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 [], 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. For convenience, we define two orders for this chapter. First, we introduce ✓ as an "infinity" to the natural numbers and define the **obligation order** $\leq_{\checkmark} \subseteq (\mathbb{N} \cup \{\checkmark\}) \times (\mathbb{N} \cup \{\checkmark\})$ as $0 \leq_{\checkmark} 1 \leq_{\checkmark} 2 \leq_{\checkmark} \cdots \leq_{\checkmark} \checkmark$.

Second, we define an order of "goodness" on parity priorities $\preceq_{\mathrm{p}} \subseteq \mathbb{N} \times \mathbb{N}$ as $0 \preceq_{\mathrm{p}} 2 \preceq_{\mathrm{p}} 4 \preceq_{\mathrm{p}} \cdots \preceq_{\mathrm{p}} 5 \preceq_{\mathrm{p}} 3 \preceq_{\mathrm{p}} 1$.

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

- $Q_{\mathrm{de}} = Q \times Q \times (\mathrm{img}(c) \cup \{\checkmark\})$, 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 ✓.
- The alphabet remains Σ.
- The starting state is a triple $(p, q, \gamma(c(p), c(q), \checkmark))$, where $p, q \in Q$ are parameters given to the automaton, and γ is defined below.
- $\delta_{\mathrm{de}}((p, q, k), a) = (p', q', \gamma(c(p'), c(q'), k)$, where $p' = \delta(p, a)$, $q' = \delta(q, a)$, and γ is the same function as used in the initial state. The first two components behave like a regular product automaton.
- $F_{\mathrm{de}} = Q \times Q \times \{\checkmark\}$.

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

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

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

$\equiv_{\mathrm{de}}$ is a congruence relation.

Our overall goal is to use $\equiv_{\mathrm{de}}$ to build a quotient automaton of our original DPA. The first step towards this goal is to show that the result is actually a well-defined DPA, by proving that the relation is a congruence.

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 $j \preceq_{\mathrm{p}} i$, 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 $j \preceq_{\mathrm{p}} i$, then also $j \leq_{\checkmark} k'$ and $\gamma(i,j,k) = \gamma(i,j,k') = \checkmark$.

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

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 $\mathcal{A}_{\mathrm{de}}$ starting at (p,q,k) on some $\alpha \in \Sigma^\omega$ is accepting, then for all $k \leq_{\checkmark} k'$ also the run of $\mathcal{A}_{\mathrm{de}}$ 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 0.1.1. □

Lemma 0.1.3. *Let $\mathcal{A}$ be a DPA and $\rho \in Q_{\mathrm{de}}^\omega$ be a run of $\mathcal{A}_{\mathrm{de}}$ on some word. Let $k \in (\mathbb{N} \cup \{\checkmark\})^\omega$ be the third component during ρ. For all i, $k(i+1) \leq_{\checkmark} k(i)$ or $k(i+1) = \checkmark$.*

Proof. Follows directly from the definition of γ. □

Lemma 0.1.4. *Let $\mathcal{A}$ be a DPA with states $p, q \in Q$. For a word $\alpha \in \Sigma^\omega$, let $\rho : i \mapsto (p_i, q_i, k_i)$ be the run of $\mathcal{A}_{\mathrm{de}}(p,q)$ on α. If ρ is not accepting, there is a position n such that*

- *$Occ(\{p_i \mid i \geq n\}) = Inf(\{p_i \mid i \in \mathbb{N}\})$,*
- *$Occ(\{q_i \mid i \geq n\}) = Inf(\{q_i \mid i \in \mathbb{N}\})$,*
- *For all $i \geq j \geq n$, $k_i = k_j$ and $k_i \neq \checkmark$.*

In other words, from n on, p and q only see states that are seen infinitely often, and the obligations of ρ do not change anymore.

Proof. The first two requirements are clear. Since states not in $\mathrm{Inf}(\{p_i \mid i \in \mathbb{N}\})$ only occur finitely often, there must be positions n_p and n_q from which on they do not occur anymore at all.

