Unit-5: ω-regular properties

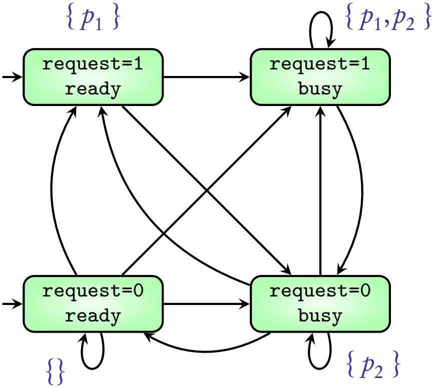
B. Srivathsan

Chennai Mathematical Institute

NPTEL-course

July - November 2015

Module 1:
Specifying properties



Traces: $\{\ \}\,\{\ \}\,\{\ \}\,\{\ \}\,\{\ \}\,\{\ \}\,\{\ \}\,\{\ \}\ldots$

$\{\ \}\,\{ p_2 \}\,\{ p_2 \}\,\{ p_2 \}\,\{ p_2 \}\,\{ p_2 \}\,\{ p_2 \}\ldots$

$\{ p_1 \}\,\{ p_1,p_2 \}\,\{ p_2 \}\,\{ p_1,p_2 \}\,\{ p_2 \}\,\{ p_1,p_2 \}\ldots$

$\{\}\,\{ p_1,p_2 \}\,\{ p_1,p_2 \}\,\{ p_1,p_2 \}\,\{ p_1,p_2 \}\,\{ p_1,p_2 \}\ldots$

$\vdots$

$$
\begin{aligned}
\mathbf{AP} &= \{\ p_1, p_2,\ \dots\ , p_k\ \} \\
\mathit{PowerSet}(\mathbf{AP}) &= \{\ \{\ \},\ \{p_1\},\ \dots\ ,\ \{p_k\}, \\
&\qquad \{\ p_1, p_2\ \},\ \{\ p_1, p_3\ \},\ \dots\ ,\ \{\ p_{k-1}, p_k\ \}, \\
&\qquad\qquad \dots \\
&\qquad \{\ p_1, p_2,\ \dots,\ p_k\ \}\ \}
\end{aligned}
$$

Trace(Execution) is an **infinite word** over *PowerSet*(**AP**)

Traces(TS) is the $\{\ \text{Trace}(\sigma)\ \mid\ \sigma$ is an execution of the TS $\}$

AP-INF = set of **infinite words** over *PowerSet*(**AP**)

Property 1: p_1 is always true

$$\{\, A_0A_1A_2 \cdots \in \text{AP-INF} \mid \textbf{each } A_i \text{ contains } p_1 \,\}$$

$\{p_1\}\{p_1\}\{p_1\}\{p_1\}\{p_1\}\{p_1\}\{p_1\}\ldots$

$\{p_1\}\{p_1,p_2\}\{p_1\}\{p_1,p_2\}\{p_1\}\{p_1,p_2\}\ldots$

$\vdots$

Property 2: p_1 is true at least once and p_2 is always true

$$\{\, A_0A_1A_2 \cdots \in \text{AP-INF} \mid \textbf{exists } A_i \text{ containing } p_1 \text{ and } \textbf{every } A_j \text{ contains } p_2 \,\}$$

$\{p_2\}\{p_1,p_2\}\{p_2\}\{p_2\}\{p_2\}\{p_1,p_2\}\{p_2\}\ldots$

$\{p_1,p_2\}\{p_2\}\{p_2\}\{p_2\}\{p_2\}\{p_2\}\ldots$

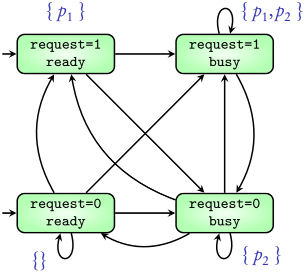
$\vdots$

AP-INF = set of **infinite words** over *PowerSet*(**AP**)

A property over AP is a **subset** of AP-INF

$$\mathrm{AP} = \{\ p_1,\ p_2\ \ \}$$

Transition System | **Property**



P

Transition system TS satisfies property P if

$$\textbf{Traces}(TS) \subseteq P$$

Specifying properties

G, F, X, GF

Finite Automata

Specifying properties

G, F, X, GF

Finite Automata

ω-regular expressions

Specifying properties

G, F, X, GF **Finite Automata** **ω-regular expressions**

- Use ω-regular expressions to specify properties
- An algorithm for model-checking ω-regular expressions on transition systems

Unit-5: ω-regular properties

B. Srivathsan

Chennai Mathematical Institute

NPTEL-course

July - November 2015

Module 2:

