cse303
ELEMENTS OF THE THEORY OF COMPUTATION

Professor Anita Wasilewska

LECTURE 7

CHAPTER 2
FINITE AUTOMATA

3. Finite Automata and Regular Expressions
4. Languages that are Not Regular
5. State Minimization
6. Algorithmic Aspects of Finite Automata

CHAPTER 2
PART 3: Finite Automata and Regular Expressions

Finite Automata and Regular Expressions

The goal of this part of chapter 2 is to prove a theorem that establishes a **relationship** between Finite Automata and Regular languages, i.e to prove that following

MAIN THEOREM

A language L is regular if and only if it is accepted by a finite automaton, i.e.

A language L is regular if and only if there is a finite automaton M, such that

$$L = L(M)$$

Closure Theorem

To achieve our goal we first prove the following

CLOSURE THEOREM

The class of languages accepted by Finite Automata (FA) is **closed** under the following operations

1. union
2. concatenation
3. Kleene's Star
4. complementation
5. intersection

Observe that we used the term Finite Automata (FA) so in the proof we have can choose a DFA or NDFA, as we have already proved their EQUIVALENCY

Closure Theorem

Remember that languages are sets, so we have the set operations $\cup$, $\cap$, $-$, defined for any $L_1, L_2 \subseteq \Sigma^*$, i.e the languages

$$L = L_1 \cup L_2, \quad L = L_1 \cap L_2, \quad L = \Sigma^* - L_1$$

We also defined the languages specific operations of concatenation and Kleene's Star , i.e. the languages

$$L = L_1 \circ L_2 \quad \text{and} \quad L = {L_1}^*$$

Closure Under Union

1. The class of languages accepted by Finite Automata (FA) is **closed** under union

Proof

Let $M_1,\ M_2$ be two NDFA finite automata

We **construct** a NDF automaton M, such that

$$L(M) = L(M_1) \cup L(M_2)$$

Let $M_1 = (K_1,\ \Sigma,\ \Delta_1,\ s_1,\ F_1)$ and

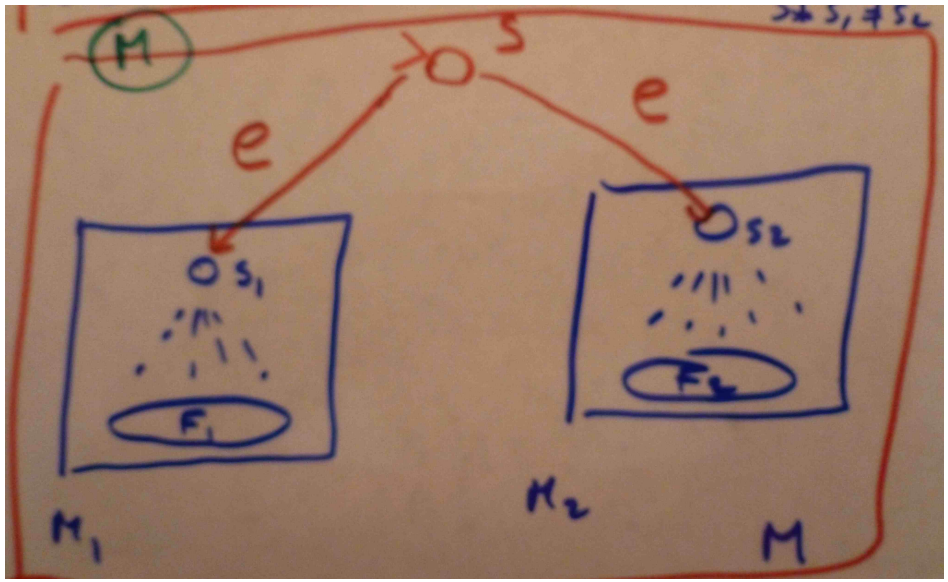
$M_2 = (K_2,\ \Sigma,\ \Delta_2,\ s_2,\ F_2)$

Where (we rename the states, if needed)

$\Sigma = \Sigma_1 \cup \Sigma_2, \quad s_1 \neq s_2, \quad K_1 \cap K_2 = \emptyset \quad F_1 \cap F_2 = \emptyset$

Closure Under Union

We **picture** M, such that $L(M) = L(M_1) \cup L(M_2)$ as follows



M goes nondeterministically to M_1 or to M_2 reading nothing so we get

$$w \in L(M) \quad \text{if and only if} \quad w \in M_1 \ \text{ or } \ w \in M_2$$

and hence

$$L(M) = L(M_1) \cup L(M_2)$$

Closure Under Union

We **define formally**

$$M = M_1 \cup M_2 = (K,\ \Sigma,\ \Delta,\ s,\ F)$$

where

$K = K_1 \cup K_2 \cup \{s\}$ for $s \notin K_1 \cup K_2$

s is a NEW state and

$F = F_1 \cup F_2, \quad \Delta = \Delta_1 \cup \Delta_2 \cup \{(s, e, s_1),\ (s, e, s_2)\}$

for s_1 - initial state of M_1 and

s_2 the initial state of M_2

Observe that by Mathematical Induction we construct,

for any $n \geq 2$ an automaton $M = M_1 \cup M_2 \cup \ \ldots\ M_n$ such that

$$L(M) = L(M_1) \cup L(M_2) \cup \ \ldots\ L(M_n)$$

Closure Under Union

Formal proof

Directly from the definition we get

$w \in L(M)$ iff $\exists_q((q \in F = F_1 \cup F_2) \cap ((s,w) \vdash_M{}^*(q,e))$ iff

$\exists_q(((q \in F_1) \cup (q \in F_2)) \cap ((s,w) \vdash_M{}^*(q,e))$ iff

$\exists_q((q \in F_1) \cap ((s,w) \vdash_M{}^*(q,e)) \cup$

$\exists_q((q \in F_2) \cap ((s,w) \vdash_M{}^*(q,e)))$ iff

$w \in L(M_1) \ \cup \ w \in L(M_2)$, what proves that

$$L(M) = L(M_1) \cup L(M_2)$$

We used the following Law of Quantifiers

$\exists_x(A(x) \cup B(x)) \ \equiv \ (\exists_x A(x) \cup \exists_x B(x))$

Examples

Example 1

Diagram of M_1 such that $L(M_1) = aba^*$ is

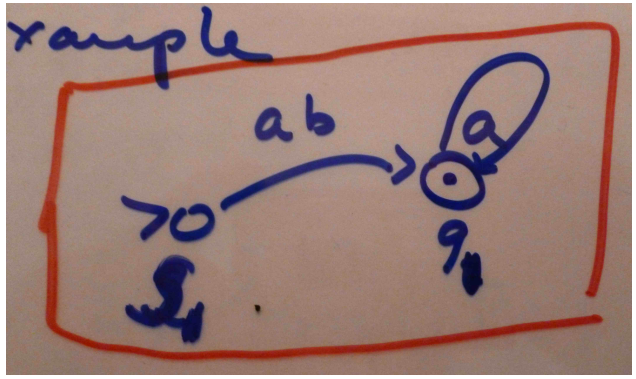
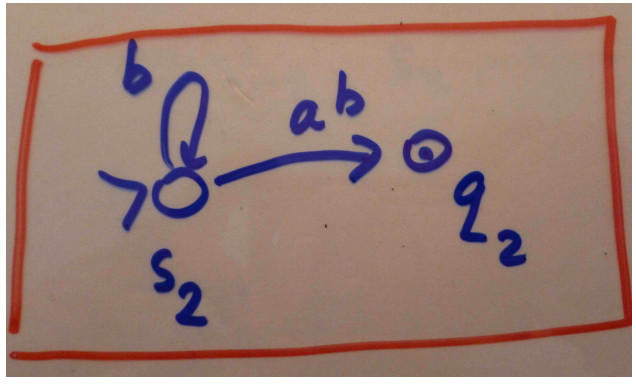


Diagram of M_2 such that $L(M_2) = b^*ab$ is



We construct $M = M_1 \cup M_2$ such that

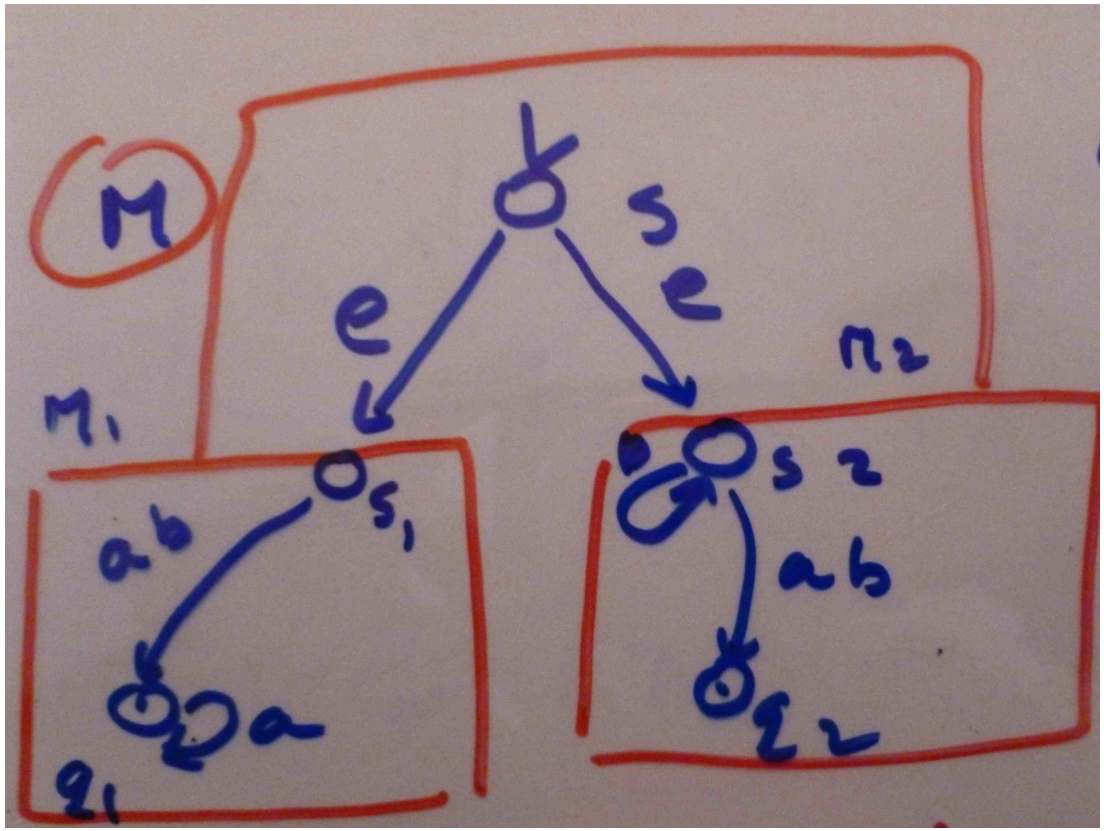
$$L(M) = aba^* \cup b^*ab = L(M_1) \cup L(M_2)$$

as follows

Examples

Example 1

Diagram of M such that $L(M) = aba^* \cup b^*ab$ is



Examples

Example 2

Diagram of M_1 such that $L(M_1) = b^*abc$ is

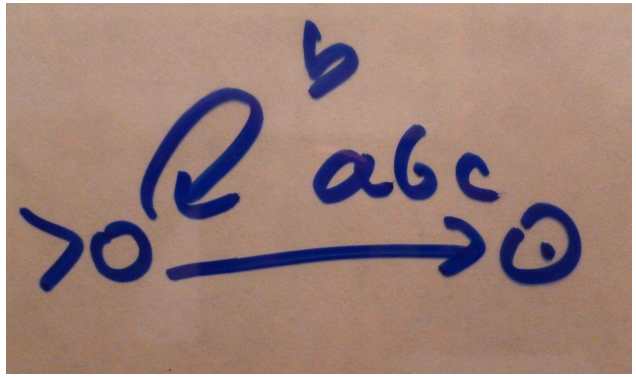
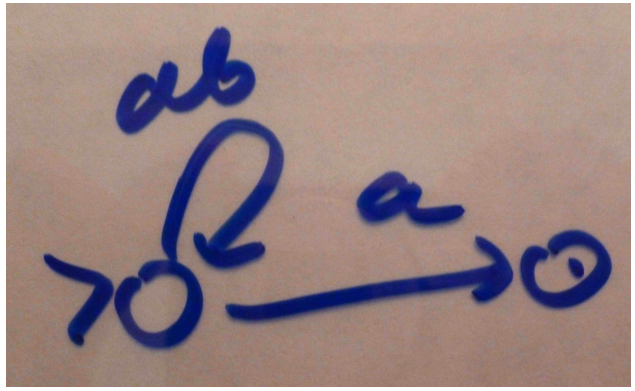


