

Figure 1: Example automaton in which the states could be merged but delayed simulation separates them.

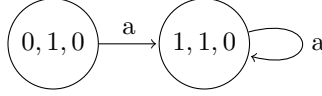


Figure 2: Example automaton in which the states could be merged but delayed simulation separates them.

0.0.1 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 ?? . In the following we will consider alternations to the delayed simulation algorithm with the goal to increase the number of removed states or to reduce computation time.

Resetting obligations

In the delayed simulation automaton, “obligations” correspond to good priorities that the first state has accumulated or bad priorities that the second state has accumulated and the need for the respective other state to compensate in some way. The intuitive idea behind this concept is that an obligation that cannot be compensated, stands for an infinite run in which the acceptance differs between the two states that are being compared. The issue with the original definition is that obligations carry over, even if they can only be caused finitely often. This is demonstrated in figure 1; the two states could be merged into one, but they are not \equiv_{de} -equivalent as can be seen in the delayed simulation automaton in figure 2.

As a solution to this, we propose a simple change to the definition of the automaton which resets the obligations every time, either state moves to a new SCC.

Definition 0.0.1. Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ be a DPA. We define the *delayed simulation automaton with SCC resets* $\mathcal{A}_{deR}(p, q) = (Q_{de}, \Sigma, (p, q), \gamma(c(p), c(q), \checkmark), \delta_{deR}, F_{de})$ with $\delta_{deR}((p, q, k), a) = \delta_{de}((p, q, \text{reset}(p, q, k, a)), a)$. Except for the addition of the reset function, this automaton is the same as \mathcal{A}_{de} .

If p and $\delta(p, a)$ lie in the same SCC, as well as q and $\delta(q, a)$, then we simply set $\text{reset}(p, q, k, a) = k$. Otherwise, i.e. if any state changes its SCC, the reset comes into play and we set $\text{reset}(p, q, k, a) = \checkmark$.

We write $p \leq_{deR} q$ if $L(\mathcal{A}_{deR}(p, q)) = \Sigma^\omega$. If also $q \leq_{deR} p$ holds, we write $p \equiv_{deR} q$.

As the definition is so similar to the original delayed simulation, most results that we have already proven translate directly to the new relation.

Theorem 0.0.1. \equiv_{deR} is a congruence relation.

Lemma 0.0.2. Let \mathcal{A} be a DPA with two states p and q . If $p \leq_{de} q$, then $p \leq_{deR} q$.

Proof. Consider the two simulation automata $\mathcal{A}_{\text{de}}(p, q)$ and $\mathcal{A}_{\text{deR}}(p, q)$ and let $\alpha \in \Sigma^\omega$ be an arbitrary word. Let $(p_i, q_i, k_i)_{i \in \mathbb{N}}$ and $(p_i, q_i, l_i)_{i \in \mathbb{N}}$ be the runs of these two automata on α . We claim that $k_i \leq_{\checkmark} l_i$ at every position. Then, since $L(\mathcal{A}_{\text{de}}(p, q)) = \Sigma^\omega$, both runs must be accepting.

We know $k_0 = l_0$ by definition. For the sake of induction, we look at position $i + 1$. If neither p_i nor q_i change their SCC in this step, the statement follows from lemma ?? . Otherwise, $l_{i+1} = \checkmark \geq_{\checkmark} k_{i+1}$. \square

The real difference comes up when looking at theorem ?? . If we consider again the example given in figure 1, if that theorem would hold the same for \equiv_{deR} we could assign both states the priority 0, which would then make the automaton accept every word. This is obviously not the same as the original automaton, so this statement deserves some additional inspection.

