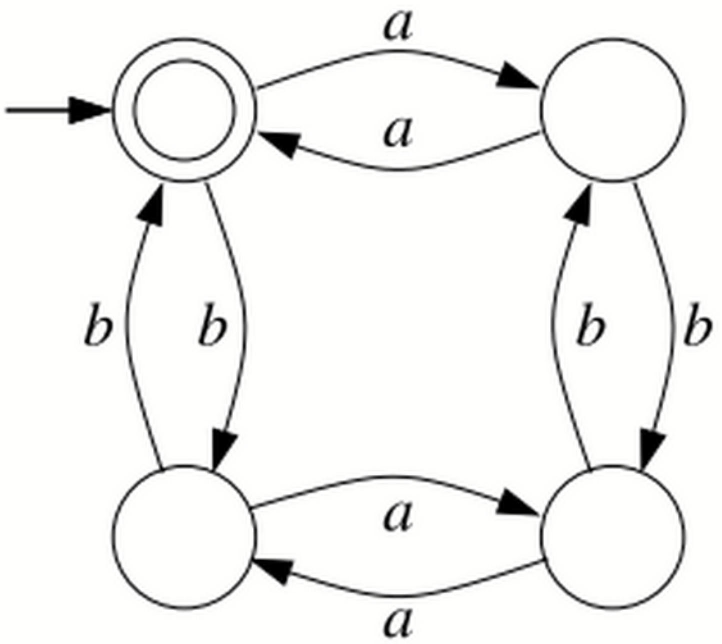
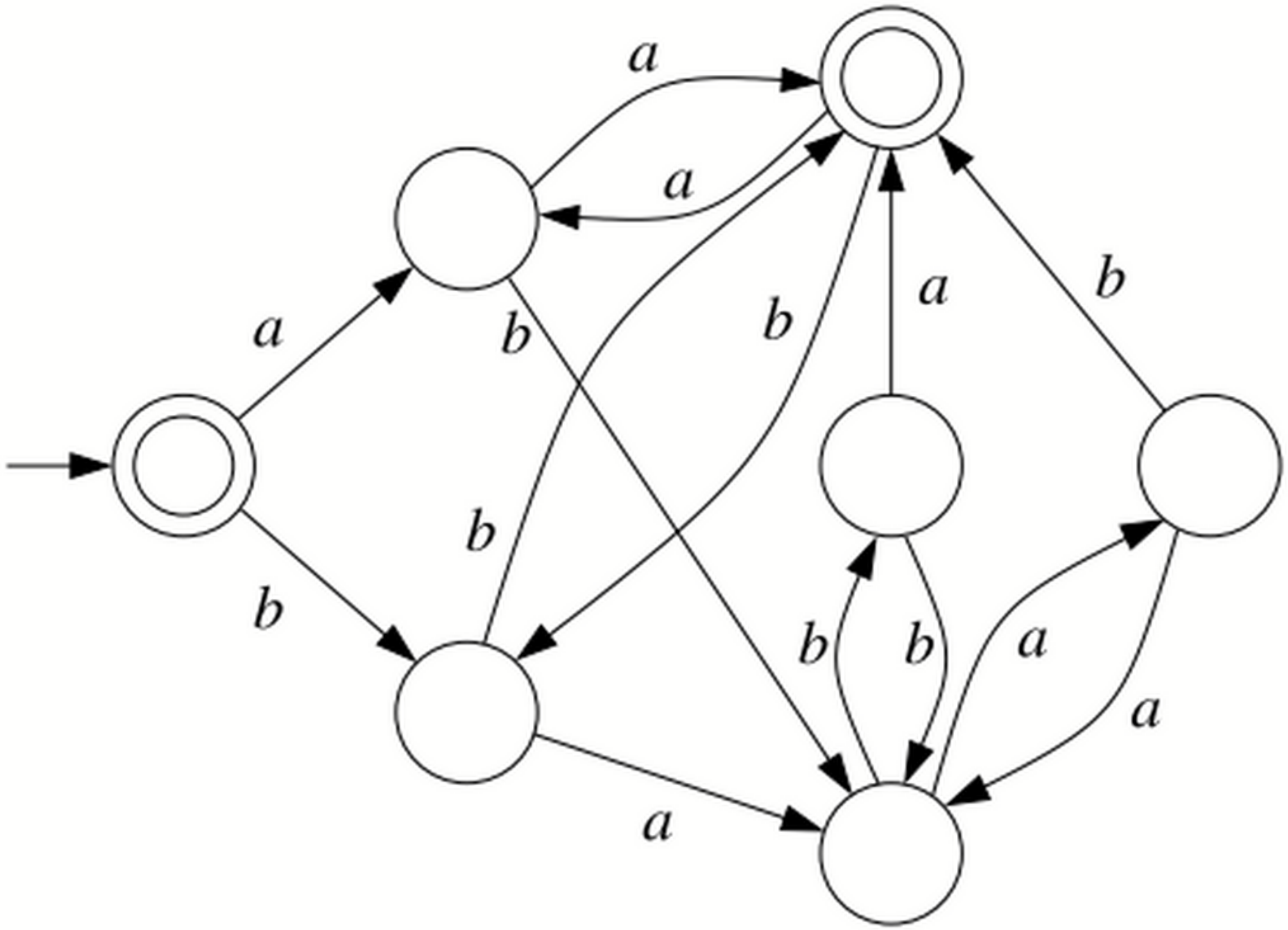
Minimization and Reduction