Diagram of M_2 such that $L(M_2) = (ab)^*a$ is



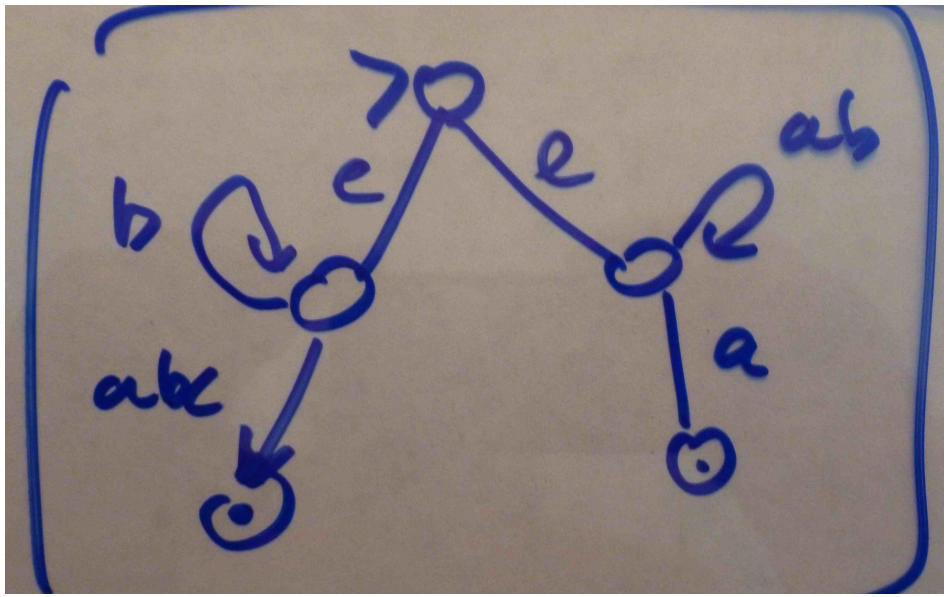
We construct $M = M_1 \cup M_2$ such that

$$L(M) = b^*abc \cup (ab)^*a = L(M_1) \cup L(M_2)$$

as follows

Examples

Diagram of M such that $L(M) = b^*abc \cup (ab)^*a$ is



This is a schema diagram

If we need to specify the components we put names on states on the diagrams

Closure Under Concatenation

2. The class of languages accepted by Finite Automata is **closed** under concatenation

Proof

Let M_1, M_2 be two NDFA

We **construct** a NDF automaton M, such that

$$L(M) = L(M_1) \circ L(M_2)$$

Let $M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1)$ and

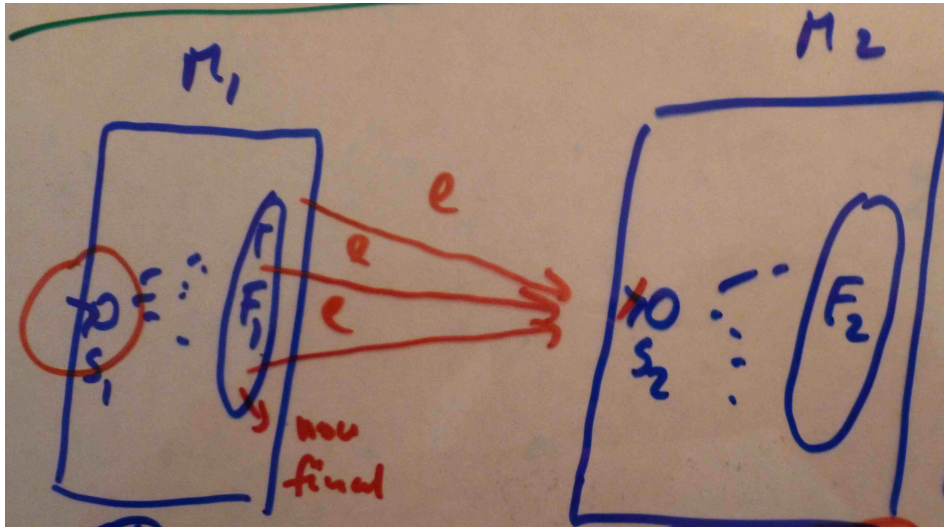
$M_2 = (K_2, \Sigma, \Delta_2, s_2, F_2)$

Where (if needed we re-name states)

$\Sigma = \Sigma_1 \cup \Sigma_2, \quad s_1 \neq s_2, \quad K_1 \cap K_2 = \emptyset \quad F_1 \cap F_2 = \emptyset$

Closure Under Concatenation

We **picture** M, such that $L(M) = L(M_1) \circ L(M_2)$ as follows



The final states from F_1 of M_1 become internal states of M

The initial state s_2 of M_2 becomes an internal state of M

M goes nondeterministically from ex-final states of M_1 to the ex-initial state of M_2 reading nothing

Closure Under Concatenation

We **define formally**

$$M = M_1 \circ M_2 = (K,\ \Sigma,\ \Delta,\ s_1,\ F_2)$$

where

$K = K_1 \cup K_2$

s_1 of M_1 is the initial state

F_2 of M_2 is the set of final states

$\Delta = \Delta_1 \cup \Delta_2 \cup \{(q, e, s_2) : \quad \text{for} \quad q \in F_1\}$

Directly from the definition we get

$w \in L(M)$ iff $w = w_1 \circ w_2$ for $w_1 \in L_1,\ w_2 \in L_2$

and hence

$$L(M) = L(M_1) \circ L(M_2)$$

Examples

Diagram of M_1 such that $L(M_1) = aba^*$ is

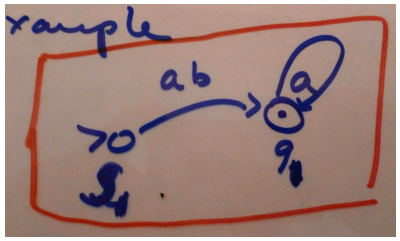
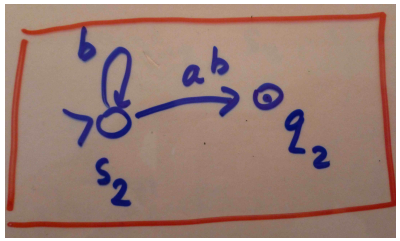


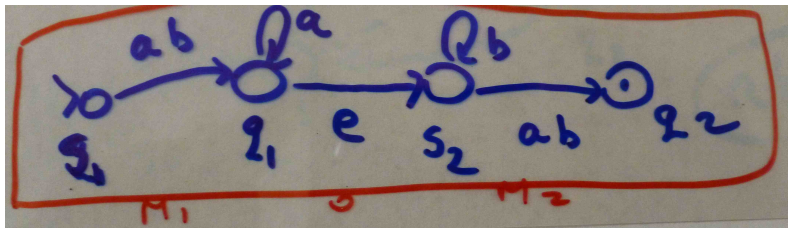
Diagram of M_2 such that $L(M_2) = b^*ab$ is



We construct $M = M_1 \circ M_2$ such that

$$L(M) = aba^* \circ b^*ab = L(M_1) \circ L(M_2)$$

as follows



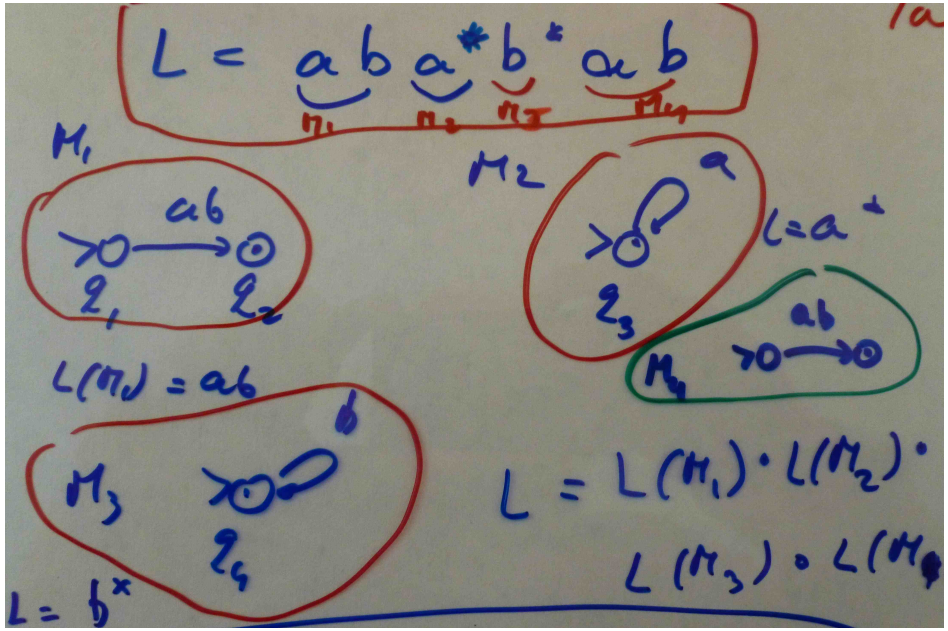
Examples

Given a language $L = aba^*b^*ab$

Observe that we can reprezent L as, for example, the following concatenation

$$L = ab \circ a^* \circ b^* \circ ab$$

Then we construct "easy" automata $M_1,\ M_2,\ M_3,\ M_4$ as follows



Examples

We know, by Mathematical Induction that we can construct, for any $n \geq 2$ an automaton

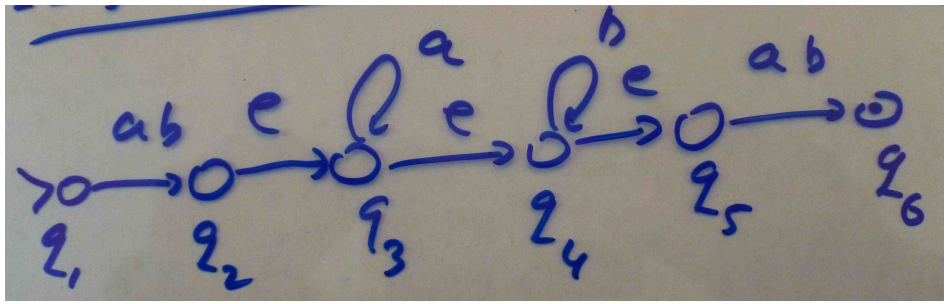
$$M = M_1 \circ M_2 \circ \ \circ M_n$$

such that

$$L(M) = L(M_1) \circ \ \ldots \ \circ \ L(M_n)$$

In our case n=4 and we get

Diagram of M



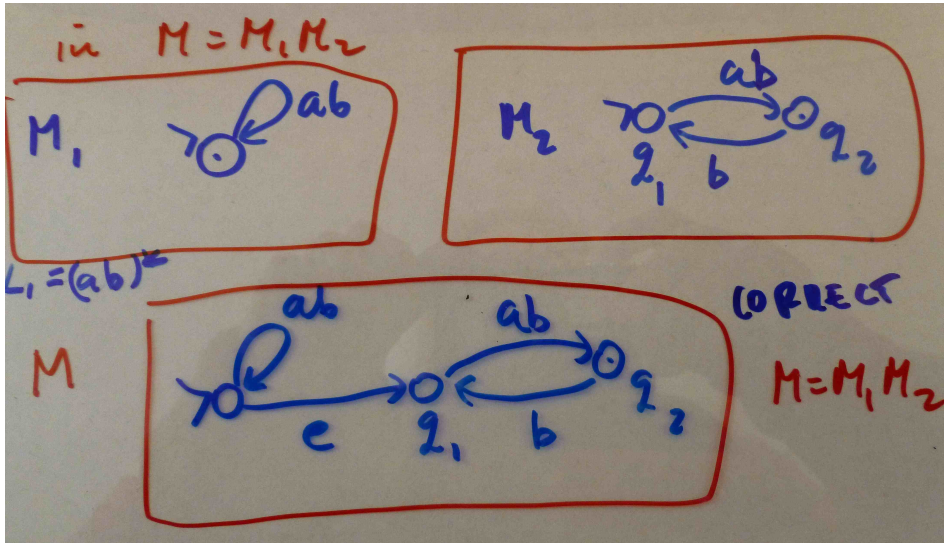
and $L(M) = aba^*b^*ab$

Question

Why we have to go be the transactions (q, e, s_2) between M_1 and M_2 while constructing $M = M_1 \circ M_2$?