Lemma 0.0.3. *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. Let off_π and off_ρ be the positions after which π and ρ respectively only stay in one single SCC. Let $\text{off} = \max\{\text{off}_\pi, \text{off}_\rho\}$. If $\pi(0) \equiv_{\text{deR}} \rho(0)$, then $\min \text{Occ}(c(\pi[\text{off}, \omega])) = \min \text{Occ}(c(\rho[\text{off}, \omega]))$.*

Proof. Let $k = \min \text{Occ}(c(\pi[\text{off}, \omega]))$ and $l = \min \text{Occ}(c(\rho[\text{off}, \omega]))$. 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 $\mathcal{A}_{\text{de}}(\pi(0), \rho(0))$ on α . Let $n \geq \text{off}$ 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 . There is also never a change in SCCs anymore, by choice of n . 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. \square

Theorem 0.0.4. *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 $c'(s) = \begin{cases} c(p) & \text{if } s = q \\ c(s) & \text{else} \end{cases}$. If $\{p\}$ is not a trivial SCC in \mathcal{A} , 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 occurring 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 ?? , 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. \square

Delayed simulation with normalized priorities

Additional properties of delayed simulation can be found if we look at only a subclass of all DPAs. In particular, we use automata with normalized priorities here.

Definition 0.0.2. 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 .

Algorithm 1 shows how an equivalent normalized priority function can be computed in $\mathcal{O}(nk)$, where n and k are the number of states and priorities respectively. The algorithm is a slight adaption of that presented in [1], which is why we will not go into further details here and just refer to the original source.

Lemma 0.0.5. 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 $p \equiv_{de} q$ if and only if $p \sim_M q$.

Proof. The “if”-implication was shown in [1]. 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 [1], that means $p \not\equiv_{de} q$. \square

Iterated Moore equivalence

Our next approach differs greatly in its computation from the delayed simulation (or rather, it is not related at all to the delayed simulation automaton anymore) but will yield a result that is at least as good. As before, we focus on normalized DPAs here.

Definition 0.0.3. Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ and $\mathcal{B} = (P, \Sigma, p_0, \varepsilon, d)$ be DPAs and $S \subseteq Q$. We say that \mathcal{A} *prepends* S to \mathcal{B} if

- $q_0 \in S$
- $Q = P \dot{\cup} S$
- $\delta \upharpoonright_{P \times \Sigma} = \varepsilon$
- $c \upharpoonright_P = d$

We assume S to be an SCC in these use cases, i.e. from every $s \in S$, every other $s' \in S$ is reachable in \mathcal{A} .

Definition 0.0.4. Let \mathcal{A} be a DPA with SCCs $\mathcal{S} \subseteq 2^Q$ without unreachable states. Let $\preceq \subseteq \mathcal{S} \times \mathcal{S}$ be a total preorder such that $S \preceq S'$ implies that S' is reachable from S . For the i -th element w.r.t. this order, we write S_i , i.e. $S_0 \prec S_1 \prec \dots \prec S_{|\mathcal{S}|}$.

For every state q in \mathcal{A} , let $\text{SCC}(q)$ be the SCC of q and let $\text{SCCi}(q)$ be the index of that SCC, i.e. $\text{SCC}(q) = S_{\text{SCCi}(q)}$. Let $\preceq_Q \subseteq Q \times Q$ be a total preorder on the states such that $q \preceq_Q q'$ implies $\text{SCCi}(q) \leq \text{SCCi}(q')$.

We inductively define a sequence of automata $(\mathcal{B}_i)_{0 \leq i \leq |\mathcal{S}|}$. For every i , we write $\mathcal{B}_i = (Q_i, \Sigma, q_0^i, \delta_i, c_i)$.

Algorithm 1 Normalizing the priority function of a DPA.

```

1: function NORMALIZE( $\mathcal{A}$ )
2:    $c' : Q \rightarrow \mathbb{N}, q \mapsto c(q)$ 
3:    $M(\mathcal{A}, c')$ 
4:   return  $c'$ 
5: end function

6: function  $M(\mathcal{A} \upharpoonright_P, c')$ 
7:   if  $P = \emptyset$  then
8:     return 0
9:   end if
10:   $min \leftarrow 0$ 
11:  for SCC  $S$  in  $\mathcal{A} \upharpoonright_P$  do
12:     $m := \min c(S) \bmod 2$ 
13:     $X := c^{-1}(m)$ 
14:    for  $q \in X$  do
15:       $c'(q) \leftarrow m$ 
16:    end for
17:     $S' := S \setminus X$ 
18:     $m' \leftarrow M(\mathcal{A} \upharpoonright_{S'}, c')$ 
19:    if  $m'$  even then
20:      if  $m$  even then
21:         $\delta := m$ 
22:      else
23:         $\delta := m - 2$ 
24:      end if
25:    else
26:       $\delta := m - 1$ 
27:    end if
28:    for  $q \in S'$  do
29:       $c'(q) \leftarrow c'(q) - \delta$ 
30:    end for
31:     $min \leftarrow \min\{min, m\}$ 
32:  end for
33:  return  $min$ 
34: end function

