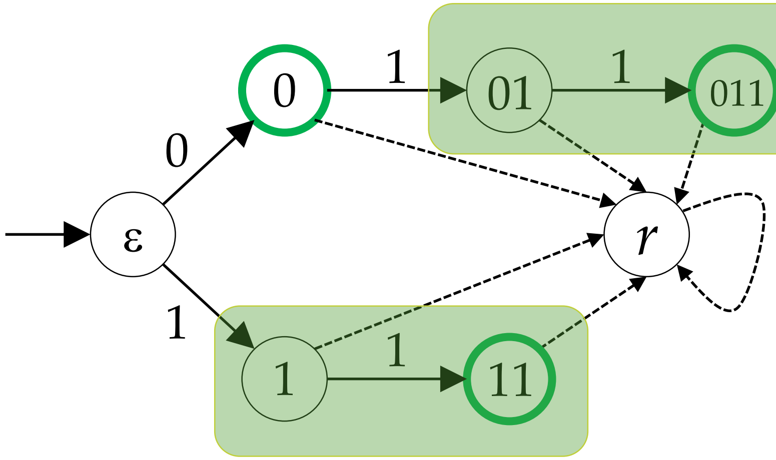
Minimizing DFA

- We had constructed the following DFA for the language $L = \{0, 11, 011\}$



- We want an efficient algorithm to minimize DFAs

Recap of Myhill-Nerode Theorem

- For any language $L \subseteq A^*$, define the following relation over A^*

$$x \equiv_L y \quad \text{iff} \quad \forall z \in A^*, \quad x.z \in L \Leftrightarrow y.z \in L$$

- **Theorem** (*Myhill-Nerode*): L is regular iff $\equiv_L$ has finite index

- **Key Lemma** (Claim 3 in earlier lecture): If $\exists M = (Q, s, \delta, F)$ such that $L(M) = L$ then $\equiv_M$ refines $\equiv_L$, where $x \equiv_M y \quad \text{iff} \quad \hat{\delta}(s, x) = \hat{\delta}(s, y)$
- Prove that if $x \equiv_M y$ then $\forall z \in A^*, \ \hat{\delta}(s, x.z) = \hat{\delta}(s, y.z)$
  - Proof by induction on z

Inductive proof

- If $x \equiv_M y$ then $\forall z \in A^*,\ \hat{\delta}(s, x.z) = \hat{\delta}(s, y.z)$
- Base case ($z = \varepsilon$): By definition, if $x \equiv_M y$ then $\hat{\delta}(s, x.\varepsilon) = \hat{\delta}(s, y.\varepsilon)$
- Inductive step ($z = w.a$ for some $w \in A^*$, $a \in A$): Suppose $x \equiv_M y$.
- Now, $\hat{\delta}(s, x.z) = \hat{\delta}(s, x.(w.a)) = \delta\left(\hat{\delta}(s, x.w), a\right)$

  and $\hat{\delta}(s, y.z) = \hat{\delta}(s, y.(w.a)) = \delta\left(\hat{\delta}(s, y.w), a\right)$
- By the IH, $\hat{\delta}(s, x.w) = \hat{\delta}(s, y.w)$
- Since δ is well-defined, the result holds.

Induced equivalences on DA states

- Let $M = (Q, s, \delta, F)$ be a DA for $L \subseteq A^*$ with no unreachable states
  - $\forall q \in Q, \exists x_q \in A^*$ such that $\hat{\delta}(s, x_q) = q$
- Define this equivalence over Q:

"Two states are L-equivalent iff their access strings are L-equivalent"

$p \approx_L q$ iff $\forall x, y \in A^*$ such that $\hat{\delta}(s, x) = p$ and $\hat{\delta}(s, y) = q$, $x \equiv_L y$

- **Claim 1**: The mapping $f\left([x]_{\equiv_L}\right) = \left[\hat{\delta}(s, x)\right]_{\approx_L}$ is a bijection
- **Proof**:
  - Well-defined
  - Injective
  - Surjective

Corollary 1: L is regular iff $\approx_L$ has finite index

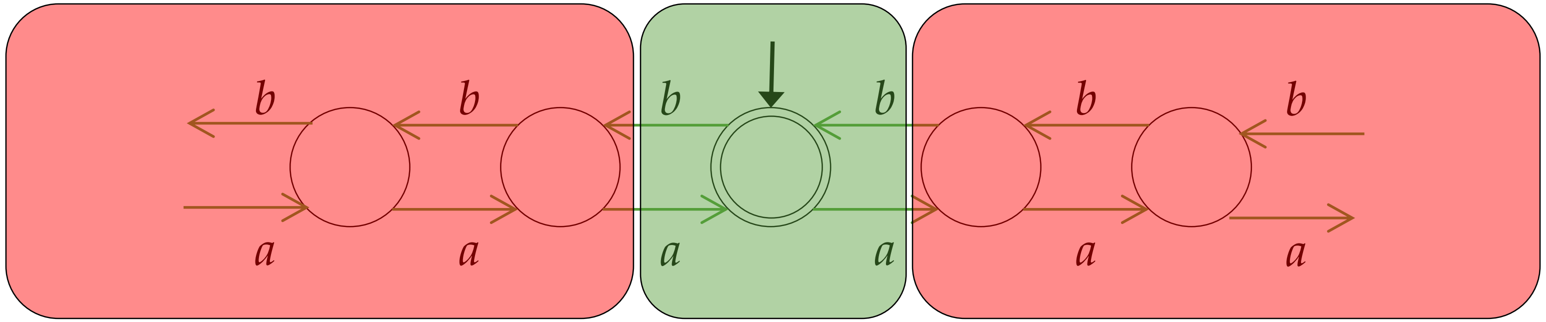
Corollary 2: This is a canonical DA for L:

