

0.1 Fritz & Wilke

0.1.1 Delayed Simulation Game

In this section we consider delayed simulation games and variants thereof on DPAs. This approach is based on the paper [1] which considered the games for alternating parity automata. The DPAs we use are a special case of these APAs and therefore worth examining.

Definition 0.1.1. We define $\leq_{\checkmark} \subseteq (\mathbb{N} \cup \{\checkmark\}) \times (\mathbb{N} \cup \{\checkmark\})$ as follows:

- For $i, j \in \mathbb{N}$, we set $i \leq_{\checkmark} j$ iff $i \leq j$.
- For all $i \in \mathbb{N}$, we have $i \leq_{\checkmark} \checkmark$ and $\checkmark \not\leq_{\checkmark} i$.
- $\checkmark \leq_{\checkmark} \checkmark$.

Further, we define an order of “goodness” on parity priorities as $\preceq_p \subseteq \mathbb{N} \times \mathbb{N}$ as $0 \preceq_p 2 \preceq_p 4 \preceq_p \dots \preceq_p 5 \preceq_p 3 \preceq_p 1$.

Definition 0.1.2. Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ be a DPA. We define the *delayed simulation automaton* $\mathcal{A}_{de}(p, q) = (Q_{de}, \Sigma, (p, q, \checkmark), \delta_{de}, F_{de})$, which is a deterministic Büchi automaton, as follows.

- $Q_{de} = Q \times Q \times (\text{img}(c) \cup \{\checkmark\})$, i.e. the states are given as triples in which the first two components are states from \mathcal{A} and the third component is either a priority from \mathcal{A} or \checkmark .
- The alphabet remains Σ .
- The starting state is a triple (p, q, \checkmark) , where $p, q \in Q$ are parameters given to the automaton.
- $\delta_{de}((p, q, i), a) = (p', q', \gamma(i, c(p'), c(q')))$, where $p' = \delta(p, a)$, $q' = \delta(q, a)$, and γ is defined in detail below. The first two components behave like a regular product automaton.
- $F_{de} = Q \times Q \times \{\checkmark\}$.

$\gamma : \mathbb{N} \times \mathbb{N} \times (\mathbb{N} \cup \{\checkmark\}) \rightarrow \mathbb{N} \cup \{\checkmark\}$ is the update function of the third component and defines the “obligations” as they are called in [1]. It is defined as

$$\gamma(i, j, k) = \begin{cases} \checkmark & \text{if } i \text{ is odd and } i \leq_{\checkmark} k \text{ and } j \preceq_p i \\ \checkmark & \text{if } j \text{ is even and } j \leq_{\checkmark} k \text{ and } j \preceq_p i \\ \min_{\leq_{\checkmark}} \{i, j, k\} & \text{else} \end{cases}$$

Definition 0.1.3. Let \mathcal{A} be a DPA and let \mathcal{A}_{de} be the delayed simulation automaton of \mathcal{A} . We say that a state p *de-simulates* a state q if $L(\mathcal{A}_{de}(p, q)) = \Sigma^\omega$. In that case we write $p \leq_{de} q$. If also $q \leq_{de} p$ holds, we write $p \equiv_{de} q$.

Lemma 0.1.1. γ is monotonous in the third component, i.e. if $k \leq_{\checkmark} k'$, then $\gamma(i, j, k) \leq_{\checkmark} \gamma(i, j, k')$ for all $i, j \in \mathbb{N}$.

Proof. We consider each case in the definition of γ . If i is odd, $i \leq_{\checkmark} k$ and $j \preceq_p i$, then also $i \leq_{\checkmark} k'$ and $\gamma(i, j, k) = \gamma(i, j, k') = \checkmark$.

If j is even, $j \leq_{\checkmark} k$ and $j \preceq_p i$, then also $j \leq_{\checkmark} k'$ and $\gamma(i, j, k) = \gamma(i, j, k') = \checkmark$.

Otherwise, $\gamma(i, j, k) = \min_{\leq_{\checkmark}} \{i, j, k\}$ and $\gamma(i, j, k') = \min_{\leq_{\checkmark}} \{i, j, k'\}$. Since $k \leq_{\checkmark} k'$, $\gamma(i, j, k) \leq_{\checkmark} \gamma(i, j, k')$. \square

Lemma 0.1.2. *Let \mathcal{A} be a DPA and let $p, q \in Q$, $k \in \mathbb{N} \cup \{\checkmark\}$. If the run of \mathcal{A}_{de} starting at (p, q, k) on some $\alpha \in \Sigma^\omega$ is accepting, then for all $k \leq_{\checkmark} k'$ also the run of \mathcal{A}_{de} starting at (p, q, k') on α is accepting.*

Proof. Let ρ be the run starting at (p, q, k) and let ρ' be the run starting at (p, q, k') . Further, let p_i, q_i, k_i , and k'_i be the components of the states of those runs in the i -th step. Via induction we show that $k_i \leq_{\checkmark} k'_i$ for all i . Since k_i is \checkmark infinitely often, the same must be true for k'_i and ρ' is accepting.

For $i = 0$, we have $k_0 = k \leq_{\checkmark} k' = k'_0$. Otherwise, we have $k_{i+1} = \gamma(c(p_{i+1}), c(q_{i+1}), k_i)$ and k'_{i+1} analogously. The rest follows from Lemma ??.

Lemma 0.1.3. *Let \mathcal{A} be a DPA. Then \leq_{de} is reflexive and transitive.*

Proof. For reflexivity, we need to show that $q \leq_{\text{de}} q$ for all states q . This is rather easy to see. For a word $\alpha \in \Sigma^\omega$, the third component of states in the run of $\mathcal{A}_{\text{de}}(q, q)$ on α is always \checkmark , as $\gamma(i, i, \checkmark) = \checkmark$.

For transitivity, let $q_1 \leq_{\text{de}} q_2$ and $q_2 \leq_{\text{de}} q_3$. Our goal is to prove $q_1 \leq_{\text{de}} q_3$.

Lemma 0.1.4. *Let \mathcal{A} be a DPA. Then \equiv_{de} is a congruence relation.*

Proof. The three properties that are required for \equiv_{de} to be an equivalence relation are rather easy to see. Reflexivity and transitivity have been shown for \leq_{de} already and symmetry follows from the definition. Congruence requires more elaboration.

Let $p \equiv_{\text{de}} q$ be two equivalent states. Let $a \in \Sigma$ and $p' = \delta(p, a)$ and $q' = \delta(q, a)$. We have to show that also $p' \equiv_{\text{de}} q'$. Towards a contradiction, assume that $p' \not\leq_{\text{de}} q'$, so there is a word $\alpha \notin L(\mathcal{A}_{\text{de}}(p', q'))$. Let $(p', q', k) = \delta_{\text{de}}((p, q, \checkmark), a)$. By Lemma ??, the run of \mathcal{A}_{de} on α from (p', q', k) cannot be accepting; otherwise, the run of \mathcal{A}_{de} from (p', q', \checkmark) would be accepting and $\alpha \in L(\mathcal{A}_{\text{de}}(p', q'))$. Hence, $a\alpha \notin L(\mathcal{A}_{\text{de}}(p, q))$, which means that $p \not\equiv_{\text{de}} q$.