Example of a construction when we can't SKIP the transaction (q, e, s_2)

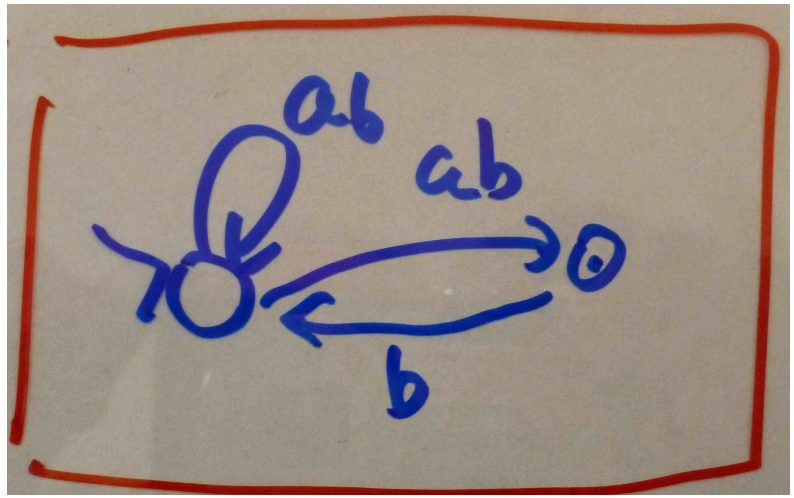
Here is a **correct** construction of $M = M_1 \circ M_2$



Observe that $abbabab \notin L(M)$

Question

Here is a construction of $M' = M_1 \circ M_2$ without the transaction (q, e, s_2)



Observe that $abbabab \in L(M')$ and $abbabab \notin L(M)$

We hence proved that skipping the transactions (q, e, s_2) between M_1 and M_2 leads to automata accepting different languages

Closure Under Kleene's Star

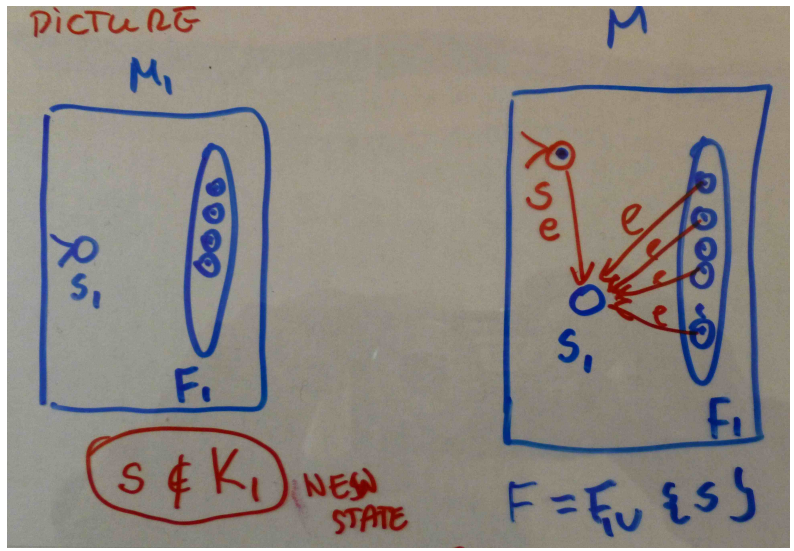
3. The class of languages accepted by Finite Automata is **closed** under Kleene's Star

Proof Let $M_1 = (K_1,\ \Sigma,\ \Delta_1,\ s_1,\ F_1)$

We **construct** a NDF automaton $M = {M_1}^*$, such that

$$L(M) = L(M_1)^*$$

Here is a **diagram**



Closure Under Kleene's Star

Given $M_1 = (K_1,\ \Sigma,\ \Delta_1,\ s_1,\ F_1)$

We **define formally**

$$M = M_1{}^* = (K,\ \Sigma,\ \Delta,\ s,\ F)$$

where

$K = K_1 \cup \{s\}$ for $s \notin K_1$

s is new initial state, s_1 becomes an internal state

$F = F_1 \cup \{s\}$

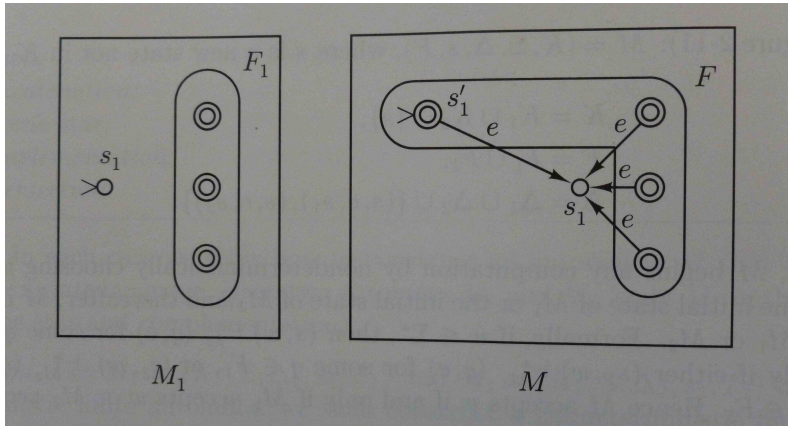
$\Delta = \Delta_1 \cup \{(s, e, s_1)\} \cup \{(q, e, s_1) :$ for $q \in F_1\}$

Directly from the definition we get

$$L(M) = L(M_1)^*$$

Closure Under Kleene's Star

The Book **diagram** is



Given $M_1 = (K_1,\ \Sigma,\ \Delta_1,\ s_1,\ F_1)$

We define

$$M_1{}^* = (K_1 \cup \{s\},\ \Sigma,\ \Delta,\ s,\ F_1 \cup \{s\})$$

where s is a new initial state and

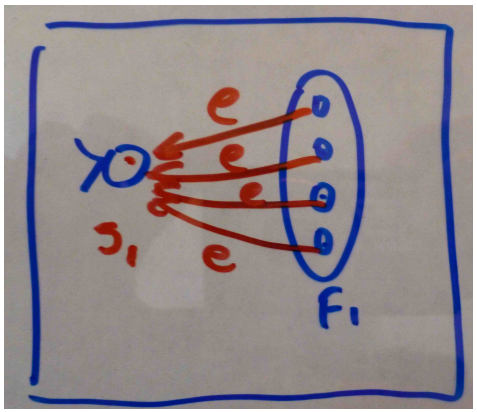
$\Delta = \Delta_1 \cup \{(s, e, s_1)\} \cup \{(q, e, s_1) :$ for $q \in F_1\}$

Two Questions

Here **two questions** about the construction of $M = M_1{}^*$

Q1 Why do we need to make the NEW initial state s of M also a FINAL state?

Q2 Why can't SKIP the introduction of the NEW initial state and design $M = M_1{}^*$ as follows



Q1 + Q2 give us answer why we construct $M = M_1{}^*$ as we did, i.e. provides the motivation for the correctness of the construction

Question 1 Answer

Observe that the definition of $M = M_1{}^*$ must be correct for ALL automata M_1 and hence in particular for M_1 such that $F_1 = \emptyset$,

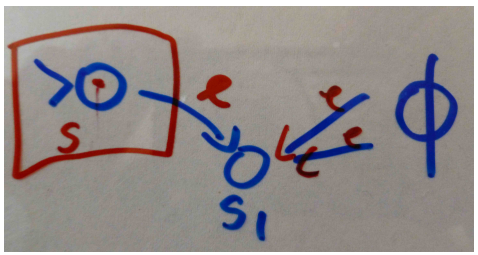
In this case we have that $L(M_1) = \emptyset$

But we know that

$$L(M) = L(M_1)^* = \emptyset^* = \{e\}$$

This proves that $M = M_1{}^*$ must accept e, and hence we must make s of M also a FINAL state

Diagram



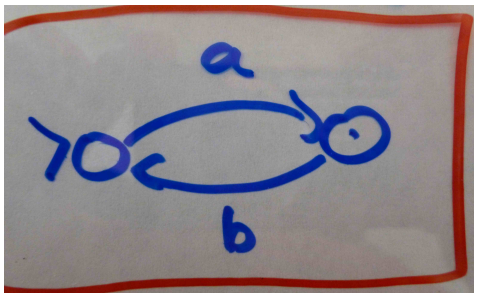
Question 2 Answer

Q2 Why can't SKIP the introduction of the NEW initial state and design $M = M_1{}^*$

Here is an **example**

Let M_1 , such that $L(M_1) = a(ba)^*$

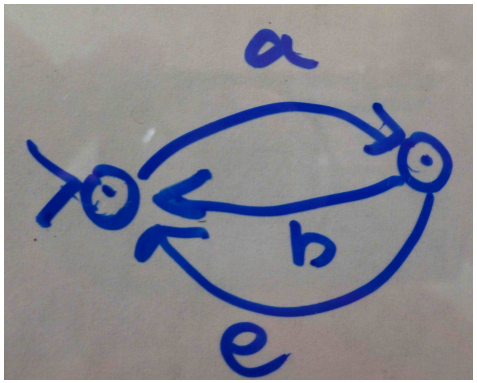
M_1 is defined by a **diagram**



$$L(M_1)^* = (a(ba)^*)^*$$

Question 2 Answer

Here is a **diagram** of M where we skipped the introduction of a new initial state



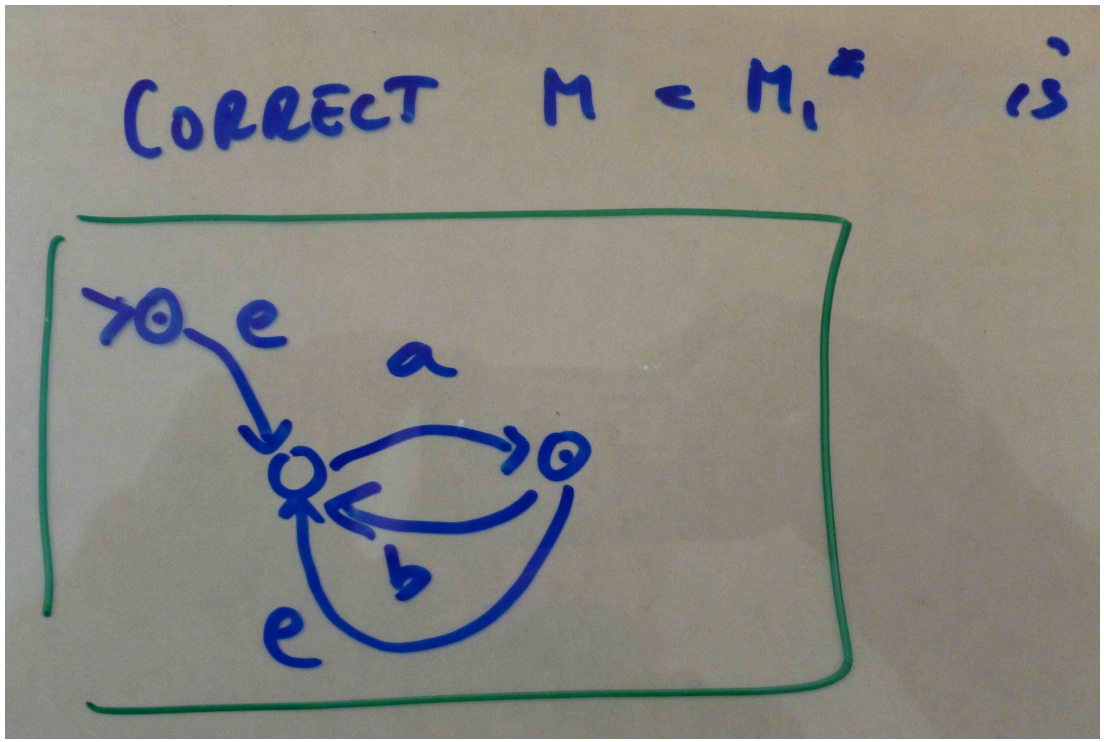
Observe that $ab \in L(M)$, but

$$ab \notin (a(ba)^*)^* = L(M_1)^*$$

This proves **incorrectness** of the above construction

Correct Diagram

The CORRECT **diagram** of $M = M_1{}^*$ is



Exercise 1

Exercise 1

Construct M such that

$$L(M) = (ab^*ba \cup a^*b)^*$$

Observe that

$$L(M) = (L(M_1) \cup L(M_2))^*$$

and

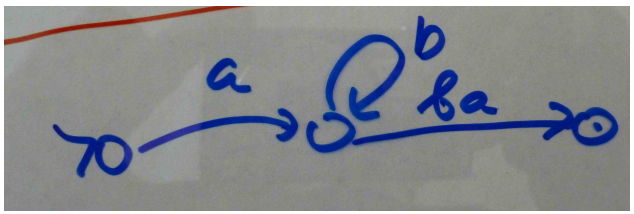
$$M = (M_1 \cup M_2)^*$$

Exercise 1

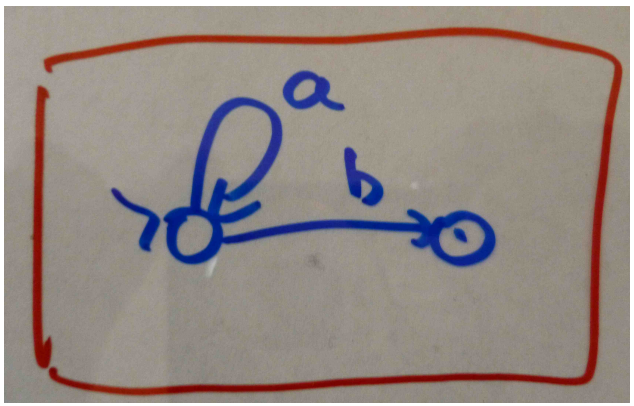
Solution

We construct M such that $L(M) = (ab^*ba \cup a^*b)^*$ in the following steps using the **Closure Theorem** definitions