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

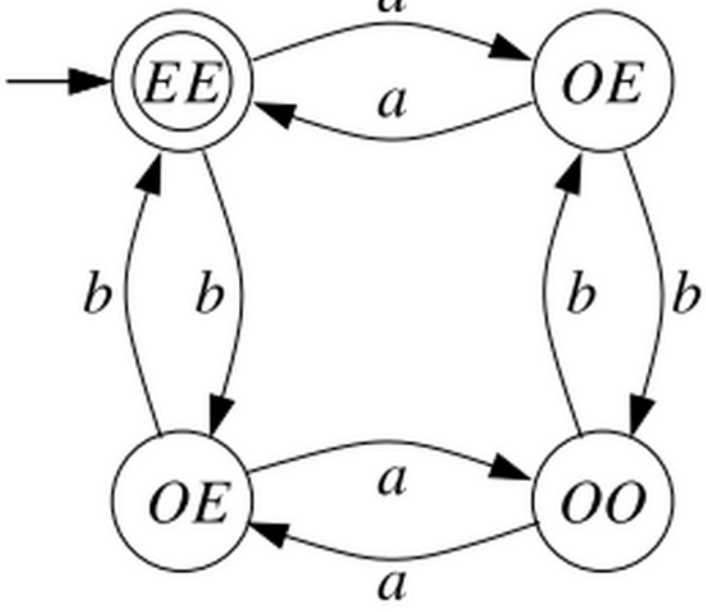
Relation between residuals and states

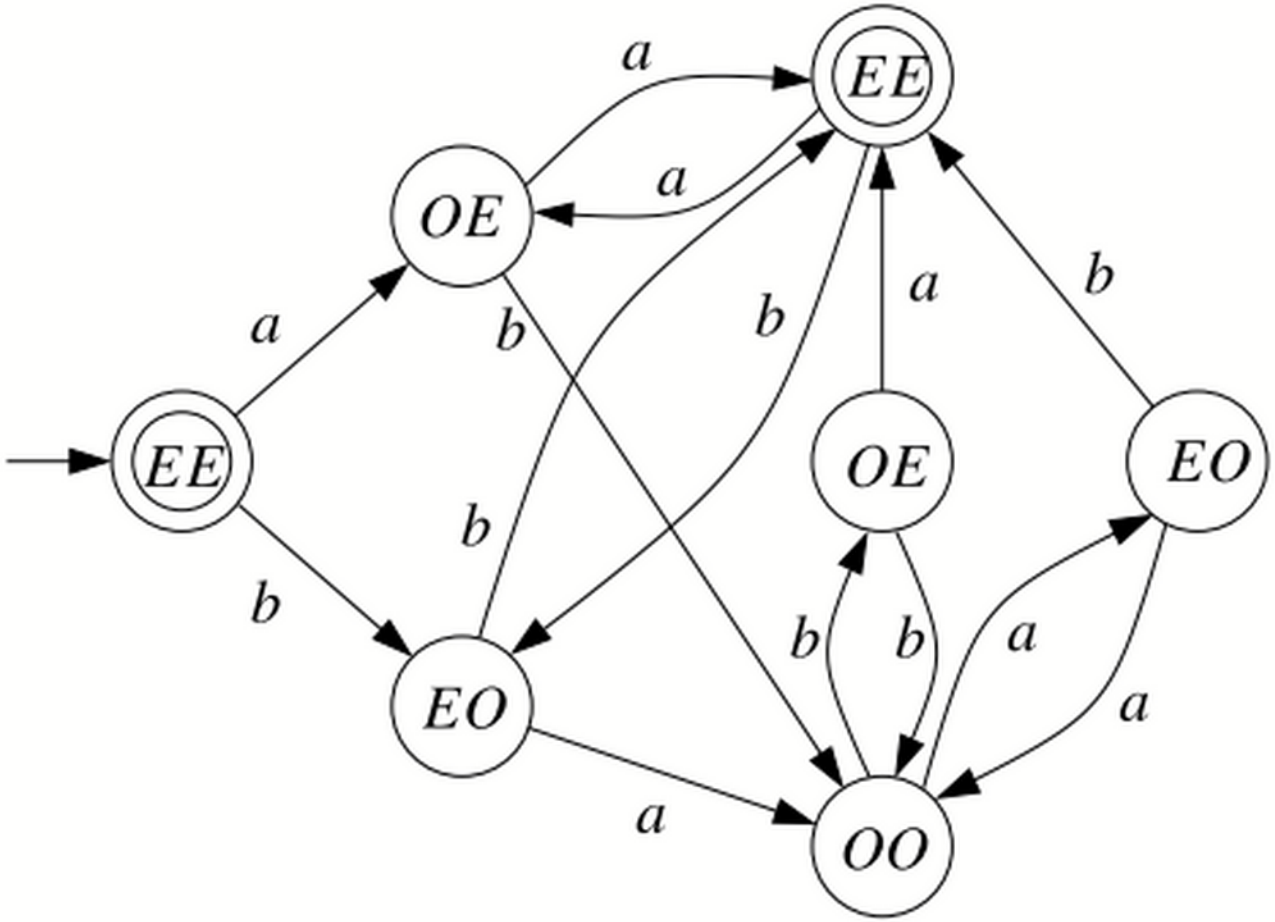
Let A be a (finite or infinite) deterministic automaton.

Def: The language of a state q of A, denoted by L_A(q) or L(q), is the language recognized by A with q as initial state

- State-languages are residuals.
  For every state q of A: L(q) is a residual of L(A)

- Residuals are state-languages:
  For every residual R of L(A): there is a state q such that R=L(A).


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


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

- 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 *the canonical DA for L recognizes L.*

Proof: Let C_L be the canonical DA for L. We prove $L(C_L) = L$.

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

$$\begin{array}{lll} & \varepsilon \in L & (w = \epsilon) \\ \Leftrightarrow & L \in F_L & (\text{definition of } F_L) \\ \Leftrightarrow & q_{0L} \in F_L & (q_{0L} = L) \\ \Leftrightarrow & \varepsilon \in L(C_L) & (q_{0L} \text{ is the initial state of } C_L) \end{array}$$

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

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

Proof:

(1) C_L is a DFA for L with a minimal number of states.
Because C_L has as many states as residuals, and every DFA for L has at least as many states as residuals.

(2) Every minimal DFA for L is isomorphic to C_L
Let A be an arbitrary DFA for L. Then:
- the states of A are in bijection with the residuals of L
- the transitions are completely determined by this bijection:
  if q corresponds to L^w, then delta(q, a) corresponds to L^wa
- the initial state is the state corresponding to L
- the final states are the states for the residuals containing the empty word.

