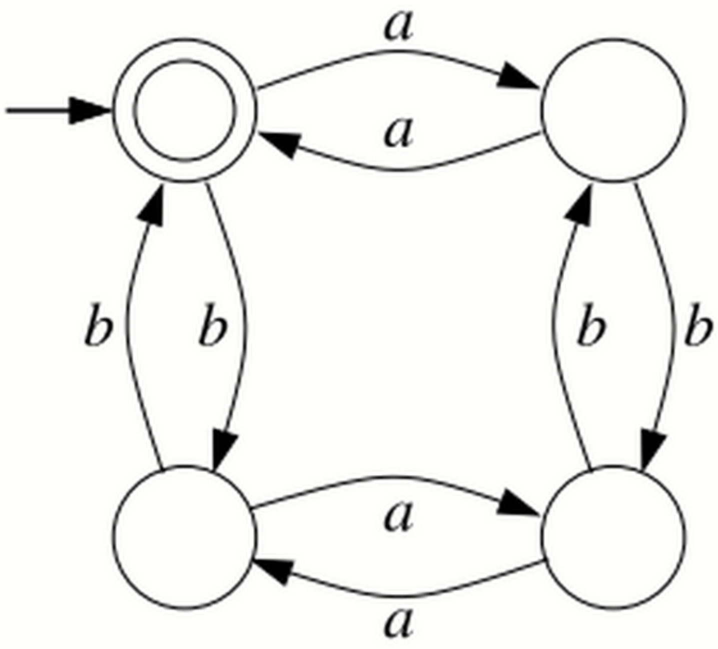
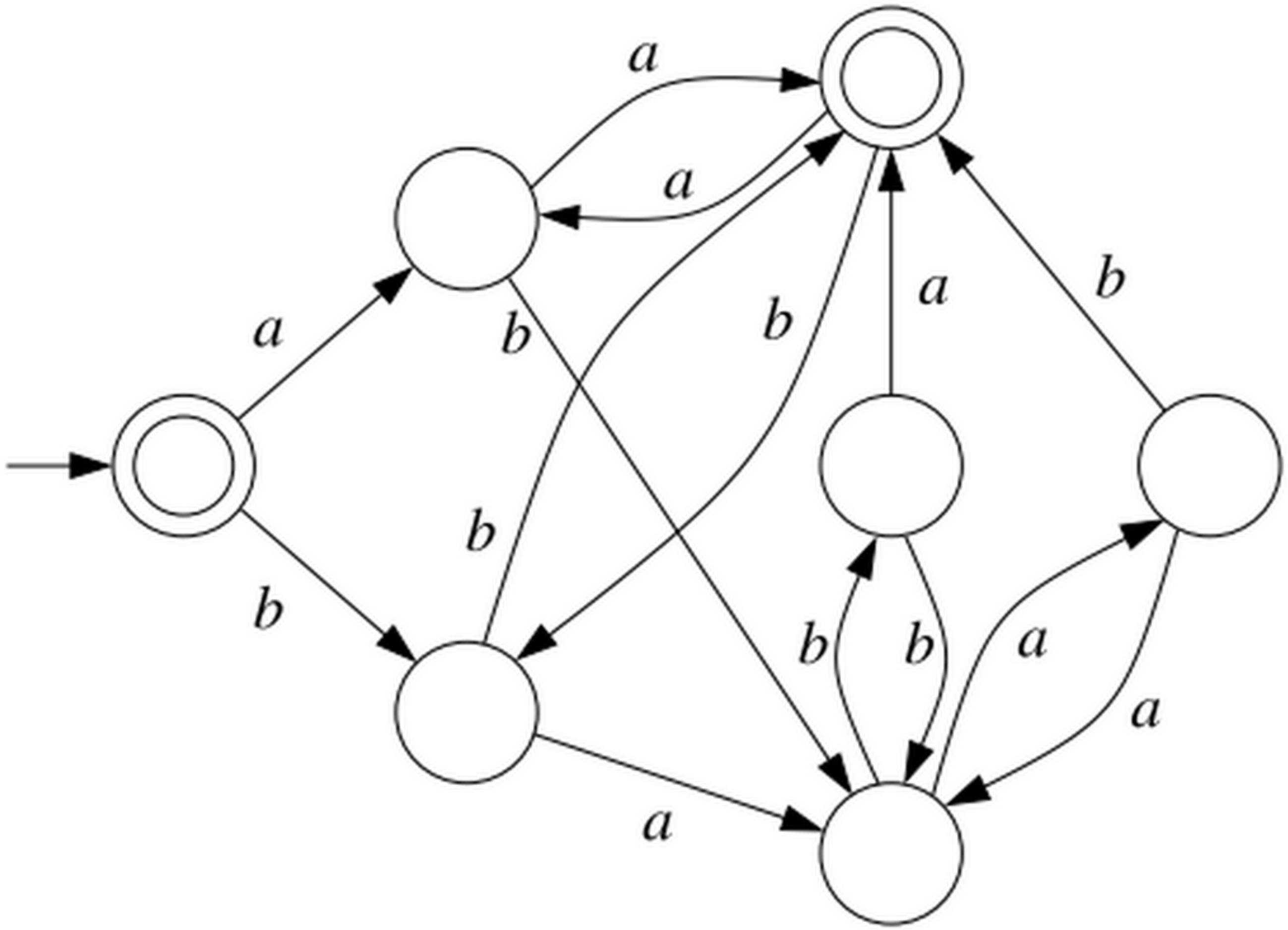
Minimization and Reduction





Residual

Definition 3.1 *Given a language $L \subseteq \Sigma^*$ and $w \in \Sigma^*$, the w-*residual *of L is the language $L^w = \{u \in \Sigma^* \mid wu \in L\}$. A language $L' \subseteq \Sigma^*$ is a* residual *of L if $L' = L^w$ for at least one $w \in \Sigma^*$.*

Observe : $(L^w)^u = L^{wu}$

Relation between residuals and states in DFAs

Definition 3.2 *Let* $A = (Q, \Sigma, \delta, q_0, F)$ *be a DA and let* $q \in Q$. *The* language recognized by q, *denoted by* $\mathcal{L}_A(q)$, *or just* $\mathcal{L}(q)$ *if there is no risk of confusion, is the language recognized by A with* q *as initial state, i.e., the language recognized by the DA* $(Q, \Sigma, \delta, q, F)$.