Step 1 Construct M_1 for $L(M_1) = ab^*ba$

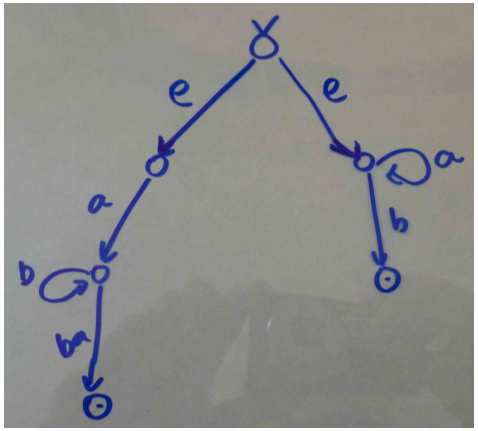


Step 2 Construct M_2 for $L(M_2) = a^*b$

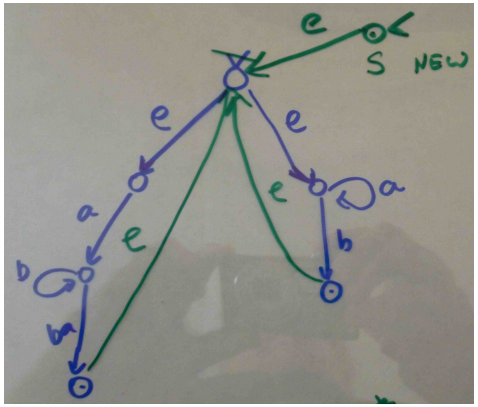


Exercise

Step 3 Construct $M_1 \cup M_2$



Step 4 Construct $M = (M_1 \cup M_2)^*$



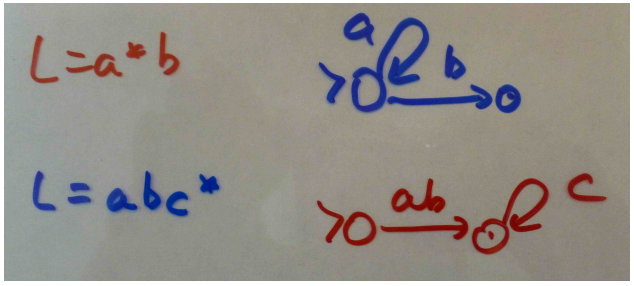
$$L(M) = (ab^*ba \cup a^*b)^*$$

Exercise 2

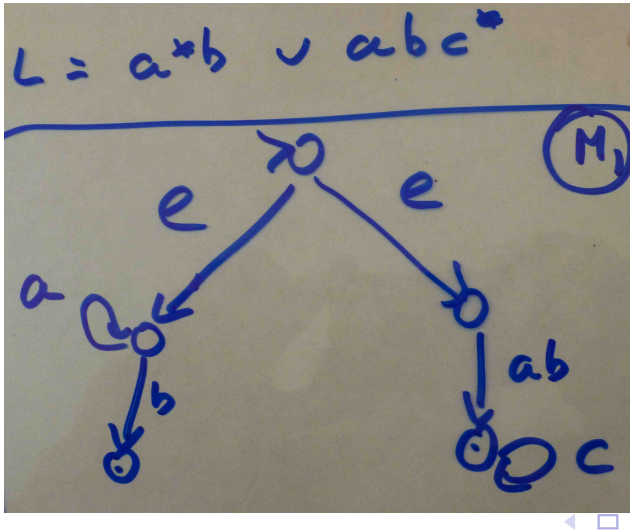
Construct M such that $L(M) = (a^*b \cup abc^*)a^*b^*$

Solution We construct M in the following steps using the **Closure Theorem** definitions

Step 1 Construct N_1, N_2 for $L = a^*b$ and $L = abc^*$

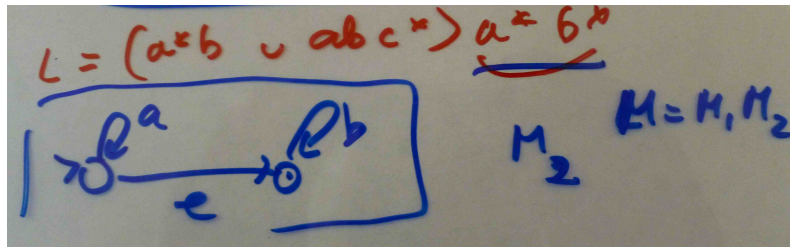


Step 2 Construct $M_1 = N_1 \cup N_2$

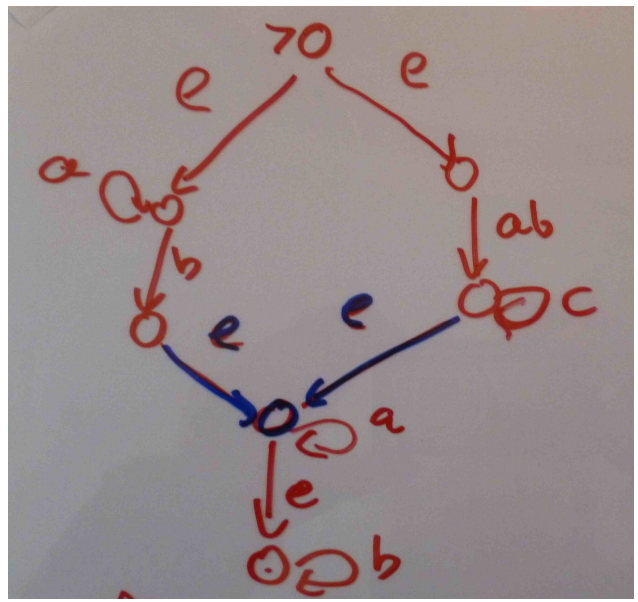


Exercise 2

Step 3 Construct M_2 for $L = a^*b^*$



Step 4 Construct $M = (M_1 \circ M_2)^*$



$$L(M) = (a^*b \cup abc^*)a^*b^*$$

Back to Closure Theorem

CLOSURE THEOREM

The class of languages accepted by Finite Automata (FA) is **closed** under the following operations

1. union **proved**
2. concatenation **proved**
3. Kleene's Star **proved**
4. complementation
5. intersection

Observe that we used the term Finite Automata (FA) so in the proof we have can choose a DFA or NDFA, as we have already proved their EQUIVALENCY

Closure Under Complementation

4. The class of languages accepted by Finite Automata is **closed** under complementation

Proof Let

$$M = (K,\ \Sigma,\ \delta,\ s,\ F)$$

be a **deterministic** finite automaton DFA

The complementary language $\overline{L} = \Sigma^* - L(M)$ is accepted by the DFA denoted by $\overline{M}$ that is identical with M except that final and nonfinal states are interchanged, i.e. we define

$$\overline{M} = (K,\ \Sigma,\ \delta,\ s,\ K - F)$$

and we have

$$L(\overline{M}) = \Sigma^* - L(M)$$

Closure Under Intersection

4. The class of languages accepted by Finite Automata is **closed** under intersection

Proof 1

Languages are sets so we have have the following property

$$L_1 \cap L_2 = \Sigma^* - ((\Sigma^* - L_1) \cup (\Sigma^* - L_2))$$

Given finite automata M_1, M_2 such that

$$L_1 = L(M_1) \quad \text{and} \quad L_2 = L(M_2)$$

We construct M such that $L(M) = L_1 \cap L_2$ as follows

1. Transform M_1, M_2 into equivalent DFA automata N_1, N_2

2. Construct $\overline{N_1}$, $\overline{N_2}$ and then $N = \overline{N_1} \cup \overline{N_2}$

3. Transform NDF automaton N into equivalent DFA automaton N'

4. $M = \overline{N'}$ is the required finite automata

This is an indirect Construction

Homework: describe the direct construction

Closure Theorem

We have proved all cases of the

CLOSURE THEOREM

The class of languages accepted by Finite Automata (FA) is **closed** under the following operations

1. union **proved**
2. concatenation **proved**
3. Kleene's Star **proved**
4. complementation **proved**
5. intersection **proved**

Observe that we used the term Finite Automata (FA) so in the proof we have can choose a DFA or NDFA, as we have already proved their EQUIVALENCY

Intersection Direct Construction

Direct Construction

Case 1 deterministic

Given **deterministic** automata $M_1,\ M_2$ such that

$$M_1 = (K_1,\ \Sigma_1,\ \delta_1,\ s_1,\ F_1), \qquad M_2 = (K_2,\ \Sigma_2,\ \delta_2,\ s_2,\ F_2)$$

We construct $M = M_1 \cap M_2$ such that $L(M) = L(M_1) \cap L(M_2)$ as follows

$$M = (K,\ \Sigma, \delta,\ s,\ F)$$

where . $\Sigma = \Sigma_1 \cup \Sigma_2$

$$K = K_1 \times K_2, \qquad s = (s_1, s_2), \qquad F = F_1 \times F_2$$

$$\delta((q_1, q_2),\ \sigma) = (\delta_1(q_1,\ \sigma),\ \delta_2(q_2,\ \sigma))$$

Intersection Direct Construction

Proof of correctness of the construction

$w \in L(M)$ if and only if

$((s_1, s_2), w) \vdash_M^* ((f_1, f_2), e))$ and $f_1 \in F_1$, $f_2 \in F_2$

if and only if

$(s_1, w) \vdash_{M_1}^* (f_1, e)$ for $f_1 \in F_1$ and

$(s_2, w) \vdash_{M_2}^* (f_2, e)$ for $f_2 \in F_2$

if and only if

$w \in L(M_1)$ and $w \in L(M_2)$

if and only if

$w \in L(M_1) \cap L(M_2)$

Intersection Direct Construction

Direct Construction

Case 2 nondeterministic

Given **nondeterministic** automata $M_1,\ M_2$ such that

$$M_1 = (K_1,\ \Sigma_1,\ \Delta_1,\ s_1,\ F_1), \qquad M_2 = (K_2,\ \Sigma_2,\ \Delta_2,\ s_2,\ F_2)$$

We construct $M = M_1 \cap M_2$ such that $L(M) = L(M_1) \cap L(M_2)$ as follows

$$M = (K,\ \Sigma, \Delta,\ s,\ F)$$

where $\Sigma = \Sigma_1 \cup \Sigma_2$

$$K = K_1 \times K_2, \quad s = (s_1, s_2), \quad F = F_1 \times F_2$$

and Δ is defined as follows

Intersection Direct Construction

Δ is defined as follows

$$\Delta = \Delta' \cup \Delta'' \cup \Delta'''$$

$\Delta' = \{((q_1, q_2), \sigma, (p_1, p_2)) : \ (q_1, \sigma, p_1) \in \Delta_1$ and $(q_2, \sigma, p_2) \in \Delta_2, \ \sigma \in \Sigma\}$

$\Delta'' = \{((q_1, q_2), \sigma, (p_1, p_2)) : \ \sigma = e, \ (q_1, \ e, \ p_1) \in \Delta_1$ and $q_2 = p_1\}$

$\Delta'' = \{((q_1, q_2), \sigma, (p_1, p_2)) : \ \sigma = e, \ (q_2, e, p_2) \in \Delta_2$ and $q_1 = p_1\}$

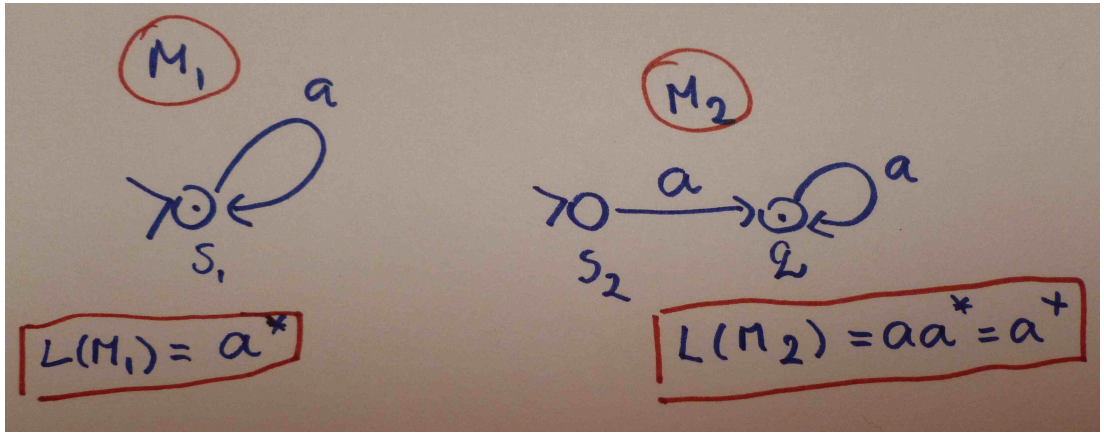
Observe that if M_1, M_2 have each at most n states, our direct construction of produces $M = M_1 \cap M_2$ with at most n^2 states.