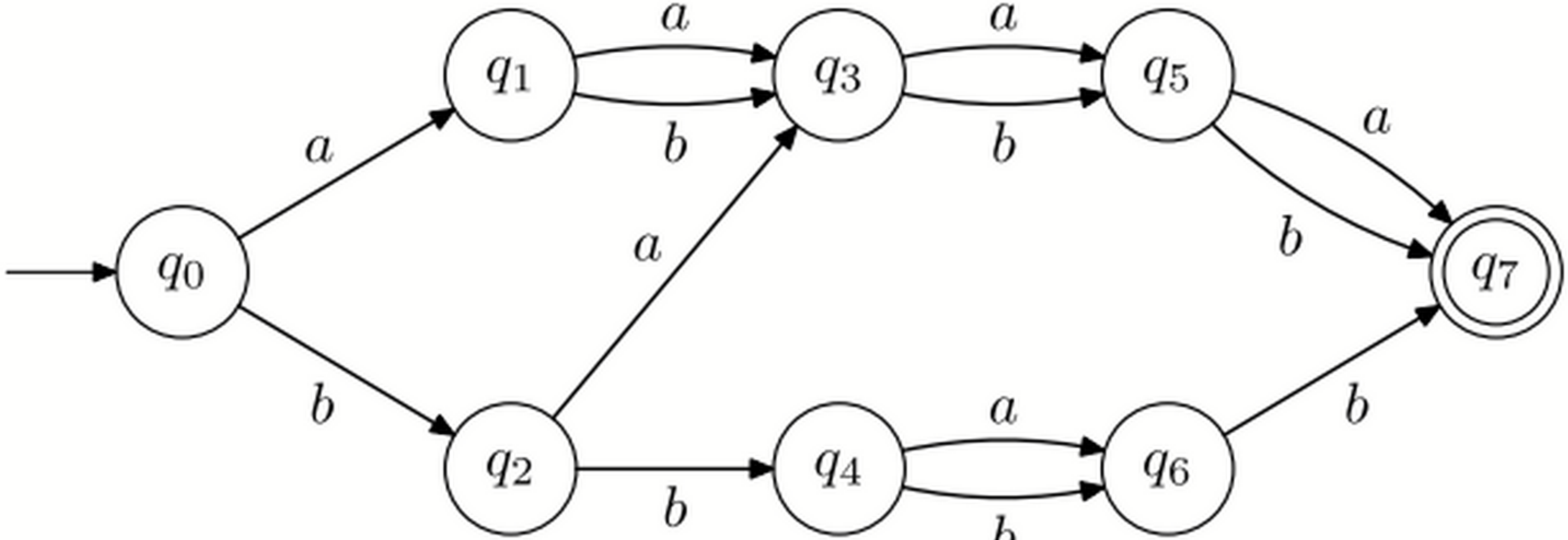
Corollary 3.8 *A DFA is minimal if and only if $L(q) \neq L(q')$ for every two distinct states q and q'.*

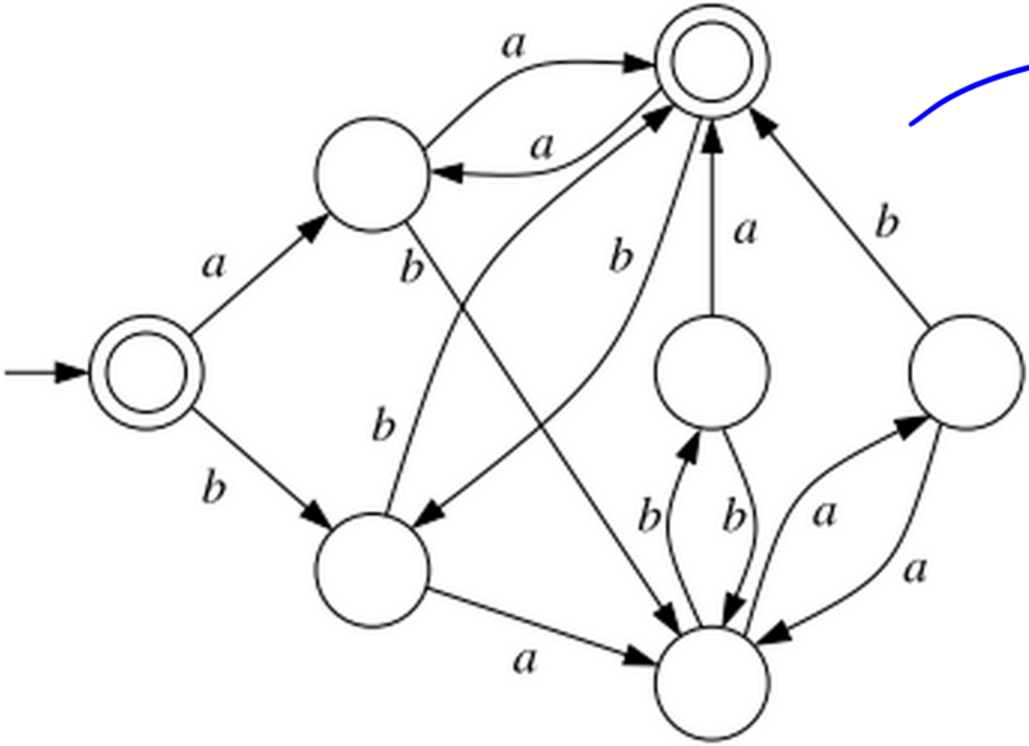
Proof:

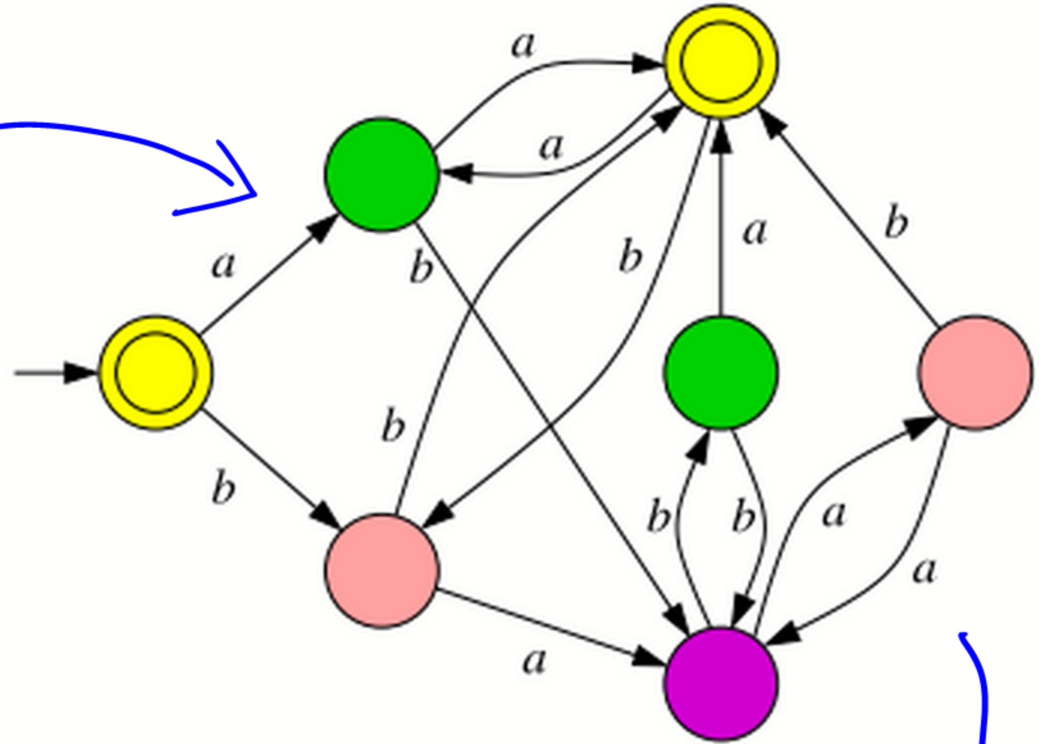
(=>) The states of a DFA recognize all residuals.
Minimal DFA have exactly as many states as residuals.
So each state recognizes a different residual.