Relation between residuals and states in DFAs

Lemma 3.3 *Let L be a language and let $A = (Q, \Sigma, \delta, q_0, F)$ be a DA recognizing L.*

(1) *For every $w \in \Sigma^*$ some state of A recognizes L^w (i.e., for every $w \in \Sigma^*$ there is $q \in Q$ such that $\mathcal{L}_A(q) = L^w$).*

(2) *For every $q \in Q$, $\mathcal{L}_A(q)$ is a residual of L.*

(1) *For every $w \in \Sigma^*$ some state of A recognizes L^w (i.e., for every $w \in \Sigma^*$ there is $q \in Q$ such that $\mathcal{L}_A(q) = L^w$).*

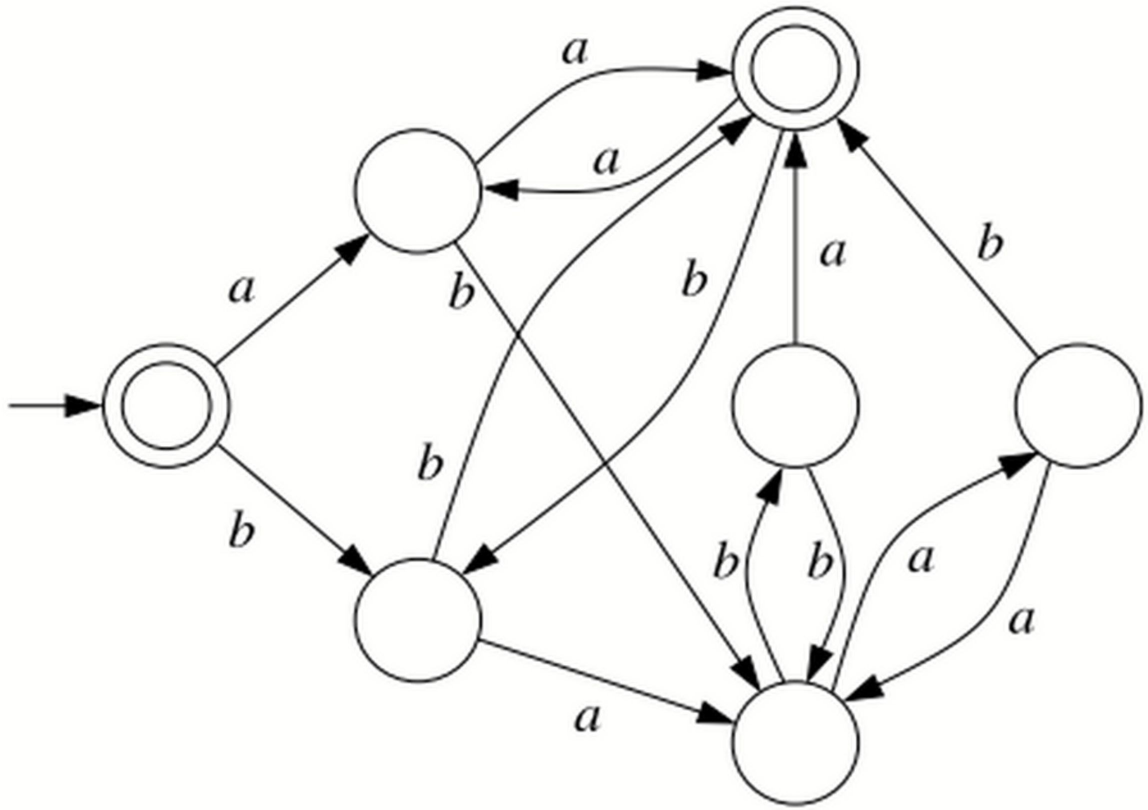
(2) *For every $q \in Q$, $\mathcal{L}_A(q)$ is a residual of L.*

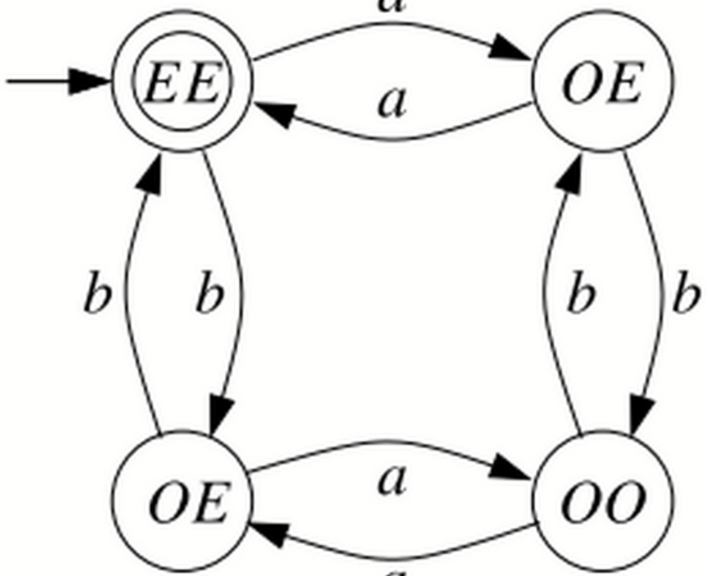
Relation between residuals and states in DFAs

Lemma 3.3 *Let L be a language and let $A = (Q, \Sigma, \delta, q_0, F)$ be a DA recognizing L.*

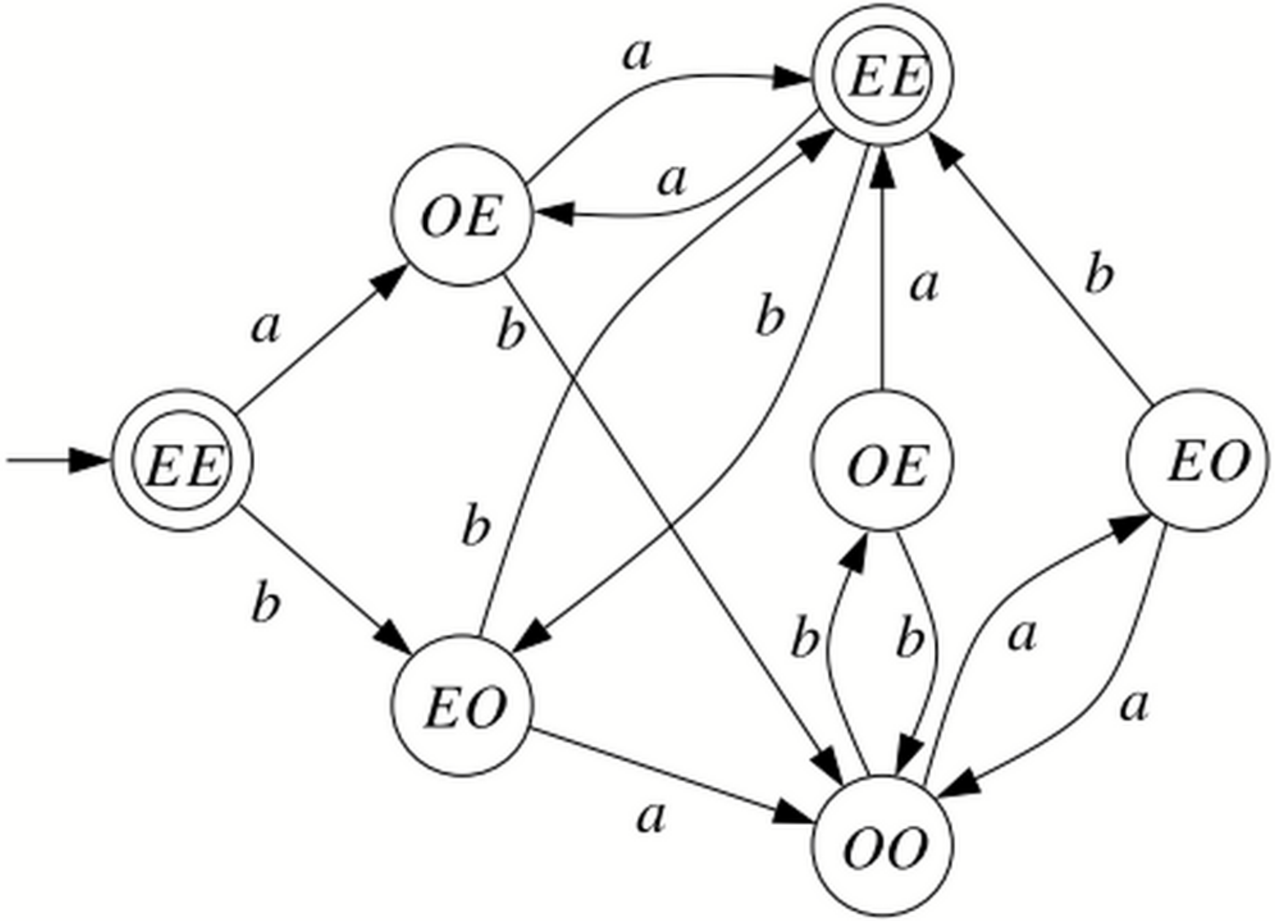
(1) For every $w \in \Sigma^*$ some state of A recognizes L^w (i.e., for every $w \in \Sigma^*$ there is $q \in Q$ such that $\mathcal{L}_A(q) = L^w$).

