

## 0.1 Congruence Path Refinement

In this section we present an algorithm that uses an existing congruence relation and refines it to the point where equivalent states can be “merged”.

**Definition 0.1.1.** Let  $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$  be a DPA and let  $\equiv \subseteq Q \times Q$  be an equivalence relation on the state set. For every equivalence class  $\kappa \subseteq Q$ , let  $r_\kappa \in \kappa$  be an arbitrary representative of that class. For a DPA  $\mathcal{A}' = (Q', \Sigma, q'_0, \delta', c')$ , we say that  $\mathcal{A}'$  is a *representative merge* of  $\mathcal{A}$  w.r.t.  $\equiv$  if it satisfies the following:

- $Q' = \{r_{[q]_\equiv} \mid q \in Q\}$
- $q'_0 = r_{[q_0]_\equiv}$
- For all  $q \in Q'$  and  $a \in \Sigma$ :  $\delta'(q, a) = r_{[\delta(q, a)]_\equiv}$
- $c' = c \upharpoonright_{Q'}$

**Definition 0.1.2.** Let  $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$  be a DPA and let  $R \subseteq Q \times Q$  be a congruence relation on the state space. For any states  $p, q \in Q$ , let  $L_{[p]_R \rightarrow [q]_R} \subseteq \Sigma^*$  be the set of words  $w$  such that for any  $u \sqsubseteq w$ ,  $(\delta(p, u), q) \in R$  iff  $u \in \{\varepsilon, w\}$ . In other words, the set contains all minimal words by which the automaton reaches  $[q]_R$  from  $[p]_R$ .

Let  $\kappa \subseteq Q$  be an equivalence class of  $R$  and let  $p, q \in \kappa$ . We define  $R_\kappa \subseteq \kappa \times \kappa$  as the largest set  $(p, q) \in R_\kappa$  iff the following holds for all words  $w \in L_{\kappa \rightarrow \kappa}$ :

- $(\delta^*(p, w), \delta^*(q, w)) \in R_\kappa$
- $\min\{c(\delta^*(p, u)) \mid u \sqsubset w\} = \min\{c(\delta^*(q, u)) \mid u \sqsubset w\}$

Finally, we call  $\equiv_{PR}^R = \bigcup_{q \in Q} R_{[q]_R}$  the *path refinement* of  $R$ .

**Lemma 0.1.1.** *The path refinement is a well defined equivalence relation.*

*Proof.* We have to consider the sets  $L_{[p]_R \rightarrow [q]_R}$  and the sets  $R_\kappa$ . For  $L_{[p]_R \rightarrow [q]_R}$ , the definition works because  $R$  has the congruence property.

For  $R_\kappa$ , consider the following function  $f : 2^{Q \times Q} \rightarrow 2^{Q \times Q}$ :

$$f(X) = \{(p, q) \in X \mid \text{for all } w \in L_{\kappa \rightarrow \kappa} : (\delta^*(p, w), \delta^*(q, w)) \in X\}$$

$$Y_\kappa = \{(p, q) \in Q \times Q \mid \text{for all } w \in L_{\kappa \rightarrow \kappa} : \min\{c(\delta^*(p, u)) \mid u \sqsubset w\} = \min\{c(\delta^*(q, u)) \mid u \sqsubset w\}\}$$

Now Let  $X_0 = Y_\kappa$  and  $X_{i+1} = f(X_i)$ .  $f$  is monotone w.r.t.  $\subseteq$ , so there must be a fixed point  $X_\infty$ . By Kleene’s fixed point theorem and from the definition of  $R_\kappa$ , we have  $X_\infty = \text{gfp}(f) = R_\kappa$ .

Every  $X_i$  is an equivalence relation on  $\kappa$ : for  $i = 0$ , every state is only equivalent to itself, and for  $i > 0$ , the three properties can easily be verified via induction. Hence,  $X_\infty = R_\kappa$  is also an equivalence relation. All  $R_\kappa$  are disjoint and thus  $\equiv_{PR}^R$  has to be an equivalence relation as well.  $\square$

**Theorem 0.1.2.** *Let  $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$  be a DPA and let  $R \subseteq Q \times Q$  be a congruence relation that implies language equivalence. Let  $\mathcal{B}$  be a representative merge of  $\mathcal{A}$  w.r.t.  $\equiv_{PR}^R$ . Then  $L(\mathcal{A}) = L(\mathcal{B})$ .*

*Proof.* Let  $n$  be the number of non-trivial equivalence classes in  $\equiv_{\text{PR}}^R$ , i.e. classes with size greater than 1. If  $n = 0$ , then  $p \equiv_{\text{PR}}^R q$  iff  $p = q$  and therefore  $\mathcal{B} = \mathcal{A}$ .