ω-regular expressions

Languages over finite words

Σ : finite alphabet $\quad \Sigma^* \;=\;$ set of **all words** over Σ

Language: A set of **finite words**

$\{\, ab,\ abab,\ ababab,\ \ldots \}$

finite words starting with an a

finite words starting with a b

$\{\, \epsilon,\ b,\ bb,\ bbb,\ \ldots \}$

$\{\, \epsilon,\ ab,\ abab,\ ababab,\ \ldots \}$

$\{\, \epsilon,\ bbb,\ bbbbbb,\ (bbb)^3,\ \ldots \}$

words starting and ending with an a

$\{\, \epsilon,\ ab,\ aabb,\ aaabbb, a^4 b^4\ \ldots \}$

Regular expressions

Σ : finite alphabet $\quad \Sigma^* \ = \$ set of **all words** over Σ

Language: A set of **finite words**

$ab(ab)^*$ $\quad \{\, ab,\ abab,\ ababab,\ \ldots \}$

$a\Sigma^*$ $\quad$ finite words starting with an a

$b\Sigma^*$ $\quad$ finite words starting with a b

b^* $\quad \{\, \epsilon,\ b,\ bb,\ bbb,\ \ldots \}$

$(ab)^*$ $\quad \{\, \epsilon,\ ab,\ abab,\ ababab,\ \ldots \}$

$(bbb)^*$ $\quad \{\, \epsilon,\ bbb,\ bbbbbb,\ (bbb)^3,\ \ldots \}$

$a\Sigma^*a$ $\quad$ words starting and ending with an a

$\{\, \epsilon,\ ab,\ aabb,\ aaabbb, a^4b^4\ \ldots \}$

Alphabet $\Sigma = \{ a, b \}$

$$\Sigma \cdot \Sigma = \{ a, b \} \cdot \{ a, b \} = \{ aa, ab, ba, bb \}$$

$$aba \cdot \epsilon = aba$$

$$\epsilon \cdot bbb = bbb$$

$$w \cdot \epsilon = w$$

$$\epsilon \cdot w = w$$

$\Sigma^0 = \{ \epsilon \}$ (empty word, with length 0)

$\Sigma^1 =$ words of length 1

$\Sigma^2 =$ words of length 2

$\Sigma^3 =$ words of length 3

$\vdots$

$\Sigma^k =$ words of length k

$\vdots$

$\Sigma^* = \bigcup_{i \geq 0} \Sigma^i$

$=$ set of all finite length words

Regular expressions

Regular expressions

ϵ

Regular expressions

$$\epsilon \mid a \mid b$$

Regular expressions

$$\epsilon \;\mid\; a \;\mid\; b \;\mid r_1\, r_2$$

Regular expressions

$$\epsilon \mid a \mid b \mid r_1\, r_2 \mid r_1 + r_2$$

Regular expressions

$$\epsilon \;\mid\; a \;\mid\; b \;\mid r_1\, r_2 \;\mid r_1 + r_2 \;\mid r^*$$

Regular expressions

$$\epsilon \mid a \mid b \mid r_1\, r_2 \mid r_1 + r_2 \mid r^*$$

where r_1, r_2, r are regular expressions themselves

Regular expressions

$$\epsilon \;\mid\; a \;\mid\; b \;\mid r_1\, r_2 \;\mid r_1 + r_2 \;\mid r^*$$

where r_1, r_2, r are regular expressions themselves

$$a^* + b^*$$
$$ab + bb + baa$$
$$(a+b)^* ab(ba+bb)$$
$$(ab+bb)^*$$
$$\vdots$$

Theorem

1. Every **regular expression can be converted to an NFA** accepting the language of the expression
2. Every **NFA can be converted to a regular expression** describing the language of the NFA

Coming next: Languages over **infinite words**

$$\Sigma = \{ a, b \}$$

$$\Sigma = \{ a,\ b \}$$

Example 1: Infinite word consisting only of a

$$\{ aaaaaaaaaaaaaaaa \dots \}$$

$$\Sigma = \{\, a,\ b \,\}$$

Example 1: Infinite word consisting only of a

$$\{\, aaaaaaaaaaaaaaaa\ldots \,\}$$

Example 2: Infinite words containing only a or only b

$$\{\, aaaaaaaaaaaaaaa\ldots\,,\ \ bbbbbbbbbbbb\ldots \,\}$$

$$\Sigma = \{ a, b \}$$

Example 1: Infinite word consisting only of a

$$\{ aaaaaaaaaaaaaaaa \ldots \}$$

Example 2: Infinite words containing only a or only b

$$\{ aaaaaaaaaaaaaaa \ldots , \; bbbbbbbbbbb \ldots \}$$

Example 3: a word in $aa\Sigma^* aa$ followed by only b-s

$$\{ aaaabbbbbb \ldots , \; aababaabbbbb \ldots , \; aabbbbaabbbbbbb \ldots , \; \ldots \}$$

$$\Sigma = \{ a, b \}$$

Example 1: Infinite word consisting only of a

$$\{ aaaaaaaaaaaaaaa\ldots \}$$

Example 2: Infinite words containing only a or only b

$$\{ aaaaaaaaaaaaaaa\ldots,\ bbbbbbbbbbb\ldots \}$$

Example 3: a word in $aa\Sigma^*aa$ followed by only b-s

$$\{ aaaabbbbbb\ldots,\ aababaabbbbb\ldots,\ aabbbbaabbbbbb\ldots,\ \ldots \}$$

Example 4: Infinite words where b occurs **only finitely often**

$$\{ aaaaaaaaaaaaaaa\ldots,\ baaaaaaaaa\ldots,\ babbaaaaaaaaaa\ldots,\ \ldots \}$$

$$\Sigma = \{ a,\ b \}$$

Example 1: Infinite word consisting only of a

$$\{ aaaaaaaaaaaaaaa\ldots \}$$

Example 2: Infinite words containing only a or only b

$$\{ aaaaaaaaaaaaaaa\ldots,\ bbbbbbbbbbb\ldots \}$$

Example 3: a word in $aa\Sigma^*aa$ followed by only b-s

$$\{ aaaabbbbbb\ldots,\ aababaabbbbb\ldots,\ aabbbbaabbbbbb\ldots,\ \ldots \}$$

Example 4: Infinite words where b occurs **only finitely often**

$$\{ aaaaaaaaaaaaaaa\ldots,\ baaaaaaaaa\ldots,\ babbaaaaaaaaaaa\ldots,\ \ldots \}$$

Example 5: Infinite words where b occurs **infinitely often**

$$\{ ababababab\ldots,\ bbbabbbabbbabbba\ldots,\ bbbbbbbbbbbb\ldots,\ \ldots \}$$

$$\Sigma = \{\, a,\ b \,\}$$

Example 1: Infinite word consisting only of a a^ω

$$\{\ aaaaaaaaaaaaaaa\ldots\ \}$$

Example 2: Infinite words containing only a or only b

$$\{\ aaaaaaaaaaaaaaa\ldots\ ,\ \ bbbbbbbbbbb\ldots\ \}$$

Example 3: a word in $aa\Sigma^*aa$ followed by only b-s

$$\{\ aaaabbbbbbb\ldots\ ,\ \ aababaabbbbbb\ldots\ ,\ \ aabbbbaabbbbbbb\ldots,\ \ \ldots\ \}$$