```

- The state sets are defined as $Q_i = \bigcup_{j=i}^{|S|} S_j$.
- The base case is $\mathcal{B}_{|S|} = \mathcal{A} \upharpoonright_{S_{|S|}}$.
- Given that \mathcal{B}_{i+1} is defined, let $\mathcal{B}'_i = (Q_i, \Sigma, q_0^i, \delta'_i, c'_i)$ be a DPA that prepends S_i to \mathcal{B}_{i+1} such that $\delta \upharpoonright_{Q_i \times \Sigma} = \delta'_i$ and $c \upharpoonright_{S_i} = c'_i \upharpoonright_{S_i}$. If $i = 0$, we require $q_0^i = q_0$.
- Let $q_0^i = q_0^i$ and $\delta_i = \delta'_i$.
- Let M'_i be the Moore equivalence on \mathcal{B}'_i . If $S_i = \{q\}$ is a trivial SCC, q is not M'_i -equivalent to any other state, and there is a $p \in Q_{i+1}$ such that for all $a \in \Sigma$ $(\delta'_i(q, a), \delta'_i(p, a)) \in M'_i$, then let p_0 be \preceq_Q -maximal among those p and let $c_i(r) = \begin{cases} c_{i+1}(p_0) & \text{if } r = q \\ c_{i+1}(r) & \text{else} \end{cases}$. If any of the three conditions is false, simply set $c_i = c'_i$.

Let M_i be the Moore equivalence on \mathcal{B}_i . We define $\sim_{IM} := M_0$ and call this the *iterated Moore equivalence* of \mathcal{A} .

At first, this definition might seem complex and confusing when written down formally like this. More casually explained, we continuously add the SCCs of \mathcal{A} starting from the “back”. In addition to computing the usual Moore equivalence on our automaton, we also take trivial SCCs into special consideration; as their priority cannot appear infinitely often on any run, its value is effectively arbitrary. We can therefore perform extra steps to more liberally merge it with other states.

In the upcoming statements we prove the important properties of iterated Moore equivalence.

Lemma 0.0.6. *Let variables be as in definition 0.0.4. For all i and all $p, q \in Q_{i+1}$, $(p, q) \in M'_i$ iff $(p, q) \in M_{i+1}$. For all i and all $p, q \in Q_{i+1}$, $(p, q) \in M_i$ iff $(p, q) \in M'_i$.*

Proof.

□

Theorem 0.0.7. *Let variables be defined as in definition 0.0.4 and let \sim_M be the Moore equivalence of \mathcal{A} . Then $\sim_M \subseteq \sim_{IM}$.*

Proof. Let p, q be two states with $p \sim_M q$ but $p \not\sim_{IM} q$ such that $\max\{\text{SCCi}(p), \text{SCCi}(q)\}$ is maximal among all possible pairs; if there are multiple pairs with the highest value, choose a pair which has maximal $\min\{\text{SCCi}(p), \text{SCCi}(q)\}$ as well.

Without loss of generality we can assume that $c_0(p) \neq c_0(q)$ as we can always find such a pair with the definition of Moore equivalence. Since $c(p) = c(q)$, at least one of the two states must have a different priority in c_0 compared to c . By symmetry, assume that state is q . $\{q\}$ must be a trivial SCC in \mathcal{A} . Let q' be the \preceq_Q -maximal state s.t. $(\delta(q, a), \delta(q', a)) \in M'_{\text{SCCi}(q)}$ for all $a \in \Sigma$, that is the state whose priority was copied to q .

