Informed Learners

Presented by: Joe Scott

Automata+Learning
Uppsala University

12 February, 2014
(Charles Darwin's 205^{th} birthday!)

Outline

1. informed learners
   - background
   - the general case
2. RPNI
   - overview
   - algorithm
   - example

Outline

Learning from an informant

Definition (informed learning sample)

An *informed learning sample* S is composed of a set S_+ of positive strings, and a set S_- of negative strings, such that $S_+ \cap S_- = \emptyset$.

Definition (deterministic finite automaton)

A *dfa* A is a tuple $A = \langle \Sigma, Q, q_\lambda, F_a, F_r, \delta \rangle$ where:

- Σ is an alphabet
- Q is a finite set of states
- $q_\lambda \in Q$ is the initial state
- $\delta \colon Q \times \Sigma \to Q$ is a transition function
- $F_a, F_r \subseteq Q$ are sets of final accepting states and final rejecting states

Definition

$\mathcal{L}_{F_a}(A)$ is the language *recognized* by A;
$\mathcal{L}_{F_r}(A)$ is the language *recognized by rejection* in A.

Consistency

Definition (weakly consistent)

A is *weakly consistent* with the sample $S = \langle S_+, S_- \rangle$ if

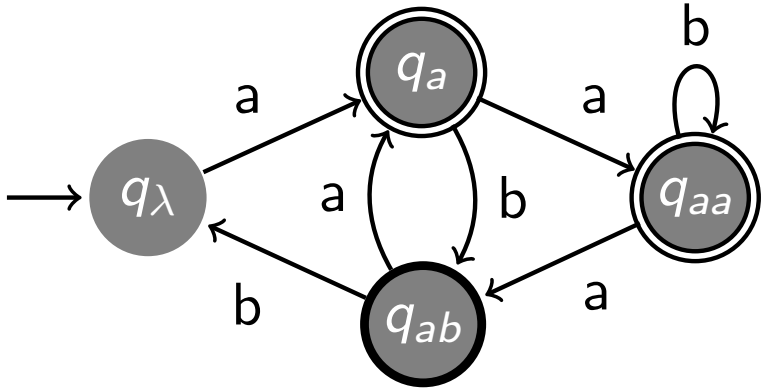
$$\forall x \in S_+ : \delta(q_\lambda, x) \in F_a \text{ and } \forall x \in S_- : \delta(q_\lambda, x) \notin F_a$$

Definition (strongly consistent)

A is *strongly consistent* with the sample $S = \langle S_+, S_- \rangle$ if

$$\forall x \in S_+ : \delta(q_\lambda, x) \in F_a \text{ and } \forall x \in S_- : \delta(q_\lambda, x) \in F_r$$

$S_+ = \{ aa \}$
$S_- = \{ abb, b \}$

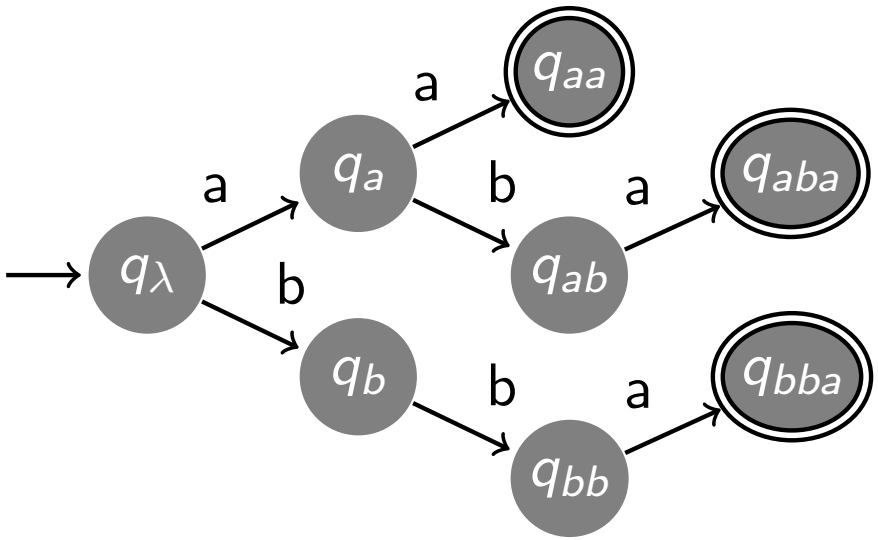


$S_+ = \{ aa, aba \}$
$S_- = \{ abab \}$

Definition (prefix tree acceptor)

$PTA(S_+)$, is the DFA $A = \langle \Sigma, Q, q_\lambda, F_a, F_r, \delta \rangle$ such that:

- $Q = \{ q_u \mid u \in \text{PREF}(S_+) \}$
- $\forall ua \in \text{PREF}(S_+) \colon \delta(q_u, a) = q_{ua}$
- $F_a = \{ q_u \mid u \in S_+ \}$
- $F_r = \emptyset$



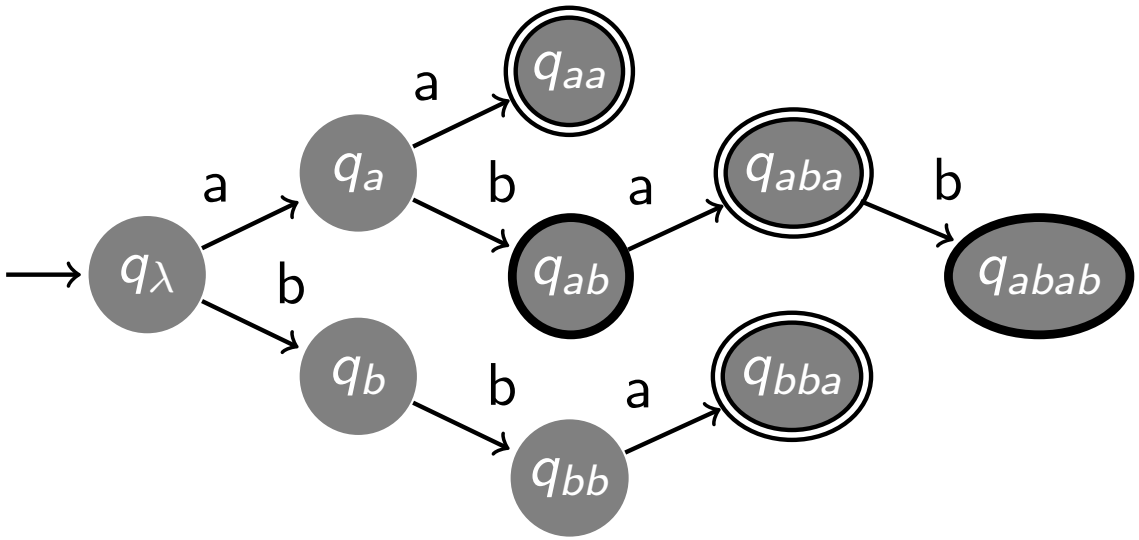
The smallest DFA recognizing S_+ that is a tree.

$S_+ = \{ aa, aba, bba \}$ $S_- = \{ ab, abab \}$

Definition (prefix tree acceptor)

$PTA(S_+, S_-)$ is the DFA $A = \langle \Sigma, Q, q_\lambda, F_a, F_r, \delta \rangle$ such that:

- $Q = \{ q_u \mid u \in \text{PREF}(S_+ \cup S_-) \}$
- $\forall ua \in \text{PREF}(S_\cup S_-)\colon \delta(q_u, a) = q_{ua}$
- $F_a = \{ q_u \mid u \in S_+ \}$
- $F_r = \{ q_u \mid u \in S_- \}$



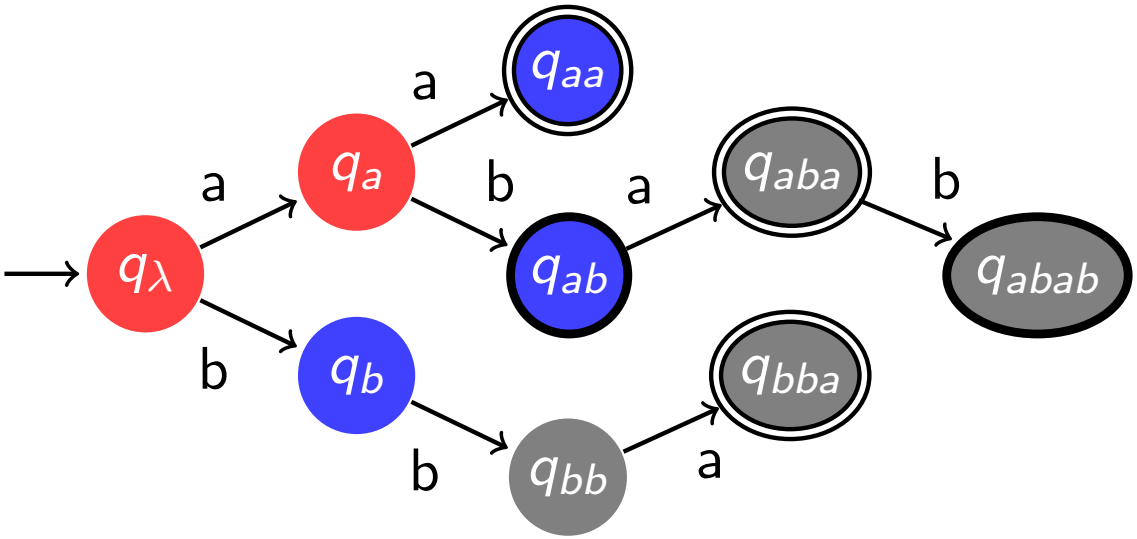
The smallest DFA that is a tree, with $S_+ \subseteq \mathcal{L}_{F_a}$ and $S_- \subseteq \mathcal{L}_{F_r}$.

$S_+ = \{ aa, aba, bba \}$ $\quad$ $S_- = \{ ab, abab \}$

Outline

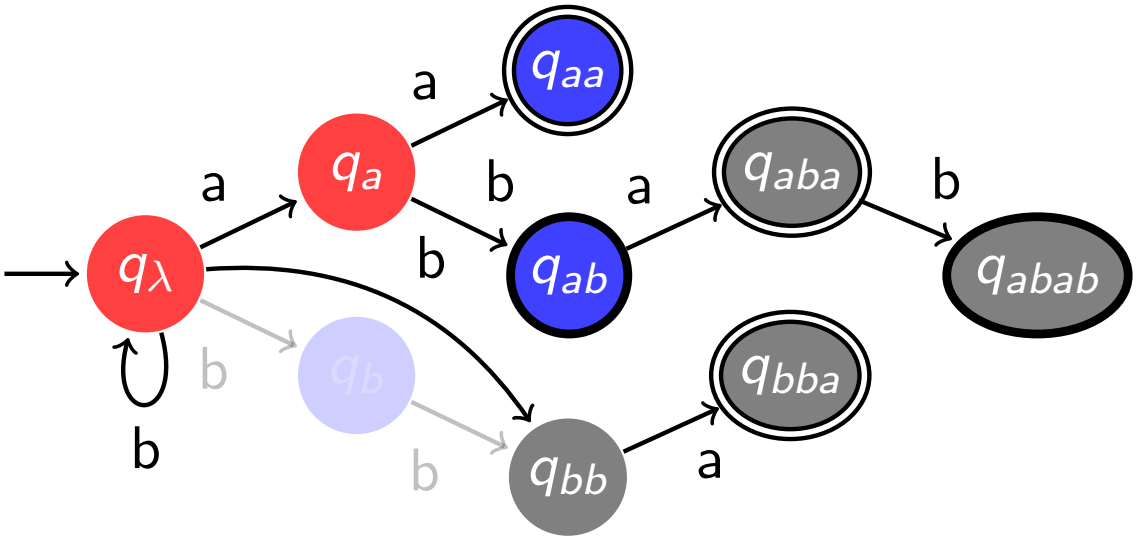
General view of informed learners

- algorithms for informed learning generally start with a PTA
- generalize by merging states
- divide states into three categories:
  - RED states: already determined to form part of the final automaton,
  - BLUE states: candidate states, to either merge with a RED state, or add to the RED states, and
  - WHITE states: everything else



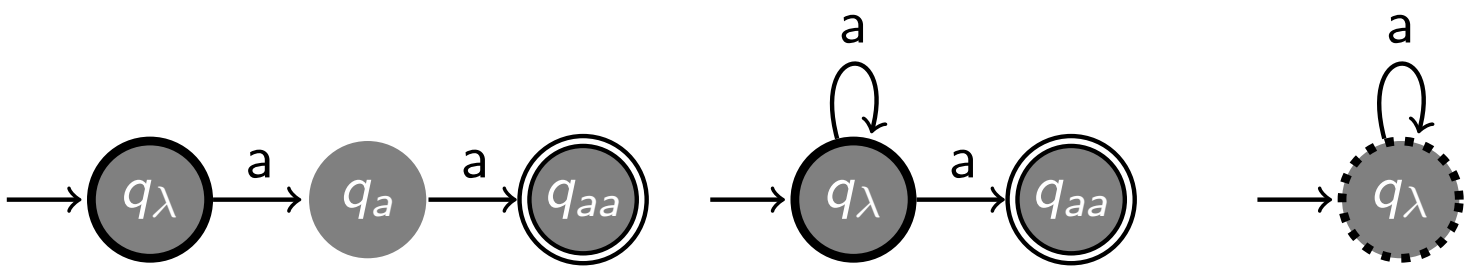
General view of informed learners

- algorithms for informed learning generally start with a PTA
- generalize by merging states
- divide states into three categories:
  - RED states: already determined to form part of the final automaton,
  - BLUE states: candidate states, to either merge with a RED state, or add to the RED states, and
  - WHITE states: everything else



generic operations for informed learners

- compatible
  - two states are compatible if they can be merged without creating confusion between accepting and rejecting states
- merge
  - merging may lead to non-determinism
  - can result in a state which is both accepting and rejecting
  - such merges have to be rejected



- promote
  - promote a BLUE state which is different then all RED states

Outline

Regular Positive and Negative Inference (RPNI)[1]

- Greedily merges states in the PTA
- result is always consistent, and more general than the PTA

- starts with a PTA constructed from S_+

- avoids non-determinism following merges by adding a fold operation

[1] Oncina & García. Identifying regular languages in polynomial time. In H. Burke, ed., *Advances in Structural and Syntactic Pattern Recognition*, v. 5 of *Series in Machine Perception and Artificial Intelligence*, pp. 99–108. World Scientific, 1992.