Example 4: Infinite words where b occurs **only finitely often**

$$\{\ aaaaaaaaaaaaaaaa\ldots\ ,\ \ baaaaaaaaaa\ldots\ ,\ \ babbaaaaaaaaaaaa\ldots,\ \ \ldots\ \}$$

Example 5: Infinite words where b occurs **infinitely often**

$$\{\ abababababab\ldots\ ,\ \ bbbabbbabbbabbba\ldots\ ,\ \ bbbbbbbbbbbbbb\ldots,\ \ \ldots\ \}$$

$$\Sigma = \{ a, b \}$$

Example 1: Infinite word consisting only of a $\quad a^\omega$

$$\{ aaaaaaaaaaaaaaa\dots \}$$

Example 2: Infinite words containing only a or only b $\quad a^\omega + b^\omega$

$$\{ aaaaaaaaaaaaaaa\dots,\ bbbbbbbbbbb\dots \}$$

Example 3: a word in $aa\Sigma^*aa$ followed by only b-s

$$\{ aaaabbbbbbb\dots,\ aababaabbbbbb\dots,\ aabbbbaabbbbbbb\dots,\ \dots \}$$

Example 4: Infinite words where b occurs **only finitely often**

$$\{ aaaaaaaaaaaaaaa\dots,\ baaaaaaaaa\dots,\ babbaaaaaaaaaaa\dots,\ \dots \}$$

Example 5: Infinite words where b occurs **infinitely often**

$$\{ abababababab\dots,\ bbbabbbabbbabbba\dots,\ bbbbbbbbbbbbb\dots,\ \dots \}$$

$$\Sigma = \{ a, b \}$$

Example 1: Infinite word consisting only of a $\quad a^\omega$

$$\{ aaaaaaaaaaaaaaa \ldots \}$$

Example 2: Infinite words containing only a or only b $\quad a^\omega + b^\omega$

$$\{ aaaaaaaaaaaaaaa \ldots , \; bbbbbbbbbbbb \ldots \}$$

Example 3: a word in $aa\Sigma^*aa$ followed by only b-s $\quad aa\Sigma^*aa \cdot b^\omega$

$$\{ aaaabbbbbb \ldots , \; aababaabbbbb \ldots , \; aabbbbaabbbbbbb \ldots, \; \ldots \}$$

Example 4: Infinite words where b occurs **only finitely often**

$$\{ aaaaaaaaaaaaaaa \ldots , \; baaaaaaaaa \ldots , \; babbaaaaaaaaaaa \ldots, \; \ldots \}$$

Example 5: Infinite words where b occurs **infinitely often**

$$\{ ababababab \ldots , \; bbbabbbabbbabbba \ldots , \; bbbbbbbbbbbbb \ldots, \; \ldots \}$$

$$\Sigma = \{ a, b \}$$

Example 1: Infinite word consisting only of a $\quad a^\omega$

$$\{ aaaaaaaaaaaaaaa\ldots \}$$

Example 2: Infinite words containing only a or only b $\quad a^\omega + b^\omega$

$$\{ aaaaaaaaaaaaaaa\ldots,\ bbbbbbbbbbb\ldots \}$$

Example 3: a word in $aa\Sigma^*aa$ followed by only b-s $\quad aa\Sigma^*aa \cdot b^\omega$

$$\{ aaaabbbbbb\ldots,\ aababaabbbbb\ldots,\ aabbbbaabbbbbb\ldots,\ \ldots \}$$

Example 4: Infinite words where b occurs **only finitely often** $(a+b)^* \cdot b^\omega$

$$\{ aaaaaaaaaaaaaaa\ldots,\ baaaaaaaaa\ldots,\ babbaaaaaaaaaaa\ldots,\ \ldots \}$$

Example 5: Infinite words where b occurs **infinitely often**

$$\{ ababababab\ldots,\ bbbabbbabbbabbba\ldots,\ bbbbbbbbbbbb\ldots,\ \ldots \}$$

$$\Sigma = \{ a, b \}$$

Example 1: Infinite word consisting only of a $\quad a^{\omega}$

$$\{ aaaaaaaaaaaaaaa\dots \}$$

Example 2: Infinite words containing only a or only b $\quad a^{\omega} + b^{\omega}$

$$\{ aaaaaaaaaaaaaaa\dots ,\ bbbbbbbbbbb\dots \}$$

Example 3: a word in $aa\Sigma^* aa$ followed by only b-s $\quad aa\Sigma^* aa \cdot b^{\omega}$

$$\{ aaaabbbbbb\dots ,\ aababaabbbbb\dots ,\ aabbbbaabbbbbb\dots ,\ \dots \}$$

Example 4: Infinite words where b occurs **only finitely often** $(a+b)^* \cdot b^{\omega}$

$$\{ aaaaaaaaaaaaaaa\dots ,\ baaaaaaaaa\dots ,\ babbaaaaaaaaaaa\dots ,\ \dots \}$$

Example 5: Infinite words where b occurs **infinitely often** $\quad (a^* b)^{\omega}$

$$\{ ababababab\dots ,\ bbbabbbabbbabbba\dots ,\ bbbbbbbbbbbb\dots ,\ \dots \}$$

ω-regular expressions

$$G = E_1 \cdot F_1^\omega + E_2 \cdot F_2^\omega + \cdots + E_n \cdot F_n^\omega$$

$E_1, \ldots, E_n, F_1, \ldots, F_n$ are **regular expressions** and $\epsilon \notin L(F_i)$ for all $1 \leq i \leq n$