(2) For every $q \in Q$, $\mathcal{L}_A(q)$ is a residual of L.





EE
a
OE
a
b
b
b
b
OE
a
OO
a



a
EE
a
OE
a
b
b
a
b
EE
OE
EO
b
b
b
a
b
EO
a
a
OO

- Important consequence:

  A regular language has finitely many residuals

  or, equivalently

  Languages with infinitely many residuals are not regular

Canonical DFA for a regular language

Definition 3.4 *Let $L \subseteq \Sigma^*$ be a language. The* canonical DA *for L is the DA $C_L = (Q_L, \Sigma, \delta_L, q_{0L}, F_L)$, where:*

- *Q_L is the set of residuals of L; i.e., $Q_L = \{L^w \mid w \in \Sigma^*\}$;*
- *$\delta(K, a) = K^a$ for every $K \in Q_L$ and $a \in \Sigma$;*
- *$q_{0L} = L$; and*
- *$F_L = \{K \in Q_L \mid \varepsilon \in K\}$.*

Example 1: The language EE $\subseteq \{a, b\}^*$

$Q_{EE} =$

$q_{0EE} =$

$F_{EE} =$

$\delta_{EE} =$

Example 2: The language a*b*

$Q_{a^*b^*} =$

$q_0 a^* b^* =$

$F a^* b^* =$

$\delta a^* b^* =$

Proposition 3.6 *For every language* $L \subseteq \Sigma^*$*:* $\mathcal{L}(C_L) = L$.

Proof: Let $w \in \Sigma^*$. We prove by induction on $|w|$ that $w \in L$ iff $w \in \mathcal{L}(C_L)$.

If $|w| = 0$ then $w = \varepsilon$. By the definition of F_L we have $\varepsilon \in L$ iff $L \in F_L$. Since $q_{0L} = L$, we have $L \in F_L$ iff $q_{0L} \in F_L$. Finally, since q_{0L} is the initial state of C_L, we have $q_{0L} \in F_L$ iff $\varepsilon \in \mathcal{L}(C_L)$.

If $|w| > 0$, then $w = aw'$ for some $a \in \Sigma$ and $w' \in \Sigma^*$. So

$$
\begin{array}{lll}
 & w \in L & \\
\Leftrightarrow & w' \in L^a & (w = aw') \\
\Leftrightarrow & w' \in \mathcal{L}(C_{L^a}) & \text{(induction hypothesis)} \\
\Leftrightarrow & aw' \in \mathcal{L}(C_L) & (\delta_L(L, a) = L^a) \\
\Leftrightarrow & w \in \mathcal{L}(C_L) & (w = aw')
\end{array}
$$

Theorem 3.7 *If L is regular, then C_L is the unique minimal DFA up to isomorphism recognizing L.*

Proof: Let L be a regular language, and let $A = (Q, \Sigma, \delta, q_0, F)$ be an arbitrary DFA recognizing L. By Lemma 3.3 the number the number of states of A is greater than or equal to the number of states of C_L, and so C_L is a minimal automaton for L. To prove uniqueness of the minimal automaton up to isomorphism, assume A is minimal. By Lemma 3.3(2), $\mathcal{L}_A$ is a mapping that assigns to each state of A a residual of L, and so $\mathcal{L}_A : Q \to Q_L$. We prove that $\mathcal{L}_A$ is an isomorphism. $\mathcal{L}_A$ is bijective because it is surjective (Lemma 3.3(2)), and $|Q| = |Q_L|$ (A is minimal by assumption). Moreover, if $\delta(q, a) = q'$, then $\mathcal{L}_A(q') = (\mathcal{L}_A(q))^a$, and so $\delta_L(\mathcal{L}_A(q), a) = \mathcal{L}_A(q')$. Also, $\mathcal{L}_A$ maps the initial state of A to the initial state of C_L: $\mathcal{L}_A(q_0) = L = q_{0L}$. Finally, $\mathcal{L}_A$ maps final to final and non-final to non-final states: $q \in F$ iff $\varepsilon \in \mathcal{L}_A(q)$ iff $\mathcal{L}_A(q) \in F_L$. □

(1) Any DFA A for L has at least as many states as the canonical DFA for L.
Because every residual of L must be recognized by some state of A, and so A has at least as many states as L has residuals.

(2) If a DFA for L has as many states as the canonical DFA for L, then it is isomorphic to the canonical DFA for L.