promoting a state

Algorithm 1: PROMOTE$(A, \mathrm{RED}, \mathrm{BLUE}, q_u)$

input : dfa $A = \langle \Sigma, Q, q_\lambda, F_a, F_r, \delta \rangle$, sets $\mathrm{RED}, \mathrm{BLUE} \subseteq Q$, $q_u \in \mathrm{BLUE}$
output: $A, \mathrm{RED}, \mathrm{BLUE}$
$\mathrm{RED} \leftarrow \mathrm{RED} \cup \{ q_u \}$
$\mathrm{BLUE} \leftarrow \mathrm{BLUE} \setminus \{ q_u \}$
$\mathrm{BLUE} \leftarrow \mathrm{BLUE} \cup \{ \delta(q_u, a) \mid a \in \Sigma \}$
return $A, \mathrm{RED}, \mathrm{BLUE}$

- promotes a BLUE state to RED
- all successors of the promoted state become BLUE

state equivalence

Algorithm 2: COMPATIBLE(A, S_-)

input : dfa $A = \langle \Sigma, Q, q_\lambda, F_a, F_r, \delta \rangle$, sample S_-
output: Boolean
for $w \in S_-$ **do**
    if $\delta(q_\lambda, w) \cap F_a \neq \emptyset$ **then**
        return *false*
return *true*

- returns false iff some counter-example is accepted by the automaton

merging two states, part 1

Algorithm 3: $\textsc{merge}(A, q, q')$

input : dfa $A = \langle \Sigma, Q, q_\lambda, F_a, F_r, \delta \rangle$, states $q \in \text{RED}, q' \in \text{BLUE}$
output: A
Let (q_f, a) be such that $\delta(q_f, a) = q'$
$\delta(q_f, a) \leftarrow q$
return $\textsc{fold}(A, q, q')$

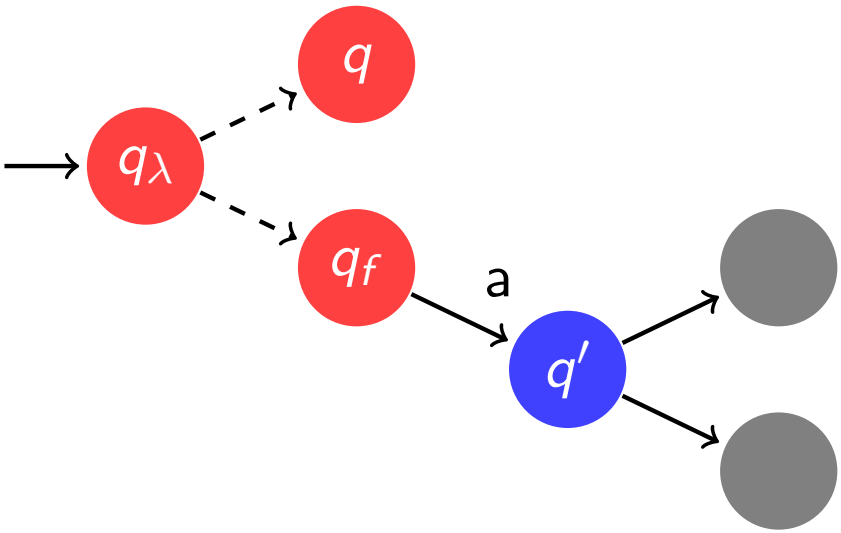
- a unique (q_f, a) must exist, because $q' \in \text{BLUE}$:
  - q' has exactly one predecessor,
  - and q' is the root of a tree.
- the single incoming transition to q' is redirected to q
- yields a disconnected subtree with root q'

merging two states, part 1

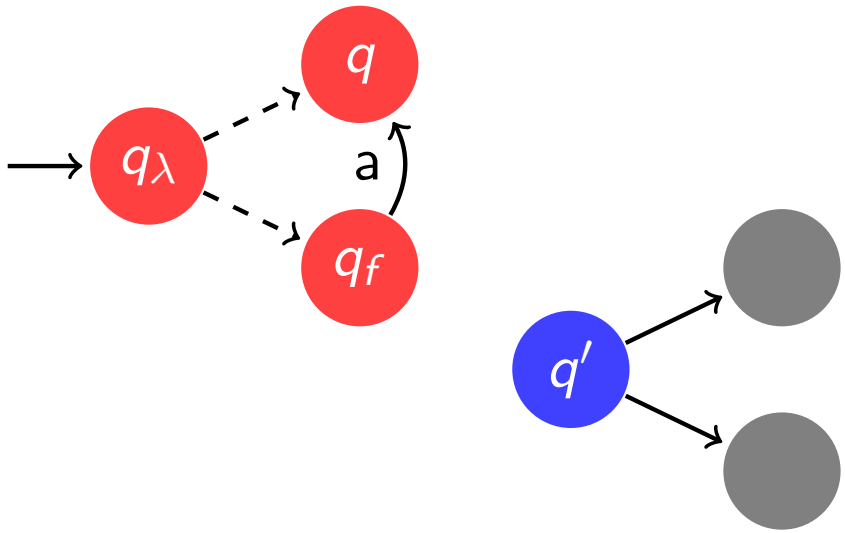
Algorithm 4: $\textsc{merge}(A, q, q')$

input : dfa $A = \langle \Sigma, Q, q_\lambda, F_a, F_r, \delta \rangle$, states $q \in \text{RED}, q' \in \text{BLUE}$
output: A
Let (q_f, a) be such that $\delta(q_f, a) = q'$
$\delta(q_f, a) \leftarrow q$
return $\textsc{fold}(A, q, q')$

before merging:



after merging:



merging two states, part 2

Algorithm 5: $\textsc{fold}(A, q, q')$

input : dfa $A = \langle \Sigma, Q, q_\lambda, F_a, F_r, \delta \rangle$, states $q, q' \in Q$
output: A with subtree of q' folded into q

```
if q' ∈ F_a then
    F_a ← F_a ∪ { q }
for a ∈ Σ do
    if δ(q', a) is defined then
        if δ(q, a) is defined then
            A ← FOLD(A, δ(q, a), δ(q', a))
        else
            δ(q, a) ← δ(q', a)
return A
```

- the fold simultaneously traverses the PTA and the subtree
  - adds missing states, transitions from subtree to PTA
  - updates states that are final in subtree, but not in PTA

Outline

Algorithm 6: RPNI

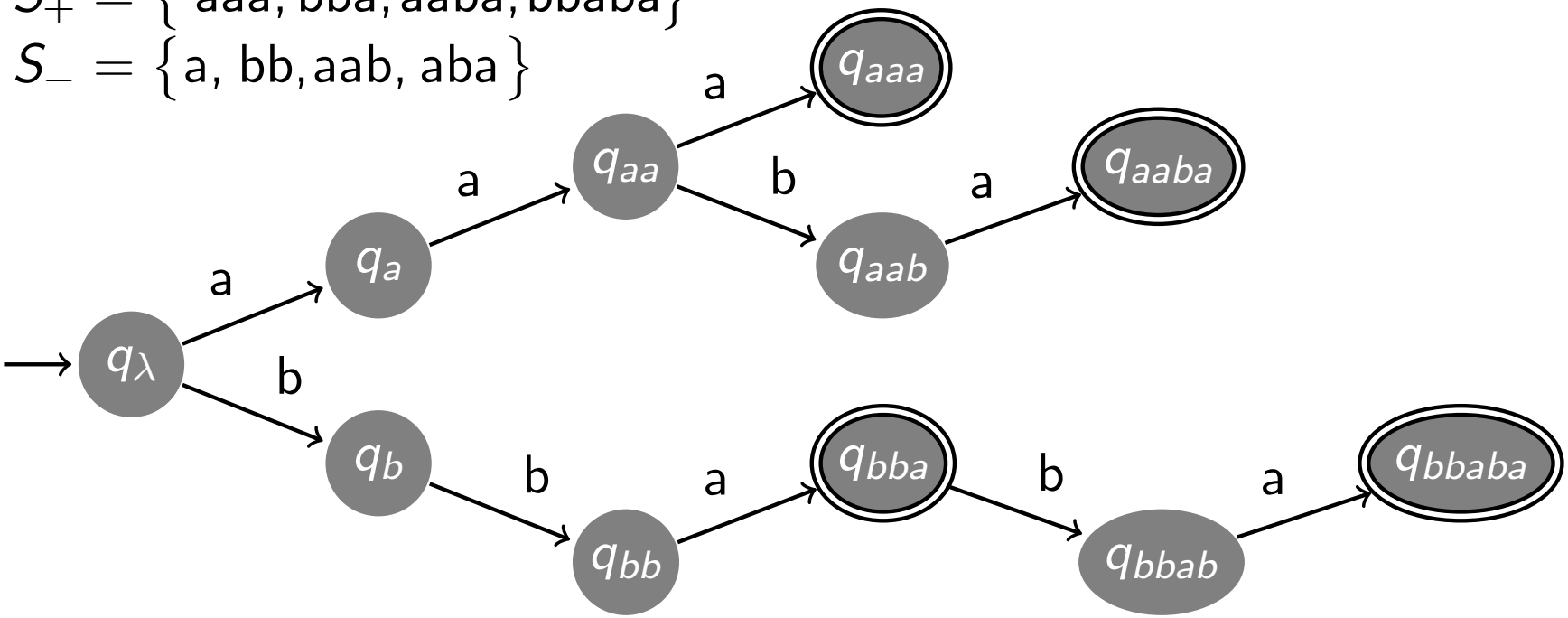
input : sample $S = \langle S_+, S_- \rangle$
output: a dfa $A = \langle \Sigma, Q, q_\lambda, F_a, F_r, \delta \rangle$
$A \leftarrow \text{BUILDPTA}(S_+)$
$\text{RED} \leftarrow \{ q_\lambda \}$
$\text{BLUE} \leftarrow \{ q_a \mid a \in \Sigma \cap \text{PREF}(S_+) \}$
while $\text{BLUE} \neq \emptyset$ **do**
    $\text{CHOOSE}(q_b \in \text{BLUE})$
    if $\exists q_r \in \text{RED}: \text{COMPATIBLE}(\text{MERGE}(A, q_r, q_b), S_-)$ **then**
        $A \leftarrow \text{MERGE}(A, q_r, q_b)$
        $\text{BLUE} \leftarrow \text{BLUE} \cap \{ \delta(q, a) \mid q \in \text{RED}, a \in \Sigma, \delta(q, a) \notin \text{RED} \}$
    else
        $A, \text{RED}, \text{BLUE} \leftarrow \text{PROMOTE}(A, \text{RED}, \text{BLUE}, q_b)$
parse S_- and mark rejecting states

Outline

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

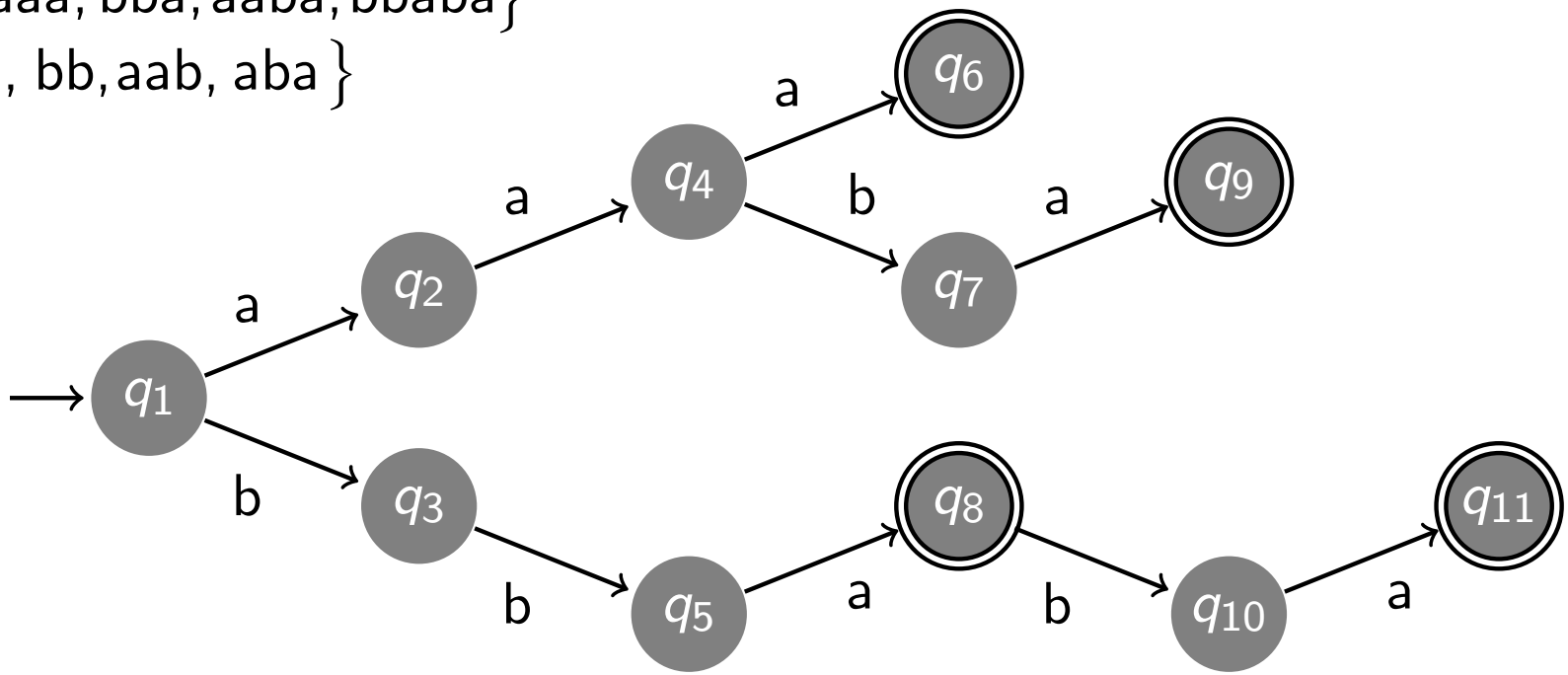
$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