The **indirect** construction from the proof of the theorem might generate M with up to $2^{2^{n+1}+1}$ states

Direct Construction Example

Example

Let M_1, M_2 be given by the following **diagrams**



Observe that $L(M_1) \cap L(M_2) = a^* \cap a^+ = a^+$

Direct Construction Example

Formally $M_1,\ M_2$ are defined as follows

$$M_1 = (\{s_1\},\ \{a\},\ \delta_1,\ s_1,\ \{s_1\}),\quad M_2 = (\{s_2, q\},\ \{a\},\ \delta_2,\ s_2,\ \{q\})$$

for $\delta_1(s_1, a) = s_1$ and $\delta_2(s_2, a) = q,\ \ \delta_2(q, a) = q$

By the deterministic case **definition** we have that
$M = M_1 \cap M_2$ is

$$M = (K,\ \Sigma, \delta,\ s,\ F)$$

for $\Sigma = \{a\}$

$$K = K_1 \times K_2 = \{s_1\} \times \{s_2, q\} = \{(s_1, s_2),\ (s_1, g)\ \}$$

$$s = (s_1, s_2), \quad F = \{s_1\} \times \{q\} = \{(s_1, q)\}$$

Direct Construction Example

By definition

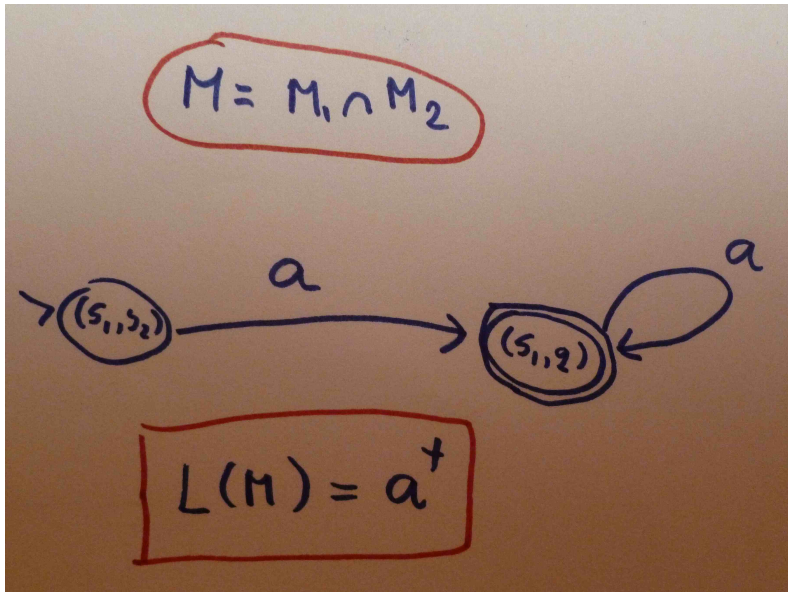
$$\delta((q_1, q_2),\ \sigma) = (\delta_1(q_1,\ \sigma),\ \delta_2(q_2,\ \sigma))$$

In our case we have

$$\delta((s_1, s_2),\ a) = (\delta_1(s_1,\ a),\ \delta_2(s_2,\ a)) = (s_1, q),$$

$$\delta((s_1, q),\ a) = (\delta_1(s_1,\ a),\ \delta_2(q,\ a)) = (s_1, q)$$

The **diagram** of $M = M_1 \cap M_2$ is



Main Theorem

Now our goal is to prove a theorem that established the relationship between languages and finite automata

This is the most important Theorem of this section so we call it a Main Theorem

Main Theorem

A language L is regular

if and only if

L is accepted by a finite automata

Main Theorem

The Main Theorem consists of the following two parts

Theorem 1

For any a regular language L

there is a finite automata M, such that $L = L(M)$

Theorem 2

For any a finite automata M, the language L(M) is regular

Main Theorem

Definition

A language $L \subseteq \Sigma^*$ is regular iff there is a regular expression $r \in \mathcal{R}$ that represents L, i.e. such that

$$L = \mathcal{L}(r)$$

Reminder: the function $\mathcal{L} : \mathcal{R} \longrightarrow 2^{\Sigma^*}$ is defined recursively as follows

1. $\mathcal{L}(\emptyset) = \emptyset, \quad \mathcal{L}(\sigma) = \{\sigma\}$ for all $\sigma \in \Sigma$
2. If $\alpha, \beta \in \mathcal{R}$, then

$$\mathcal{L}(\alpha\beta) = \mathcal{L}(\alpha) \circ \mathcal{L}(\beta) \quad \text{concatenation}$$

$$\mathcal{L}(\alpha \cup \beta) = \mathcal{L}(\alpha) \cup \mathcal{L}(\beta) \quad \text{union}$$

$$\mathcal{L}(\alpha^*) = \mathcal{L}(\alpha)^* \quad \text{Kleene's Star}$$

Regular Expressions Definition

Reminder

We define a $\mathcal{R}$ of **regular expressions** over an alphabet Σ as follows

$\mathcal{R} \subseteq (\Sigma \cup \{(,\ ),\ \emptyset,\ \cup,\ *\})^*$ and $\mathcal{R}$ is the smallest set such that

1. $\emptyset \in \mathcal{R}$ and $\Sigma \subseteq \mathcal{R}$, i.e. we have that

$$\emptyset \in \mathcal{R} \text{ and } \forall_{\sigma \in \Sigma}\ (\sigma \in \mathcal{R})$$

2. If $\alpha, \beta \in \mathcal{R}$, then

$$(\alpha\beta) \in \mathcal{R} \quad \text{concatenation}$$

$$(\alpha \cup \beta) \in \mathcal{R} \quad \text{union}$$

$$\alpha^* \in \mathcal{R} \quad \text{Kleene's Star}$$

Proof of Main Theorem Part 1

Now we are going to **prove** the first part of the Main Theorem, i.e.

Theorem 1

For any a regular language L

there is a finite automata M, such that $L = L(M)$

Proof

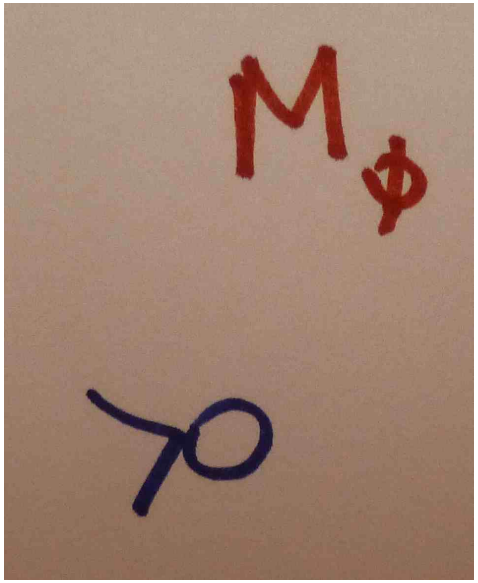
By definition of regular language, L is regular iff there is a regular expression $r \in \mathcal{R}$ that represents L, what we write in **shorthand** notation as L = r

Given a regular language, L, we **construct** a finite automaton M such that L(M) = L recursively following the definition of the set $\mathcal{R}$ of **regular expressions** as follows

Proof Theorem 1

1. $r = \emptyset$, i.e. the language is $L = \emptyset$

Diagram of M, such that $L(M) = \emptyset$ is

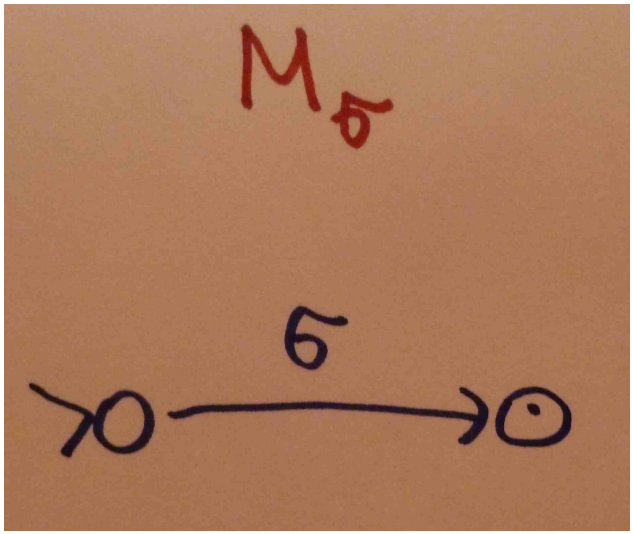


We denote M as $M = M_{\emptyset}$

Proof Theorem 1

2. $r = \sigma$, for any $\sigma \in \Sigma$ i.e. the language is $L = \sigma$

Diagram of M, such that $L(M) = \emptyset$ is



We denote M as $M = M_\sigma$

Proof Theorem 1

3. $r \neq \emptyset, \quad r \neq \sigma$

By the recursive definition, we have that $L = r$ where

$$r = \alpha \cup \beta, \quad r = \alpha \circ \beta, \quad r = \alpha^*$$

for any $\alpha, \beta \in \mathcal{R}$

We construct as in the proof of the **Closure Theorem** the automata

$$M_r = M_\alpha \cup M_\beta, \quad M_r = M_\alpha \circ M_\beta, \quad M_r = (M_r)^*$$

respectively and it ends the proof

Example

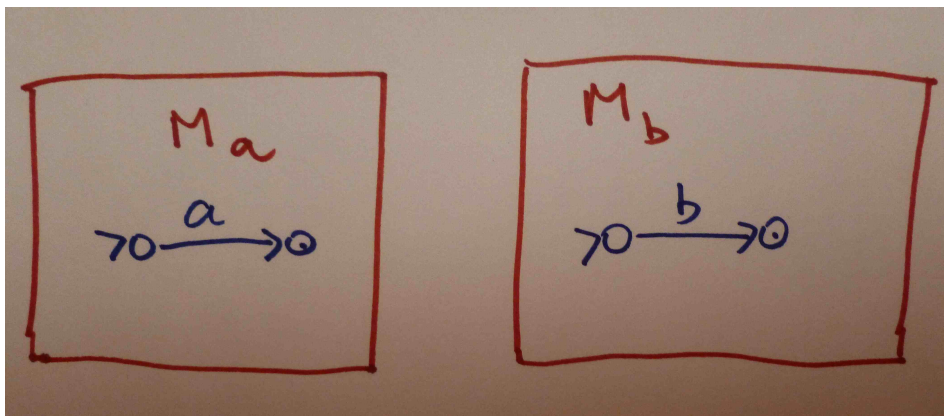
Use construction defined in the proof of **Theorem 1** to construct an automaton M such that

$$L(M) = (ab \cup aab)^*$$

We construct M in the following stages

Stage 1

For $a, b \in \Sigma$ we construct M_a and M_b

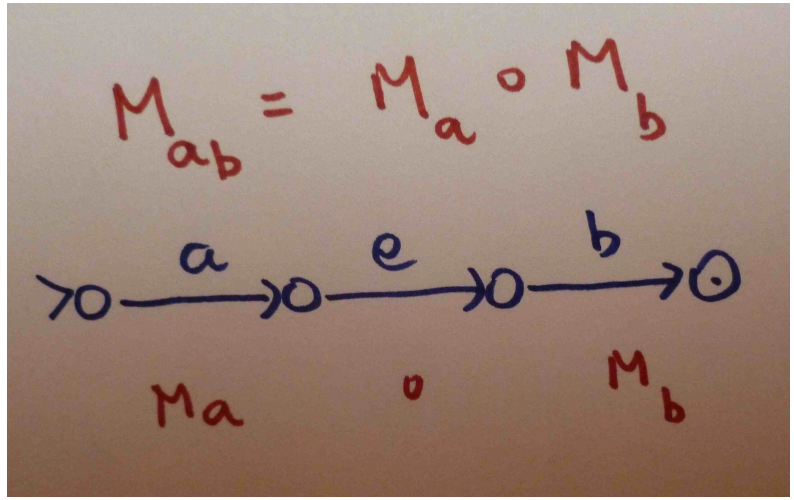


frame

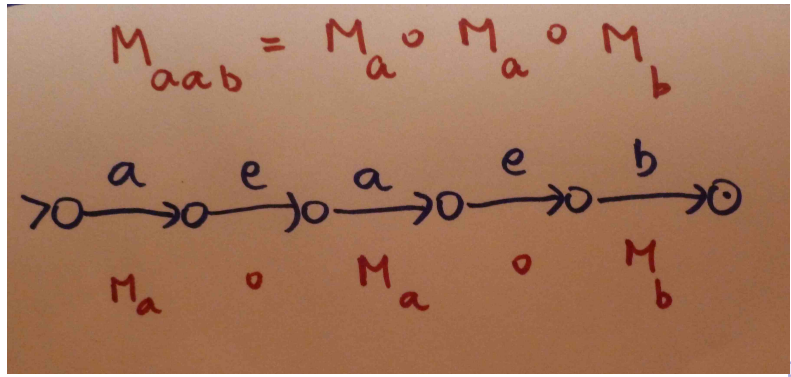
Example

Stage 2

For ab, aab we use M_a and M_b and **concatenation** construction to construct M_{ab}