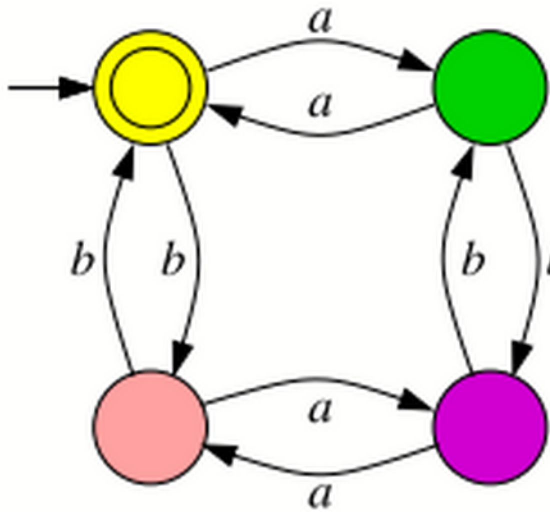
(<=) Every state of a DFA recognizes a residual.
If all state-languages are distinct, then the number of residuals is equal to the number of states, and equal to the number of states of C_L.

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.

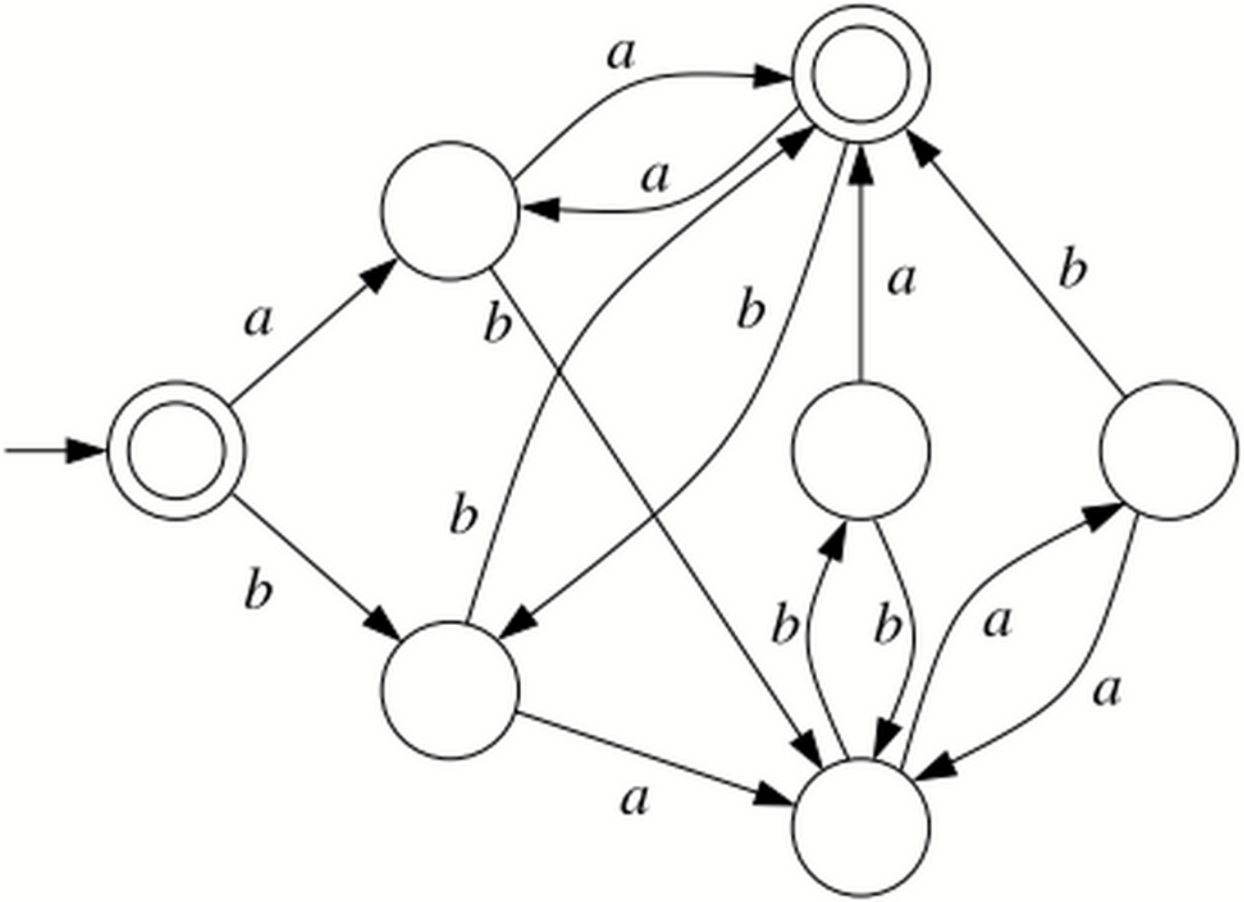
Ref_P(B,a,B') denotes the result of splitting block B
o partition P in two parts, as follows:
- states whose a-transition leads to B' (e.g. q1)
- states whose a-transition leads elsewhere (e.g. q2)

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
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.