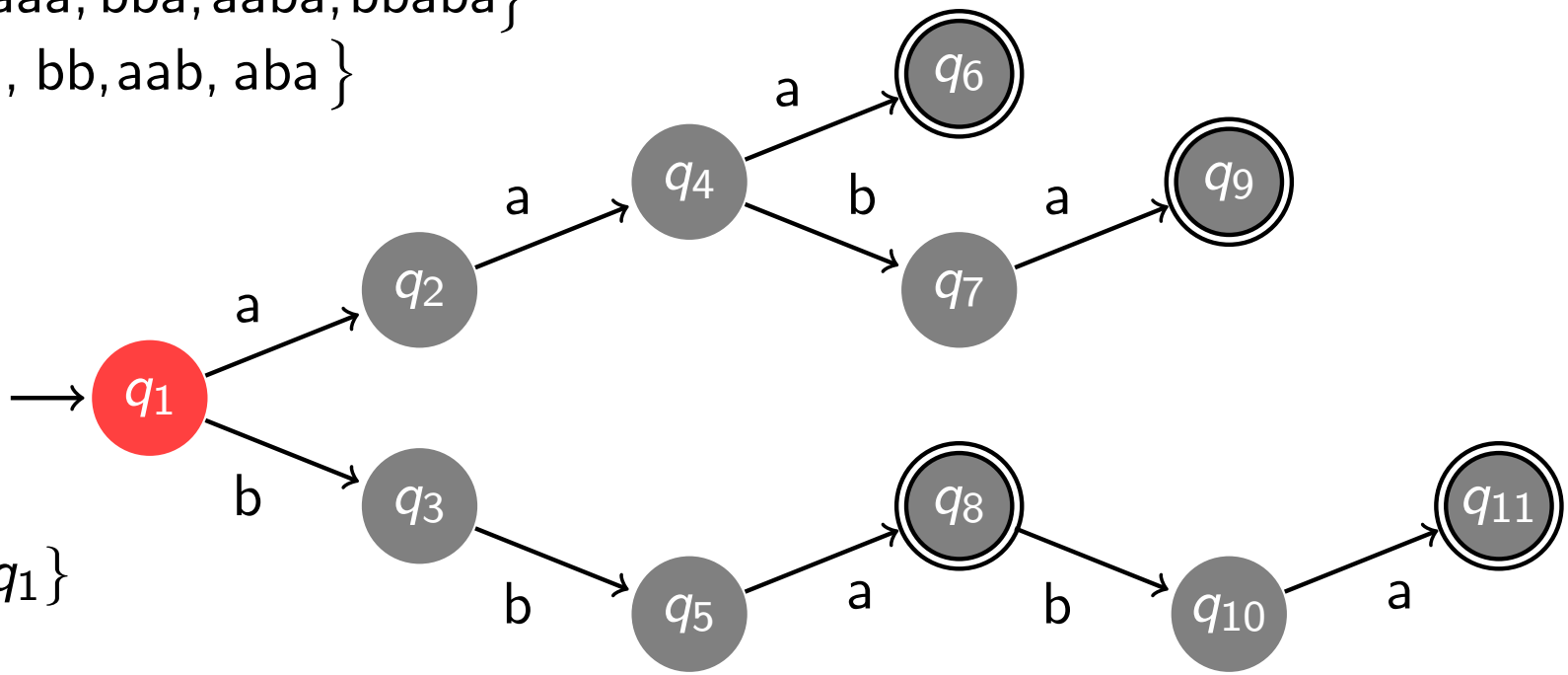
$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$

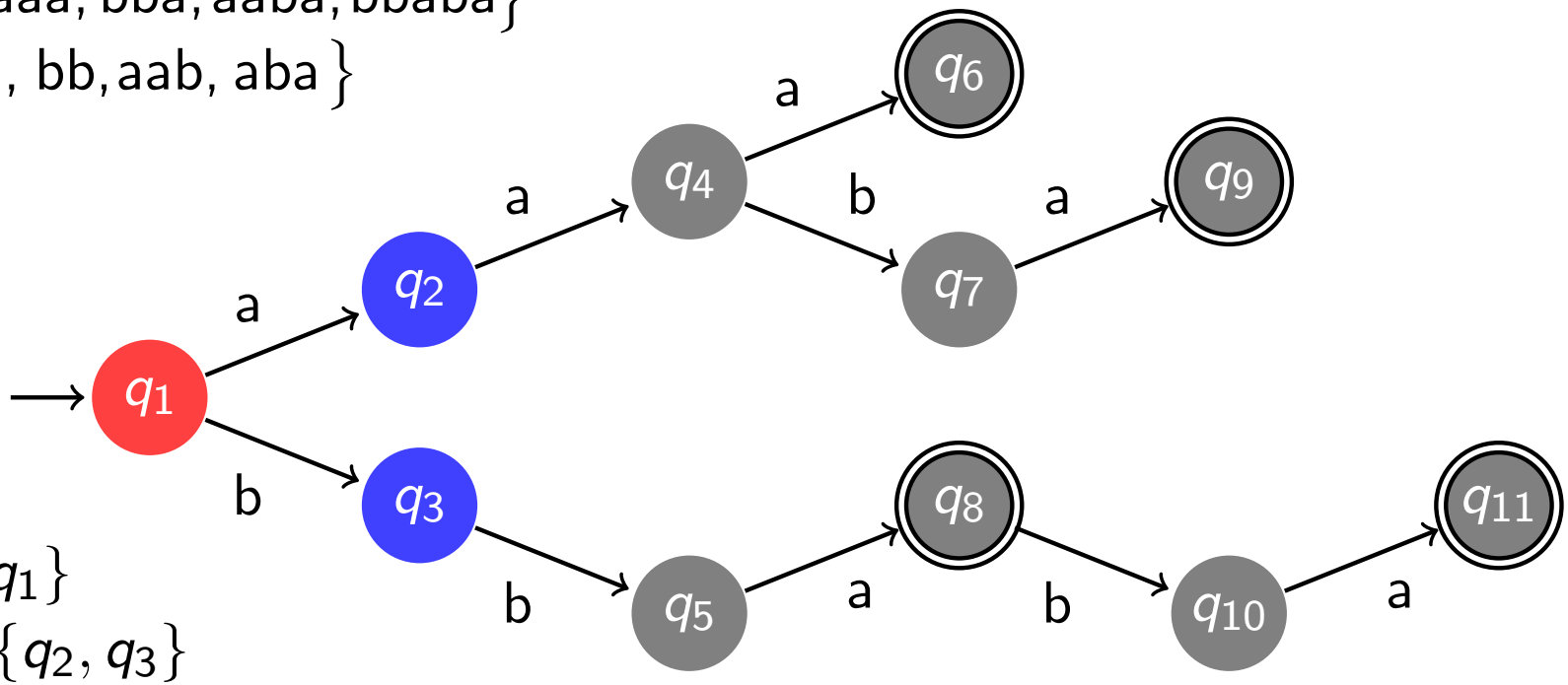


$\text{RED} = \{q_1\}$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



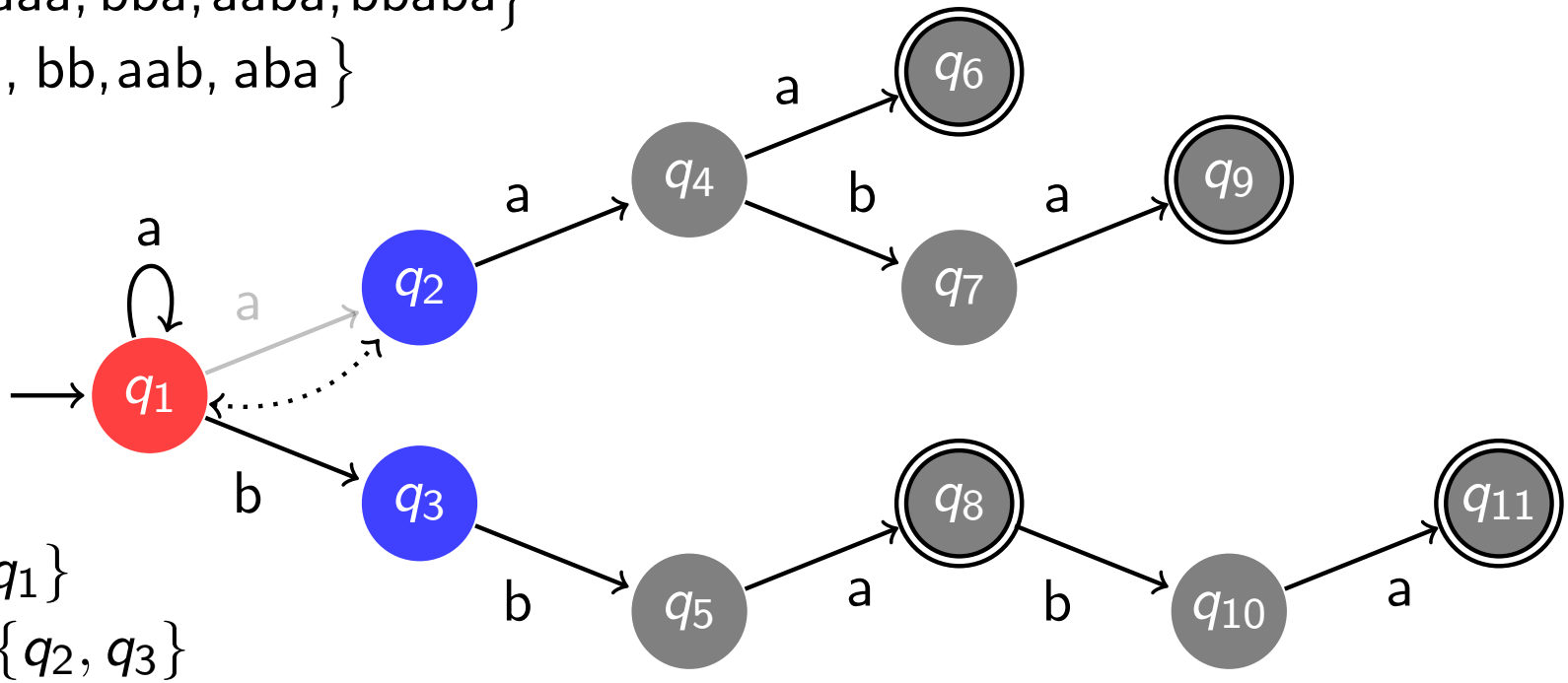
$\text{RED} = \{q_1\}$

$\text{BLUE} = \{q_2, q_3\}$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1\}$

$\text{BLUE} = \{q_2, q_3\}$

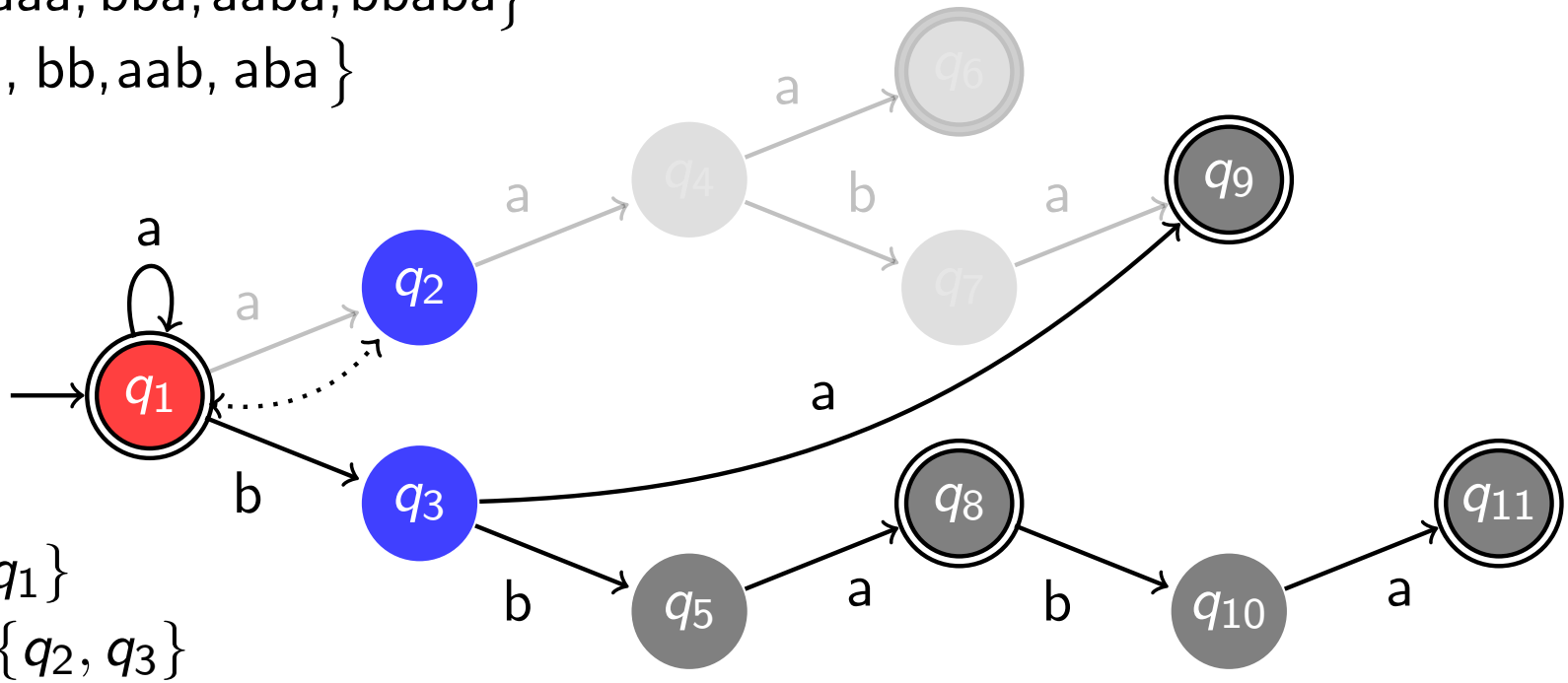
q_2 :

$\text{MERGE}(q_1, q_2)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1\}$

$\text{BLUE} = \{q_2, q_3\}$

q_2 :