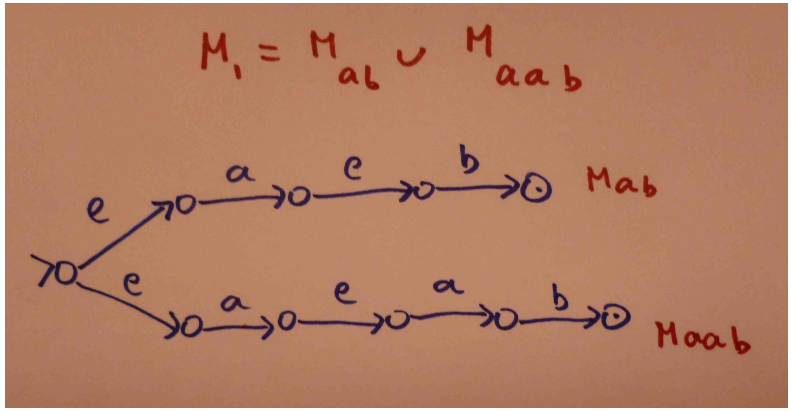
and M_{aab}



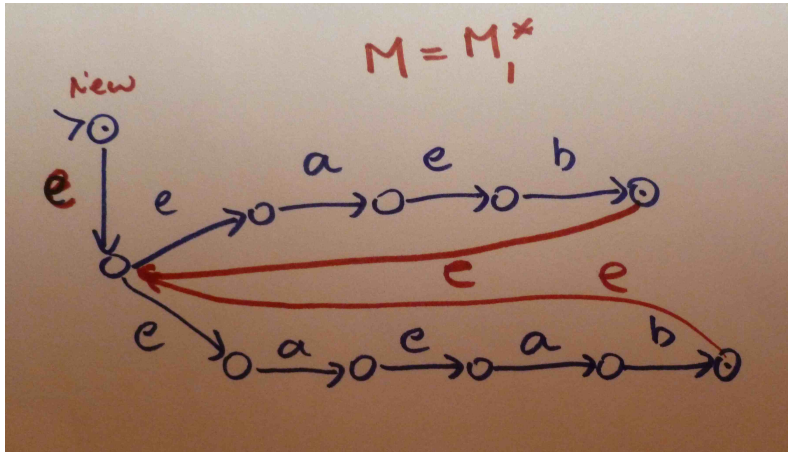
Example

Stage 3

We use **union** construction to construct $M_1 = M_{ab} \cup M_{aab}$



Stage 4 We use Kleene's **star** construction to construct $M = M_1{}^*$



Exercise

Use construction defined in the proof of **Theorem 1** to construct an automaton M such that

$$L(M) = (a^* \cup abc \cup a^*b)^*$$

We construct (draw diagrams) M in the following stages

Stage 1

Construct $M_a, \quad M_b, \quad M_c$

Stage 2

Construct $M_1 = M_{abc}$

Stage 3

Construct $M_2 = {M_a}^*$

Stage 4

Construct $M_3 = {M_a}^* M_b$

Stage 5

Construct $M_4 = M_1 \cup M_2 \cup M_3$

Stage 6

Construct $M = {M_4}^*$

Main Theorem Part 2

Theorem 2

For any a finite automaton M there is a regular expression $r \in \mathcal{R}$, such that

$$L(M) = r$$

Proof

The proof is **constructive**; given M we will give an algorithm how to recursively generate the regular expression r , such that $L(M) = r$

We assume that M is nondeterministic

$$M = (K,\ \Sigma,\ \Delta,\ s,\ F)$$

We use the BOOK definition, i.e.

$$\Delta \subseteq K \times (\Sigma \cup \{e\}) \times K$$

Proof of Theorem 2

We put states of M into a one- to - one sequence

$$K: \quad s = q_1,\ q_2,\ \ldots\ q_n \ \text{ for } \ n \geq 1$$

We build r using the following expressions

$$R(i,\ j,\ k) \quad \text{for } \ i, j = 1,\ 2,\ \ldots\ n, \quad k = 0,\ 1,\ 2,\ \ldots\ n$$

$$R(i,\ j,\ k) = \{w \in \Sigma^*; \quad (q_i,\ w) \vdash_{M,k}{}^* (q_j, w')\}$$

R(i, j, k) is the set of all words "spelled" by all PATHS from q_i to q_j in such way that we **do not pass** through an intermediate state numbered k+1 or greater

Observe that $\neg(m \geq k+1) \equiv m \leq k$ so we get the following

Proof of Theorem 2

We say that a PATH has a RANK k when

$$(q_i,\ w) \vdash_{M,k}{}^{*} (q_j, w')$$

I.e. when M can pass ONLY through states numbered $m \leq k$ while going from q_i to q_j

RANK 0 **case** $k = 0$

$$R(i,\ j,\ 0) = \{w \in \Sigma^{*};\quad (q_i,\ w) \vdash_{M,0}{}^{*} (q_j, w')\}$$

This means; M "goes" from q_i to q_j only through states numbered $m \leq 0$

There is **no** such states as $K = \{q_1,\ q_2,\ \ldots\ \ q_n\}$

Proof of Theorem 2

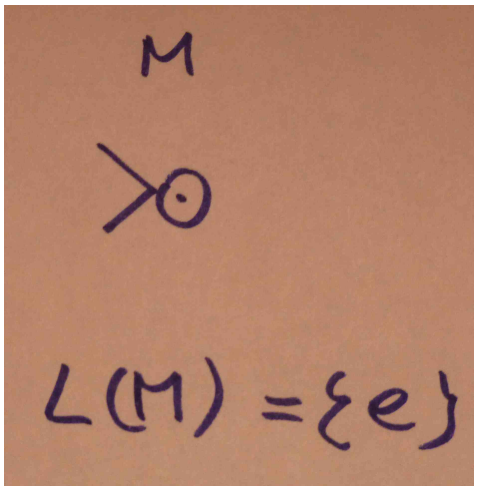
Hence $R(i,\ j,\ 0)$ means that M "goes" from q_i to q_j DIRECTLY, i.e. that

$$R(i,\ j,\ 0) = \{w \in \Sigma^*;\quad (q_i,\ w) \vdash_M{}^*(q_j, w')\}$$

Reminder: we use the BOOK definition so

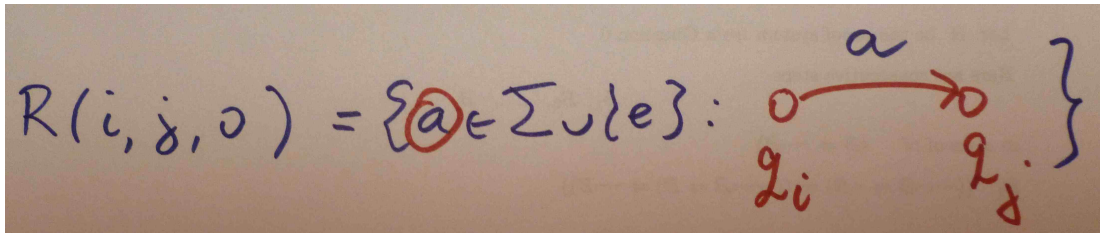
$$R(i,\ j,\ 0) = \begin{cases} a \in \Sigma \cup \{e\} & \text{if } i \neq j \text{ and } (q_i, a, q_j) \in \Delta \\ \{e\} \cup a \in \Sigma \cup \{e\} & \text{if } i = j \text{ and } (q_i, a, q_j) \in \Delta \end{cases}$$

Observe that we need $\{e\}$ in the second equation to include the following special case

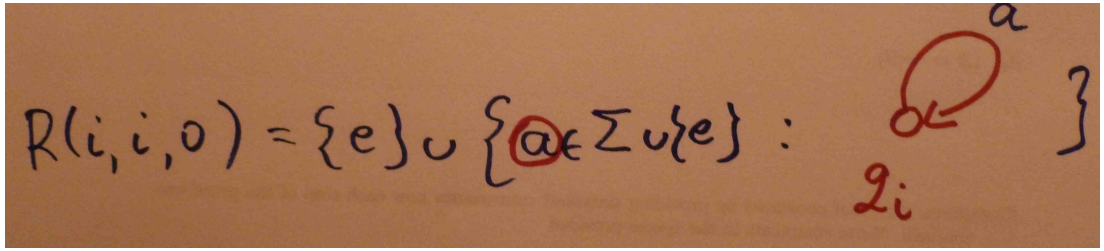


Proof of Theorem 2

We read $R(i, j, 0)$ from the **diagram** of M as follows



and



Proof of Theorem 2

RANK n **case** $k = n$

$$R(i,\ j,\ n) = \{w \in \Sigma^*;\quad (q_{\,i},\ w) \vdash_{M,n}{}^*(q_{\,j}, w')\}$$

This means; M "goes" from q_i to q_j through states numbered $m \leq n$

It means that M "goes" all states as $|K| = n$

It means that M will read any $w \in \Sigma$ and hence

$$R(i,\ j,\ n) = \{w \in \Sigma^*;\quad (q_i,\ w) \vdash_M{}^*(q_j, e)\}$$

Observe that

$$w \in L(M) \quad \text{iff} \quad w \in R(1,\ j,\ n) \quad \text{and} \quad q_j \in F$$

Proof of Theorem 2

By definition of the $L(M)$ we get

$$L(M) = \bigcup \{R(1, j, n) : \quad q_j \in F\}$$

Fact

All sets $R(i, j, k)$ are regular and hence L(M) is also regular

Proof by induction on k

Base case: k =0

All sets R(i, j, 0) are FINITE, hence are regular

Proof of Theorem 2

Inductive Step

The **recursive formula** for $R(i,\ j,\ k)$ is

$$R(i,j,k) = R(i,j,k-1) \cup R(i,k,k-1)R(k,k,k-1)^*R(k,j,k-1)$$

where n is the number of states of M and
$k = 0,\ldots,n,\ \ i,j = 1,\ldots,n$

By Inductive assumption, all sets
$R(i,j,k-1),\ R(i,k,k-1),\ R(k,k,k-1),\ R(k,j,k-1)$ are regular and by the **Closure Theorem** so is the set $R(i,j,k)$

This **ends** the proof of **Theorem 2**

Observe that the recursive formula for $R(i,\ j,\ k)$ computes r such that L(M) = r

Example

Example

For the automaton M such that

$$M = (\{q_1, q_2, q_3\},\ \{a, b\},\ s = q_1,$$

$$\Delta = \{(q_1, b, q_2), (q_1, a, q_3), (q_2, a, q_1), (q_2, b, q_1),$$

$$(q_3, a, q_1), (q_3, b, q_1)\},\ F = \{q_1\})$$

Evaluate 4 steps, in which you must include at least one $R(i, j, 0)$, in the construction of regular expression that defines $L(M)$

Example

Reminder

$$L(M) = \bigcup \{R(1,\ j,\ n) :\ \ q_j \in F\}$$

$$R(i,j,k) = R(i,j,k-1) \cup R(i,k,k-1)R(k,k,k-1)^* R(k,j,k-1)$$

$$R(i,\ j,\ 0) = \begin{cases} a \in \Sigma \cup \{e\} & \text{if } i \neq j \text{ and } (q_i, a, q_j) \in \Delta \\ \{e\} \cup a \in \Sigma \cup \{e\} & \text{if } i = j \text{ and } (q_i, a, q_j) \in \Delta \end{cases}$$

Example Solution

Solution

Step 1 $L(M) = R(1,1,3)$

Step 2

$R(1,1,3) = R(1,1,2) \cup R(1,3,2)R(3,3,2)^*R(3,1,2)$

Step 3

$R(1,1,2) = R(1,1,1) \cup R(1,2,1)R(2,2,1)^*R(2,1,1)$

Step 4

$R(1,1,1) = R(1,1,0) \cup R(1,1,0)R(1,1,0)^*R(1,1,0)$ and $R(1,1,0) = \{e\} \cup \emptyset = \{e\}$, so we get

$R(1,1,1) = \{e\} \cup \{e\}\{e\}^*\{e\} = \{e\}$

Generalized Automaton

Definition

We define now a **Generalized Automaton** GM as the following generalization of of a nondeterministic automaton $M = (K,\ \Sigma,\ \Delta,\ s,\ F)$ as follows

$$GM = (K_G,\ \Sigma_G,\ \Delta_G,\ s_G,\ F_G)$$

1. GM has a single final state, i,e. $F_G = \{f\}$

2. $\Sigma_G = \Sigma \cup \mathcal{R}_0$ where $\mathcal{R}_0$ is a FINITE subset of $\mathcal{R}$ over Σ

3. Transitions of GM may be labeled not only by symbols in $\Sigma \cup \{e\}$ but also by **regular expressions** $r \in \mathcal{R}$, i.e. Δ_G is a FINITE set such that

$$\Delta_G \subseteq K \times (\Sigma \cup \{e\} \cup \mathcal{R}) \times K$$

4. There is no transition going into the initial state s nor out of the final state f

if $(q,\ u,\ p) \in \Delta_G$, then $q \neq f,\ \ p \neq s$

Generalized Automata

Given a nondeterministic automaton

$$M = (K,\ \Sigma,\ \Delta,\ s,\ F)$$

We present now a new method of construction of a regular expression $r \in \mathcal{R}$ that defines $L(M)$, i.e. such that $L(M) = r$ by the use of the notion of of **Generalized Automaton**

The method consists of a construction of a sequence of generalized automata that are all equivalent to M

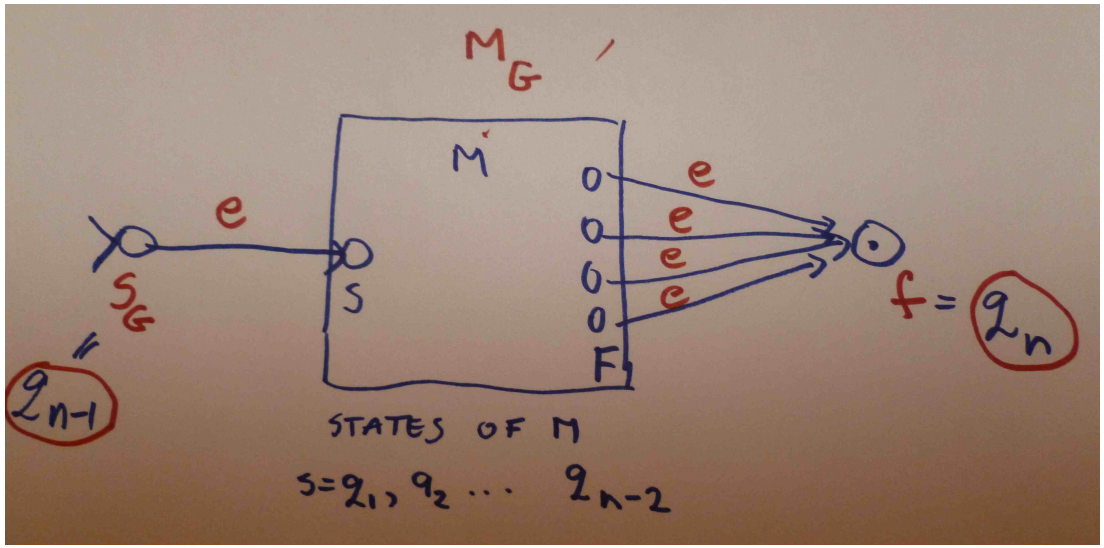
Construction

Steps of construction are as follows

Step 1

We **extend** M to a generalized automaton M_G, such that $L(M) = L(M_G)$ as depicted on the diagram below

