \Big) \quad \text{with}$$

$$\rightarrow_a = \{ (q_+, (q_-, q_-)) \},$$
$$\rightarrow_b = \{ (q_+, (q_+, q_-)),\ (q_+, (q_-, q_+)),\ (q_-, (q_-, q_-)) \}.$$

Transform the automaton to a language-equivalent automaton that has at least one transition $(q, \vec{q}) \in \rightarrow_a$ for each source state q and symbol a. (This will ensure that the parity game is deadlock-free.)

Construct the parity game $\mathcal{G}(A)$ and identify a positional winning strategy for refuter. How does the tree described by the strategy look like?

Hint: Restrict yourself to positions Q^2 of prover that can actually occur during a play of the game. This prevents the game arena from becoming excessively large.

Exercise 3: Describing tree languages using S2S

We will introduce and discuss S2S in the exercise class on June 20. Afterwards, I will also add an explanation to the lecture notes.

Consider the Alphabet $\Sigma = \{a_{/2}, b_{/2}\}$. Our goal is to create a closed S2S-formula for the language $\mathcal{L}$ of trees in which exactly one branch contains infinitely many as (known from Part c) of Exercise 1 from the last exercise sheet).

a) Consider the following S2S formula that has the free second-order variable X.

$$\begin{aligned}
\text{Branch}(X) \quad = \quad & \varepsilon \in X & (1)\\
\wedge \quad & \forall x\colon x \in X \rightarrow \exists y \exists z\colon S_0(x,y) \wedge S_1(x,z) \wedge (y \in X \oplus z \in X) & (2)\\
\wedge \quad & \forall y\colon (y \in X \wedge y \neq \varepsilon) \rightarrow \exists x\colon x \in X \wedge \big(S_0(x,y) \vee S_1(x,y)\big) & (3)
\end{aligned}$$

Here, $\oplus$ is XOR and $\rightarrow$ is implication. They can be easily rewritten using negation, conjunction, and disjunction.

Argue that Branch(X) evaluates to true under a structure $\mathcal{S}(\mathcal{T})$ and an interpretation $\mathcal{I}_\mathcal{T}$ if and only if $\mathcal{I}_\mathcal{T}(X)$ is a set of positions that forms a branch of $\mathcal{T}$. Explain the purpose of each Line (1) - (3).

b) In S2S, we only have an equality predicate for first-order terms. Construct a formula Equal(X, Y) with two free second-order variables X, Y that evaluates to true under a structure $\mathcal{S}(\mathcal{T})$ and an interpretation $\mathcal{I}_\mathcal{T}$ if and only if $\mathcal{I}_\mathcal{T}(X) = \mathcal{I}_\mathcal{T}(Y)$.

c) Construct formulas $\text{Fin}_a(X)$ respectively $\text{Inf}_a(X)$ with one free second-order variable X that evaluate to true under a structure $\mathcal{S}(\mathcal{T})$ and an interpretation $\mathcal{I}_\mathcal{T}$ if and only if $\mathcal{I}_\mathcal{T}(X)$ contains only finitely many respectively infinite many nodes labeled by a.

For simplicity, you may suppose that $\mathcal{I}_\mathcal{T}(X)$ is a branch of $\mathcal{T}$.

d) Combine the previous parts of this exercises to construct a closed S2S-formula $\varphi_\mathcal{L}$ that evaluates to true under a structure $\mathcal{S}(\mathcal{T})$ if and only if $\mathcal{T} \in \mathcal{L}$.