For the third requirement, we know that from some point on, the obligations k never become $\checkmark$ anymore, as the run would be accepting otherwise. By Lemma 0.1.3, k can only become lower from there on. As $\leq_{\checkmark}$ is a well-ordering, a minimum must be reached at some point n_k.

The position $n_0 = \max\{n_p, n_q, n_k\}$ satisfies the statement. □

Lemma 0.1.5. *Let $\mathcal{A}$ be a DPA with two states $p, q \in Q$. Let $\alpha \in \Sigma^\omega$ be an ω-word and let ρ_p and ρ_q be the respective runs of $\mathcal{A}$ on α starting in p and q. If $\min \mathit{Inf}(c(\rho_q)) \preceq_p \min \mathit{Inf}(c(\rho_p))$, then $\alpha \in L(\mathcal{A}_{\mathrm{de}}(p, q))$.*

Proof. We write $l_q = \min \mathrm{Inf}(c(\rho_q))$ and $l_p = \min \mathrm{Inf}(c(\rho_p))$. Assume that the Lemma is false, so $l_q \preceq_p l_p$ but $\alpha \notin L(\mathcal{A}_{\mathrm{de}}(p, q))$. Let $k_{13} \in (\mathbb{N} \cup \{\checkmark\})^\omega$ be the third component of the run of $\mathcal{A}_{\mathrm{de}}(p, q)$ on α. Let n_0 be a position as described in Lemma 0.1.4 (for ρ).

Case 1: l_q is even and $l_q \leq l_p$ We know $k_{13}(n_0) = l_q$, as that is the smaller value. Let $m > n_0$ be a position with $c(\rho_q(m)) = l_q$. Then $c(\rho_q(m))$ is even and $c(\rho_q(m)) = l_q \leq l_p \leq c(\rho_p(m))$, so $c(\rho_q(m)) \preceq_p c(\rho_p(m))$. Also we have $c(\rho_q(m)) \leq k_{13}(m-1) = l_q$ and therefore $k_{13}(m) = \checkmark$, which contradicts the choice of n_0.

Case 2: l_p is odd and $l_q \geq l_p$ We know $k_{13}(n_0) = l_p$. Let $m > n_0$ be a position with $c(\rho_p(m)) = l_p$. Then $c(\rho_p(m))$ is odd and $c(\rho_p(m)) = l_p \leq l_q \leq c(\rho_q(m))$, so $c(\rho_q(m)) \preceq_p c(\rho_p(m))$. By the same argumentation as above, we deduce $k_{13}(m) = \checkmark$. $\square$

Lemma 0.1.6. *Let $\mathcal{A}$ be a DPA. Then $\leq_{\mathrm{de}}$ is reflexive and transitive.*

Proof. For reflexivitiy, we need to show that $q \leq_{\mathrm{de}} q$ 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 $\mathcal{A}_{\mathrm{de}}(q, q)$ on α is always $\checkmark$, as $\gamma(i, i, \checkmark) = \checkmark$.

For transitivity, let $q_1 \leq_{\mathrm{de}} q_2$ and $q_2 \leq_{\mathrm{de}} q_3$. Assume towards a contradiction that $q_1 \not\leq_{\mathrm{de}} q_3$, so there is a word $\alpha \notin L(\mathcal{A}_{\mathrm{de}}(q_1, q_3))$. We consider the three runs ρ_{12}, ρ_{23}, and ρ_{13} of $\mathcal{A}_{\mathrm{de}}(q_1, q_2)$, $\mathcal{A}_{\mathrm{de}}(q_2, q_3)$, and $\mathcal{A}_{\mathrm{de}}(q_1, q_3)$ respectively on α. Then ρ_{12} and ρ_{23} are accepting, whereas ρ_{13} is not.

Moreover, we use the notation $q_1(i), q_2(i), q_3(i)$ for the states of the run and $k_{12}(i), k_{23}(i), k_{13}(i)$ for the obligations. More specifically for a run ρ_{ij}, it is true that $\rho_{ij}(n) = (q_i(n), q_j(n), k_{ij}(n))$.

