Algorithm 1: Partition refinement algorithm

Input: A DFA $\mathcal{A} = (Q, q_0, \Sigma, \delta, F)$

Initialize initial partition (B_1, B_2) with $B_1 \leftarrow F$ and $B_2 \leftarrow Q \setminus F$

while *there exist sets* $B_i, B_j,$ *states* $p, q \in B_j,$ *and* $a \in \Sigma$ *such that* $\delta(p, a) \in B_j \Leftrightarrow \delta(q, a) \notin B_j$ **do**

  Split B_i into $\{p \in B_i \mid \delta(p, a) \in B_j\}$ and $\{p \in B_i \mid \delta(p, a) \notin B_j\}$

end

return $\sim_{\mathcal{A}}$ such that $p \sim_{\mathcal{A}} q$ if and only if $p, q \in B_\ell$ for some ℓ

Algorithm 2: Bisimulation marking algorithm

Input: Two LTS $\mathcal{A} = (Q_\mathcal{A}, \Sigma, \Delta_\mathcal{A})$ and $\mathcal{B} = (Q_\mathcal{B}, \Sigma, \Delta_\mathcal{B})$

Mark $(p, q) \in Q_\mathcal{A} \times Q_\mathcal{B}$ if $p \in F_\mathcal{A} \not\Leftrightarrow q \in F_\mathcal{B}$

while *some states have been marked in the last iteration* **do**

  Mark each previously unmarked pair (p, q) for which a $(p, a, p') \in \Delta_\mathcal{A}$ exists such that for each $(q, a, q') \in \Delta_\mathcal{B}$ the pair (p', q') is marked

  Mark each previously unmarked pair (p, q) for which a $(q, a, q') \in \Delta_\mathcal{B}$ exists such that for each $(p, a, p') \in \Delta_\mathcal{A}$ the pair (p', q') is marked

end

return all marked pairs of states