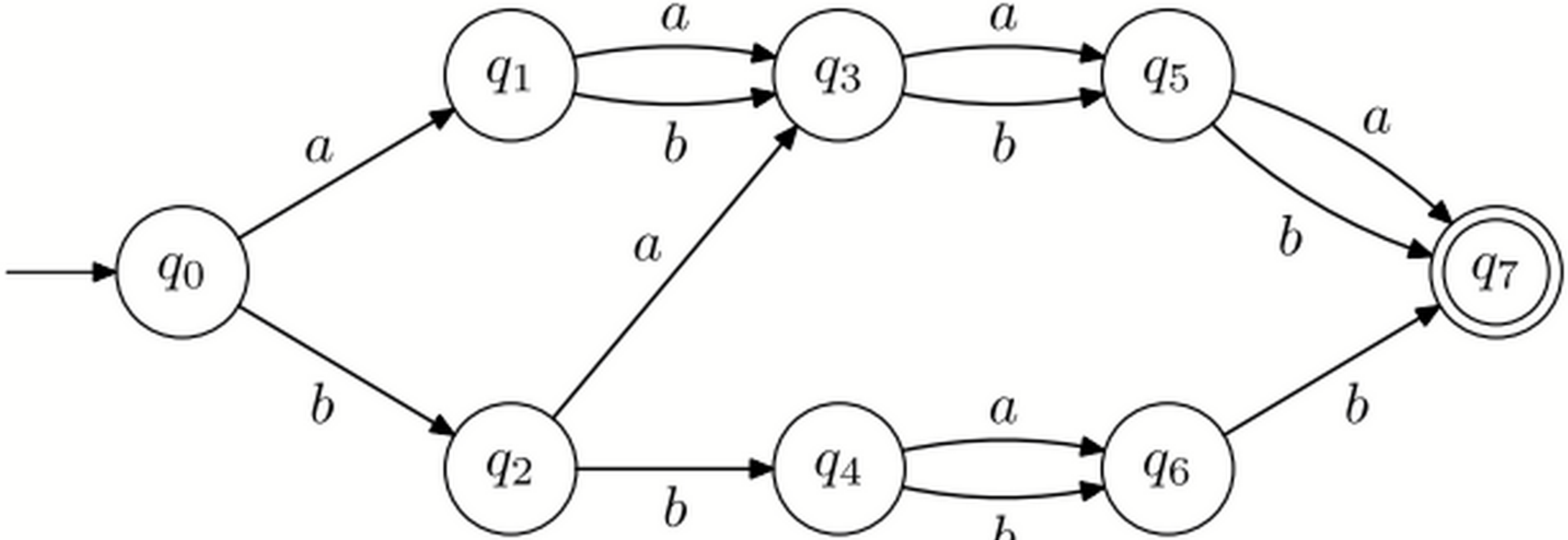
Isomorphism: bijection between states respecting initial and final states and transitions.

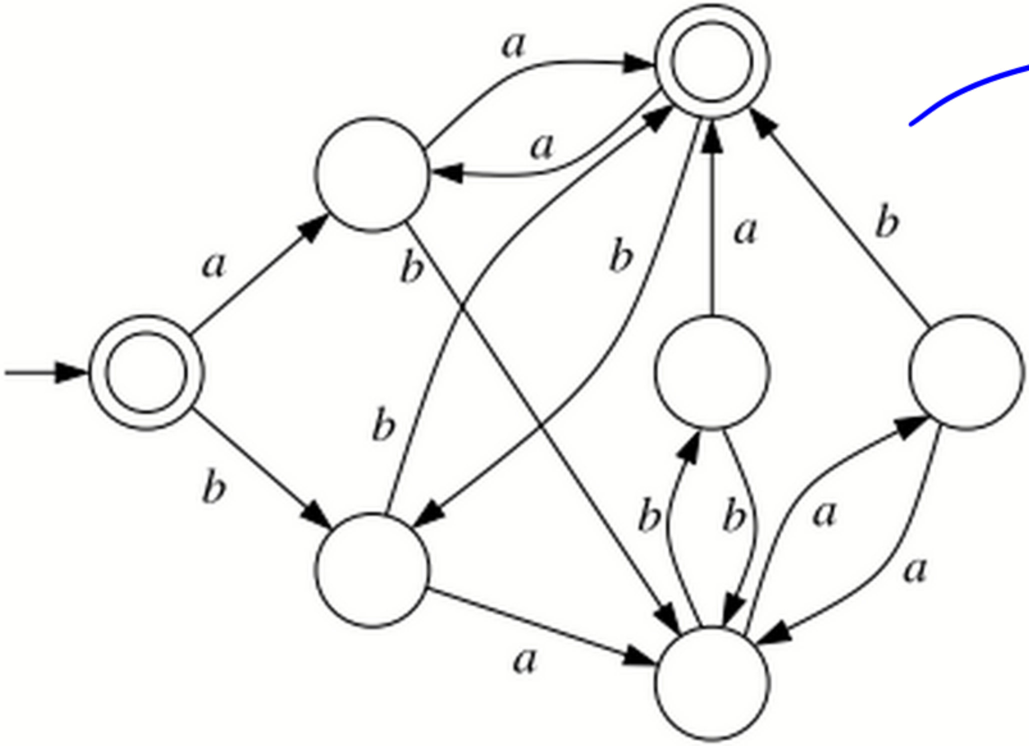
Corollary 3.8 *A DFA is minimal if and only if* $\mathcal{L}(q) \neq \mathcal{L}(q')$ *for every two distinct states* q *and* q'.

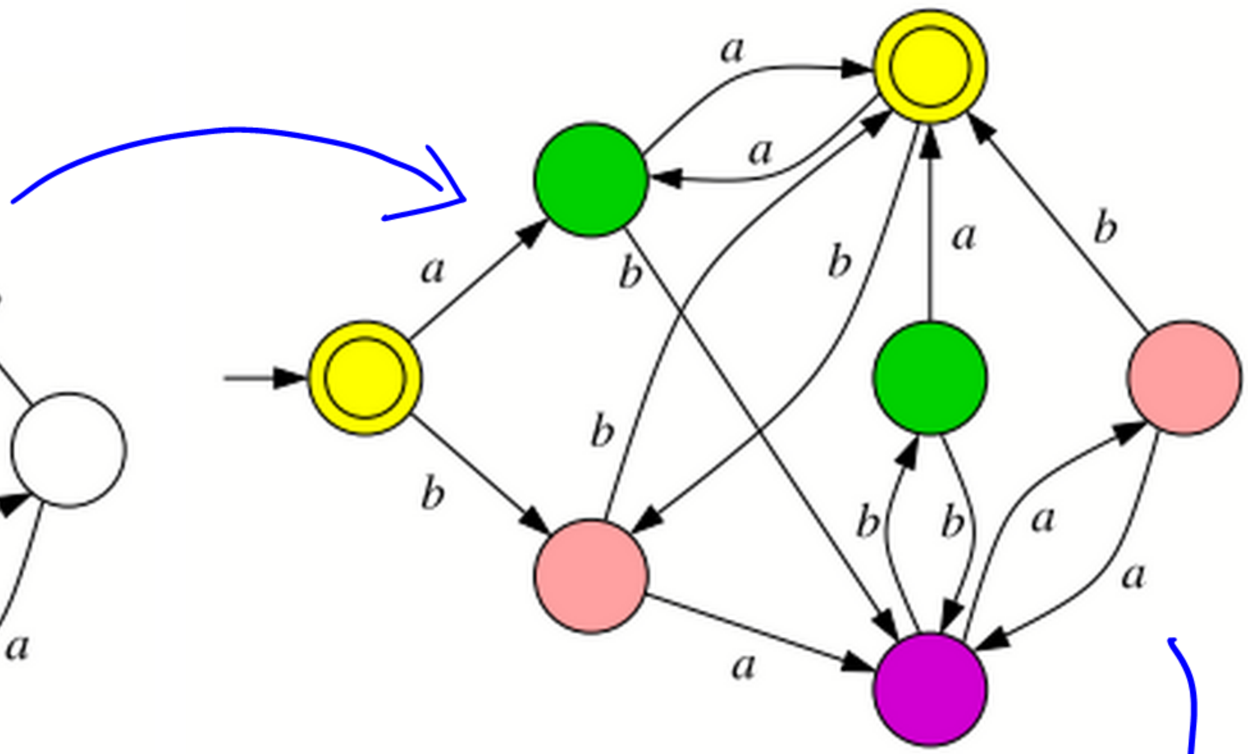
Proof: (⇐): By Theorem 3.7, the number of states of a minimal DFA is equal to the number of residuals of its language. Since every state of recognizes some residual, each state must recognize a different residual.