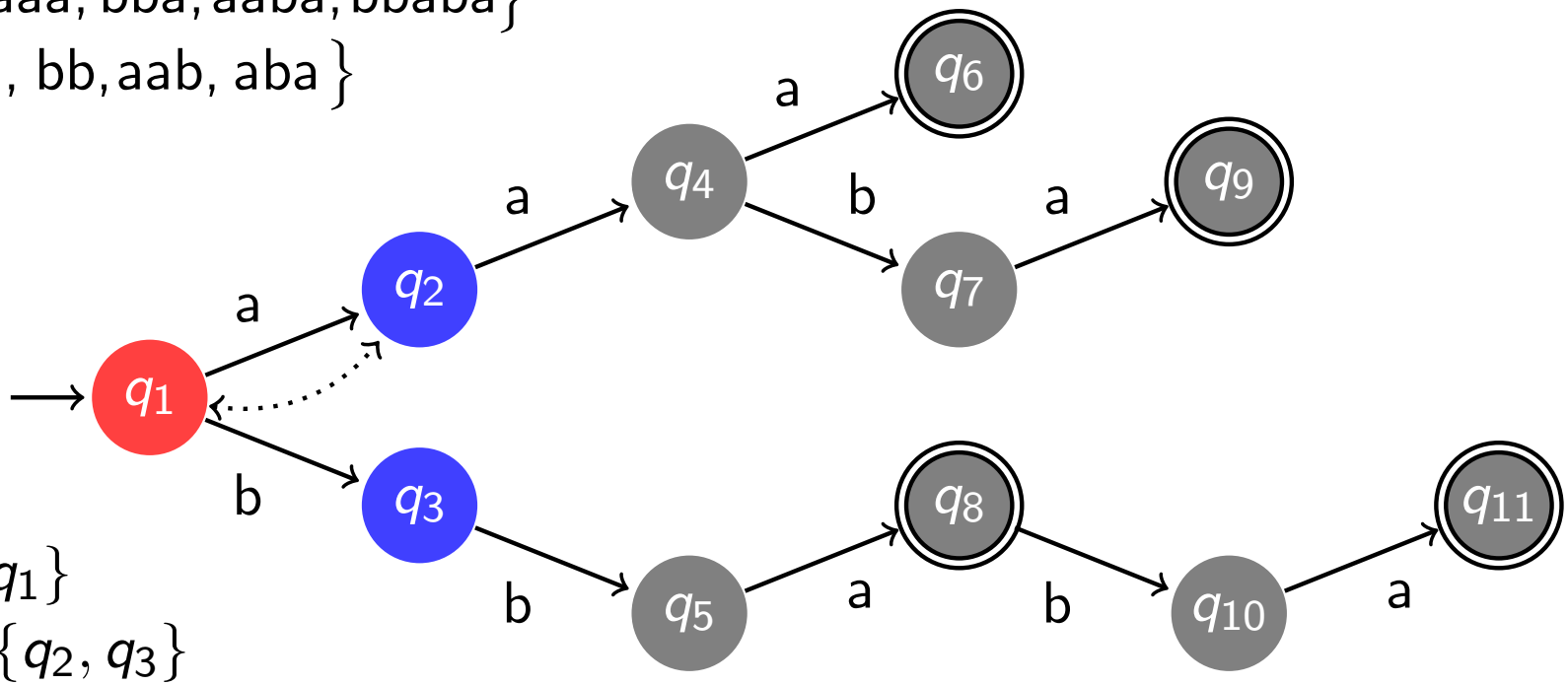
$\text{MERGE}(q_1, q_2)$

$\text{FOLD}(q_1, q_2)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1\}$

$\text{BLUE} = \{q_2, q_3\}$

q_2 :

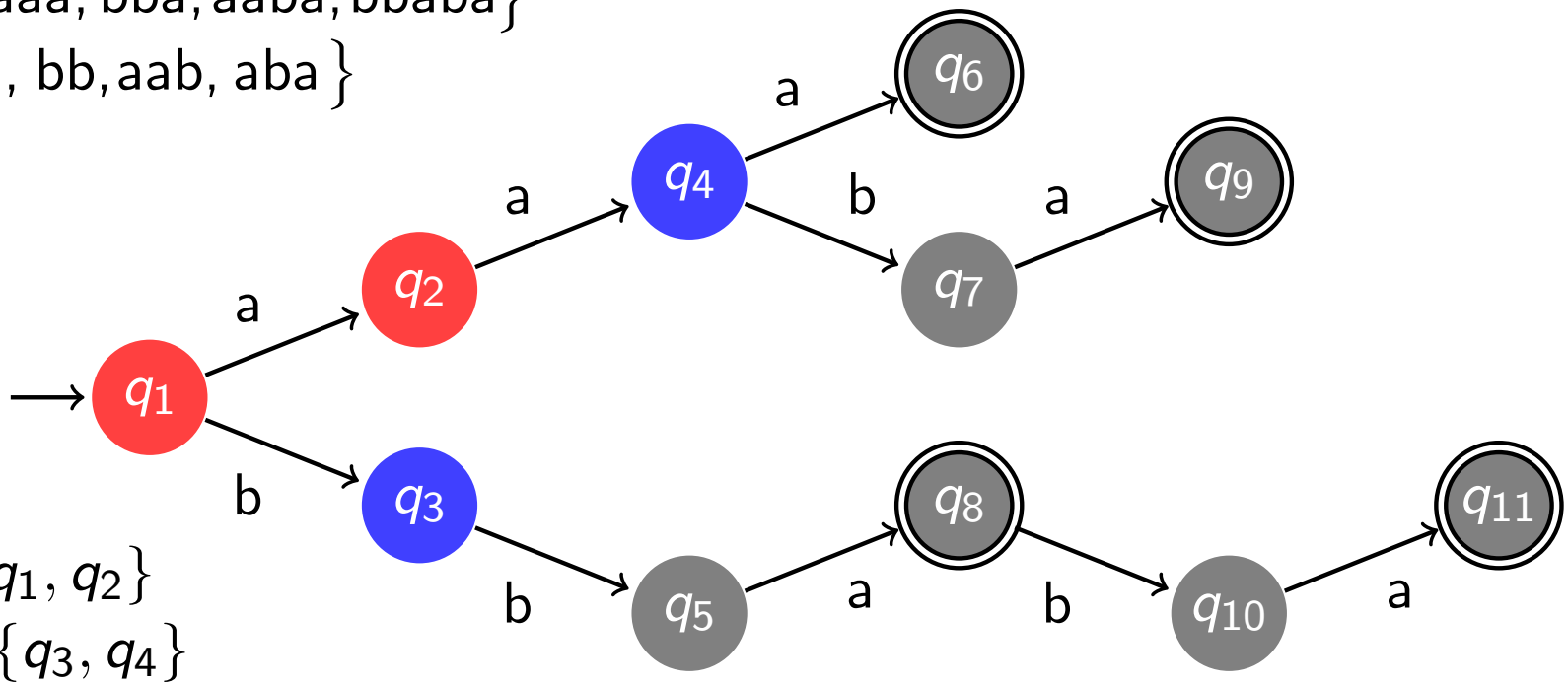
~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2\}$

$\text{BLUE} = \{q_3, q_4\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

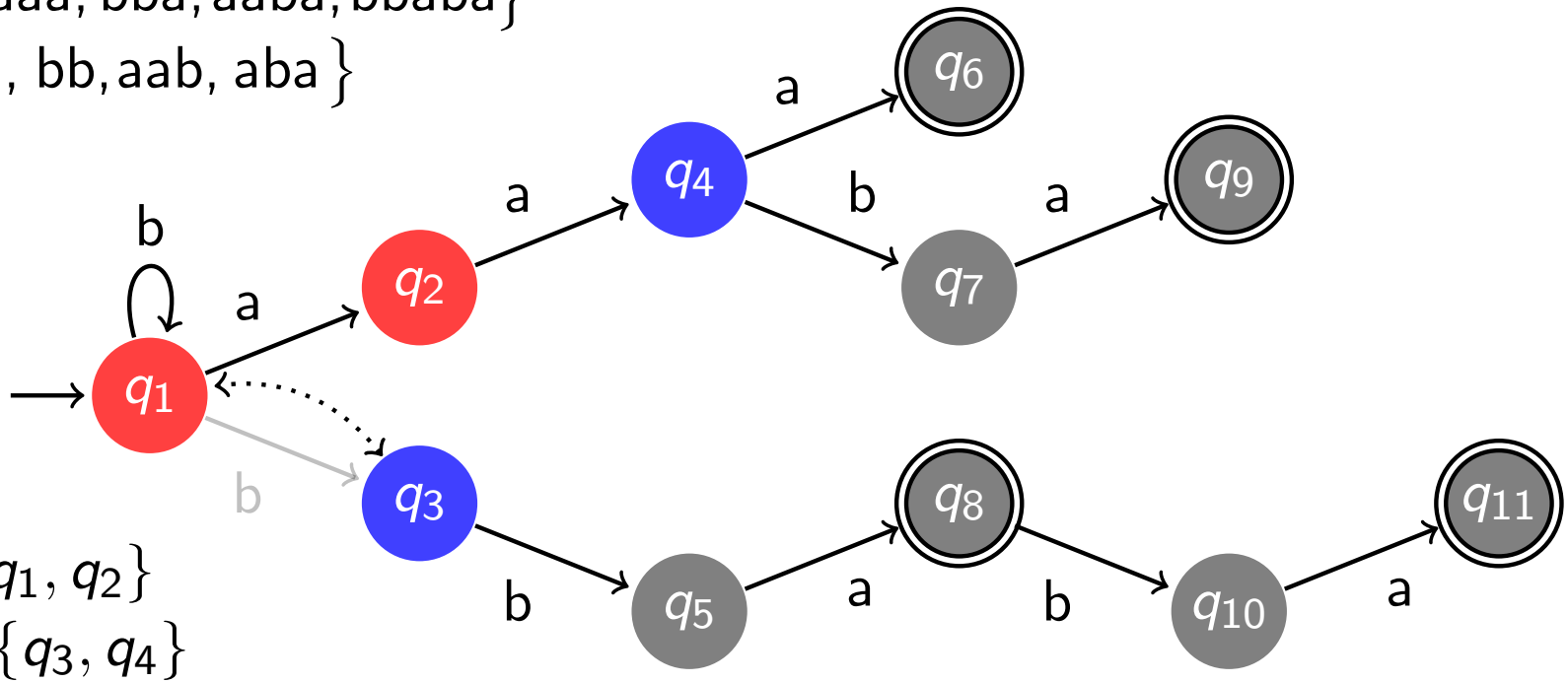
~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2\}$

$\text{BLUE} = \{q_3, q_4\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

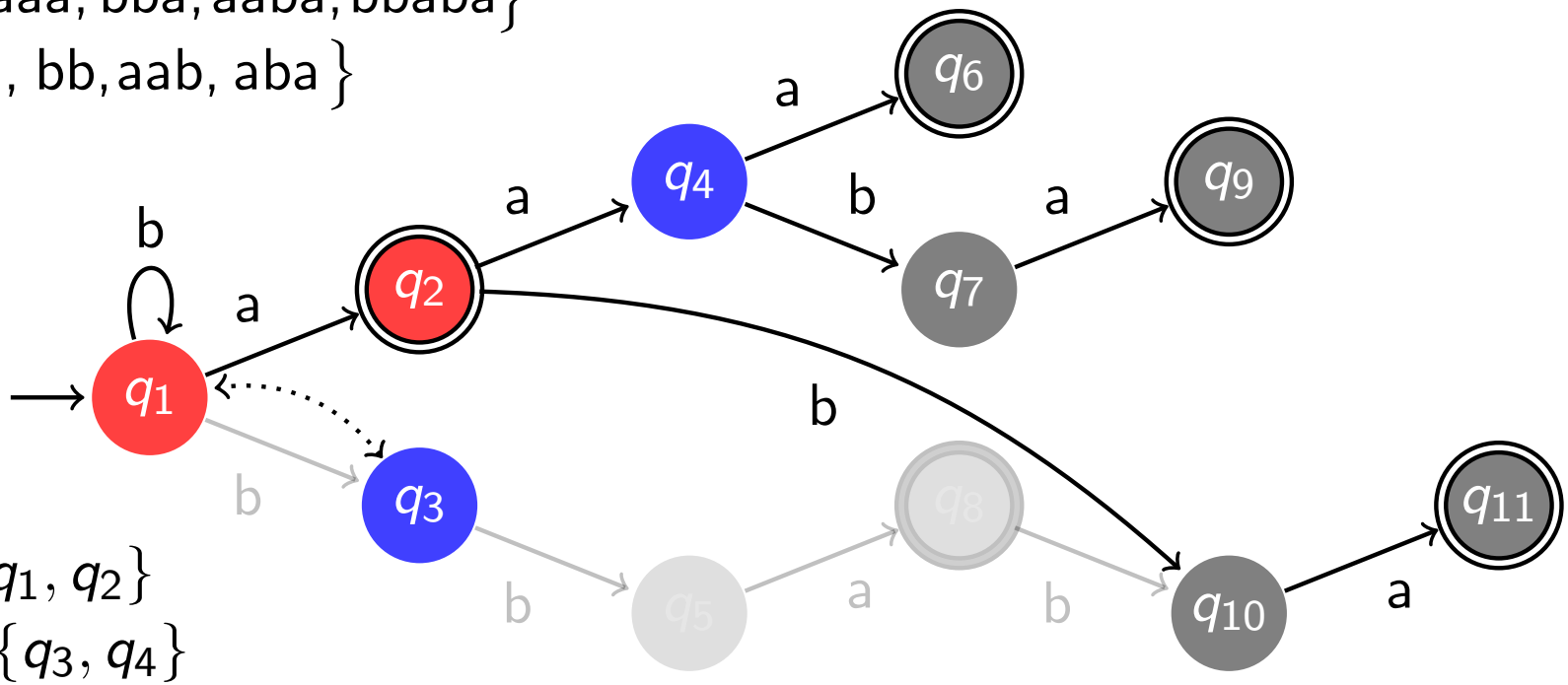
q_3 :