Let n_0 be a position as described in Lemma 0.1.4 (for ρ_{13}) and let $l_j = \min\{c(q_j(i)) \mid i \geq n_0\}$ be the lowest priority that q_j reaches after n_0. This is equivalent to $l_j = \min \mathrm{Inf}(\{c(q_j(i)) \mid i \in \mathbb{N}\})$. We now show that $l_3 \preceq_p l_1$. By Lemma 0.1.5 this gives us $\alpha \in L(\mathcal{A}_{\mathrm{de}}(q_1, q_3))$, letting us conclude in a contradiction.

Case 1: l_2 is even. We claim that l_3 is even and $l_3 \leq l_2$.

First, to show $l_3 \leq l_2$, let $m \geq n_0$ be a position with $c(q_2(m)) = l_2$ and let $n \geq m$ be the minimal position with $k_{23}(n) = \checkmark$. If $m = n$, then $c(q_3(n)) \preceq_p c(q_2(n)) = l_2$ and therefore $c(q_3(n)) \leq l_2$. Otherwise, from m to $n-1$, k_{23} only grows smaller and is at most l_2 (Lemma 0.1.3). As the priority of q_2 never becomes an odd number smaller than l_2, the only way for $k_{23}(m)$ to be $\checkmark$ is that $c(q_3(m))$ is even and $c(q_3(m)) \leq k_{23}(m-1) \leq l_2$.

Second, assume that l_3 is odd and let m be a position with $c(q_3(m)) = l_3$. As l_2 is even, we have $k_{23}(m) \leq l_3 < l_2$. At no future position can $c(q_3)$ both be even and smaller than k_{23}, so k_{23} never becomes $\checkmark$ again. Thus, ρ_{23} is not accepting.

We claim that l_1 is odd or $l_1 \geq l_2$.

Towards a contradiction assume the opposite, so $l_1 < l_2$ and l_1 is even. Let $m \geq n_0$ be a position with $c(q_1(m)) = l_1$. Then $c(q_2(m)) \not\preceq_p c(q_1(m))$ and therefore $k_{12}(m) = l_1$. At no position after m can it happen that the conditions for k_{12} to become $\checkmark$ again are satisfied. Thus, ρ_{12} would not be accepting.

If l_1 is odd and l_3 is even, $l_3 \preceq_p l_1$ follows. For the other case, l_1 and l_3 both being even with $l_3 \leq l_2 \leq l_1$, that also holds.

Case 2: l_2 is odd. We skip the details of this case as it works symmetrically to case 1. In particular, we first show that l_1 is odd and $l_1 \leq l_2$. We continue with l_3 being even or $l_3 \geq l_2$. From these two statements, $l_3 \preceq_p l_1$ again follows. □

Theorem 0.1.7. *Let $\mathcal{A}$ be a DPA. Then $\equiv_{\mathrm{de}}$ is a congruence relation.*

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

Let $p \equiv_{\mathrm{de}} q$ be two equivalent states. Let $a \in \Sigma$ and $p' = \delta(p, a)$ and $q' = \delta(q, a)$. We have to show that also $p' \equiv_{\mathrm{de}} q'$. Towards a contradiction, assume that $p' \not\leq_{\mathrm{de}} q'$, so there is a word $\alpha \notin L(\mathcal{A}_{\mathrm{de}}(p', q'))$. Let $(p', q', k) = \delta_{\mathrm{de}}((p, q, \checkmark), a)$. By Lemma 0.1.2, the run of $\mathcal{A}_{\mathrm{de}}$ on α from (p', q', k) cannot be accepting; otherwise, the run of $\mathcal{A}_{\mathrm{de}}$ from $(p', q', \checkmark)$ would be accepting and $\alpha \in L(\mathcal{A}_{\mathrm{de}}(p', q'))$. Hence, $a\alpha \notin L(\mathcal{A}_{\mathrm{de}}(p, q))$, which means that $p \not\equiv_{\mathrm{de}} q$. □

We want to mention here that $\equiv_{\mathrm{de}}$ is actually an equivalence relation on APAs as well, as was shown in the original paper. However, congruence is the key point at which deterministic automata diverge. Congruence requires something to be true for *all* successors of a state; delayed simulation only requires there to be *one* equivalent pair of successors. Only in deterministic automata is it that these two coincide.