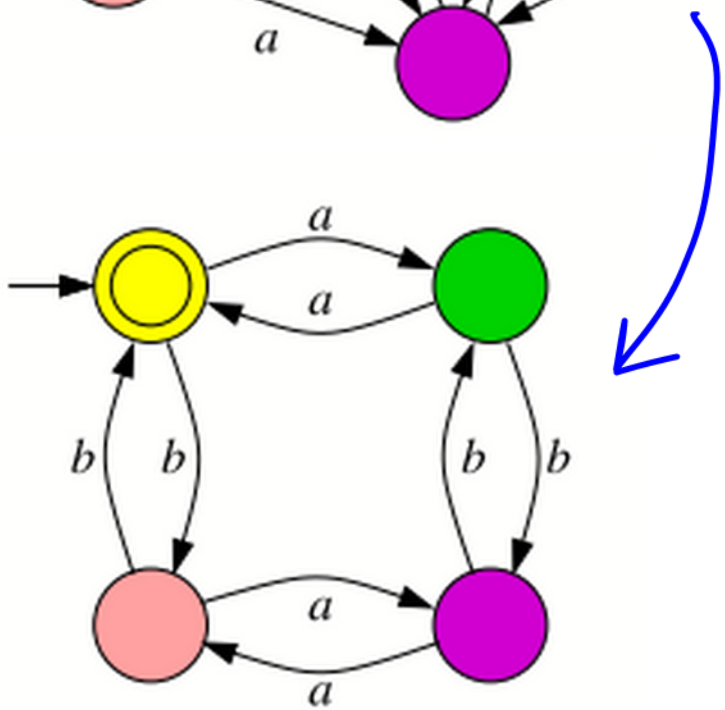
(⇒): If all states of a DFA A recognize different languages, then, since every state recognizes some residual, the number of states of A is less than or equal to the number of residuals. So A has at most as many states as $C_{\mathcal{L}(A)}$, and so it is minimal. □

Is it minimal?









1. Computing the language partition
2. Quotienting
3. Thm: the result is the canonical DFA

Computing the language partition

State partitions

Block: set of states

Partition: set of blocks such that each state belongs to exactly one block

Partition P refines partition P' if every block of P is contained in some block of P'.

If P refines P', then P is finer than P', and P' is coarser than P.

Language partition: states belong to the same block iff they recognize the same language

Computing the language partition

- Start with the partition containing two blocks:
  final states (recognize the empty word)
  non-final states (do not recognize the empty word)

- Iteratively split blocks, ensuring that states recognizing the same language always stay in the same block.

- Blocks that can be split, i.e., blocks that contain two states recognizing different languages, are called unstable.

Finding an unstable block

If q1, q2 belong to the same block B but delta(q1,a), delta(q2,a) belong to different blocks, then B is unstable.

Splitting an unstable block

If q1, q2 belong to the same block B and
some block B' contains delta(q1,a) but not delta(q2,a)
then B is unstable and we say that (a, B') splits B.

Ref_P(B,a,B') denotes the result of splitting block B
o partition P in two parts, as follows:

- states whose a-transition leads to B' (e.g. q1)
- states whose a-transition leads elsewhere (e.g. q2)

LanPar(A)

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

Output: The language partition P_ℓ for $L = \mathcal{L}(A)$.

if $F = \emptyset$ or $Q \setminus F = \emptyset$ **then return** $\{Q\}$

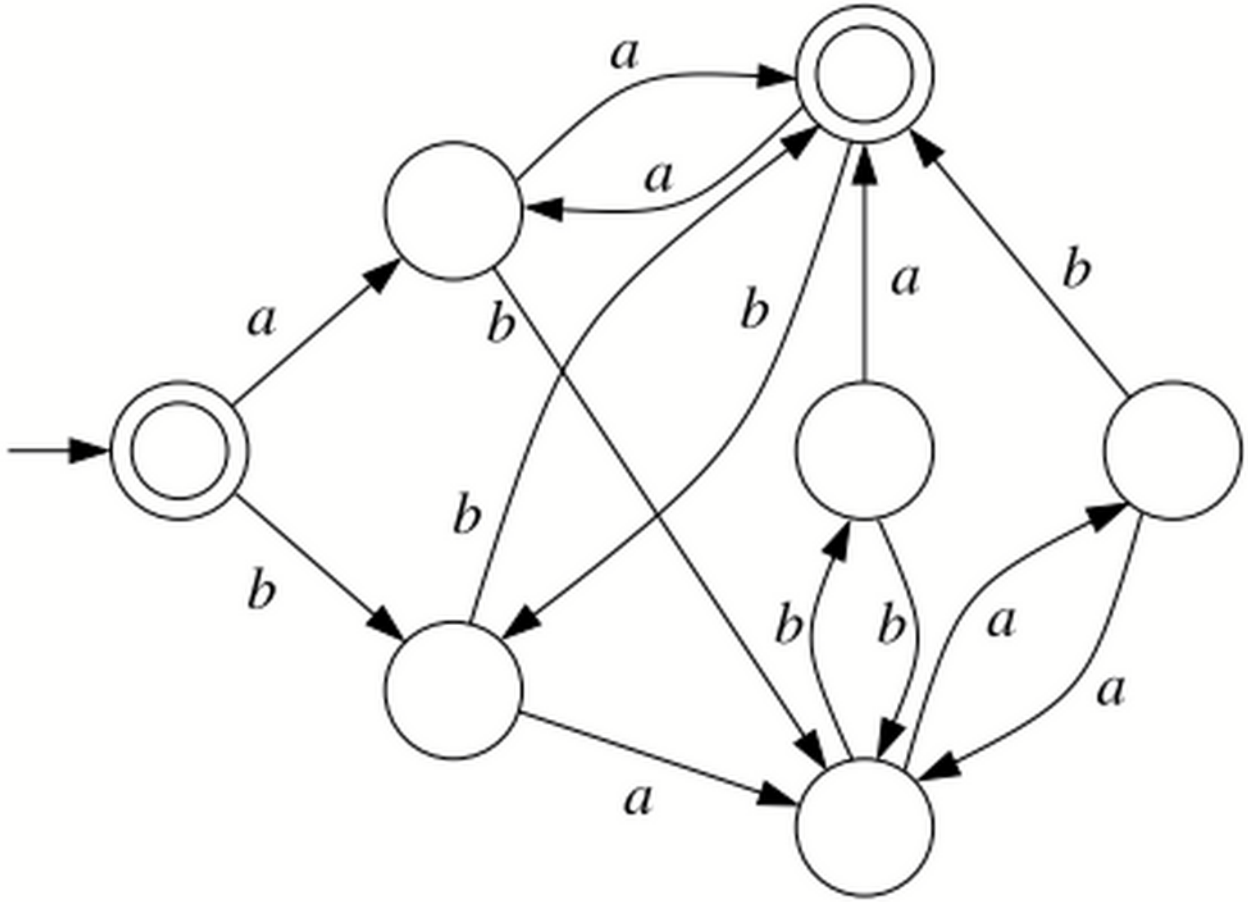
else $P \leftarrow \{F, Q \setminus F\}$

while P is unstable **do**

  pick $B, B' \in P$ and $a \in \Sigma$ such that (a, B') splits B

  $P \leftarrow \mathit{Ref}_P[B, a, B']$

return P


a
a
a
b
b
a
b
b
a
b
b
b
a
a
a

Correctness

-Termination:
Every execution of the loop increases the number of blocks by 1, and the number of blocks is bounded by the number of states.