$\text{MERGE}(q_1, q_3)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2\}$

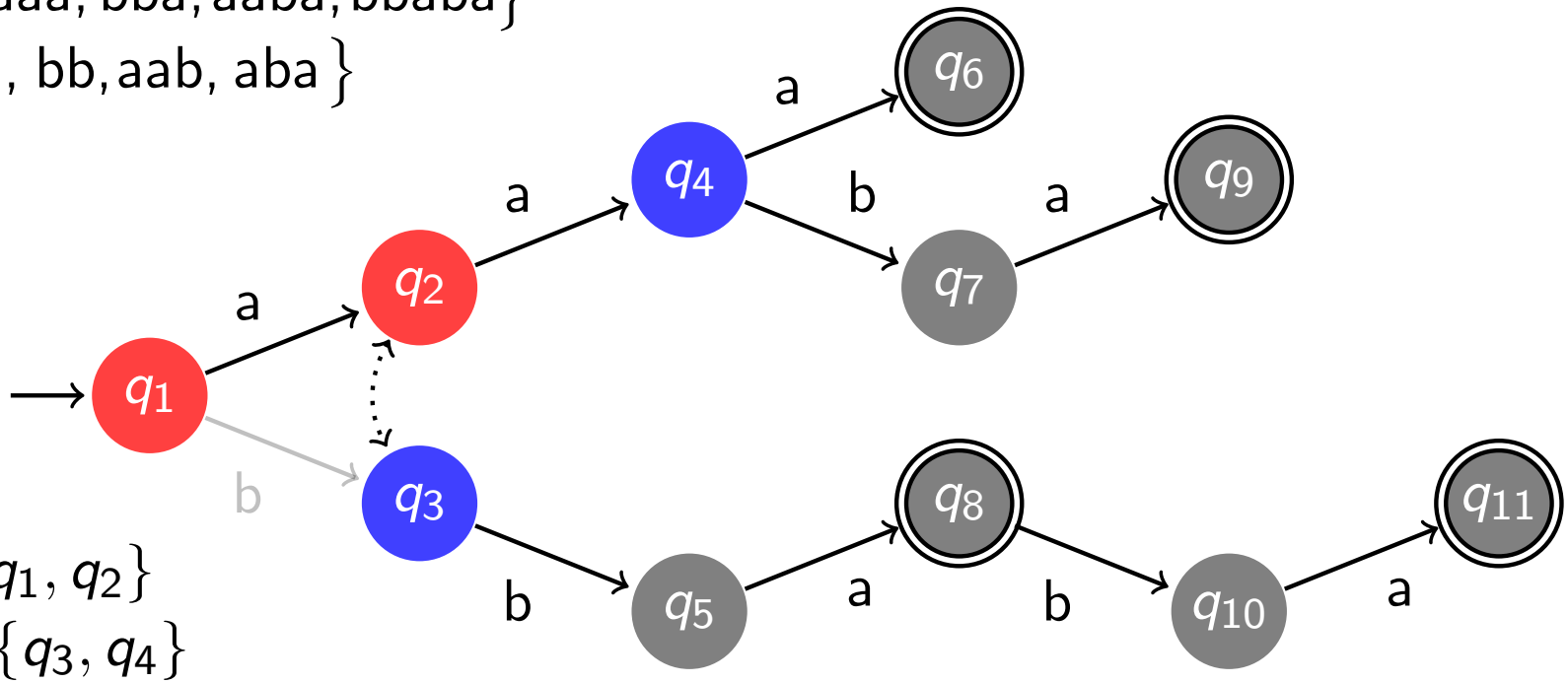
$\text{BLUE} = \{q_3, q_4\}$

q_2 :	q_3 :
~~$\text{MERGE}(q_1, q_2)$~~	$\text{MERGE}(q_1, q_3)$
~~$\text{FOLD}(q_1, q_2)$~~	$\text{FOLD}(q_1, q_3)$
$\text{PROMOTE}(q_2)$	

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2\}$

$\text{BLUE} = \{q_3, q_4\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

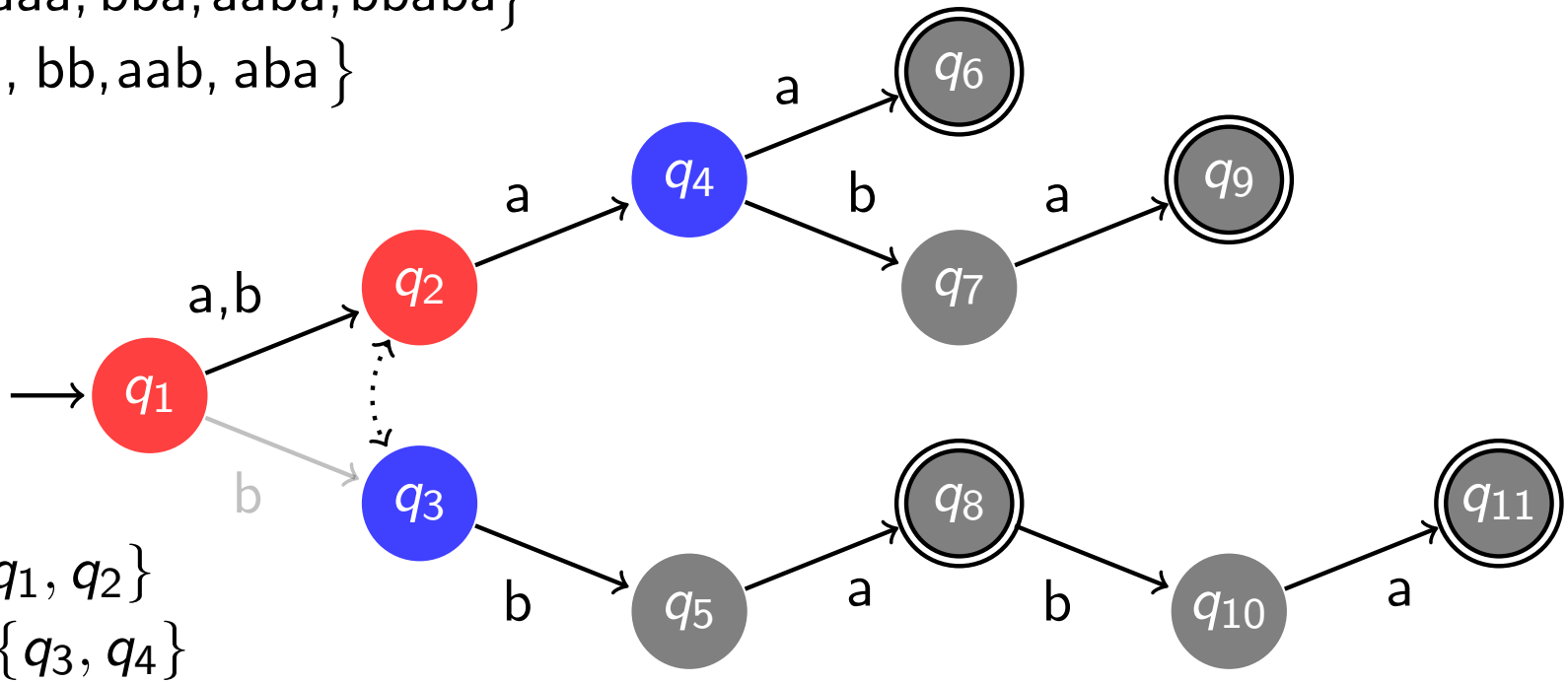
~~$\text{FOLD}(q_1, q_3)$~~

$\text{MERGE}(q_2, q_3)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2\}$

$\text{BLUE} = \{q_3, q_4\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

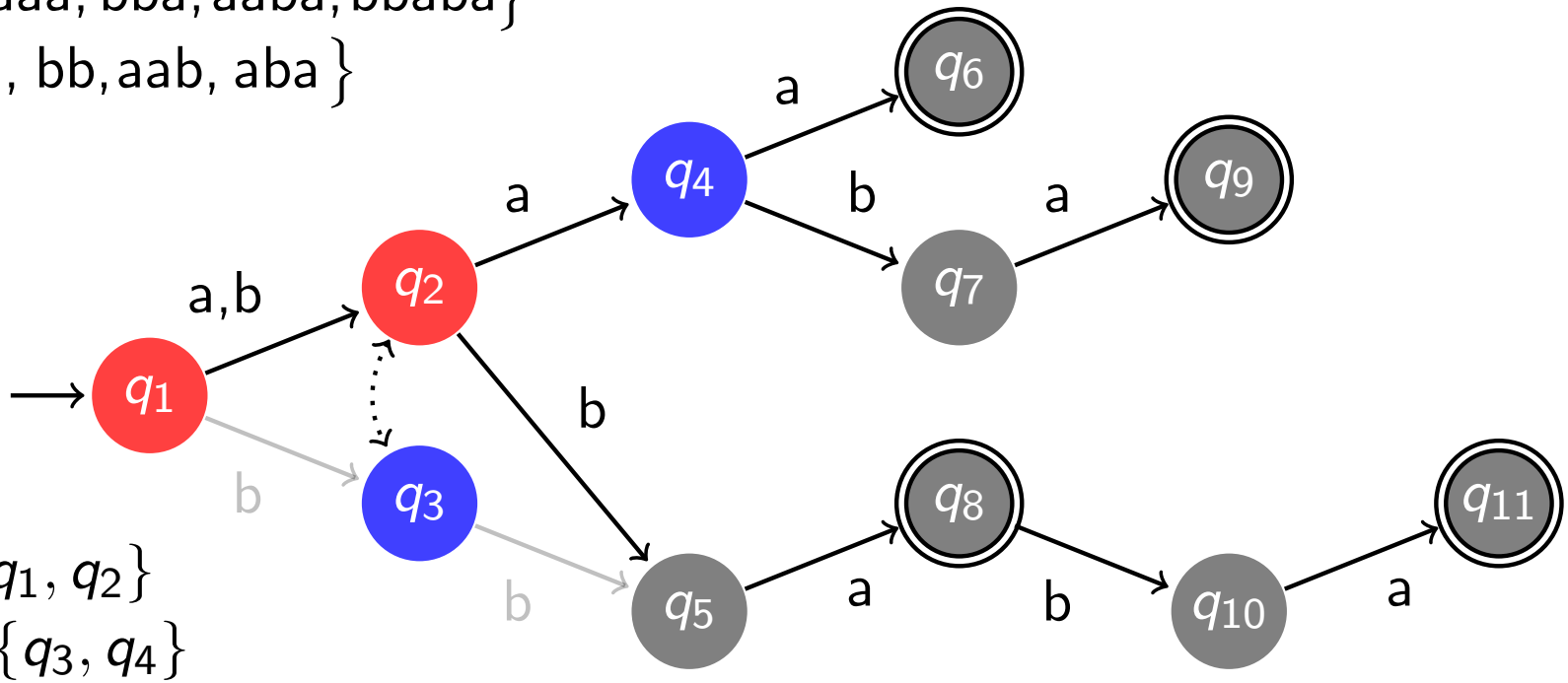
~~$\text{FOLD}(q_1, q_3)$~~

$\text{MERGE}(q_2, q_3)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2\}$

$\text{BLUE} = \{q_3, q_4\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

~~$\text{FOLD}(q_1, q_3)$~~

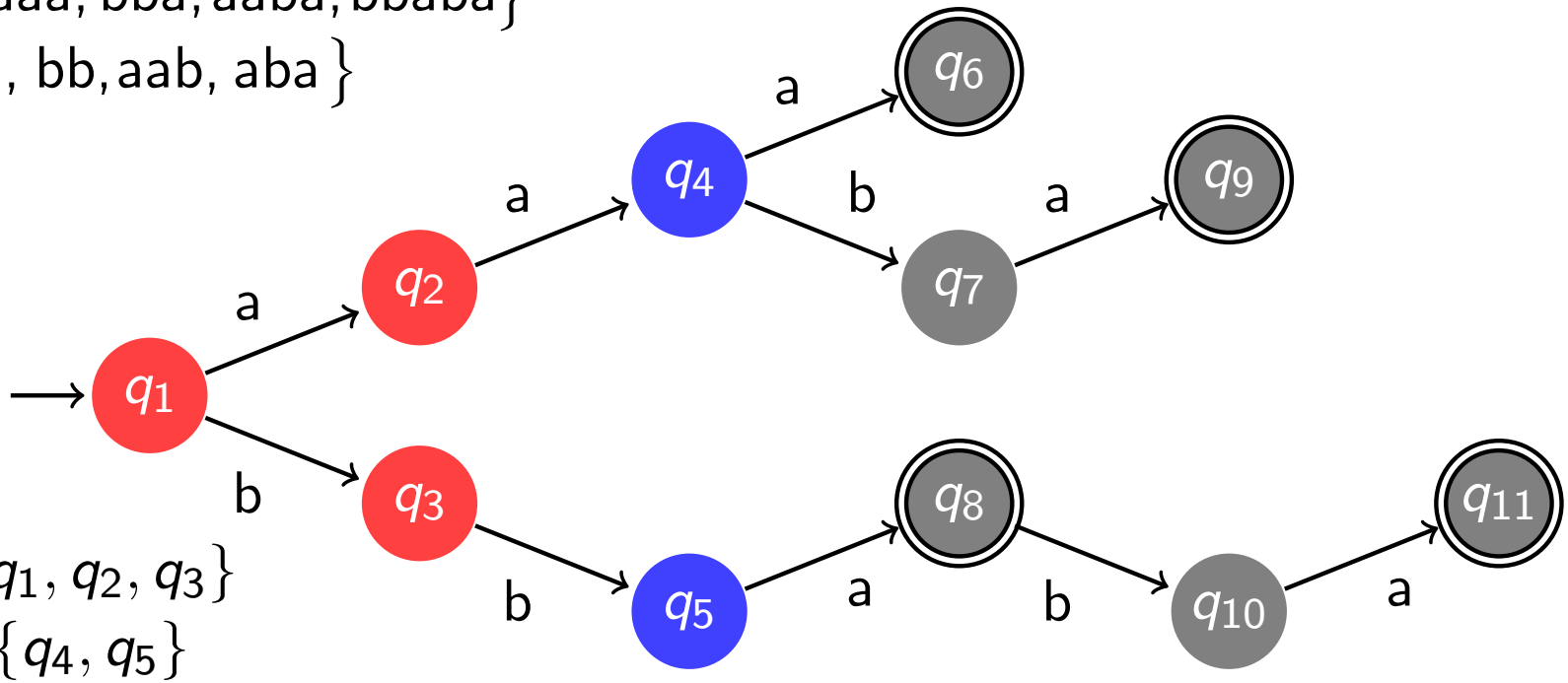
$\text{MERGE}(q_2, q_3)$

$\text{FOLD}(q_2, q_3)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_4, q_5\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