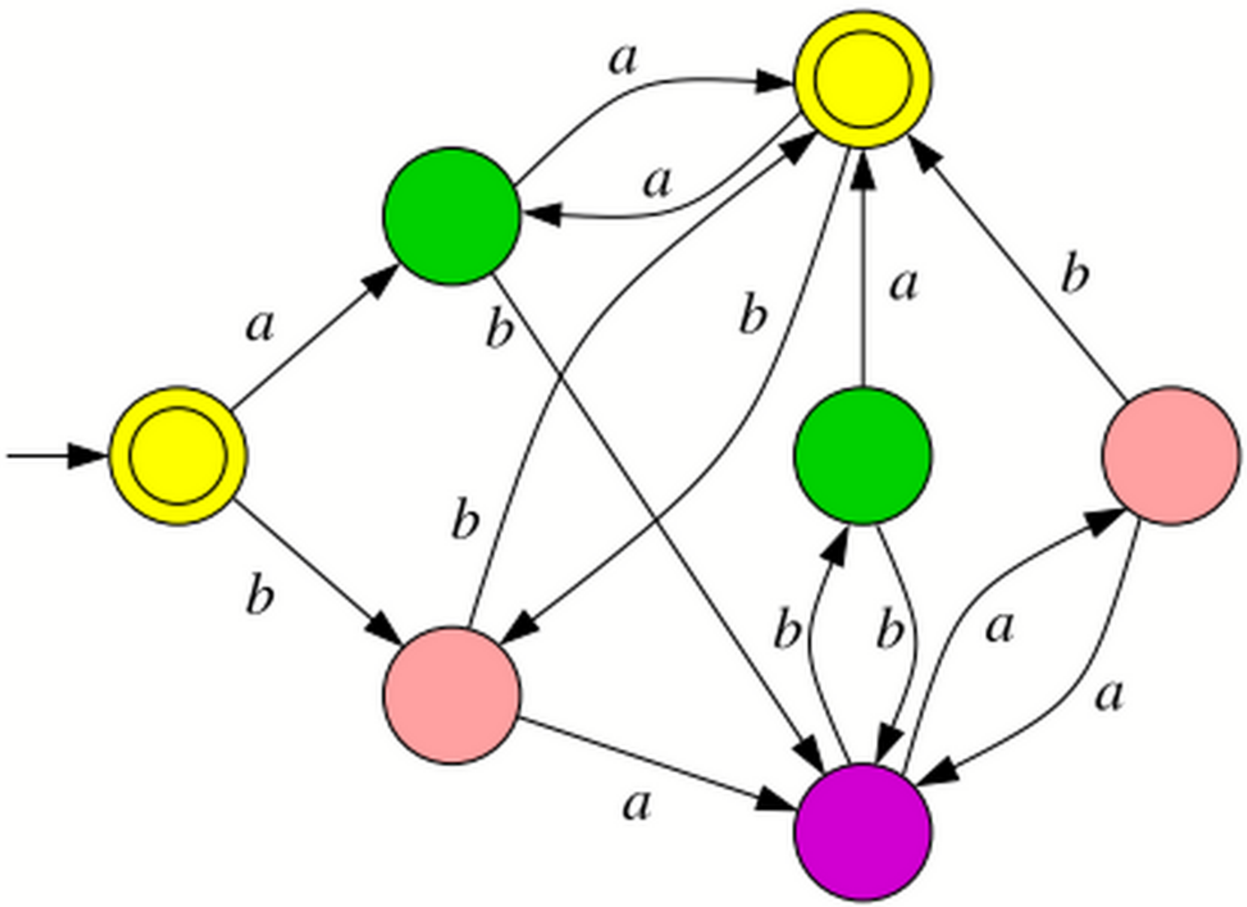
- |w|>0. Then w = aw'. Let q1'=delta(q1,a) and q2'=delta(q2,a). By induction hypothesis q1',q2' are put at some point in different blocks (DFA property). If at this point q1,q2 still belong to the same block B, then B becomes unstable and is eventually split.

Quotienting

Quotient w.r.t. a partition

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

- $Q_P = P$;
- $(B, a, B') \in \delta_P$ *if* $(q, a, q') \in \delta$ *for some* $q \in B$, $q' \in B'$;
- $q_{0B} = [q_0]_P$; *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 w.r.t. the language partition recognize the same language

Lemma 3.16 *Let P be a refinement of P_ℓ, let q be a state of A, and let B be the block of P containing q. Then $L_A(q) = L_{A/P}(B)$.*

Proof: show by induction on |w|: $w \in L_A(q)$ iff $w \in L_{A/P}(B)$

$$\begin{array}{ll} & \varepsilon \in L_A(q) \\ \text{iff} & q \in F \\ \text{iff} & B \subseteq F \qquad\qquad (P \text{ refines } P_\ell\text{, and so also } P_0) \\ \text{iff} & B \in F_P \\ \text{iff} & \varepsilon \in L_{A/P}(B) \end{array}$$

|w| > 0. Then w = w'a.

There is a transition (q,a,q') in A such that w' in L_A(q').

There is a transition (B,a,B') in A/P such that q' in B'.

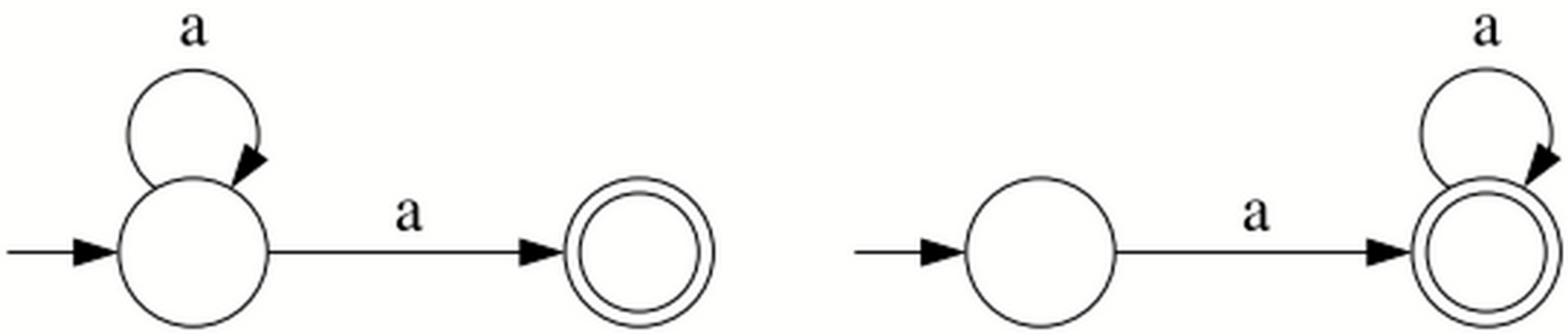
$$
\begin{array}{lll}
 & aw' \in L_A(q) & \\
\text{iff} & w' \in L_A(q') & \\
\text{iff} & w' \in L_{A/P}(B') & \text{(induction hypothesis)} \\
\text{iff} & aw' \in L_{A/P}(B) & (\ (B, a, B') \in \delta_P\ )
\end{array}
$$