- After termination: two states belong to the same block iff they recognize the same language.

(1) If two states belong to different blocks, they recognize different languages.
(2) If two states recognize different languages, then they belong to different blocks.

(1) If q1, q2 belong to different blocks, they recognize different languages.

By induction on the number k of splittings carried out until q1 and q2 are put into diferent blocks.

- k = 0. Then q1 is final and q2 is not, or vice versa, and we are done.

- k -> k+1. Then there are states q1',q2' such that q1 -a-> q1', q2 -a->q2' and q1',q2' belong to different blocks. By induction hypothesis q1', q2' recognize different languages, and (DFA property) so do q1,q2.

(2) If q1, q2 recognize different languages, then they belong to different blocks.

Let w be a shortest word that belongs to, say, L(q1) but not to L(q2). By induction on the length of w.

- |w|=0. Then w is the empty word, q1 is final, q2 is not, and so they belong to different blocks from the beginning.

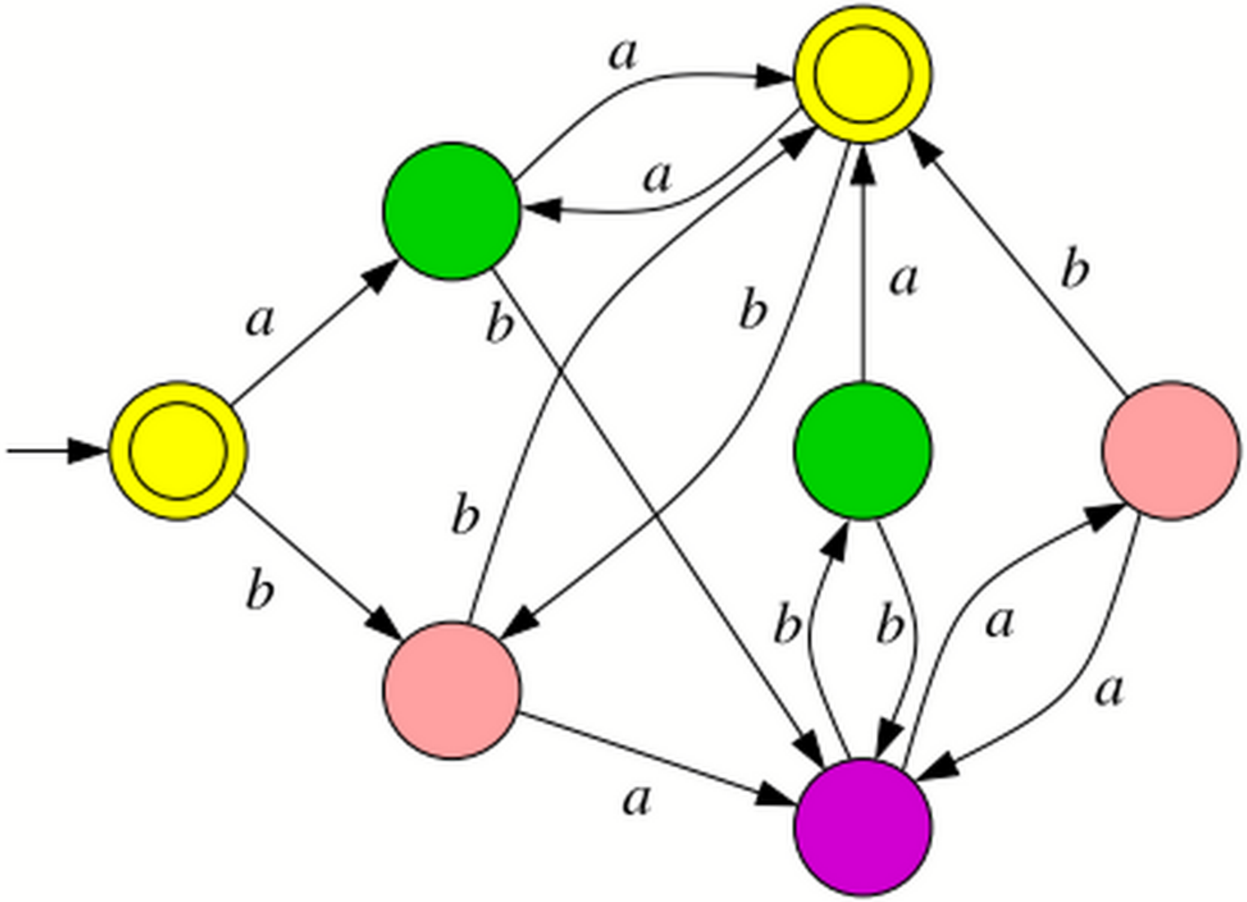
- |w|>0. Then w = aw'. Let q1'=delta(q1,a) and q2'=delta(q2,a). By induction hypothesis q1',q2' are put at some point in different blocks (DFA property). If at this point q1,q2 still belong to the same block B, then B becomes unstable and is eventually split.

Quotienting

Quotient wrt a partition

Definition 3.14 *Let P be a partition of Q. The* quotient of A with respect to P *is* $A/P = (Q_P, \Sigma, \delta_P, q_{0P}, F_P)$ *where*

- $Q_P = P$;
- $\delta_P = \{ ([q]_P, a, [q']_P) \mid (q, a, q') \in \delta \}$;
- $q_{0B} = [q_0]_P$; *and*
- $F_P = \{[q]_P \mid q \in F\}$.



a
a
a
b
b
a
b
b
b
b
b
a
a
a

The quotient wrt the language partition is the canonical DFA

(1) A DFA and its quotient wrt the language partition

Every word accepted by the DFA is accepted by the quotient.