~~$\text{FOLD}(q_1, q_3)$~~

~~$\text{MERGE}(q_2, q_3)$~~

~~$\text{FOLD}(q_2, q_3)$~~

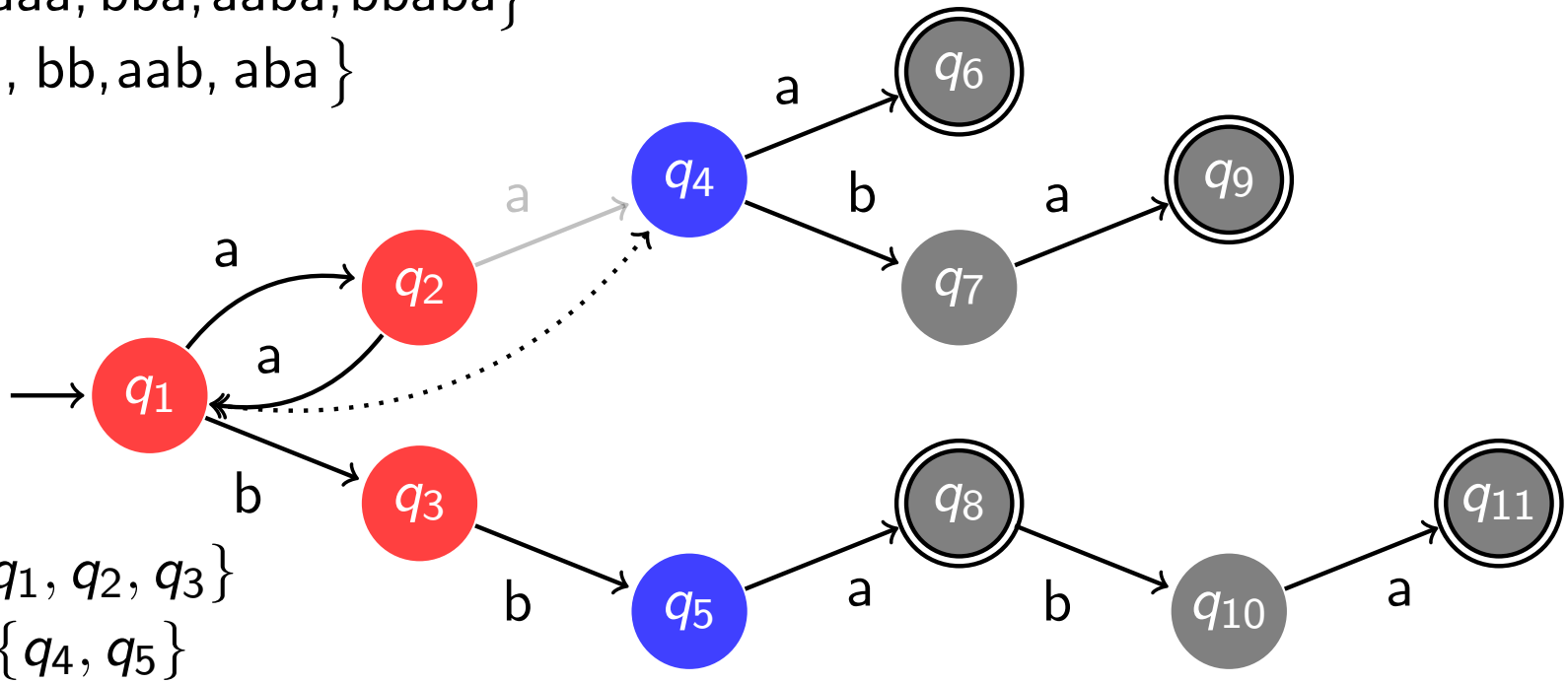
$\text{PROMOTE}(q_3)$

q_4 :

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_4, q_5\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

~~$\text{FOLD}(q_1, q_3)$~~

~~$\text{MERGE}(q_2, q_3)$~~

~~$\text{FOLD}(q_2, q_3)$~~

$\text{PROMOTE}(q_3)$

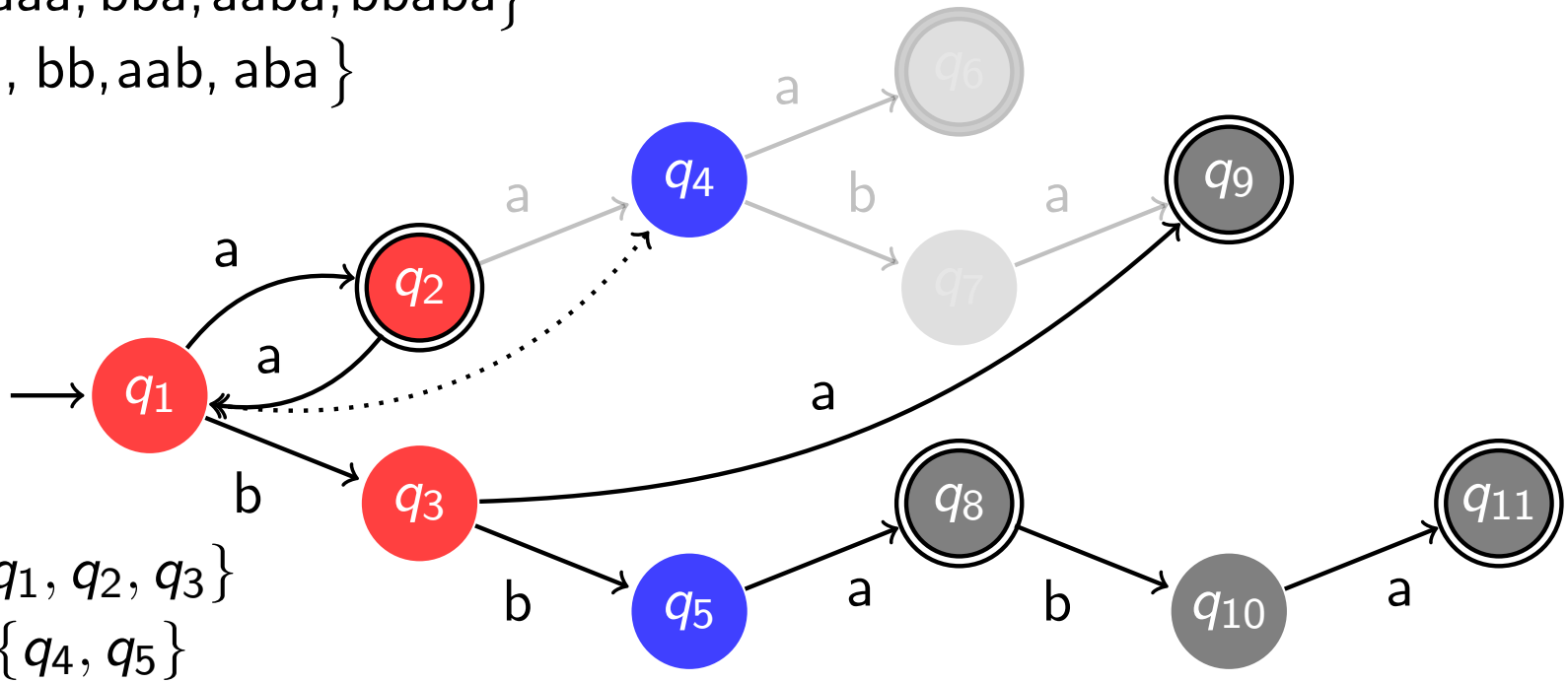
q_4 :

$\text{MERGE}(q_1, q_4)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_4, q_5\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

~~$\text{FOLD}(q_1, q_3)$~~

~~$\text{MERGE}(q_2, q_3)$~~

~~$\text{FOLD}(q_2, q_3)$~~

$\text{PROMOTE}(q_3)$

q_4 :

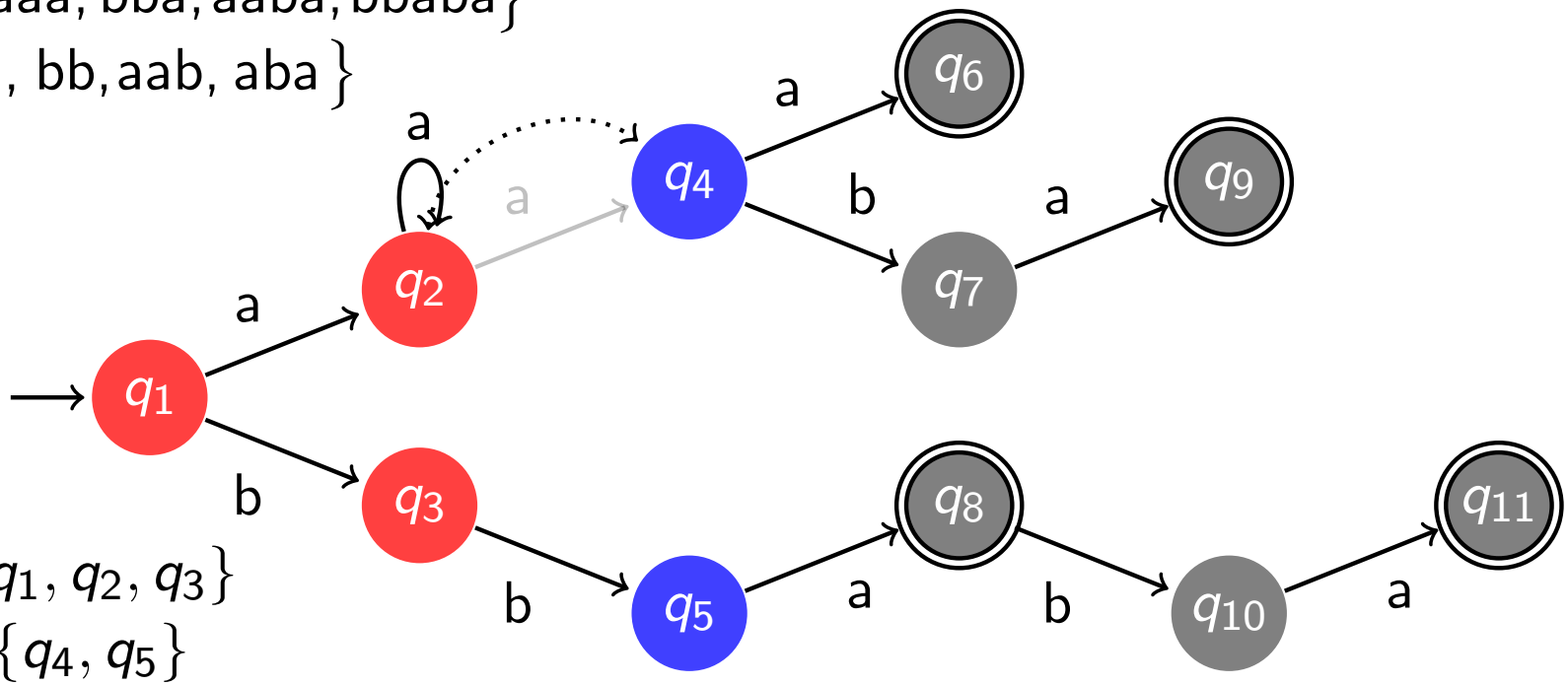
$\text{MERGE}(q_1, q_4)$

$\text{FOLD}(q_1, q_4)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_4, q_5\}$

q_2 :

~~MERGE(q_1, q_2)~~

~~FOLD(q_1, q_2)~~

PROMOTE(q_2)

q_3 :

~~MERGE(q_1, q_3)~~

~~FOLD(q_1, q_3)~~

~~MERGE(q_2, q_3)~~

~~FOLD(q_2, q_3)~~

PROMOTE(q_3)

q_4 :

~~MERGE(q_1, q_4)~~

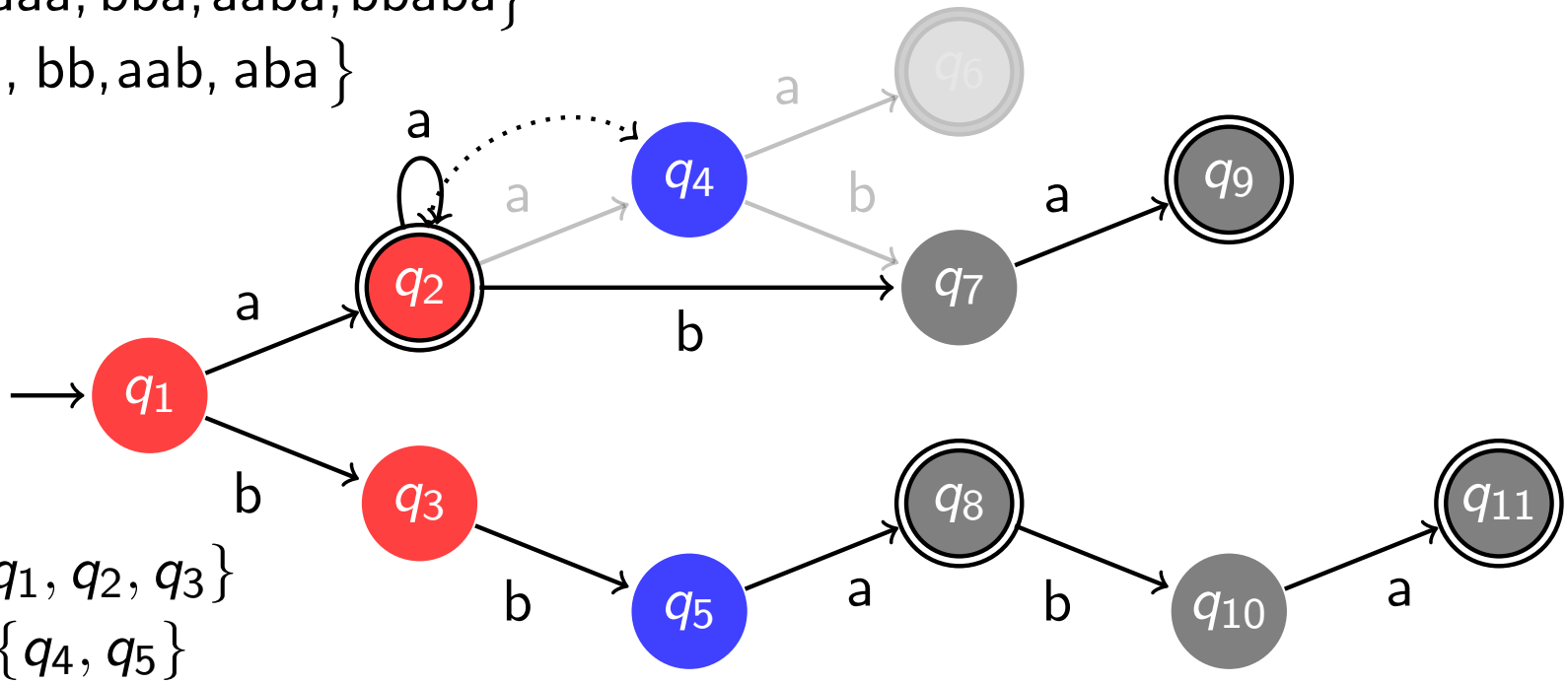
~~FOLD(q_1, q_4)~~

MERGE(q_2, q_4)