(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

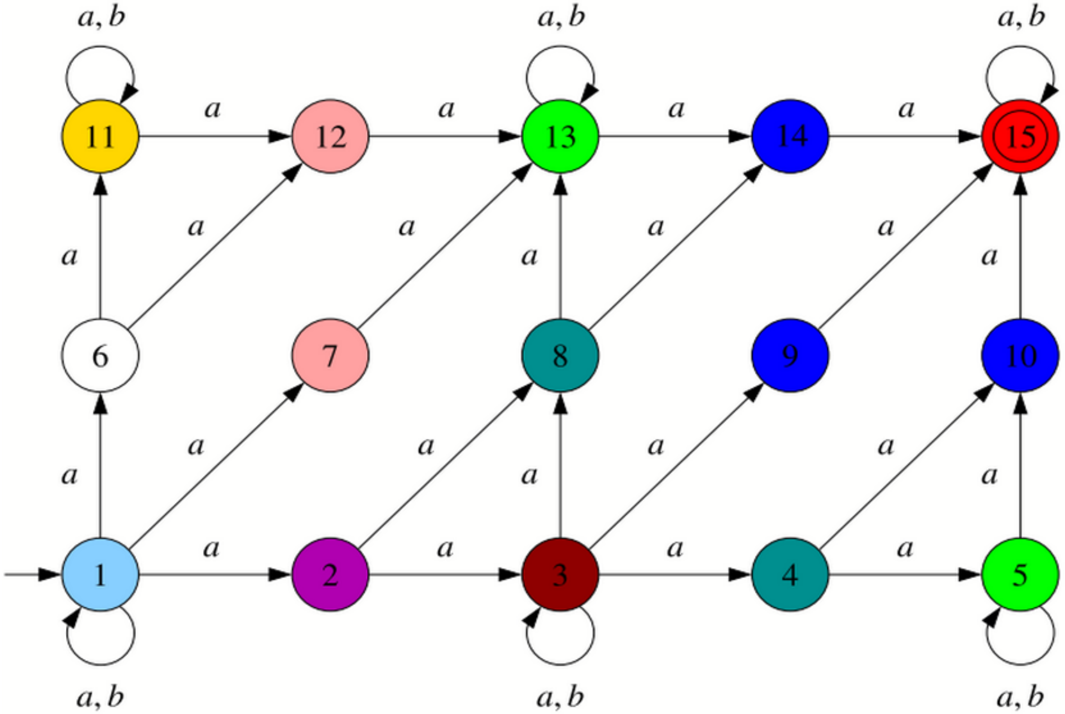
The minimal NFA is not unique

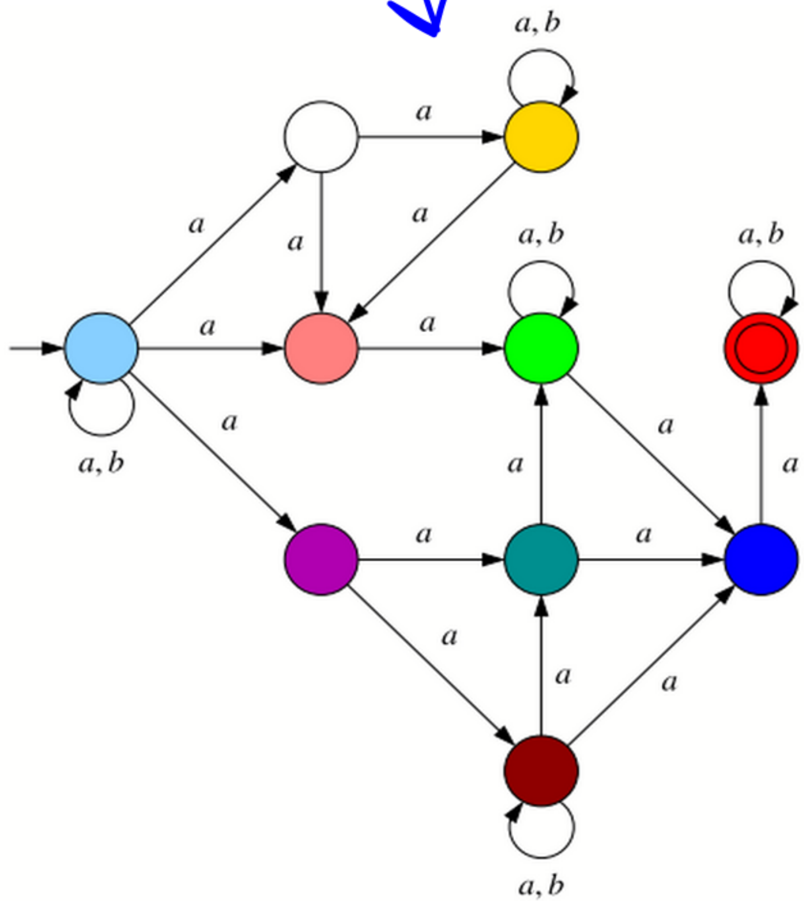


Minimal NFAs are hard to compute

Theorem 3.18 *The following problem is PSPACE-complete: given a NFA A and a number k* ≥ 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.

Lemma 3.16 *Let P be a refinement of P_ℓ, let q be a state of A, and let B be the block of P containing q. Then $L_A(q) = L_{A/P}(B)$.*

So we need to compute a partition such that states in the same block recognize the same language.