$Q^* = \{[q]_{\approx_L} | q \in Q\}$; $F^* = \{[q]_{\approx_L} | q \in F\}$

$s^* = [s]_{\approx_L}$; $\forall a \in A$, $\delta^*\left([q]_{\approx_L}, a\right) = [\delta(q, a)]_{\approx_L}$

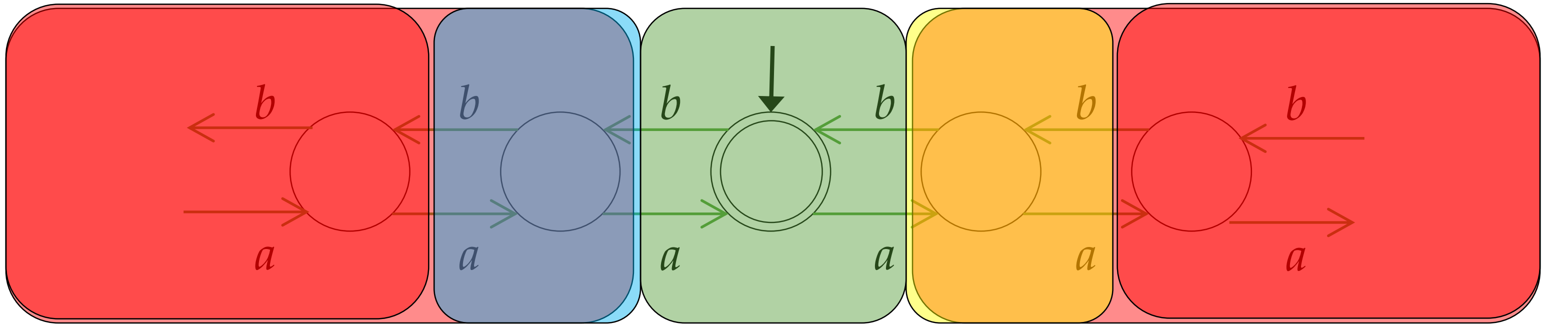
A family of equivalences

- *Example*: Let $L \subseteq \{a, b\}^*$ be strings with an equal number of a's and b's
  - Not regular
- DA for L:



A family of equivalences

- *Example*: Let $L \subseteq \{a, b\}^*$ be strings with an equal number of a's and b's
  - Not regular
- DA for L:

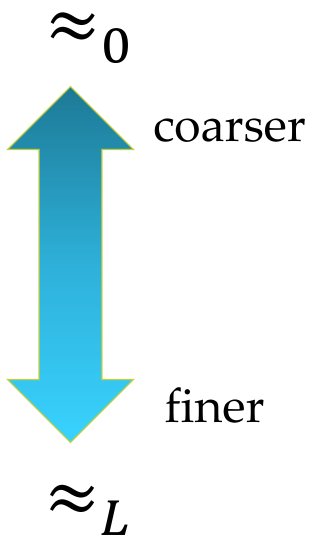


A family of equivalences

- $\forall n \in \mathbb{N}$, define these equivalences over Q:

$$p \approx_n q \quad \text{iff} \quad \forall x, y, z \in A^* \text{ such that } \hat{\delta}(s, x) = p,\ \hat{\delta}(s, y) = q \text{ and } |z| \leq n,$$
$$\hat{\delta}(s, x.z) \in F \Leftrightarrow \hat{\delta}(s, y.z) \in F$$

- **Claim 2**: $\forall n \in \mathbb{N}$, $\approx_{n+1}$ refines $\approx_n$
- **Claim 3**: $\forall n \in \mathbb{N}$, $\approx_L$ refines $\approx_n$
- **Claim 4**: $\forall n \in \mathbb{N}$, $\approx_n$ has finite index
- **Claim 5**: If $\approx_{n+1}$ is the same as $\approx_n$ for some $n \in \mathbb{N}$, then $\approx_L$ and $\approx_n$ are identical (and hence L is regular)