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_4, q_5\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

~~$\text{FOLD}(q_1, q_3)$~~

~~$\text{MERGE}(q_2, q_3)$~~

~~$\text{FOLD}(q_2, q_3)$~~

$\text{PROMOTE}(q_3)$

q_4 :

~~$\text{MERGE}(q_1, q_4)$~~

~~$\text{FOLD}(q_1, q_4)$~~

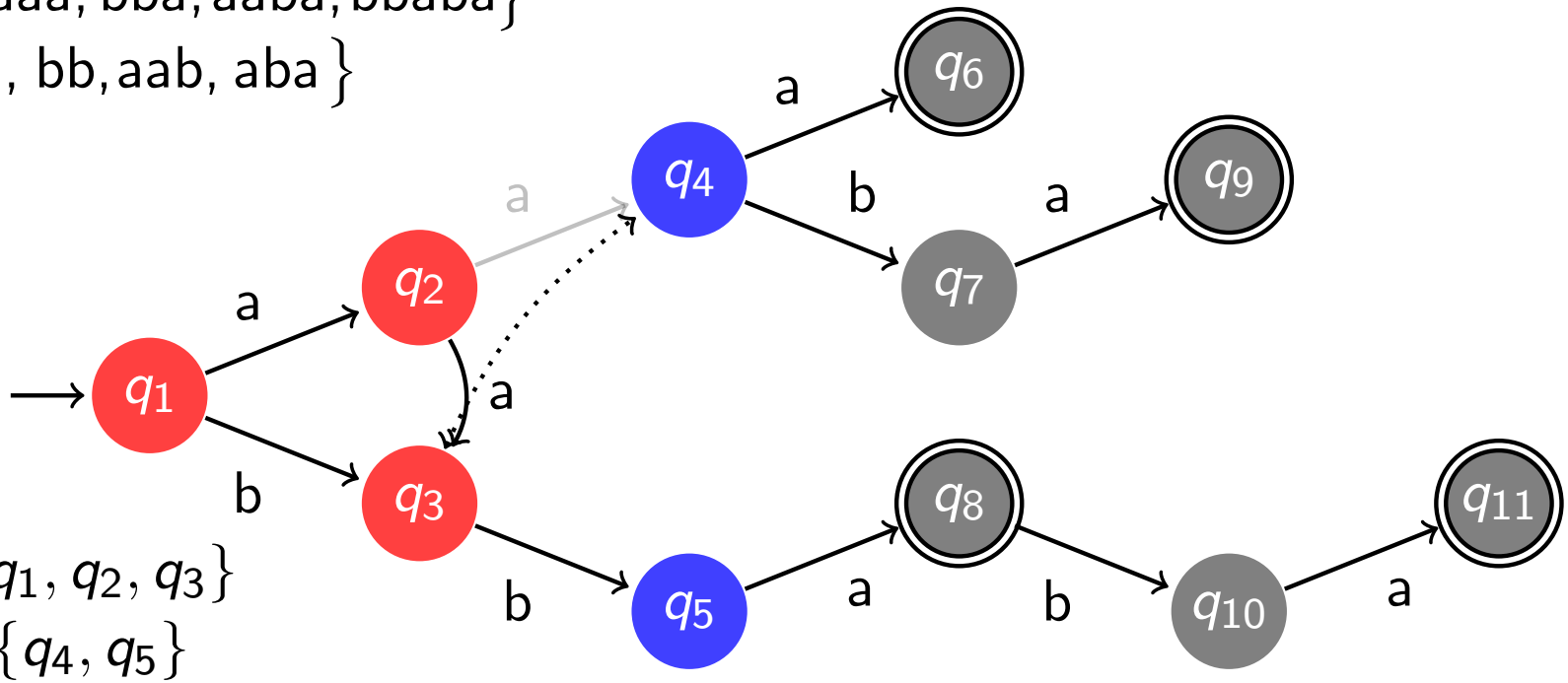
$\text{MERGE}(q_2, q_4)$

$\text{FOLD}(q_2, q_4)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_4, q_5\}$

q_2 :

~~MERGE(q_1, q_2)~~

~~FOLD(q_1, q_2)~~

PROMOTE(q_2)

q_3 :

~~MERGE(q_1, q_3)~~

~~FOLD(q_1, q_3)~~

~~MERGE(q_2, q_3)~~

~~FOLD(q_2, q_3)~~

PROMOTE(q_3)

q_4 :

~~MERGE(q_1, q_4)~~

~~FOLD(q_1, q_4)~~

~~MERGE(q_2, q_4)~~

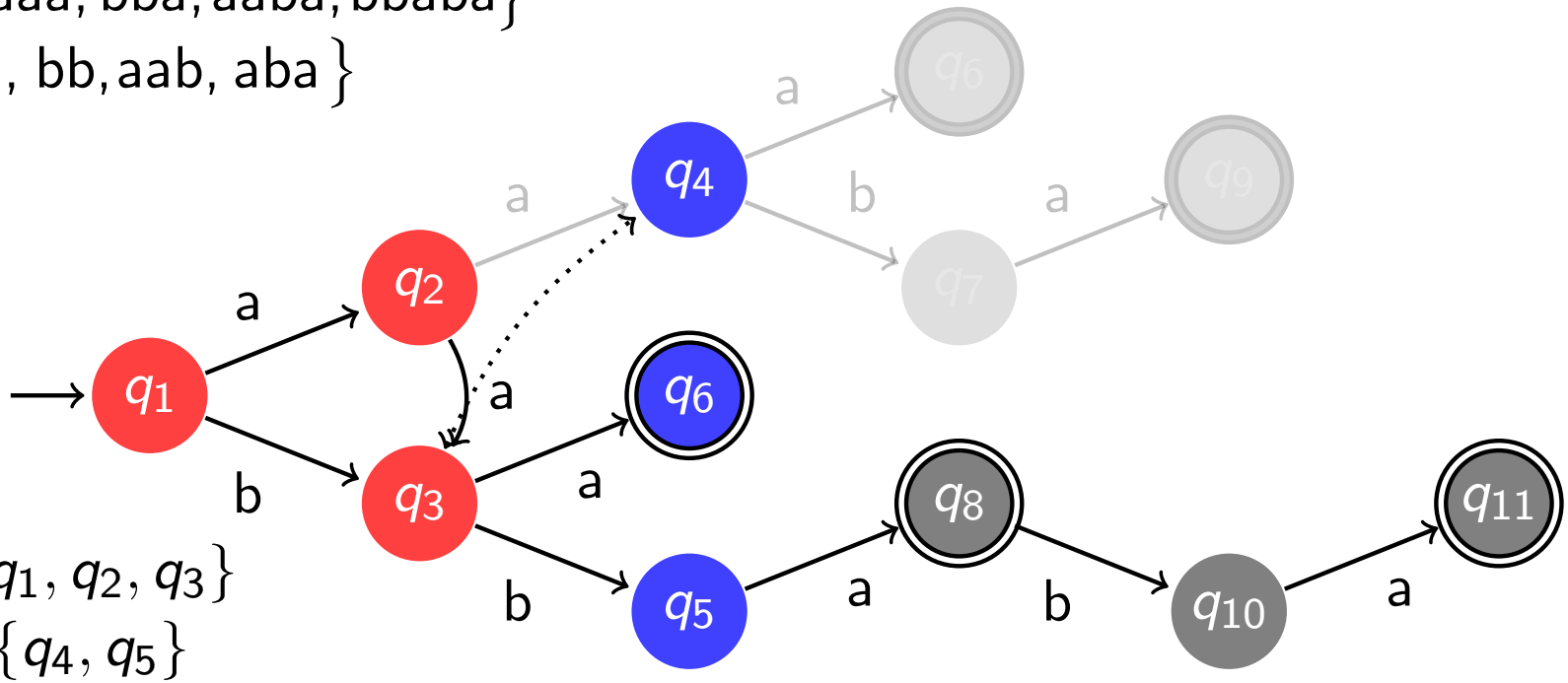
~~FOLD(q_2, q_4)~~

MERGE(q_3, q_4)

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_4, q_5\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

~~$\text{FOLD}(q_1, q_3)$~~

~~$\text{MERGE}(q_2, q_3)$~~

~~$\text{FOLD}(q_2, q_3)$~~

$\text{PROMOTE}(q_3)$

q_4 :

~~$\text{MERGE}(q_1, q_4)$~~

~~$\text{FOLD}(q_1, q_4)$~~

~~$\text{MERGE}(q_2, q_4)$~~

~~$\text{FOLD}(q_2, q_4)$~~

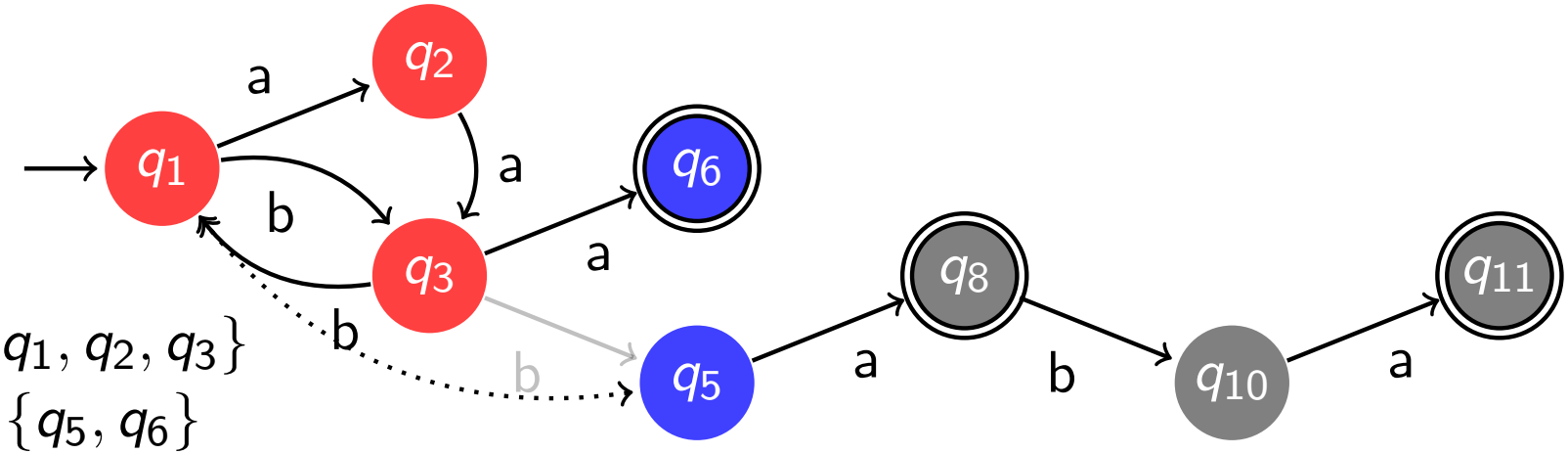
$\text{MERGE}(q_3, q_4)$

$\text{FOLD}(q_3, q_4)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_5, q_6\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

~~$\text{FOLD}(q_1, q_3)$~~

~~$\text{MERGE}(q_2, q_3)$~~

~~$\text{FOLD}(q_2, q_3)$~~

$\text{PROMOTE}(q_3)$

q_4 :

~~$\text{MERGE}(q_1, q_4)$~~

~~$\text{FOLD}(q_1, q_4)$~~

~~$\text{MERGE}(q_2, q_4)$~~

~~$\text{FOLD}(q_2, q_4)$~~

$\text{MERGE}(q_3, q_4)$

$\text{FOLD}(q_3, q_4)$

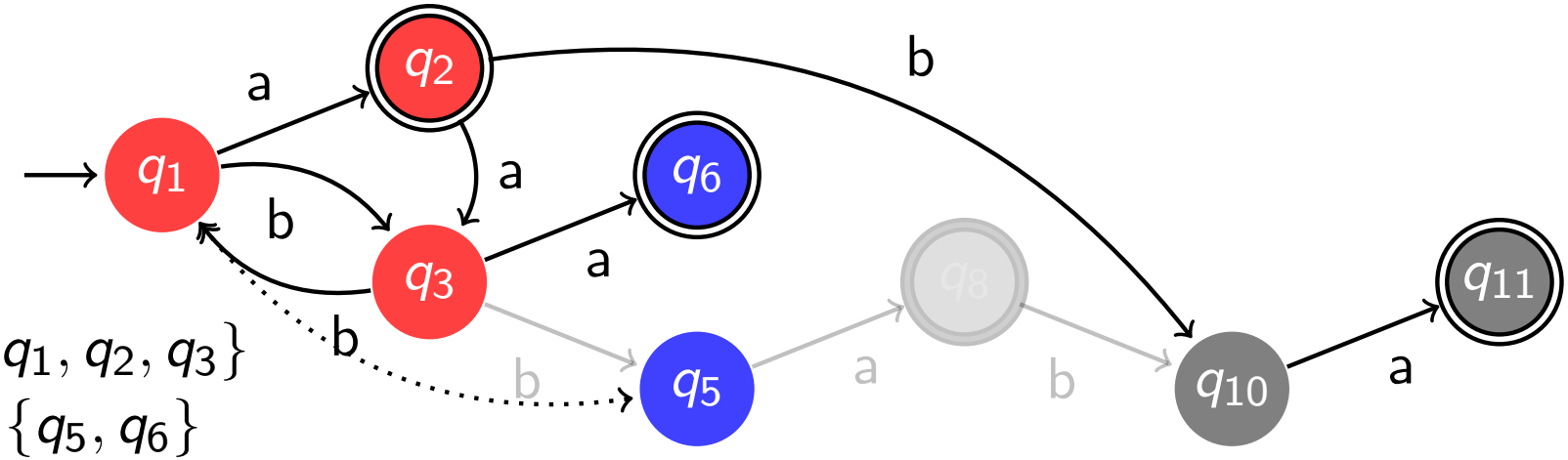
q_5 :

$\text{MERGE}(q_1, q_5)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_5, q_6\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

~~$\text{FOLD}(q_1, q_3)$~~

~~$\text{MERGE}(q_2, q_3)$~~

~~$\text{FOLD}(q_2, q_3)$~~

$\text{PROMOTE}(q_3)$

q_4 :

~~$\text{MERGE}(q_1, q_4)$~~

~~$\text{FOLD}(q_1, q_4)$~~

~~$\text{MERGE}(q_2, q_4)$~~

~~$\text{FOLD}(q_2, q_4)$~~

$\text{MERGE}(q_3, q_4)$

$\text{FOLD}(q_3, q_4)$

q_5 :