Now assume for an argument of induction that the statement is true for  $n$  and we want to show that it still holds for  $n + 1$  classes. Let  $\kappa \subseteq Q$  be an arbitrary non-trivial equivalence class of  $\equiv_{\text{PR}}^R$ . Let  $\mathcal{A}' = (Q', \Sigma, q'_0, \delta', c')$  be the representative merge of  $\mathcal{A}$  w.r.t.  $\equiv_{\text{PR}}^R \upharpoonright_{\kappa}$  with the same representative  $r_\kappa$  as in  $\mathcal{B}$ . The path refinement equivalence of  $\mathcal{A}'$  then is equal to  $\equiv_{\text{PR}}^R \upharpoonright_{Q'}$  and has  $n$  non-trivial equivalence classes (as  $\kappa$  was merged into a single state). By induction,  $L(\mathcal{A}') = L(\mathcal{B})$ . It remains to be proven that  $L(\mathcal{A}) = L(\mathcal{A}')$ .

Let  $\alpha \in \Sigma^\omega$  be a word with runs  $\rho \in Q^\omega$  and  $\rho' \in (Q')^\omega$  of  $\mathcal{A}$  and  $\mathcal{A}'$  respectively. Let  $\lambda \subseteq Q$  be the equivalence class of  $R$  from which  $\kappa$  was extracted.

**Claim 1:** At every position  $i$ ,  $\rho(i) \in \kappa$  iff  $\rho'(i) \in \kappa$ .

Let  $k_0$  be the first position at which  $\rho(k_0) \in \kappa$  is true. For all  $i < k_0$ , we have  $\rho(i) = \rho'(i)$ , and at  $k_0$  we have  $\rho(k_0) \equiv_{\text{PR}}^R r_\kappa = \rho'(k_0)$ .

Now assume that the claim holds for all  $i \leq k$ , where  $k$  is a position at which  $\rho(k) \in \kappa$ . Let  $l > k$  be the next position at which  $\rho(l) \in \lambda$ . If  $l$  does not exist, then neither  $\rho(i)$  nor  $\rho'(i)$  are elements of  $\kappa$  for any  $i > k$ .

Let  $w = \alpha[k, l]$ . Since  $\kappa \subseteq \lambda$ ,  $w \in L_{\lambda \rightarrow \lambda}$ . By definition of  $\equiv_{\text{PR}}^R$ , that means  $\delta^*(\rho(k), w) = \rho(l) \equiv_{\text{PR}}^R \delta^*(\rho'(k), w)$ . Between  $k$  and  $l - 1$ , no redirected edge is used in  $\rho'$ , so  $\delta^*(\rho'(k), \alpha[k, l - 1]) = \rho'(l - 1)$ . Finally,  $\rho'(l) = \delta'(\rho'(l - 1), \alpha(l)) = r_{[\delta(\rho'(l - 1), \alpha(l))]} \equiv_{\text{PR}}^R \delta(\rho'(l - 1), \alpha(l)) = \delta^*(\rho'(k), w)$ .

Thus,  $\rho(l) \equiv_{\text{PR}}^R \rho'(l)$ .

Now, if  $\rho(l) \in \kappa$ , then  $\rho(l + 1) \in \kappa$  and our proof of induction is complete. If  $\rho(l) \notin \kappa$ , then  $\rho'(l) = \rho(l)$ , so the runs visit the same states in all positions until  $\kappa$  is reached again. This also completes the proof of our claim.

**Claim 2:** If  $\kappa$  only occurs finitely often in  $\rho$  and  $\rho'$ , then  $\rho$  is accepting iff  $\rho'$  is accepting.

Let  $k \in \mathbb{N}$  be the last position at which  $\rho(k)$  and  $\rho'(k)$  are in  $\kappa$ . From this point on,  $\rho'[k, \omega]$  is also a valid run of  $\mathcal{A}$  on  $\alpha[k, \omega]$ .  $\rho(k) \equiv_{\text{PR}}^R \rho'(k)$ , so  $(\rho(k), \rho'(k)) \in R$ . As  $R$  implies language equivalence, reading  $\alpha[k, \omega]$  from either state in  $\mathcal{A}$  leads to the same acceptance status. This also means that  $\rho'(k)$  has the same acceptance status as  $\rho(k)$ .

**Claim 3:** If  $\kappa$  occurs infinitely often in  $\rho$  and  $\rho'$ , then  $\rho$  is accepting iff  $\rho'$  is accepting.