Corollary 0.1.8. *Let $\mathcal{A}$ be a DPA and $\equiv_{\mathrm{de}}$ the corresponding delayed simulation-relation. The quotient automaton $\mathcal{A}/_{\equiv_{\mathrm{de}}}$ is well-defined and deterministic.*

Correctness of the quotient

The quotient automaton itself is used "only" for state space reduction. The main point of delayed simulation is that the priorities of equivalent states can be made equivalent.

Theorem 0.1.9. *Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ be a DPA. Let $\sim \subseteq Q \times Q$ be a congruence relation such that $p \sim q$ implies $c(p) = c(q)$. Then $L(\mathcal{A}) = L(\mathcal{A}/_\sim)$.*

Proof. Since $\mathcal{A}$ is deterministic and $\sim$ is a congruence relation, $\mathcal{A}/_\sim = (Q_\sim, \Sigma, [q_0]_\sim, \delta_\sim, c_\sim)$ is deterministic as well. Let $\alpha \in \Sigma^\omega$ be a word and let π and ρ be the runs of $\mathcal{A}$ and $\mathcal{A}/_\sim$.

For each $i \in \mathbb{N}$, we have $\rho(i) = [\pi(i)]_\sim$ and $c_\sim(\rho(i)) = c(\pi(i))$. Thus, $\mathrm{Inf}(c(\pi)) = \mathrm{Inf}(c(\rho))$ and π is accepting iff ρ is accepting. □

Lemma 0.1.10. *Let $\mathcal{A}$ be a DPA and let π and ρ be runs of $\mathcal{A}$ on the same word but starting at different states. If $\pi(0) \equiv_{de} \rho(0)$, then $\min Occ(c(\pi)) = \min Occ(c(\rho))$.*

Proof. Let $k = \min \mathrm{Occ}(c(\pi))$ and $l = \min \mathrm{Occ}(c(\rho))$. Assume towards a contradiction without loss of generality that $k < l$. Let α be the word that is read by the two runs.

If k is even, let σ be the run of $\mathcal{A}_{\mathrm{de}}(\pi(0), \rho(0))$ on α. Let n be a position at which $c(\pi(n)) = k$. We claim that for all $i \geq n$, the third component of $\sigma(i)$ is k.

At $\sigma(n)$, this must be true because $k < l \leq c(\rho(n))$ and thus $c(\rho(n)) \not\leq_p c(\pi(n))$. At all positions after n, it can never occur that $c(\rho(i)) \leq k$ or that $c(\pi(i))$ is odd and smaller than k. The rest follows from the definition of γ.

If k is odd, we can argue similarly on the run of $\mathcal{A}_{\text{de}}(\rho(0), \pi(0))$. As soon as $c(\pi)$ reaches its minimum, the third component of the run will never change again. □

Theorem 0.1.11. *Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ be a DPA and let $p, q \in Q$ with $p \equiv_{\text{de}} q$ and $c(p) < c(q)$. Define $\mathcal{A}' = (Q, \Sigma, q_0, \delta, c')$ with $c'(s) = \begin{cases} c(p) & \text{if } s = q \\ c(s) & \text{else} \end{cases}$. Then $L(\mathcal{A}) = L(\mathcal{A}')$.*

Proof. First, consider the case that $c(p)$ is an even number. The parity of each state is at least as good in $\mathcal{A}'$ as it is in $\mathcal{A}$, so $L(\mathcal{A}) \subseteq L(\mathcal{A}')$. For the other direction, assume there is a $\alpha \in L(\mathcal{A}') \setminus L(\mathcal{A})$, so the respective run $\rho \in Q^\omega$ is accepting in $\mathcal{A}'$ but not in $\mathcal{A}$.

For this to be true, ρ must visit q infinitely often and $c'(q)$ must be the lowest priority that occurs infinitely often; otherwise, the run would have the same acceptance in both automata. Thus, there is a finite word $w \in \Sigma^*$ such that from q, $\mathcal{A}$ reaches again q via w and inbetween only priorities greater than $c'(q)$ are seen.