Proof of Claim 5

- Suppose $\approx_{n+1}$ is the same as $\approx_n$ for some $n \in \mathbb{N}$ and let $p \approx_n q$
- To show: $p \approx_L q$ i.e., $\forall x, y, z \in A^*$ such that $\hat{\delta}(s, x) = p$ and $\hat{\delta}(s, y) = q$

$$x.z \in L \text{ iff } y.z \in L$$

- **Lemma**: Suppose $\hat{\delta}(s, x) \approx_n \hat{\delta}(s, y)$ for some $x, y \in A^*$. Then $\forall z \in A^*$

$$\hat{\delta}(s, x.z) \approx_n \hat{\delta}(s, y.z)$$

- *Base case* ($z = \varepsilon$): Trivially true
- *Inductive step* ($z = w.a$): By the IH, $\hat{\delta}(s, x.w) \approx_n \hat{\delta}(s, y.w)$
- Suppose $\hat{\delta}(s, (x.w).a) \not\approx_n \hat{\delta}(s, (y.w).a)$
- So $\exists u \in A^*$ such that $|u| \leq n$ and $\hat{\delta}(s, (x.w).a.u) \in F$ XOR $\hat{\delta}(s, (y.w).a.u) \in F$
- Since $|a.u| \leq n + 1$, it follows that $\hat{\delta}(s, x.w) \not\approx_{n+1} \hat{\delta}(s, y.w)$, a contradiction

DFA minimization algorithm

- Let $M = (Q, s, \delta, F)$ be a DFA for a regular language L
  1. Remove all unreachable states
  2. $\forall p, q \in Q$, initialize $W[p, q] := (p \in F \text{ XOR } q \in F)$
  3. Repeat from $i := 0$

     Loop invariant: $\forall p, q \in Q$, $W[p, q]$ iff $p \not\approx_i q$

     a) $\forall p, q \in Q$ such that $W[p, q]$ is *false* but $W[\delta(p, a), \delta(q, a)]$ is *true* for some a
        - Set $W[p, q] :=$ *true*

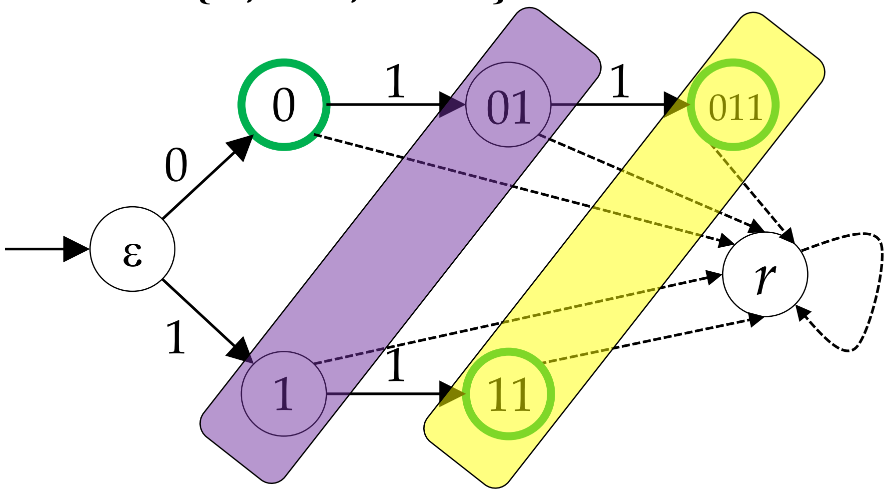
     b) Increment i
  4. Until $W[\ ]$ no longer changes
  5. Merge all states p, q such that $W[p, q]$ is *false*

Result: Canonical and minimized DFA

Polynomial run-time

Apply to this example

- $L = \{0, 11, 011\}$



	0	1	01	11	011	r
ε	√	√	√	√	√	√
0		√	√	√	√	√
1				√	√	√
01				√	√	√
11						√
011						√

- Hence state-1 $\approx_L$ state-01 and state-11 $\approx_L$ state-011

Recap: cross-product construction

- Let $M_1 = (Q_1, s_1, \delta_1, F_1)$ and $M_2 = (Q_2, s_2, \delta_2, F_2)$ be two DFAs
- $\forall F \subseteq Q_1 \times Q_2$, define $M_1 \times M_2(F) = (Q_1 \times Q_2, (s_1, s_2), \delta_1 \times \delta_2, F)$ where $\forall (q_1, q_2) \in Q_1 \times Q_2, \forall a \in A, \qquad \delta_1 \times \delta_2\big((q_1, q_2), a\big) = \big(\delta_1(q_1, a), \delta_2(q_2, a)\big)$
- **Claim**: $\forall x \in A^*, \widehat{\delta_1 \times \delta_2}\big((s_1, s_2), x\big) = \Big(\widehat{\delta_1}(s_1, x), \widehat{\delta_2}(s_2, x)\Big)$
- *Applications*: There are efficient algorithms for these problems
  1. Given two DFAs M_1 and M_2, determine whether $L(M_1) \subseteq L(M_2)$
  2. Given two DFAs M_1 and M_2, determine whether $L(M_1) = L(M_2)$
  3. Given DFA M and NFA N, determine whether $L(N) \subseteq L(M)$