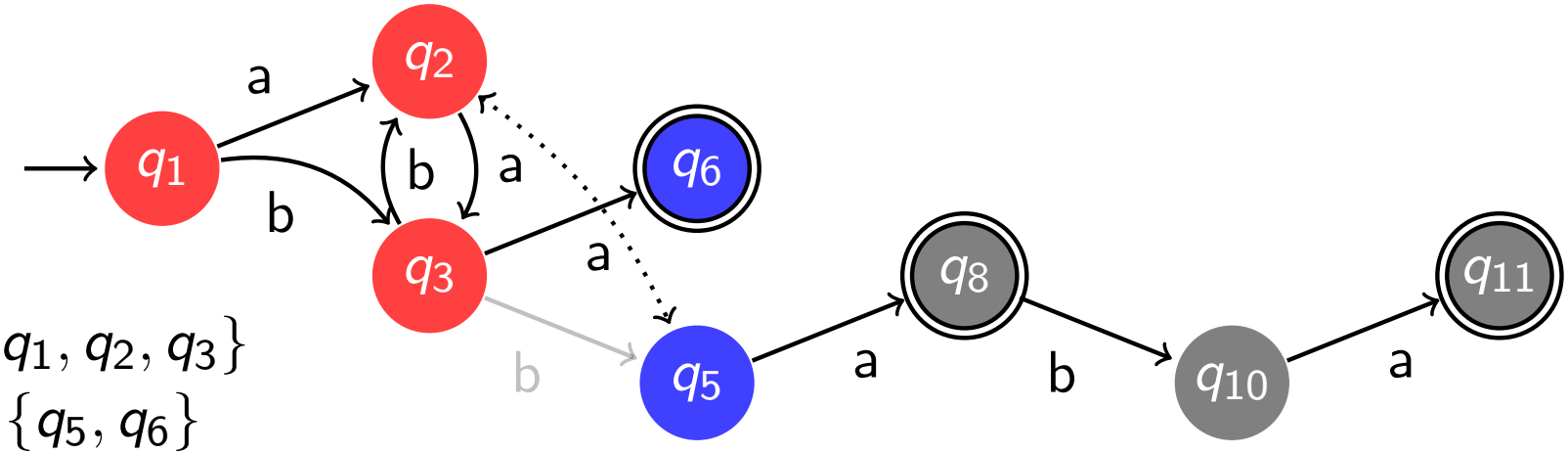
$\text{MERGE}(q_1, q_5)$

$\text{FOLD}(q_1, q_5)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_5, q_6\}$

q_2 :

~~MERGE(q_1, q_2)~~

~~FOLD(q_1, q_2)~~

PROMOTE(q_2)

q_3 :

~~MERGE(q_1, q_3)~~

~~FOLD(q_1, q_3)~~

~~MERGE(q_2, q_3)~~

~~FOLD(q_2, q_3)~~

PROMOTE(q_3)

q_4 :

~~MERGE(q_1, q_4)~~

~~FOLD(q_1, q_4)~~

~~MERGE(q_2, q_4)~~

~~FOLD(q_2, q_4)~~

MERGE(q_3, q_4)

FOLD(q_3, q_4)

q_5 :

~~MERGE(q_1, q_5)~~

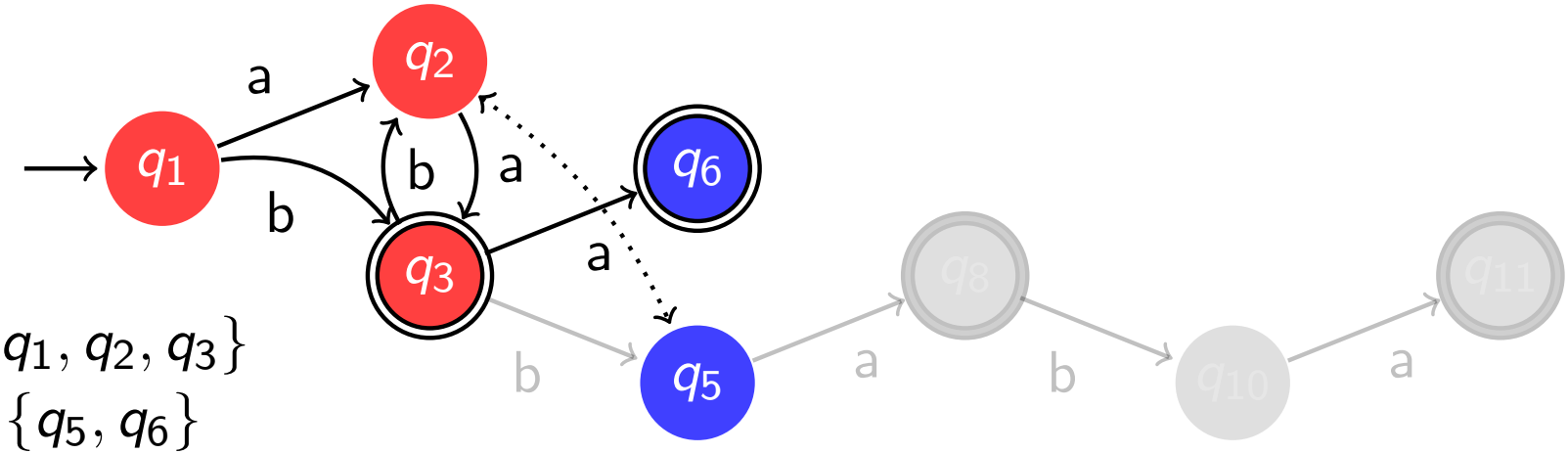
~~FOLD(q_1, q_5)~~

MERGE(q_2, q_5)

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



RED $= \{q_1, q_2, q_3\}$

BLUE $= \{q_5, q_6\}$

q_2 :

~~MERGE(q_1, q_2)~~

~~FOLD(q_1, q_2)~~

PROMOTE(q_2)

q_3 :

~~MERGE(q_1, q_3)~~

~~FOLD(q_1, q_3)~~

~~MERGE(q_2, q_3)~~

~~FOLD(q_2, q_3)~~

PROMOTE(q_3)

q_4 :

~~MERGE(q_1, q_4)~~

~~FOLD(q_1, q_4)~~

~~MERGE(q_2, q_4)~~

~~FOLD(q_2, q_4)~~

MERGE(q_3, q_4)

FOLD(q_3, q_4)

q_5 :

~~MERGE(q_1, q_5)~~

~~FOLD(q_1, q_5)~~

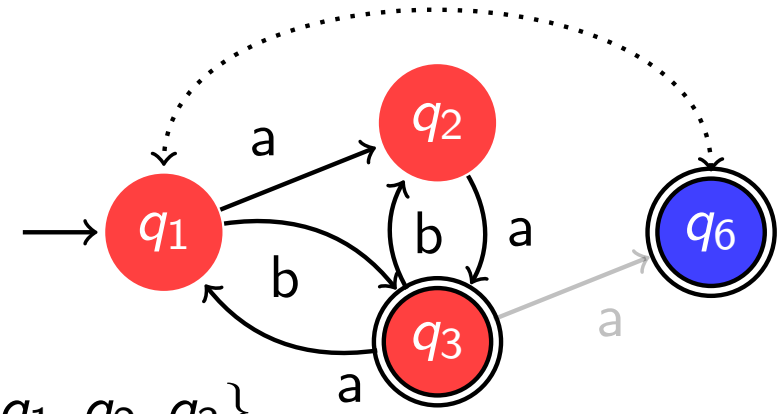
MERGE(q_2, q_5)

FOLD(q_2, q_5)

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_6\}$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

~~$\text{FOLD}(q_1, q_3)$~~

~~$\text{MERGE}(q_2, q_3)$~~

~~$\text{FOLD}(q_2, q_3)$~~

$\text{PROMOTE}(q_3)$

q_4 :

~~$\text{MERGE}(q_1, q_4)$~~

~~$\text{FOLD}(q_1, q_4)$~~

~~$\text{MERGE}(q_2, q_4)$~~

~~$\text{FOLD}(q_2, q_4)$~~

$\text{MERGE}(q_3, q_4)$

$\text{FOLD}(q_3, q_4)$

q_5 :

~~$\text{MERGE}(q_1, q_5)$~~

~~$\text{FOLD}(q_1, q_5)$~~

$\text{MERGE}(q_2, q_5)$

$\text{FOLD}(q_2, q_5)$

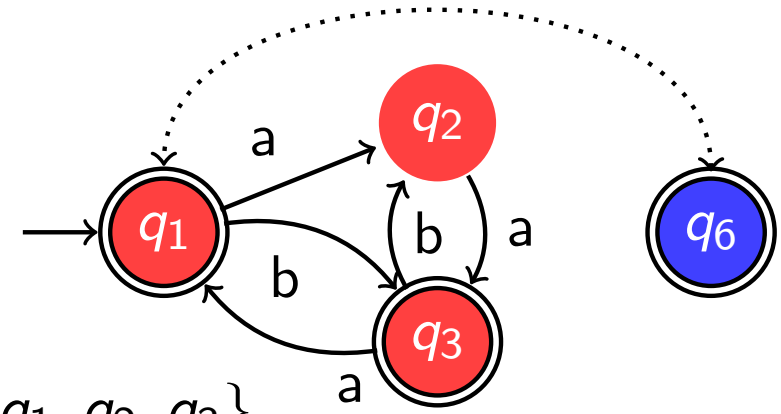
q_6 :

$\text{MERGE}(q_1, q_6)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \{q_6\}$

q_2 :

~~MERGE(q_1, q_2)~~

~~FOLD(q_1, q_2)~~

PROMOTE(q_2)

q_3 :

~~MERGE(q_1, q_3)~~

~~FOLD(q_1, q_3)~~

~~MERGE(q_2, q_3)~~

~~FOLD(q_2, q_3)~~

PROMOTE(q_3)

q_4 :

~~MERGE(q_1, q_4)~~

~~FOLD(q_1, q_4)~~

~~MERGE(q_2, q_4)~~

~~FOLD(q_2, q_4)~~

MERGE(q_3, q_4)

FOLD(q_3, q_4)

q_5 :

~~MERGE(q_1, q_5)~~

~~FOLD(q_1, q_5)~~

MERGE(q_2, q_5)

FOLD(q_2, q_5)

q_6 :

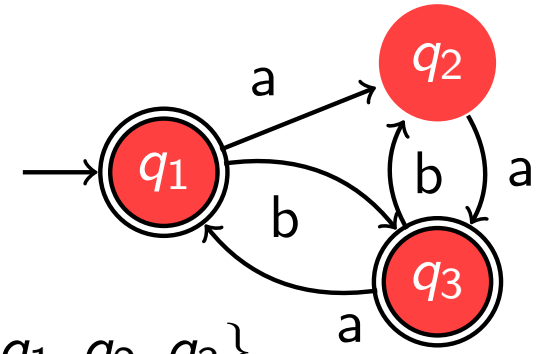
MERGE(q_1, q_6)

FOLD(q_1, q_6)

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \emptyset$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

~~$\text{FOLD}(q_1, q_3)$~~

~~$\text{MERGE}(q_2, q_3)$~~

~~$\text{FOLD}(q_2, q_3)$~~

$\text{PROMOTE}(q_3)$

q_4 :

~~$\text{MERGE}(q_1, q_4)$~~

~~$\text{FOLD}(q_1, q_4)$~~

~~$\text{MERGE}(q_2, q_4)$~~

~~$\text{FOLD}(q_2, q_4)$~~

$\text{MERGE}(q_3, q_4)$

$\text{FOLD}(q_3, q_4)$

q_5 :

~~$\text{MERGE}(q_1, q_5)$~~

~~$\text{FOLD}(q_1, q_5)$~~

$\text{MERGE}(q_2, q_5)$

$\text{FOLD}(q_2, q_5)$

q_6 :

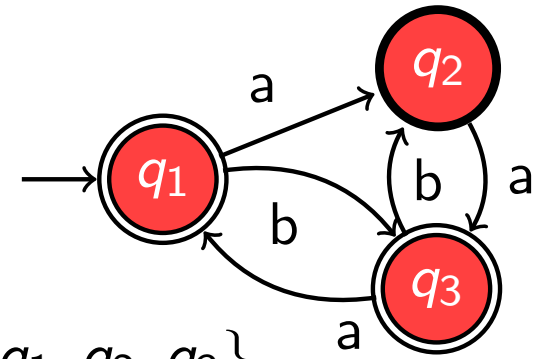
$\text{MERGE}(q_1, q_6)$

$\text{FOLD}(q_1, q_6)$

RPNI example

$S_+ = \{\text{aaa}, \text{bba}, \text{aaba}, \text{bbaba}\}$

$S_- = \{\text{a}, \text{bb}, \text{aab}, \text{aba}\}$



$\text{RED} = \{q_1, q_2, q_3\}$

$\text{BLUE} = \emptyset$

q_2 :

~~$\text{MERGE}(q_1, q_2)$~~

~~$\text{FOLD}(q_1, q_2)$~~

$\text{PROMOTE}(q_2)$

q_3 :

~~$\text{MERGE}(q_1, q_3)$~~

~~$\text{FOLD}(q_1, q_3)$~~

~~$\text{MERGE}(q_2, q_3)$~~

~~$\text{FOLD}(q_2, q_3)$~~

$\text{PROMOTE}(q_3)$

q_4 :

~~$\text{MERGE}(q_1, q_4)$~~

~~$\text{FOLD}(q_1, q_4)$~~

~~$\text{MERGE}(q_2, q_4)$~~

~~$\text{FOLD}(q_2, q_4)$~~

$\text{MERGE}(q_3, q_4)$

$\text{FOLD}(q_3, q_4)$

q_5 :

~~$\text{MERGE}(q_1, q_5)$~~

~~$\text{FOLD}(q_1, q_5)$~~

$\text{MERGE}(q_2, q_5)$

$\text{FOLD}(q_2, q_5)$

q_6 :

$\text{MERGE}(q_1, q_6)$

$\text{FOLD}(q_1, q_6)$

Closing remarks

- This simple version will not scale
- Various data structures proposed to store the intermediate automata

- Note that RPNI is not (can not be?) incremental

- worst-case: $\mathcal{O}\left((|S_+| + |S_-|) \cdot |S_+|^2\right)$
  - "The actual computation which is proposed is not convincing, and empirically, those that have consistently used the algorithm certainly do not report a cubic behaviour."