Now consider the word w^ω and the run π_q of $\mathcal{A}$ on said word starting in q. With the argument above, we know that the minimal priority occuring in $c(\pi)$ is greater than $c'(q)$. If we take the run π_p on w^ω starting at p though, we find that this run sees priority $c(p) = c'(q)$ at the very beginning. This contradicts Lemma 0.1.10, as $p \equiv_{\text{de}} q$. Thus, the described α cannot exist.

If $c(p)$ is an odd number, a very similar argumentation can be applied with the roles of $\mathcal{A}$ and $\mathcal{A}'$ reversed. We omit this repetition. □

Corollary 0.1.12. *For a DPA $\mathcal{A}$, the quotient automaton ${}^{\mathcal{A}}/_{\equiv_{\text{de}}}$ is a DPA that recognizes the same language.*

0.1.2 Using delayed simulation for APAs

0.1.3 Alternative computation

As we have seen, using delayed simulation to build a quotient automaton delivers good results in the number of removed states. The downside is the computation time which is much higher than that of our approach in section **??**. The question that we therefore deal with in the upcoming part is whether we can change the definition of $\equiv_{\mathrm{de}}$ to something that cannot be used for APAs anymore but is more efficient to compute.

Lemma 0.1.13. *Let $\mathcal{A}$ be a DPA with states $p, q \in Q$. Then $p \preceq_{\mathrm{de}} q$ if and only if the following property holds for all $w \in \Sigma^*$:*

Let $p' = \delta^*(p, w)$ and $q' = \delta^*(q, w)$. If $c(p')$ is even and $c(p') < c(q')$, then every path from q' eventually reaches a priority at most $c(p')$. On the other hand, if $c(q')$ is odd and $c(q') < c(p')$, then every path from p' eventually reaches a priority at most $c(q')$.

Proof. **If** We show the contrapositive. Let $p \not\preceq_{\mathrm{de}} q$, so there is a word $\alpha \notin L(\mathcal{A}_{\mathrm{de}}(p, q))$ with the run of $\mathcal{A}_{\mathrm{de}}(p, q)$ being $(p_0, q_0, k_0)(p_1, q_1, k_1)\dots$. By Lemma 0.1.4, there is a position n such that k_i does not change anymore for $i \geq n$. We define $w = \alpha[0, n_0]$ and $\beta = \alpha[n_0 + 1,]$ (so $\alpha = w\beta$) and claim that w is a valid counterexample for the right-side property.

The first case is: $c(p') < c(q')$ and $c(p')$ is even.
Reading β from q' induces a run that never visits a priority less or equal to $c(p')$: Let $u \sqsubset \beta$ and assume that $c(\delta^*(q', u)) \leq c(p')$. By choice of n, we know that $k_{|wu|} = k_{|wu|+1} \neq \checkmark$. This can only happen if the "else" case of γ is hit, meaning that $k_{|wu|+1} = \min\{k_{|wu|}, c(\delta^*(p', u)), c(\delta^*(q', u))\}$. Specifically, $c(\delta^*(q', u)) \geq k_{|wu|}$. By choice of w we also have $k_{|wu|} = k_{|w|} \leq c(p')$, so $c(\delta^*(q', u)) = c(p')$.

This, however, means that $c(\delta^*(q', u))$ is even, $c(\delta^*(q', u)) \leq_{\checkmark} k_{|wu|}$, and $c(\delta^*(q', u)) \preceq_p c(p')$ and thus $k_{|wu|+1} = \checkmark$ which is a contradiction.

The second case, $c(q') < c(p')$ and $c(q')$ is odd, works almost identically so we omit the proof here.

Only If Again we show the contrapositive: There is a $w \in \Sigma^*$ such that the right-side property is violated. Let this w now be chosen among all these words such that $\min\{c(p'), c(q')\}$ becomes minimal. We now show that $p \not\preceq_{\mathrm{de}} q$.