Let  $(k_i)_{i \in \mathbb{N}}$  be all positions at which  $\kappa$  is visited. For each  $k_i$ , let  $l_i > k_i$  be the minimal position at which  $\rho(l_i) \in \lambda$ . In two steps, we first show that  $c(\rho[l_i, k_{i+1}]) = c'(\rho'[l_i, k_{i+1}])$  and second that  $\min \text{Occ}(c(\rho[k_i, l_i])) = \min \text{Occ}(c'(\rho'[k_i, l_i]))$ . Together, these results mean that the minimal priority that is seen infinitely often in the two runs is the same.

First, observe that at every  $l_i$ , we either have  $l_i = k_{i+1}$  (if  $\rho(l_i) \in \kappa$ ) or  $\rho(l_i) = \rho'(l_i)$ . In the first case,  $\rho[l_i, k_{i+1}]$  is empty, so  $c(\varepsilon) = c'(\varepsilon)$  is true. In the second case,  $\rho[l_i, k_{i+1}] = \rho'[l_i, k_{i+1}]$  and therefore  $c(\rho[l_i, k_{i+1}]) = c'(\rho'[l_i, k_{i+1}])$ .

Second, let  $w_i = \alpha[k_i, l_i]$ . Then  $\alpha \in L_{\lambda \rightarrow \lambda}$  and  $\min \text{Occ}(c(\rho[k_i, l_i])) = \min \text{Occ}(c'(\rho'[k_i, l_i]))$  holds directly by definition of  $\equiv_{\text{PR}}^R$ .  $\square$

### 0.1.1 Algorithmic Definition

The definition of path refinement that we introduced is useful for the proofs of correctness. It however does not provide one with a way to actually compute the relation. That is why we now provide an alternative definition that yields the same results but is more algorithmic in nature.

**Definition 0.1.3.** Let  $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$  be a DPA and let  $R \subseteq Q \times Q$  be a congruence relation. For each equivalence class  $\lambda$  of  $R$ , we define the *path refinement automaton*  $\mathcal{G}_{\text{PR}}^{R, \lambda}(p, q) = (Q_{\text{PR}}, \Sigma, q_{0, \text{PR}}^{p, q}, \delta_{\text{PR}}^\lambda, F_{\text{PR}})$ , which is a DFA.

- $Q_{\text{PR}} = (Q \times Q \times c(Q) \times \{<, >, =\}) \cup \{\perp\}$
- $q_{0, \text{PR}}^{p, q} = (p, q, \eta_k(c(p), c(q), \checkmark), \eta_x(c(p), c(q), \checkmark, =))$
- $\delta_{\text{PR}}^\lambda((p, q, k, x), a) = \begin{cases} (p', q', \eta_k(c(p'), c(q'), k), \eta_x(c(p'), c(q'), k, x)) & \text{if } p' \notin \lambda \\ q_{0, \text{PR}}^{p', q'} & \text{if } p' \in \lambda \text{ and } (x = =) \\ \perp & \text{else} \end{cases}$   
 where  $p' = \delta(p, a)$  and  $q' = \delta(q, a)$ .  
 $\eta_k(k_p, k_q, k) = \min_{\leq_\checkmark} \{k_p, k_q, k\}$   
 $\eta_x(k_p, k_q, k, x) = \begin{cases} < & \text{if } (k_p <_\checkmark k_q \text{ and } k_p <_\checkmark k) \text{ or } (k < k_q \text{ and } (x = <)) \\ > & \text{if } (k_p >_\checkmark k_q \text{ and } k >_\checkmark k_q) \text{ or } (k_p > k \text{ and } (x = >)) \\ = & \text{else} \end{cases}$
- $F_{\text{PR}} = Q_{\text{PR}} \setminus \{\perp\}$

**Lemma 0.1.3.** Let  $\mathcal{A}$  be a DPA with a congruence relation  $R$ . Let  $\lambda$  be an equivalence class of  $R$ ,  $p, q \in \lambda$ , and  $w \in L_{\lambda \rightarrow \lambda}$ . For every  $v \sqsubset w$  and  $\oplus \in \{<, >, =\}$ , the fourth component of  $(\delta_{\text{PR}}^\lambda)^*(q_{0, \text{PR}}, v)$  is  $\oplus$  if and only if  $\min\{c(\delta^*(p, u)) \mid u \sqsubseteq v\} \oplus \min\{c(\delta^*(q, u)) \mid u \sqsubseteq v\}$ .

The proof of this Lemma is a very formal analysis of every case in the relations between the different priorities that occur and making sure that the definition of  $\eta_x$  covers these correctly. No great insight is gained, which is why we omit the proof at this point.

**Theorem 0.1.4.** Let  $\mathcal{A}$  be a DPA with a congruence relation  $R$ . Let  $\lambda$  be an equivalence class of  $R$  and  $p, q \in \lambda$ . Then  $p \equiv_{\text{PR}}^R q$  iff  $L(\mathcal{G}_{\text{PR}}^{R, \lambda}(p, q)) = \Sigma^*$ .