ω-regular expressions

$$G = E_1 \cdot F_1^\omega + E_2 \cdot F_2^\omega + \cdots + E_n \cdot F_n^\omega$$

$E_1, \ldots, E_n, F_1, \ldots, F_n$ are **regular expressions**

and $\epsilon \notin L(F_i)$ for all $1 \leq i \leq n$

$$L(F^\omega) = \{ w_1 w_2 w_3 \ldots \mid \textbf{each } w_i \in L(F) \}$$

More examples

More examples

- $(a+b)^\omega$ set of **all infinite words**

More examples

- $(a+b)^\omega$ set of **all infinite words**
- $a(a+b)^\omega$ infinite words **starting with** an a

More examples

- $(a+b)^\omega$ set of **all infinite words**
- $a(a+b)^\omega$ infinite words **starting with** an a
- $(a+bc+c)^\omega$ words where every b is **immediately followed** by c

More examples

- $(a+b)^\omega$ set of **all infinite words**
- $a(a+b)^\omega$ infinite words **starting with** an a
- $(a+bc+c)^\omega$ words where every b is **immediately followed** by c
- $(a+b)^*c(a+b)^\omega$ words with a **single occurrence** of c

More examples

- $(a+b)^\omega$ set of **all infinite words**
- $a(a+b)^\omega$ infinite words **starting with** an a
- $(a+bc+c)^\omega$ words where every b is **immediately followed** by c
- $(a+b)^*c(a+b)^\omega$ words with a **single occurrence** of c
- $((a+b)^*c)^\omega$ words where c **occurs infinitely often**

$$\mathbf{AP} = \{\ p_1, p_2,\ \cdots, p_k\ \}$$

$$\Sigma = \mathit{PowerSet}(\mathbf{AP}) = \{\ \{\ \}, \{p_1\}, \ \ldots\ , \{p_k\}, \\ \{\ p_1, p_2\ \}, \{\ p_1, p_3\ \}, \ \ldots\ , \{\ p_{k-1}, p_k\ \}, \\ \ldots \\ \{\ p_1, p_2,\ \ldots, p_k\ \}\ \}$$

A property is a **language of infinite words** over alphabet Σ

$$
\begin{aligned}
\mathbf{AP} =\; & \{\ p_1, p_2,\ \cdots\ , p_k\ \} \\
\Sigma\ =\ \mathit{PowerSet}(\mathbf{AP}) =\; & \{\ \{\ \},\ \{p_1\},\ \ldots\ ,\ \{p_k\}, \\
& \quad \{\ p_1, p_2\ \},\ \{\ p_1, p_3\ \},\ \ldots\ ,\ \{\ p_{k-1}, p_k\ \}, \\
& \qquad \ldots \\
& \quad \{\ p_1, p_2,\ \ldots,\ p_k\ \}\ \}
\end{aligned}
$$

A property is a **language of infinite words** over alphabet Σ

The property is ω-regular if it can be **described by an ω-regular expression**

$$\mathbf{AP} = \{\ \text{wait, crit}\ \}$$

$$\Sigma = \mathit{PowerSet}(\mathbf{AP}) = \{\ \{\ \},\ \{\text{wait}\},\ \{\text{crit}\}\ ,\ \{\text{wait, crit}\}\ \}$$

$$\mathbf{AP} = \{\ \text{wait, crit}\ \}$$

$$\Sigma = \mathit{PowerSet}(\mathbf{AP}) = \{\ \{\ \},\ \{\text{wait}\},\ \{\text{crit}\}\ ,\ \{\text{wait, crit}\}\ \}$$

Property: Process enters critical section infinitely often

$$\mathbf{AP} = \{ \text{wait}, \text{crit} \}$$

$$\Sigma = \mathit{PowerSet}(\mathbf{AP}) = \{ \{ \}, \{\text{wait}\}, \{\text{crit}\}, \{\text{wait}, \text{crit}\} \}$$

Property: Process enters critical section infinitely often

$$((\{ \} + \{\textbf{wait}\})^* (\{\textbf{crit}\} + \{\textbf{wait, crit}\}))^\omega$$

ω-regular properties

ω-regular expressions

Next goal: Find algorithms to model-check ω-regular properties

Unit-5: ω-regular properties

B. Srivathsan

Chennai Mathematical Institute

NPTEL-course

July - November 2015

Module 3:

Büchi automata

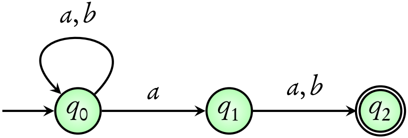
Goal

- Give some kind of an **automaton** for ω-regular expressions
- Take **synchronous product** with the transition system of the model
- Check **emptiness** of this automaton

Goal

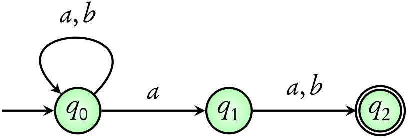
- Give some kind of an **automaton** for ω-regular expressions
- Take **synchronous product** with the transition system of the model
- Check **emptiness** of this automaton

Coming next: A short recap of **finite automata**



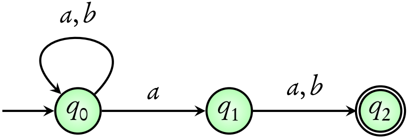
a, b
a
a, b
q_0
q_1
q_2

$a \quad b \quad b \quad a \quad a \quad b \quad a \quad b$



$a \quad b \quad b \quad a \quad a \quad b \quad a \quad b$

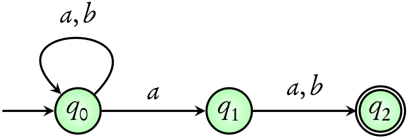
Runs:



$$a \quad b \quad b \quad a \quad a \quad b \quad a \quad b$$

Runs:

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0$$

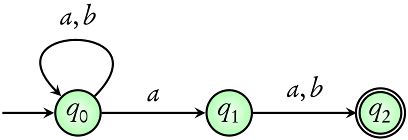


$$a \quad b \quad b \quad a \quad a \quad b \quad a \quad b$$

Runs:

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0$$

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2$$

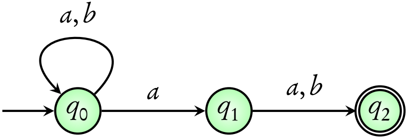


$$a \quad b \quad b \quad a \quad a \quad b \quad a \quad b$$

Runs:

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0$$

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \quad \text{accepting run}$$

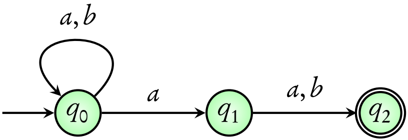


$a \quad b \quad b \quad a \quad a \quad b \quad a \quad b$

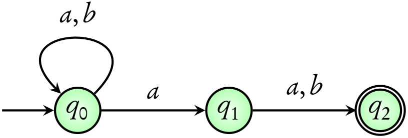
Runs:

$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0$

$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2$ accepting run

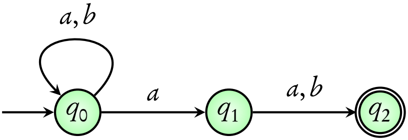


Language: set of words for which **there exists** an accepting run



Language: set of words for which **there exists** an accepting run

$a \quad b \quad b \quad b \quad a$

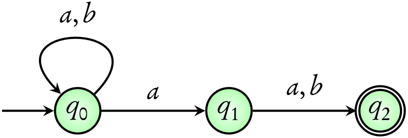


Language: set of words for which **there exists** an accepting run

$$a \quad b \quad b \quad b \quad a$$

Runs:

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0$$

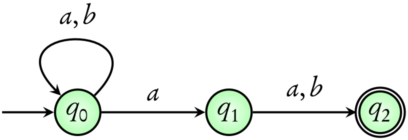


Language: set of words for which **there exists** an accepting run

$a \quad b \quad b \quad b \quad a$

Runs:

$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0$ Not accepted



Language: set of words for which **there exists** an accepting run

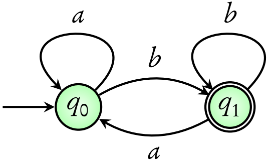
In finite words, there is an **end**

A run is accepting if it **ends in an accepting state**

In finite words, there is an **end**

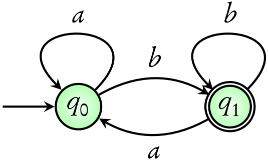
A run is accepting if it **ends in an accepting state**

How do we define **accepting runs** for **infinite words**?



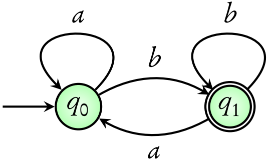
a
b
b
q_0
q_1
a

$a\ b\ a\ b\ a\ a\ b\ b\ b\ b\ b\ b\ \ldots$



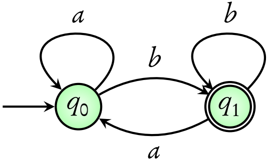
$$a\ b\ a\ b\ a\ a\ b\ b\ b\ b\ b\ b\ \dots$$

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \cdots$$



$$a\ b\ a\ b\ a\ a\ b\ b\ b\ b\ b\ b\ \ldots$$

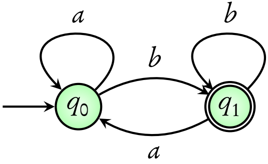
$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \cdots$$



Run is accepting if **some accepting state occurs infinitely often**

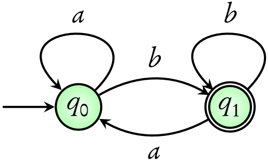
$$a\ b\ a\ b\ a\ a\ b\ b\ b\ b\ b\ b\ \ldots$$

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \cdots$$



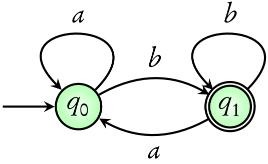
Above word is accepted by this automaton

Run is accepting if **some accepting state occurs infinitely often**



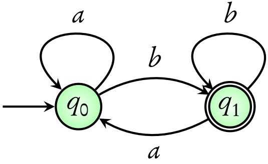
a
b
b
q_0
q_1
a

$a\ b\ a\ b\ a\ b\ a\ b\ a\ b\ a\ b\ \ldots$



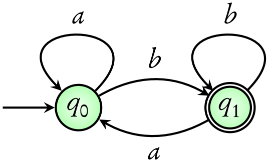
$$a\ b\ a\ b\ a\ b\ a\ b\ a\ b\ a\ b\ \ldots$$

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \cdots$$



$$a \; b \; a \; b \; a \; b \; a \; b \; a \; b \; a \; b \; \ldots$$

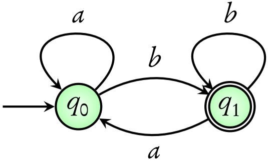
$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \cdots$$



Run is accepting if **some accepting state occurs infinitely often**

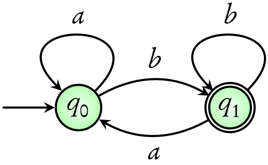
$$a\ b\ a\ b\ a\ b\ a\ b\ a\ b\ a\ b\ \ldots$$

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \cdots$$

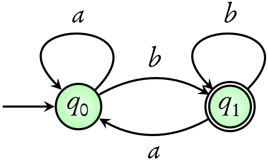


Above word is accepted by this automaton

Run is accepting if **some accepting state occurs infinitely often**

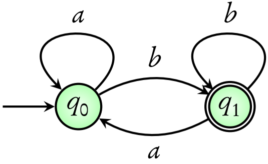

a
b
b
q_0
q_1
a

$a\ b\ a\ b\ a\ a\ a\ a\ a\ a\ a\ a\ \ldots$



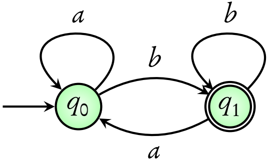
$$a\ b\ a\ b\ a\ a\ a\ a\ a\ a\ a\ a\ \ldots$$