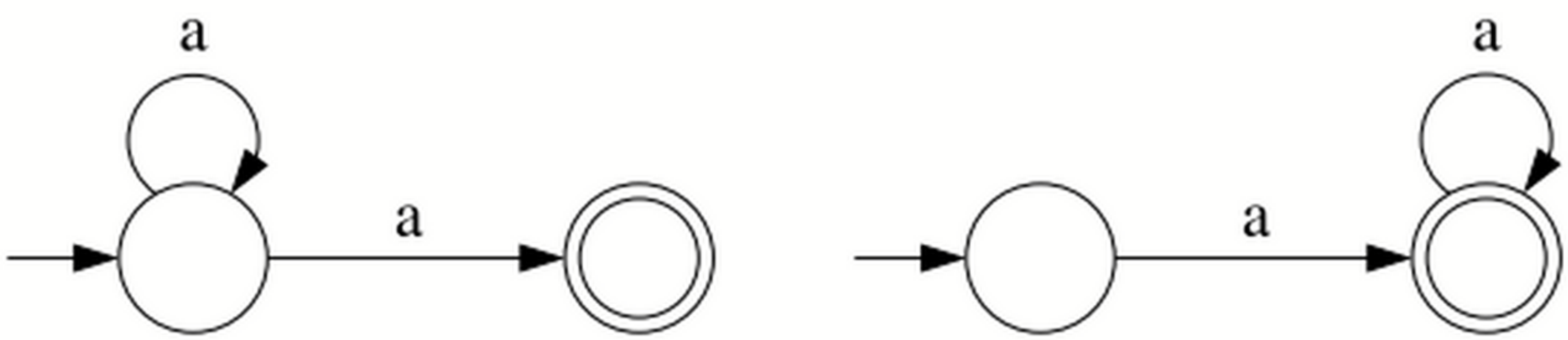
Every word accepted by the quotient is accepted by the DFA.

(2) The quotient is the canonical DFA

Easy: the quotient recognizes the same language as the initial DFA, and its states recognize different languages by definition!

Reduction of NFAs

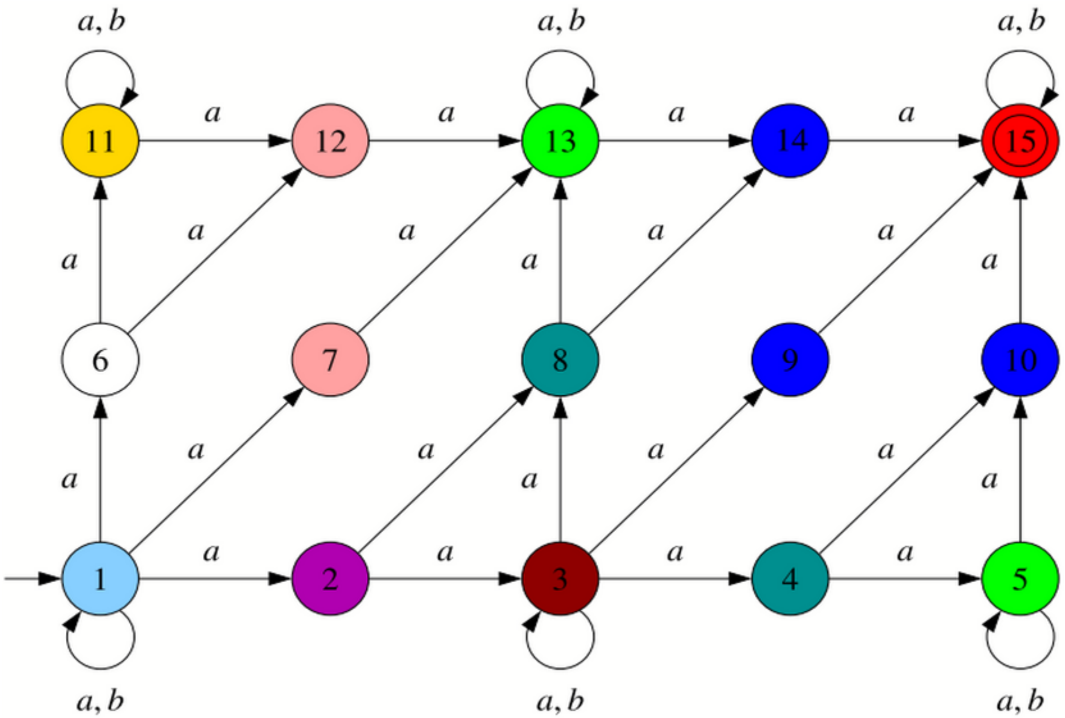
The minimal NFA is not unique

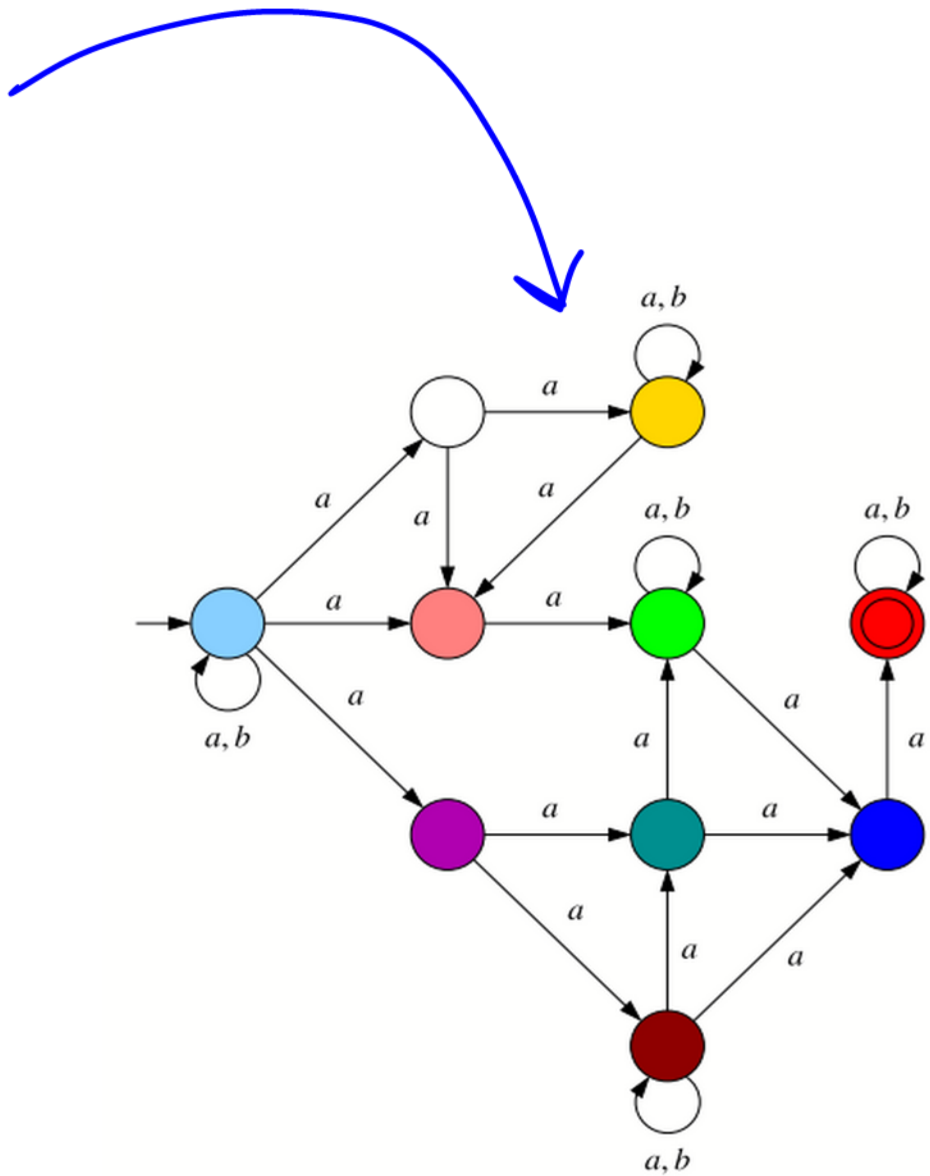


Minimal NFAs are hard to compute

Theorem 3.18 *The following problem is PSPACE-complete: given a NFA A and a number $k \geq 1$, decide if there is a NFA equivalent to A with at most k states.*

Reducing NFAs





1. Computing a suitable partition
2. Quotienting

What is a suitable partition?