We can make the assumption that $\text{SCCi}(q) > \text{SCCi}(p)$: if $c(p) \neq c_0(\text{SCCi}(p))$, then $\{p\}$ is a trivial SCC and the choice of the states is symmetric. Otherwise, if we would have $\text{SCCi}(p) > \text{SCCi}(q)$ and $c(p) = c_0(\text{SCCi}(p))$, then $(p, q) \in M_{\text{SCCi}(q)} \subseteq \sim_{IM}$.

Consider the case that $\text{SCCi}(p)$ is a non-trivial SCC, so there is a non-empty word w with $\delta^*(p, w) = p$. Because of the congruence property, we know $q \sim_M p \sim_M \delta^*(p, w) \sim_M \delta^*(q, w)$. As $\{q\}$ is a trivial SCC, $\text{SCCi}(\delta^*(q, w)) > \text{SCCi}(q)$. That means that every state that is \sim_M -equivalent to $\delta^*(q, w)$ must also be \sim_{IM} -equivalent to it; otherwise, the choice of (p, q) as a pair with maximal

value of $\text{SCCi}(q)$ would be contradicted. In particular, $q \sim_{IM} \delta(q, w) \sim_{IM} p$, which breaks our initial assumption.

Finally, we look at the case that $\text{SCC}(p) = \{p\}$ is trivial. First, if $c(p) \neq c_0(p)$, then there must be a state p' s.t. $(\delta(p, a), \delta(p', a)) \in M'_{\text{SCCi}(p)}$ for all $a \in \Sigma$; consider the \preceq_Q -maximal state that satisfies this. We can show that p' must be the same as q' and therefore $c_0(p) = c_0(q') = c_0(q)$, which would be a contradiction. The only possibility for this to be false is that there is an $a \in \Sigma$ such that $(\delta(p, a), \delta(p', a)) \in M'_{\text{SCCi}(p)}$ but $(\delta(p, a), \delta(q', a)) \notin M'_{\text{SCCi}(p)}$. Remember that $(\delta(q, a), \delta(q', a)) \in M'_{\text{SCCi}(p)}$ and $\delta(q, a) \sim_M \delta(p, a)$. If now $(\delta(p, a), \delta(q', a)) \notin M'_{\text{SCCi}(p)}$ would hold, then $(\delta(p, a), \delta(q, a)) \notin M'_{\text{SCCi}(p)} \subseteq M_0$, so $\delta(p, a) \not\sim_{IM} \delta(q, a)$. Since $\{q\}$ is a trivial SCC, $\text{SCCi}(q) < \text{SCCi}(\delta(p, a))$, meaning that the pair $(\delta(p, a), \delta(q, a))$ would contradict our choice of (p, q) .

Second, if $c(p) = c_0(p)$, then there must be a state $p' \neq p$ s.t. $(p, p') \in M'_{\text{SCCi}(p)} \subseteq M_0$. Let that p' be \preceq_Q -maximal. $\text{SCC}(p')$ cannot be a trivial SCC: note that $q \sim_{IM} q'$ and $p \sim_{IM} p'$. Furthermore, $\delta(p, a) \sim_{IM} \delta(q, a)$ for all a , as to not contradict the pair (p, q) . Put together this means $\delta(p', a) \sim_{IM} \delta(q', a)$ for all a and therefore $c_0(p) = c_0(p') = c_0(q') = c_0(q)$.

Hence, $\text{SCC}(p')$ is a non-trivial SCC. There is a non-empty word w s.t. $\delta^*(p', w) = p'$. \square

Theorem 0.0.8. For a DPA \mathcal{A} , \mathcal{A}/\sim_{IM} is also a DPA with $L(\mathcal{A}/\sim_{IM}) = L(\mathcal{A})$.

Proof. From the definition of the \mathcal{B}_i sequence in the construction of \sim_{IM} , it becomes clear that \mathcal{B}_0 and \mathcal{A} are isomorphic up to the priority function at some trivial SCCs. As those priorities are only seen finitely often anyway, they do not impact the acceptance of a word. \square

Another nice and maybe surprising result is the relation of iterated Moore equivalence to delayed simulation.

Theorem 0.0.9. Let \mathcal{A} be a DPA. Then $\equiv_{de} \subseteq \sim_{IM}$.

Proof. \square