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \cdots$$



$$a\ b\ a\ b\ a\ a\ a\ a\ a\ a\ a\ a\ \ldots$$

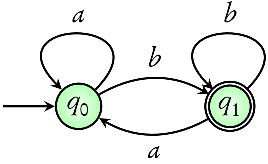
$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \cdots$$



Run is accepting if **some accepting state occurs infinitely often**

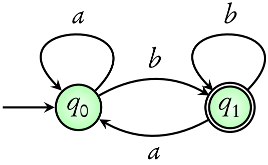
$a\ b\ a\ b\ a\ a\ a\ a\ a\ a\ a\ a\ \ldots$

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \cdots$$



Above word is **not accepted** by this automaton

Run is accepting if **some accepting state occurs infinitely often**



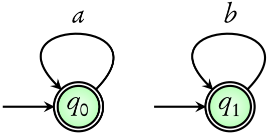
Language: set of infinite words which contain **infinitely many** b-s

Non-deterministic Büchi Automata

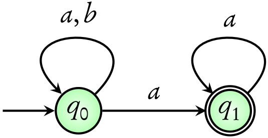
- States, transitions, initial and accepting states like an NFA
- Difference in accepting condition

Word is accepted if it has a run in which **some accepting state occurs infinitely often**

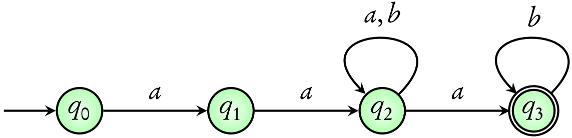
Example: $a^\omega + b^\omega$



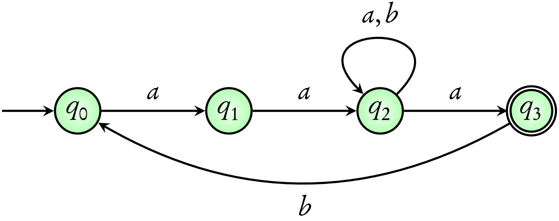
Example: $(a + b)^* a^\omega$



Example: $aa(a+b)^*ab^\omega$



Example: $(aa(a+b)^*ab)^\omega$



Non-deterministic Büchi Automaton

Accepting state occurs infinitely often

Unit-5: ω-regular properties

B. Srivathsan

Chennai Mathematical Institute

NPTEL-course

July - November 2015

Module 4:

Simple properties of NBA

Determinization

Product construction

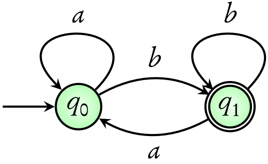
Emptiness

Complementation

Union

Deterministic Büchi Automata

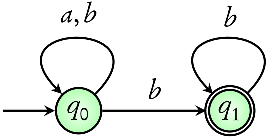
Words where b occurs infinitely often



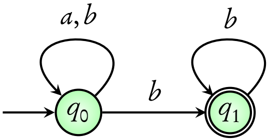
- Single initial state
- From every state - on an alphabet, there is a **unique transition**

Question: Can every NBA be converted to an **equivalent** DBA?

$(a+b)^*b^\omega$: a occurs only finitely often



$(a+b)^* b^\omega$: a occurs only finitely often



- Automaton has to **guess** the point from where only b occurs
- A deterministic Büchi automaton cannot make this guess

Unit-5: ω-regular properties

B. Srivathsan

Chennai Mathematical Institute

NPTEL-course

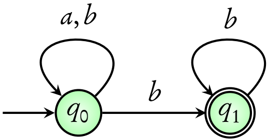
July - November 2015

Summary

- ω-regular expressions for specifying properties
- Non-deterministic Büchi automata
- Properties of NBA

Important concepts: DBA not equivalent to NBA, product construction for NBA, complementation and union

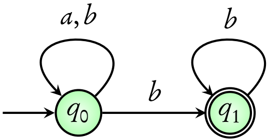
$(a+b)^*b^\omega$: a occurs only finitely often



- Automaton has to **guess** the point from where only b occurs
- A deterministic Büchi automaton cannot make this guess

The above language **cannot be** accepted by a DBA

$(a+b)^* b^\omega$: a occurs only finitely often



- Automaton has to **guess** the point from where only b occurs
- A deterministic Büchi automaton cannot make this guess

The above language **cannot be** accepted by a DBA

Theorem 4.50 (Page 190) of *Principles of Model Checking*, Baier and Katoen. MIT Press (2008)

Determinization

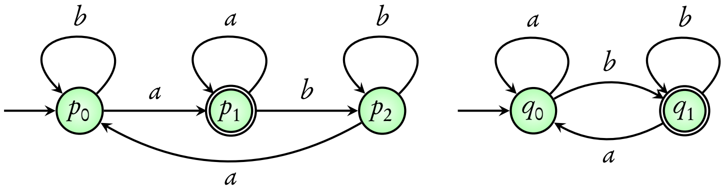
DBA less powerful than NBA

Product construction

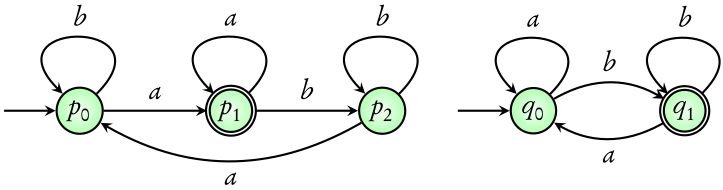
Emptiness

Complementation

Union

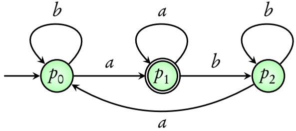


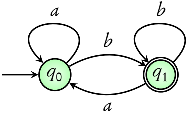
Word $(ab)^\omega$ is **accepted by both automata**