Diagram of M_G



M_G Definition

Definition of M_G

We re-name states of M as $s = q_1, q_2, \ldots, q_{n-2}$ for appropriate n and make the initial state $s = q_1$ and all final states of M the internal non-final states of G_M

We ADD TWO states: initial and one final, which me name q_{n-1}, q_n, respectively, i.e. we put

$$s_G = q_{n-1} \quad \text{and} \quad f = q_n$$

We take

$$\Delta_G = \Delta \cup \{(q_{n-1}, e, s)\} \cup \{(q, e, q_n) : \ q \in F\}$$

Obviously $L(M) = L(M_G)$, and so $M \approx M_G$

States of G_M Elimination

We construct now a sequence $GM1, GM2, \ldots, GM(n-2)$ such that

$$M \approx M_G \approx GM1 \approx \cdots \approx GM(n-2)$$

where $GM(n-2)$ has only **two states** q_{n-1} and q_n and only **one transition** (q_{n-1}, r, q_n) for $r \in \mathcal{R}$, such that

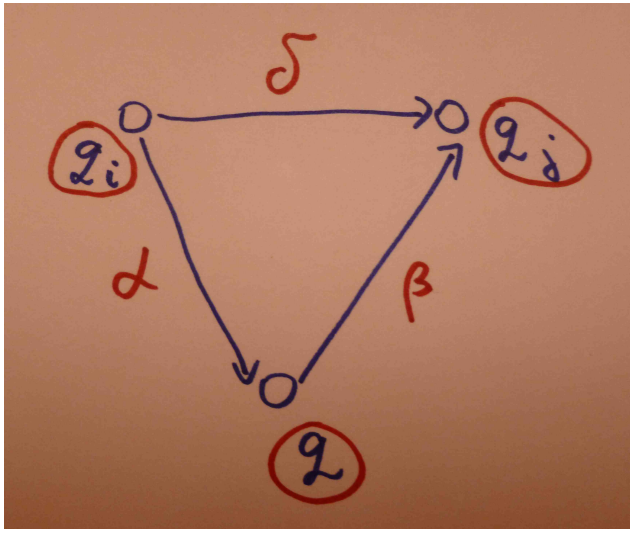
$$L(M) = r$$

We construct the sequence $GM1, GM2, \ldots, GM(n-2)$ by eliminating states of M one by one following rules given by the following diagrams

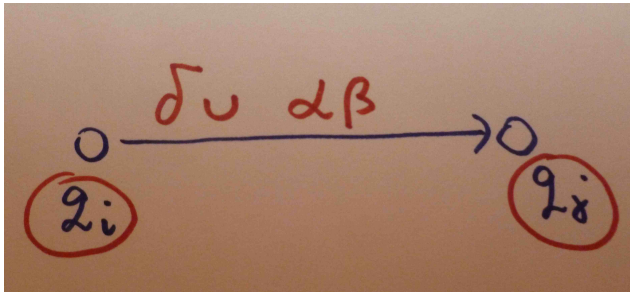
States of G_M Elimination

Case 1 of state **elimination**

Given a fragment of *GM* **diagram**



we **transform** it into

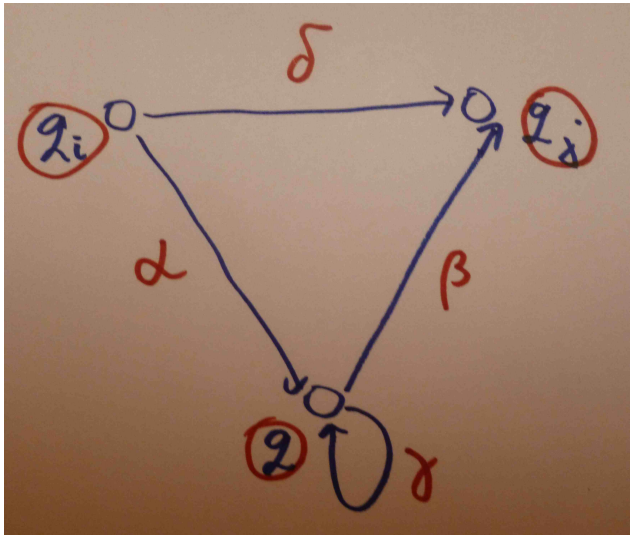


The state $q \in K$ has been **eliminated** preserving the language of *GM* and we constructed $GM' \approx GM$

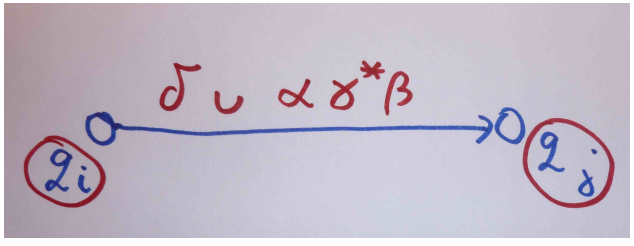
States of G_M Elimination

Case 2 of state **elimination**

Given a fragment of *GM* **diagram**



we **transform** it into



The state $q \in K$ has been **eliminated** preserving the language of *GM* and we constructed $GM' \approx GM$

Example 1

Example 1

Use the Generalized Automata Construction and States of G_M Elimination procedure to evaluate $r \in \mathcal{R}$, such that

$$\mathcal{L}(r) = L(M)$$

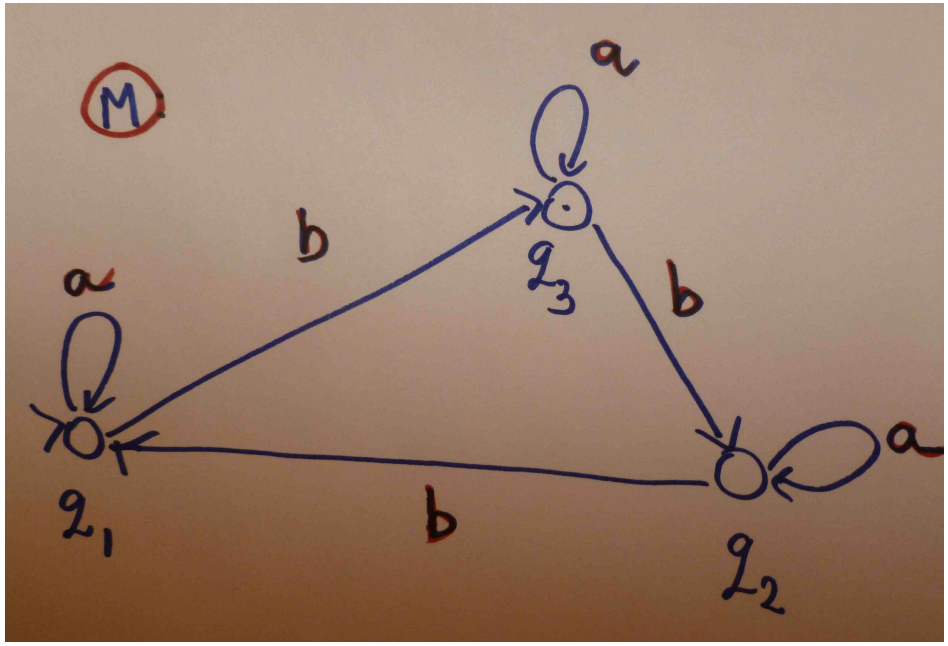
, where M is an automata that accepts the language

$$L = \{w \in \{a, b\}^* : \quad w \text{ has } \ 3k + 1 \ b'\text{s}, \ \text{for some } \ k \in N\}$$

This is the Book example, page 80

Example 1

The **Diagram** of M is



Step 1

We extend M with $K = \{q_1, q_2, q_3\}$ to a generalized M_G by adding two states

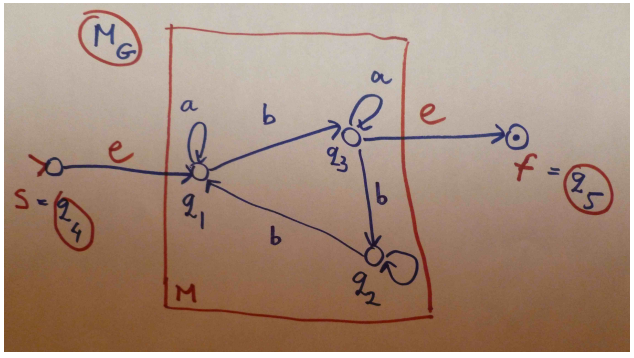
$$s_G = q_4 \quad \text{and} \quad f = q_5$$

We take

$$\Delta_G = \Delta \cup \{(q_4, e, q_1)\} \cup \{(q_3, e, q_5)\}$$

Example 1

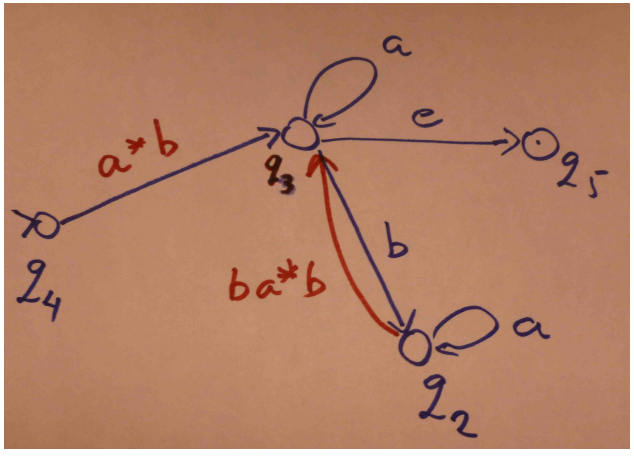
The **Diagram** of M_G is



Step 2

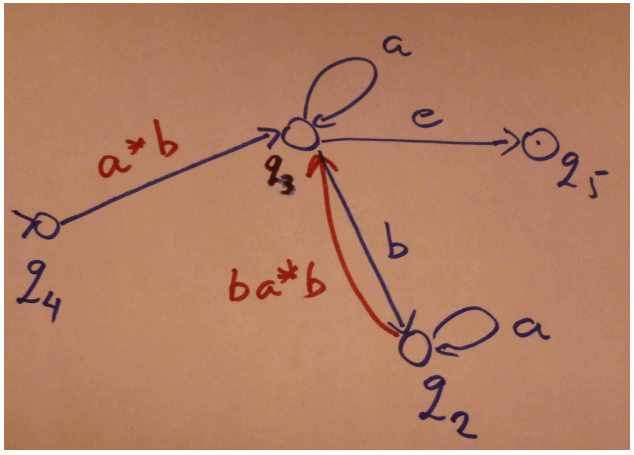
We construct $GM1 \approx M_G \approx M$ by **elimination** of q_2

The **Diagram** of $GM1$ is



Example 1

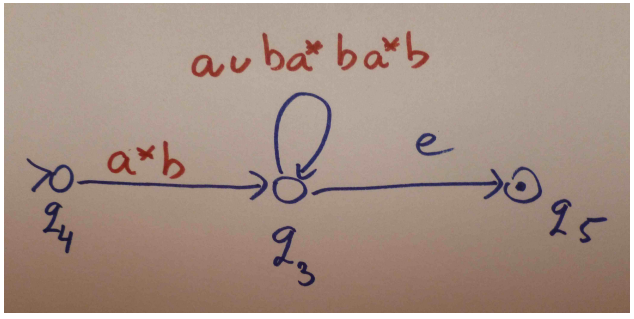
The **Diagram** of *GM1* is



Step 3

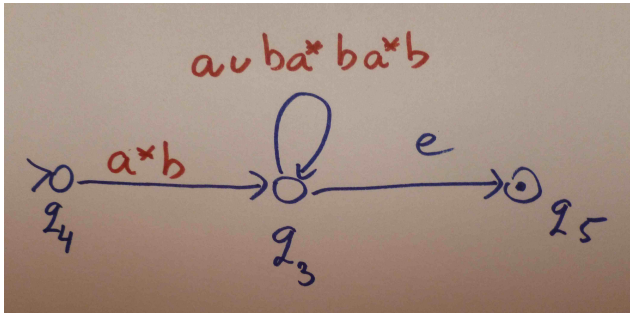
We construct $GM2 \approx GM1$ by **elimination** of q_2