The first case is: $c(p') < c(q')$ and $c(p')$ is even.
Let $\beta \in \Sigma^\omega$ be a word such that the respective run from q' only sees priorities strictly greater than $c(p')$. Let $(p_0, q_0, k_0)(p_1, q_1, k_1)\dots$ be the run of $\mathcal{A}_{\mathrm{de}}(p, q)$ on $\alpha = w\beta$. We claim that $k_i \neq \checkmark$ for all $i > |w|$. If that is true, then the run is rejecting and $\alpha \notin L(\mathcal{A}_{\mathrm{de}}(p, q))$.

Assume towards a contradiction that k_i does become $\checkmark$ again at some point. Let $j \geq |w|$ be the minimal position with $k_{j+1} = \checkmark$. Then by definition of γ, $c(q_{j+1}) \leq k_j$ is even or $c(p_{j+1}) \leq k_j$ is odd. In the former case, we would have a contradiction to the choice of β. In the latter case, we would have a contradiction to the choice of w as a word with minimal priority at $c(p')$: since $c(p')$ is even, $c(p_{j+1}) < c(p')$ and from q_{j+1} there is a run that never reaches a smaller priority. Hence, $w \cdot \beta[0, j - |w|]$ would have been our choice for w instead.

The second case, $c(q') < c(p')$ and $c(q')$ is odd, works almost identically so we omit the proof here. □

While this characterization of $\preceq_{\mathrm{de}}$ seems arbitrary, it allows for an easier definition of $\equiv_{\mathrm{de}}$ as is seen in the following statement.

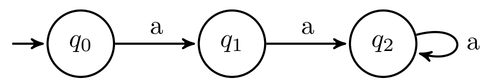

Figure 1: Example automaton in which $\sim_M$ and $\equiv_{\mathrm{de}}$ are not the same. $c(q_0) = 0, c(q_1) = 1, c(q_2) = 0$

Corollary 0.1.14. *Let $\mathcal{A}$ be a DPA with states $p, q \in Q$. Then $p \equiv_{\mathrm{de}} q$ if and only if the following holds for all words $w \in \Sigma^*$:*

Let $p' = \delta^*(p, w)$ and $q' = \delta^*(q, w)$. Every run that starts in p' or q' eventually sees a priority less than or equal to $\min\{c(p'), c(q')\}$.

This intermediate result now easily gives us the following relation of delayed simulation and Moore-equivalence.

Definition 0.1.4. Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, c)$ be a parity automaton. We call c *normalized* if for every state $q \in Q$ that does not lie in a trivial SCC and all priorities $k \leq c(q)$, there is a path from q to q such that the lowest priority visited is k.

Lemma 0.1.15. *Let $\mathcal{A}$ be a DPA with a normalized priority function and let p and q be states that do not lie in trivial SCCs. Then $p \equiv_{\mathrm{de}} q$ if and only if $p \sim_M q$.*

Proof. The "if"-implication was shown in **??**. For the other direction, let $p \not\sim_M q$, so there is a word $w \in \Sigma^*$ such that $c(p') \neq c(q')$, where $p' = \delta^*(p, w)$ and $q' = \delta^*(q, w)$. Without loss of generality, assume $c(p') < c(q')$.

As c is normalized, there is a word u such that q' reaches again q' via u and sees only priorities greater or equal to $c(q')$. That means that on the path that is obtained from q' by reading u^ω, the priority $c(p')$ is never visited. By corollary 0.1.14, that means $p \not\equiv_{\mathrm{de}} q$. □

If we can assure that our priority function is normalized, Moore-equivalence is a nice approximation of delayed simulation-equivalence. In fact, we can normalize the priority function and compute $\sim_M$-classes in $\mathcal{O}(n^2 k + n \log n)$ [].

An interesting question is whether there is an algorithm which uses this knowledge to compute exactly $\equiv_{\mathrm{de}}$ with the improved quadratic time. Figure 1 shows an example automaton with normalized c in which there are states (in trivial SCCs) that are delayed simulation-equivalent but not Moore-equivalent. In fact, all states are $\equiv_{\mathrm{de}}$-equivalent but none are $\sim_M$-equivalent.