Such partitions necessarily refine the 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:

- one of them is final and the other nonfinal, or
- one of them, say q1, satisfies: there is a letter, say a, and a state in delta(q1,a), say q1', such that for every q2' in delta(q2, a): L(q1') and L(q2') are not equal.

This suggests:

Definition 3.19 (Refinement and stability for NFAs) *Let B, B' be (not necessarily distinct) blocks of a partition P, 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 result of the split is the partition $Ref_P^{NFA}[B, a, B'] = (P \setminus \{B\}) \cup \{B_0, B_1\}$, where*

$$B_0 = \{q \in B \mid \delta(q, a) \cap B' = \emptyset\} \text{ and } B_1 = \{q \in B \mid \delta(q, a) \cap B' \neq \emptyset\} .$$

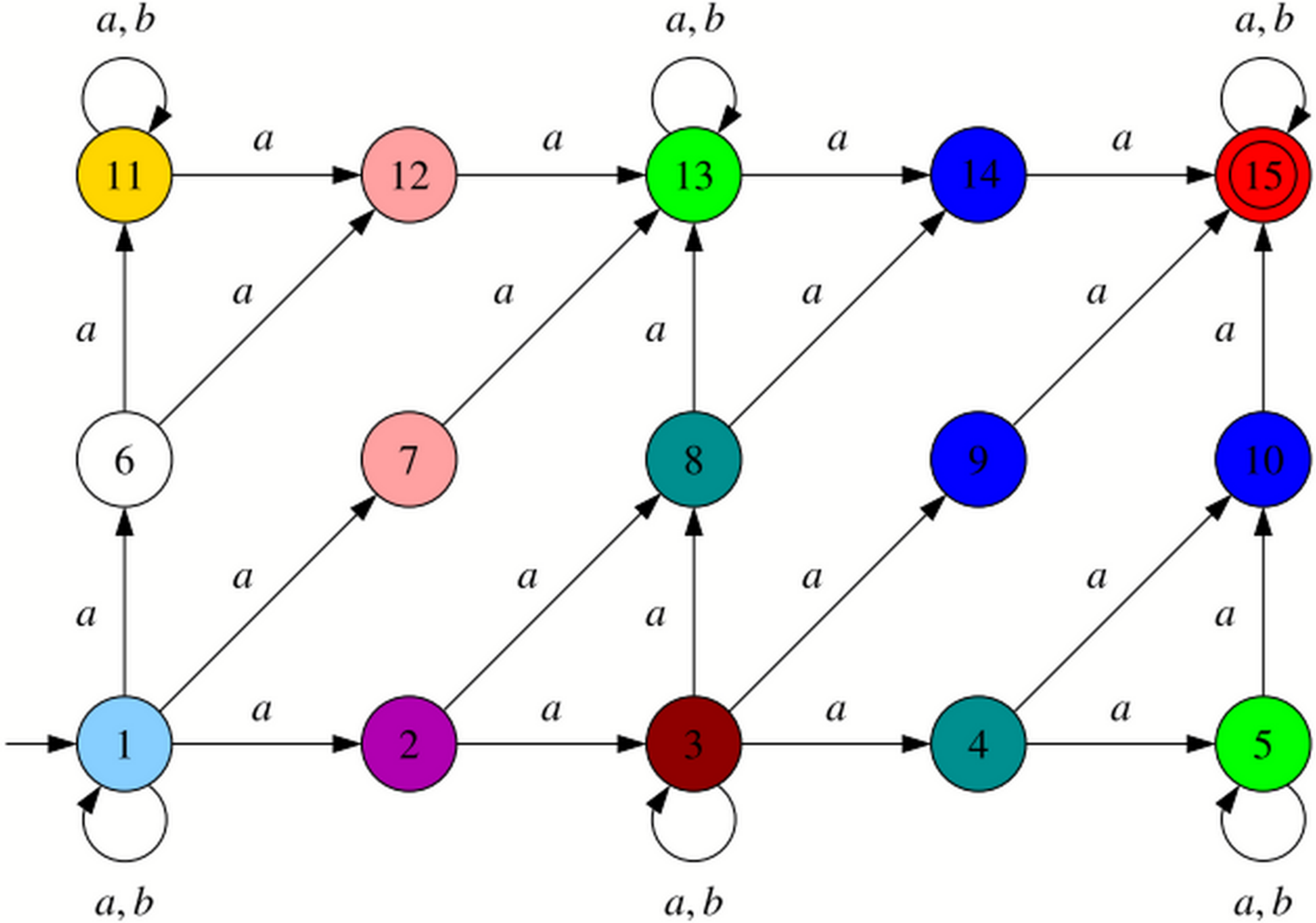
A partition is unstable *if it contains blocks B, B' such that B' splits B, and* stable *otherwise.*