The **Diagram** of *GM2* is



Example 1

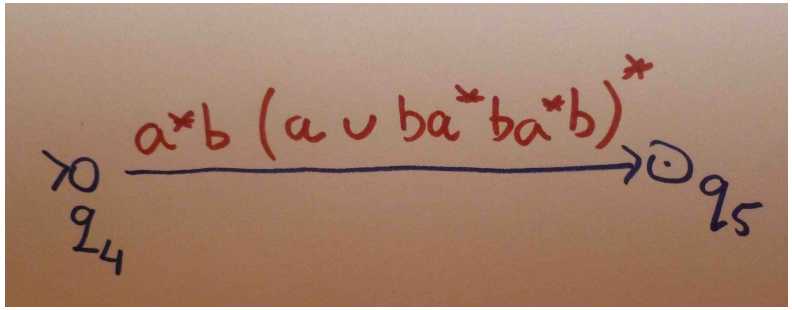
The **Diagram** of *GM2* is



Step 4

We construct $GM3 \approx GM2$ by **elimination** of q_3

The **Diagram** of *GM2* is



$$L(GM3) = a^*b(a \cup ba^*ba^*b)^* = L(M)$$

Example 2

Given the automaton

$$M = (K,\ \Sigma,\ \Delta,\ s,\ F)$$

where

$$K = \{q_1, q_2, q_3\}, \quad \Sigma = \{a, b\}, \quad s = q_1, \quad F = \{q_1\}$$

$$\Delta = \{(q_1, b, q_2),\ (q_1, a, q_3),\ (q_2, a, q_1),$$

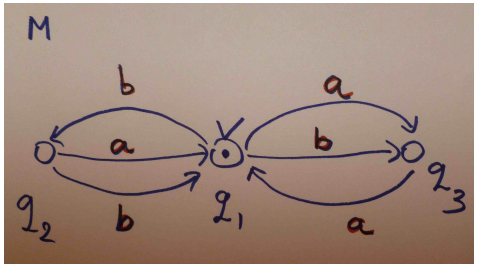
$$(q_2, b, q_1),\ (q_3, a, q_1),\ (q_3, b, q_1)$$

Use the Generalized Automata Construction and States of G_M Elimination procedure to evaluate $r \in \mathcal{R}$, such that

$$\mathcal{L}(r) = L(M)$$

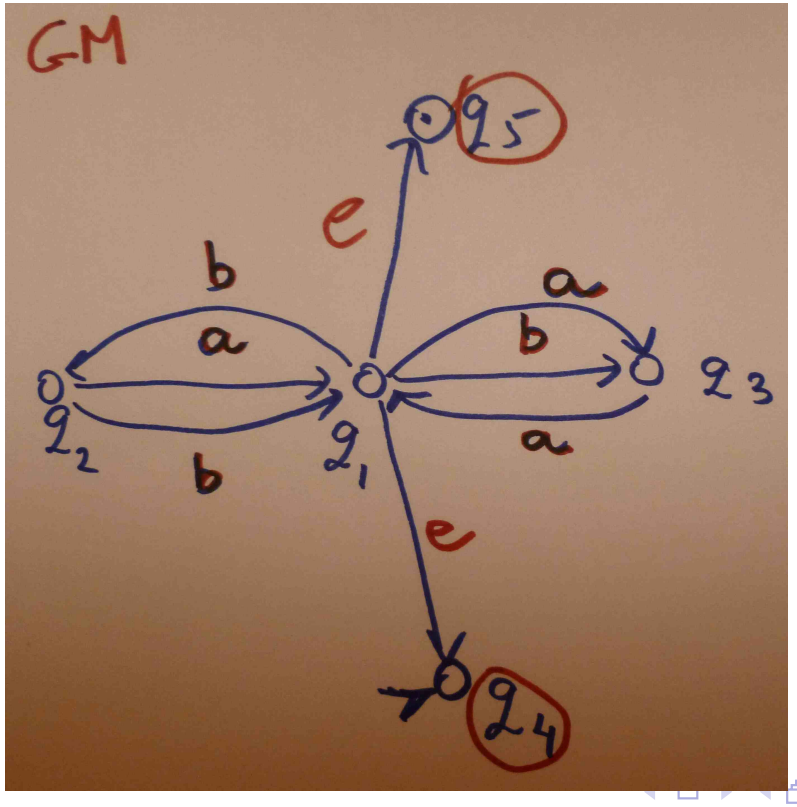
Example 2

The **diagram** of M is



Step 1

The **diagram** of $M_G \approx M$ is



Example 2

Step 1

The components of $M_G \approx M$ are

$$M_G = (K = \{q_1, q_2, q_3, q_4, q_5\}, \quad \Sigma = \{a, b\}, \quad s_G = q_4,$$

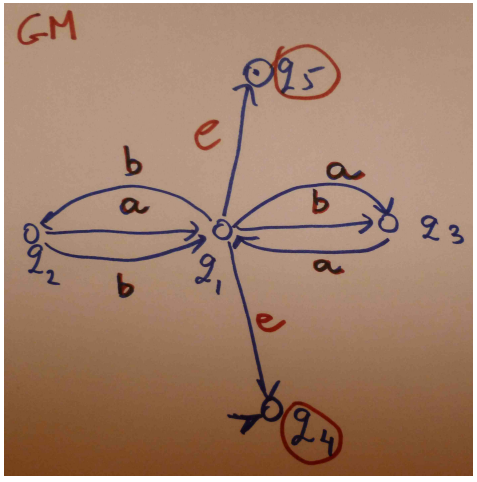
$$\Delta_G = \{(q_1, b, q_2), \ (q_1, a, q_3), \ (q_2, a, q_1),$$

$$(q_2, b, q_1), (q_3, a, q_1), (q_3, b, q_1), \ (q_4, e, q_1),$$

$$(q_1, e, q_5)\}, \quad F = \{q_5\})$$

Example 2

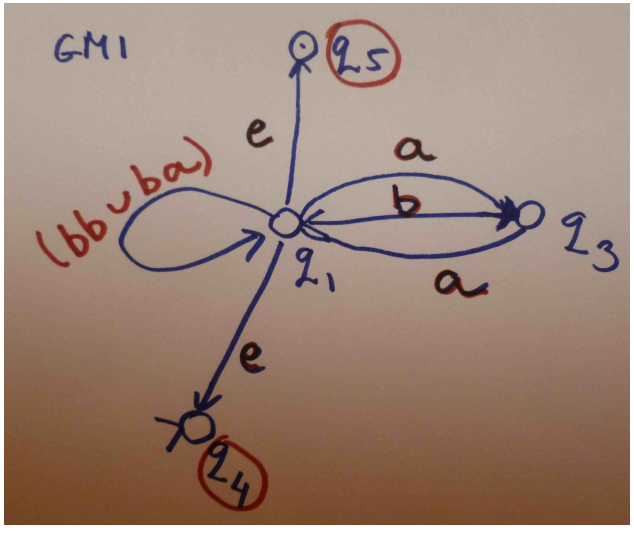
The **Diagram** of M_G is



Step 2

We construct $GM1 \approx M_G \approx M$ by **elimination** of q_2

The **Diagram** of $GM1$ is



Example 2

Step 2

The components of $GM1 \approx M_G \approx M$ are

$$GM1 = (K = \{q_1, q_3, q_4, q_5\}, \quad \Sigma = \{a, b\}, \quad s_G = q_4$$

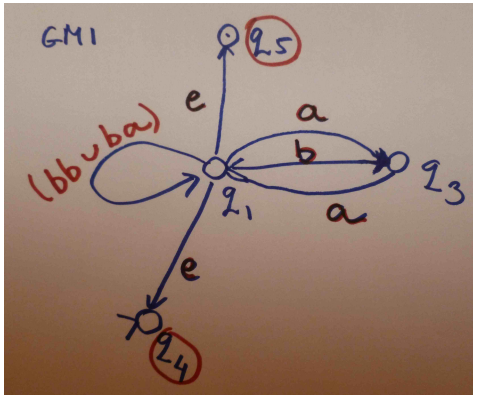
$$\Delta_G = \{(q_1, a, q_3), \ (q_1, (bb \cup ba), q_1),$$

$$(q_3, a, q_1), \ \ (q_3, b, q_1), \ \ (q_4, e, q_1),$$

$$(q_1, e, q_5)\}, \quad F = \{q_5\})$$

Example 2

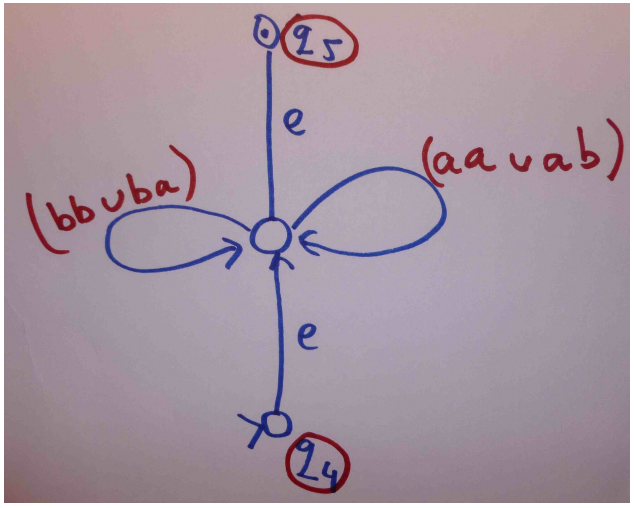
The **Diagram** of *GM1* is



Step 3

We construct $GM2 \approx GM1$ by **elimination** of q_3

The **Diagram** of *GM2* is



Example 2

Step 3

The components of $GM2 \approx GM1 \approx M_G \approx M$ are

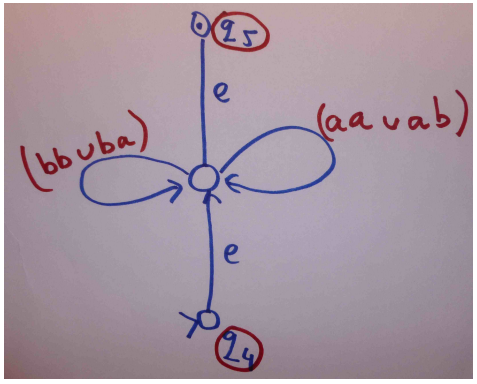
$$GM2 = (K = \{q_1,\ q_4,\ q_5\},\quad \Sigma = \{a, b\},\quad s_G = q_4$$

$$\Delta_G = \{(q_1, (bb \cup ba), q_1),\ \ (q_1, (aa \cup ab), q_1),$$

$$(q_4, e, q_1), (q_1, e, q_5)\},\quad F = \{q_5\})$$

Example 2

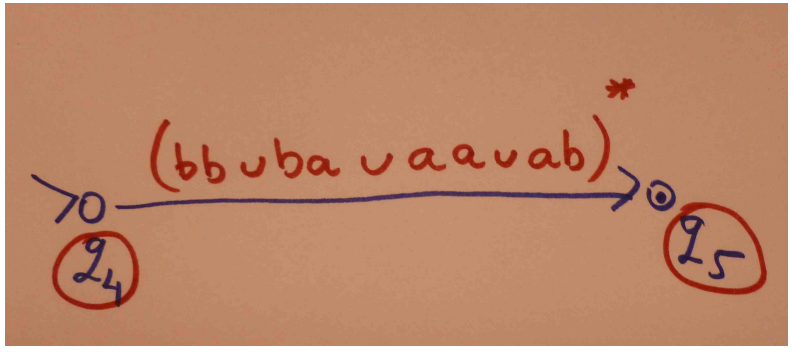
The **Diagram** of *GM2* is



Step 4

We construct $GM3 \approx GM2$ by **elimination** of q_1

The **Diagram** of *GM3* is

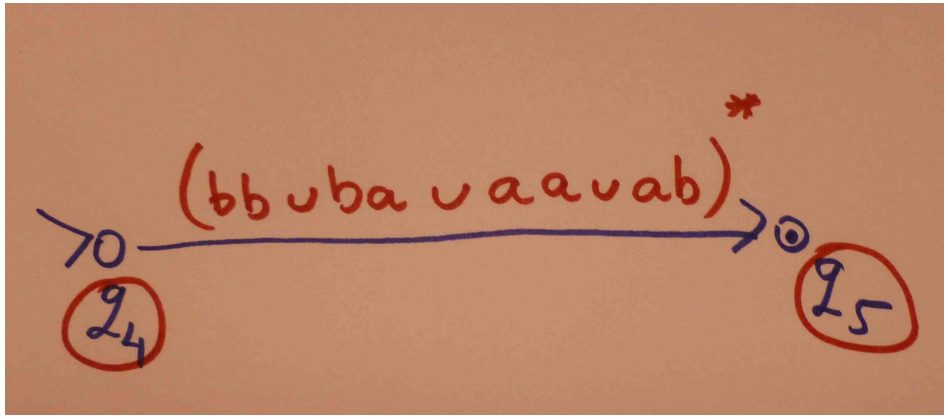


Example 2

We have constructed

$$GM3 \approx GM2 \approx GM1 \approx M_G \approx M$$

The **Diagram** of $GM3$ is



Hence the language

$$L(GM3) = (bb \cup ba \cup aa \cup ab)^* = ((a \cup b)(a \cup b))^* = L(M)$$