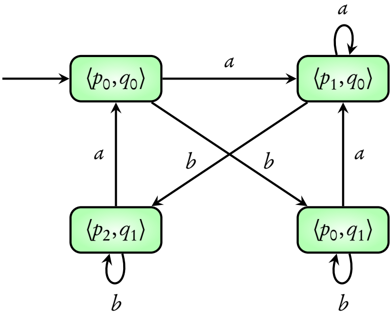


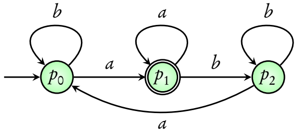
Word $(ab)^\omega$ is **accepted by both automata**

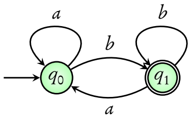
Coming next: The synchronous product construction

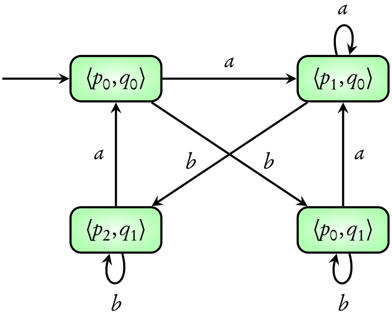

b
a
b
a
b
p_0
p_1
p_2
a


a
b
b
q_0
q_1
a

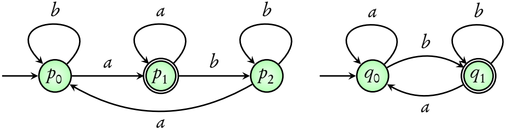

a
a
$\langle p_0, q_0 \rangle$
$\langle p_1, q_0 \rangle$
a
b
b
a
$\langle p_2, q_1 \rangle$
$\langle p_0, q_1 \rangle$
b
b

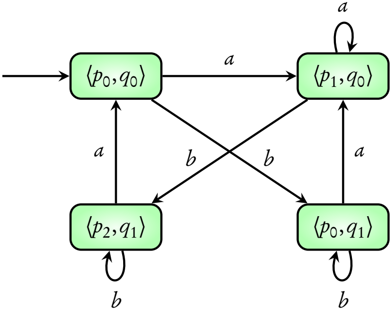






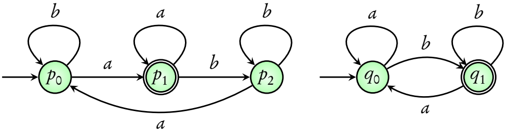
$\langle p_1, q_1 \rangle$ is not present

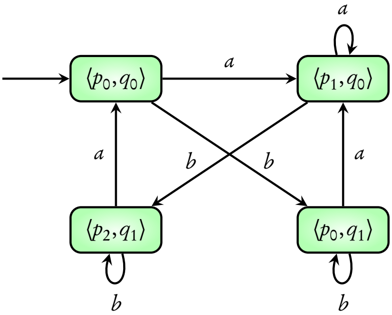




$\langle p_1, q_1\rangle$ is not present

No accepting state!



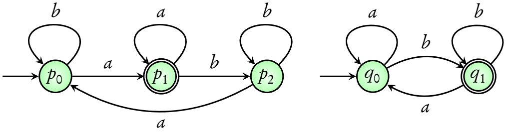


$\langle p_1, q_1 \rangle$ is not present

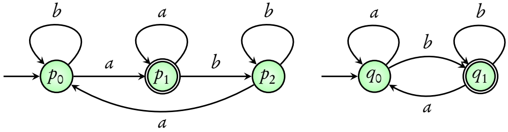
No accepting state!

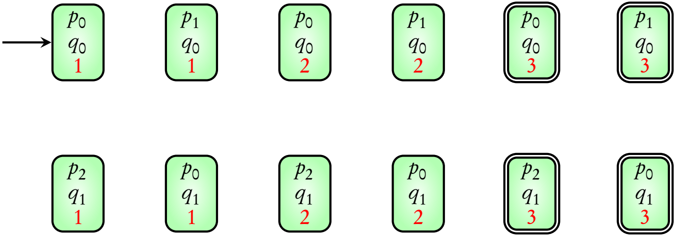
But intersection of the two automata is **not empty**

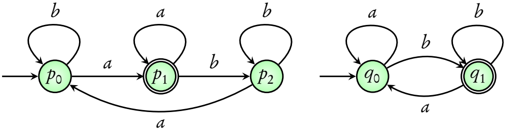
- Need to **modify** the product construction
- **Track** accepting states of **both automata**
- Ensure that **both** automata visit **accepting states infinitely often**