- Determined by step 2: the quotient must recognize the same language as the initial NFA.

Thm: Let P be any partition of an NFA A such that states of the same block recognize the same language. Then L(A) = L(A/P).

Such a partition is a refinement of the initial partition {F,Q\F}.

Computing a suitable partition

- Idea: use the same algorithm as for DFAs, but with different notions of unstable block and refinement.

- We have to guarantee: after termination, states of a block recognize the same language, or, equivalently, states recognizing different languages belong to different blocks.

Key observation:

Assume q1 and q2 recognize different languages.
Then:

- either one of them is final and the other is not, or
- there is a letter a and a state qi' of delta(qi,a) such that for every state qj' of delta(qj,a) qi' and qj' recognize different languages.

This suggests:

Definition 3.19 (Refinement and stability for NFAs) *Let B, B' be (not necessarily distinct) blocks of a partition P of Q, and let $a \in \Sigma$. The pair (a, B')* splits *B if there are $q_1, q_2 \in B$ such that $\delta(q_1, a) \cap B' \neq \emptyset$ and $\delta(q_2, a) \cap B' = \emptyset$. The (B, a, B')*-refinement *of P is the partition $Ref_P^{NFA}[B, a, B'] = (P \setminus \{B\}) \cup \{B_0, B_1\}$, where $B_0 = \{q \in B \mid \delta(q, a) \notin B'\}$ and $B_1 = B \setminus B_0$.*

A partition is unstable *if it contains blocks B, B' such that B' splits B, and* stable *otherwise.*

CSR(A)

Input: NFA $A = (Q, \Sigma, \delta, q_0, F)$

Output: The partition *CSR*.

if $F = \emptyset$ or $Q \setminus F = \emptyset$ **then return** $\{Q\}$

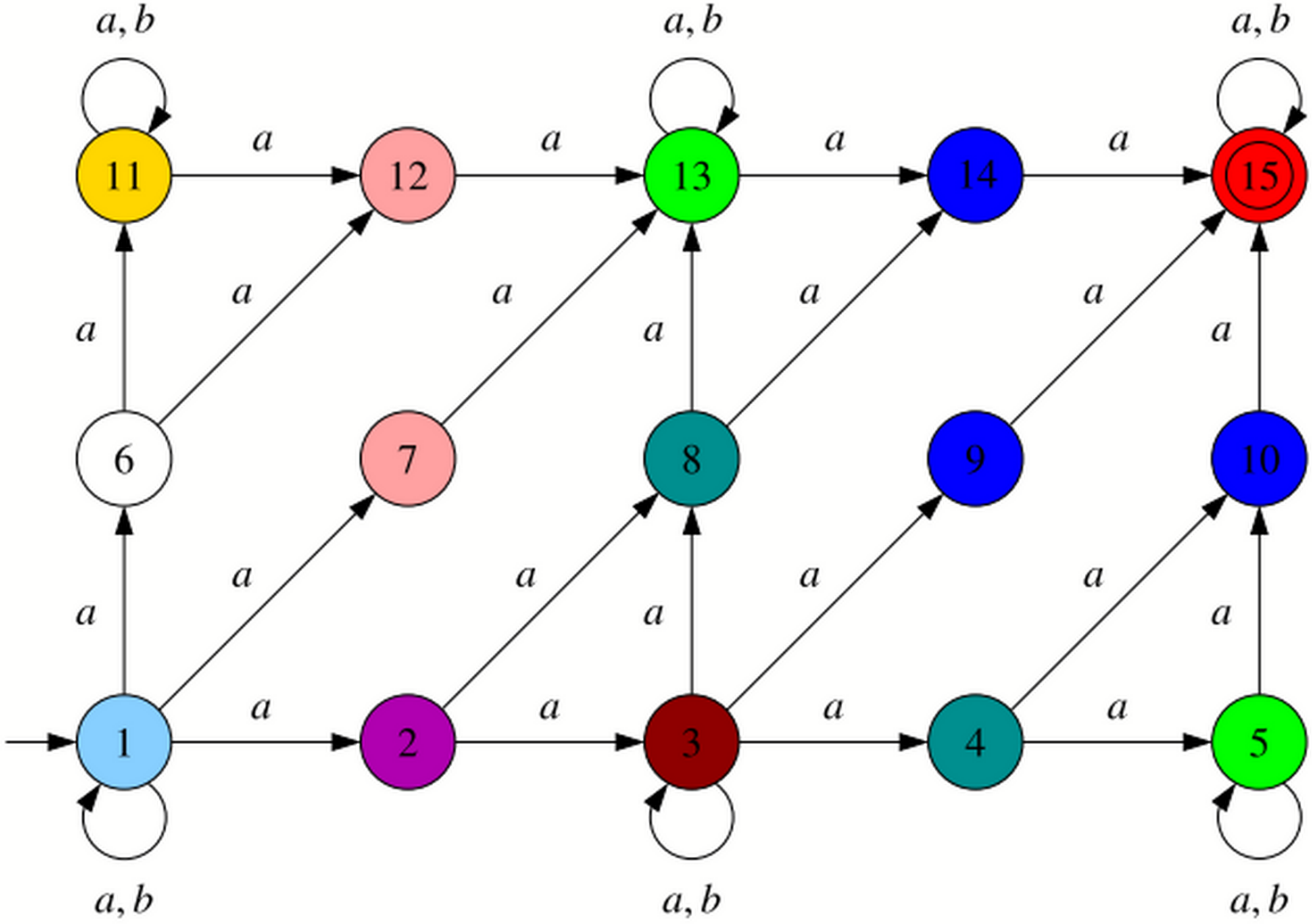
else $P \leftarrow \{F, Q \setminus F\}$

while P is unstable **do**

  pick $B, B' \in P$ and $a \in \Sigma$ such that (a, B') splits B

  $P \leftarrow Ref_P^{NFA}[B, a, B']$

return P


a, b
a, b
a, b
11
12
13
14
15
a
a
a
a
6
7
8
9
10
1
2
3
4
5
a, b
a, b
a, b