*Proof.* **If** Let  $p \not\equiv_{\text{PR}}^R q$ . Similarly to the proof of Lemma ??, we use the inductive definition of  $R_\kappa \subseteq \equiv_{\text{PR}}^R$  using  $f$  and the sets  $X_i$  here. Let  $m$  be the smallest index at which  $(p, q) \notin X_m$ . Let  $\rho = (p_i, q_i, k_i, x_i)_{0 \leq i \leq |w|}$  be the run of  $\mathcal{G}_{\text{PR}}^{R, \lambda}(p, q)$  on  $w$ . We prove that  $\rho(|w|) = \perp$  and therefore  $\rho$  is not accepting by induction on  $m$ .

If  $m = 0$ , then  $(p, q) \notin Y_\lambda$ , meaning that there is a word  $w$  such that  $\min\{c(\delta^*(p, u)) \mid u \sqsubset w\} \neq \min\{c(\delta^*(q, u)) \mid u \sqsubset w\}$ . Without loss of generality, assume  $\min\{c(\delta^*(p, u)) \mid u \sqsubset w\} < \min\{c(\delta^*(q, u)) \mid u \sqsubset w\}$ . By Lemma ??,  $x_{|w|-1} = <$ . Furthermore,  $\delta(p_{|w|-1}, w_{|w|-1}) \in \lambda$ , as  $w \in L_{\lambda \rightarrow \lambda}$ . Thus,  $\rho(|w|) = \perp$  and the run is rejecting.

Now consider  $m + 1 > 1$ . Since  $(p, q) \in X_m \setminus f(X_m)$ , there must be a word  $w \in L_{\lambda \rightarrow \lambda}$  such that  $(p', q') \notin X_m$ , where  $p' = \delta^*(p, w)$  and  $q' = \delta^*(q, w)$ . As  $R_\kappa \subseteq X_m$ ,  $(p', q') \notin R_\kappa$  and therefore  $p' \not\equiv_{\text{PR}}^R q'$ . By induction,  $w \notin L(\mathcal{G}_{\text{PR}}^{R, \lambda}(p', q'))$ ; since that run is a suffix of  $\rho$ ,  $\rho$  itself is also a rejecting run.

**Only If** Let  $L(\mathcal{G}_{\text{PR}}^{R, \lambda}(p, q)) \neq \Sigma^*$ . Since  $\varepsilon$  is always accepted, there is a word  $w \in \Sigma^+ \setminus L(\mathcal{G}_{\text{PR}}^{R, \lambda}(p, q))$ , meaning that  $\delta_{\text{PR}}^*(q_{0, \text{PR}}, w) = \perp$ . Split  $w$  into sub-words  $w = u_1 \cdots u_m$  such that  $u_1, \dots, u_m \in L_{\lambda \rightarrow \lambda}$ . Note that this partition is unique. We show  $p \not\equiv_{\text{PR}}^R q$  by induction on  $m$ . Let  $\rho = (p_i, q_i, k_i, x_i)_{0 \leq i < |w|}$  be the run of  $\mathcal{G}_{\text{PR}}^{R, \lambda}(p, q)$  on  $w$ .

If  $m = 1$ , then  $w \in L_{\lambda \rightarrow \lambda}$ . Since  $\rho(|w|) = \perp$ , it must be true that  $x_{|w|-1} \neq =$ . Without loss of generality, assume  $x_{|w|-1} = <$ . By Lemma ??,  $\min\{c(\delta^*(p, u)) \mid u \sqsubset w\} < \min\{c(\delta^*(q, u)) \mid u \sqsubset w\}$ . Therefore,  $p \not\equiv_{\text{PR}}^R q$ .

Now consider  $m + 1 > 1$ . Let  $p' = \delta^*(p, u_1)$  and  $q' = \delta^*(q, u_1)$ . By induction on the word  $u_2 \cdots u_m$ ,  $p' \not\equiv_{\text{PR}}^R q'$ . Since  $u_1 \in L_{\lambda \rightarrow \lambda}$ , that also means  $p \not\equiv_{\text{PR}}^R q$ .  $\square$

The differences between different  $\mathcal{G}_{\text{PR}}^{R, \lambda}$  for different  $\lambda$  are minor and the question whether the accepted language is universal boils down to a simple question of reachability. Thus,  $\equiv_{\text{PR}}^R$  can be computed in  $\mathcal{O}(|\mathcal{G}_{\text{PR}}^{R, \lambda}|)$  which is  $\mathcal{O}(|Q|^2 \cdot |c(Q)|)$ .