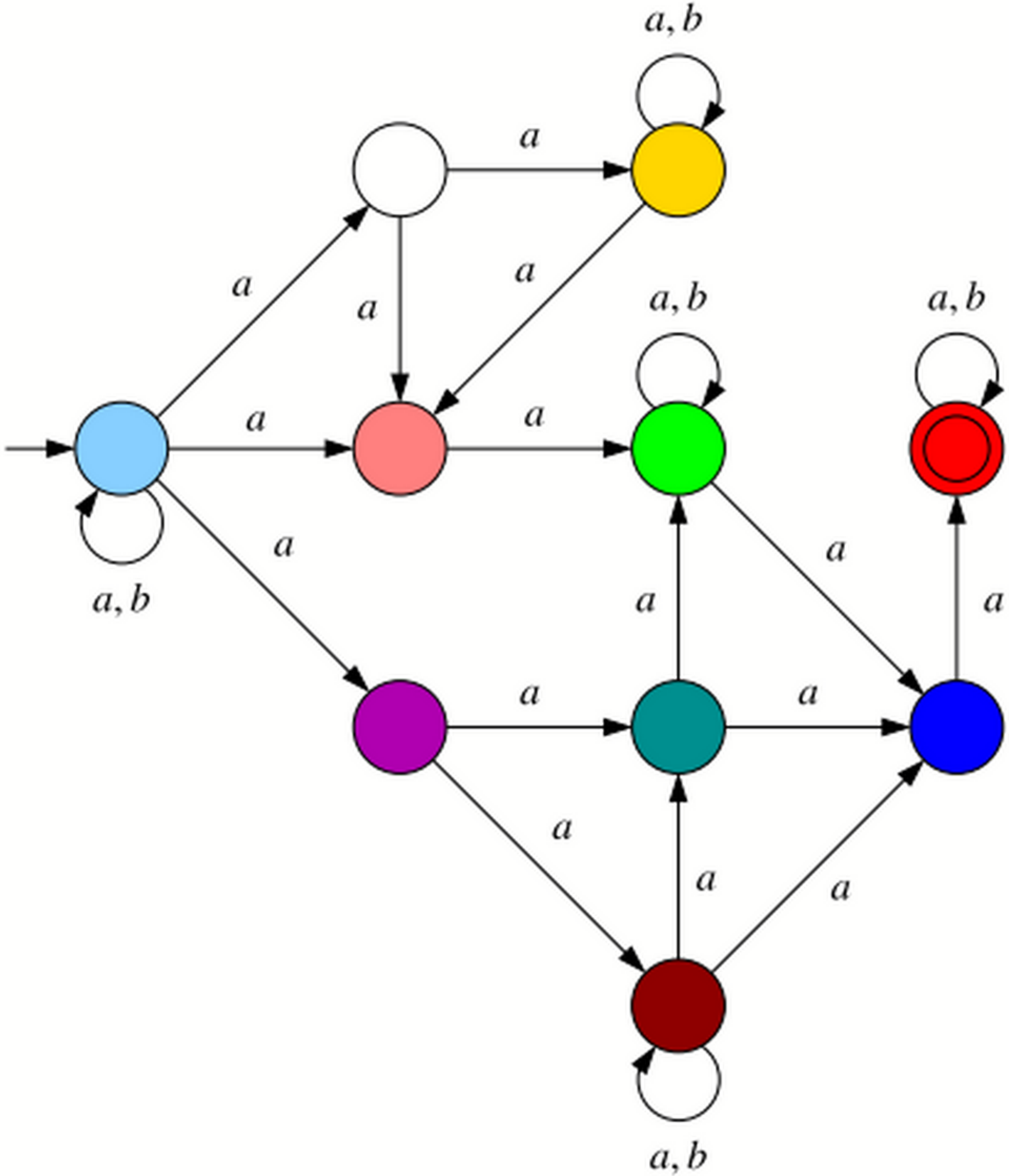
CSR(*A*)

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

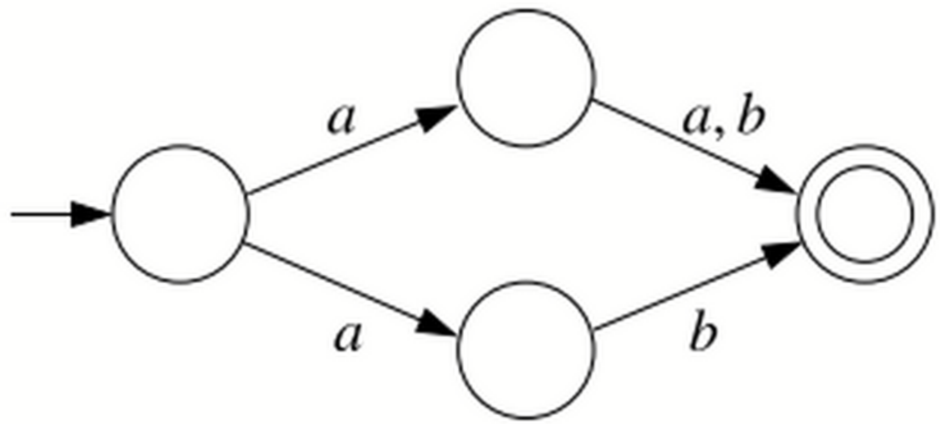
Output: The partition *CSR*.

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

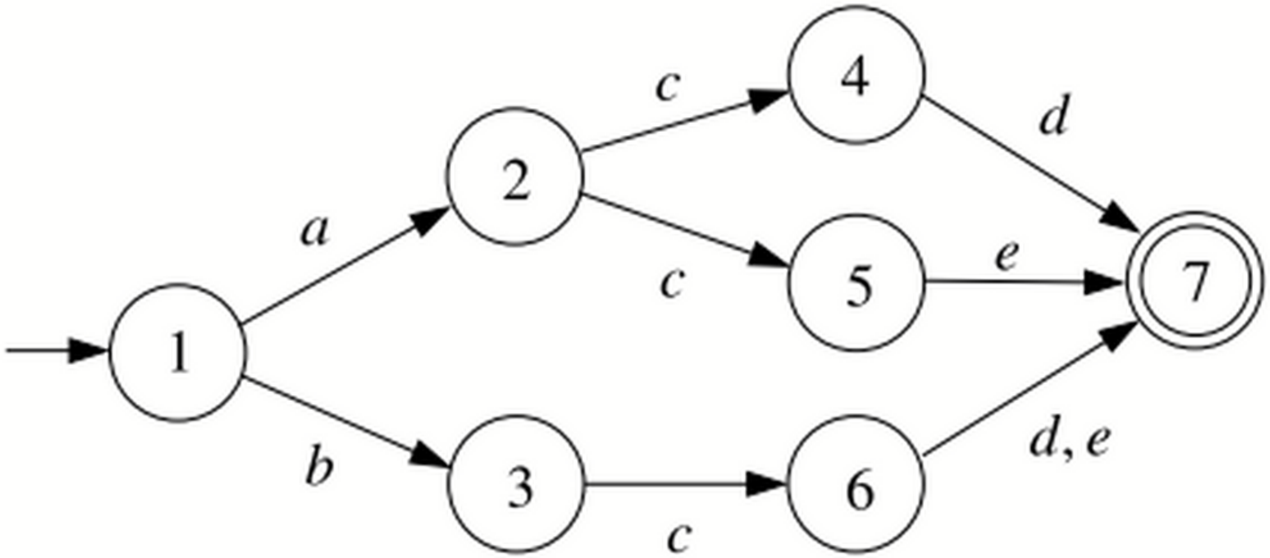

a, b
a, b
a, b
11
12
13
14
15
a
a
a
a
6
7
8
9
10
1
2
3
4
5
a, b
a, b
a, b


a, b
a
a
a
a
a, b
a, b
a
a
a, b
a
a
a
a
a
a
a
a
a
a, b

Reduction may not minimize



The algorithm does not compute the language partition



Non-regular languages