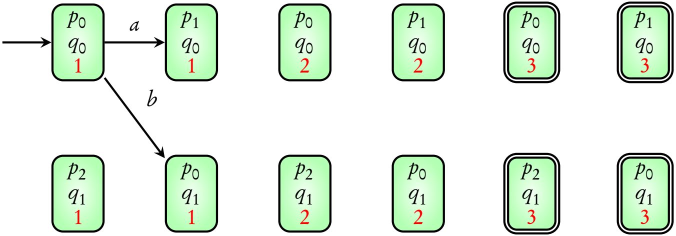


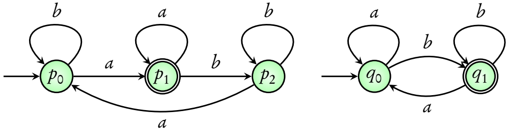
b
a
b
a
b
p0
p1
p2
a
a
b
b
q0
q1
a


b
a
b
a
b
b
p_0
p_1
p_2
q_0
q_1
a
a

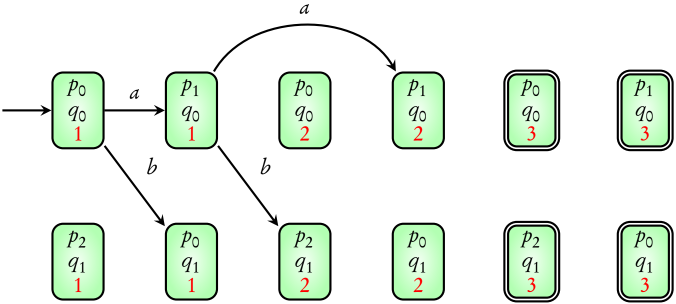

p_0 q_0 1
p_1 q_0 1
p_0 q_0 2
p_1 q_0 2
p_0 q_0 3
p_1 q_0 3
p_2 q_1 1
p_0 q_1 1
p_2 q_1 2
p_0 q_1 2
p_2 q_1 3
p_0 q_1 3


p_0
p_1
p_2
q_0
q_1
a
b

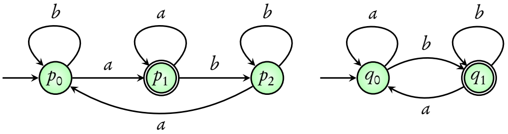

p_0 q_0 1
p_1 q_0 1
p_0 q_0 2
p_1 q_0 2
p_0 q_0 3
p_1 q_0 3
p_2 q_1 1
p_0 q_1 1
p_2 q_1 2
p_0 q_1 2
p_2 q_1 3
p_0 q_1 3
a
b

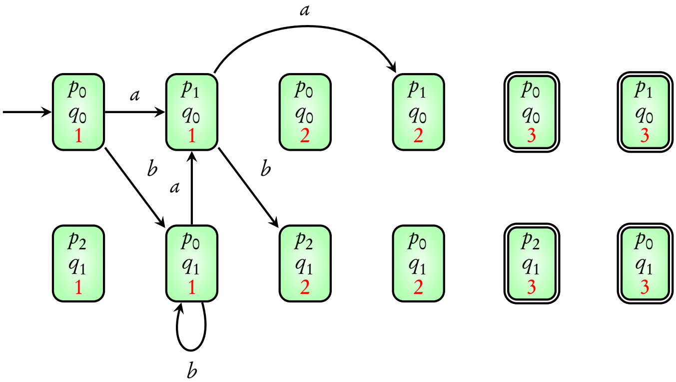


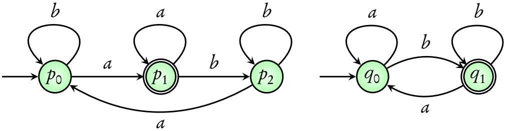
p_0 p_1 p_2 a b a b a b a
q_0 q_1 a b b a

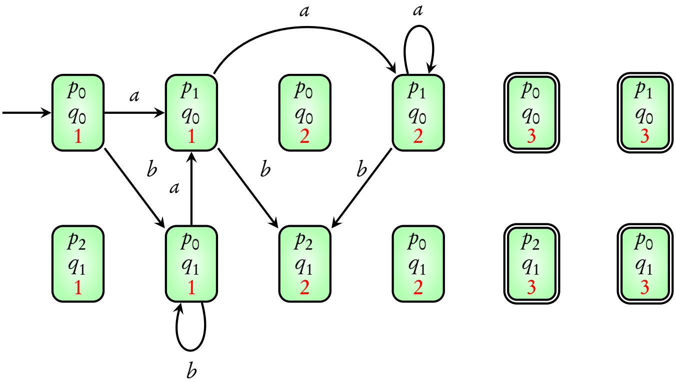


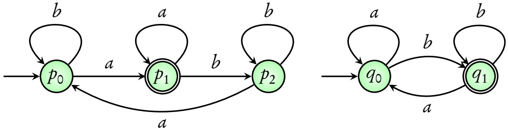
p_0 q_0 1
p_1 q_0 1
p_0 q_0 2
p_1 q_0 2
p_0 q_0 3
p_1 q_0 3
p_2 q_1 1
p_0 q_1 1
p_2 q_1 2
p_0 q_1 2
p_2 q_1 3
p_0 q_1 3
a a b b

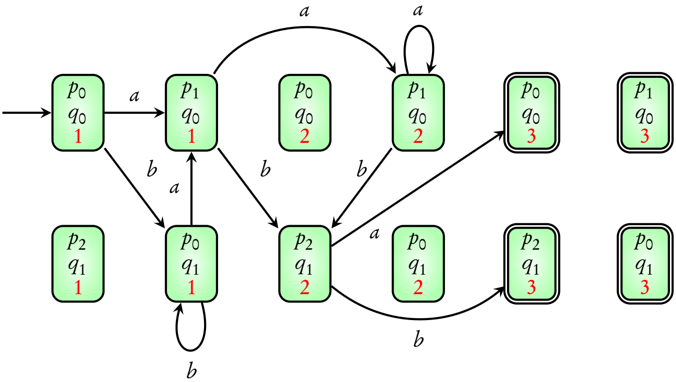

b
a
b
a
b
b
a
b
p_0
p_1
p_2
q_0
q_1
a
a

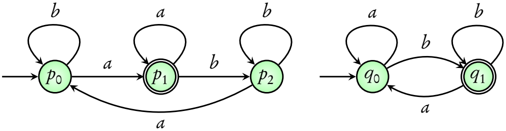

a
a
b
b
a
b
p_0 q_0 1
p_1 q_0 1
p_0 q_0 2
p_1 q_0 2
p_0 q_0 3
p_1 q_0 3
p_2 q_1 1
p_0 q_1 1
p_2 q_1 2
p_0 q_1 2
p_2 q_1 3
p_0 q_1 3

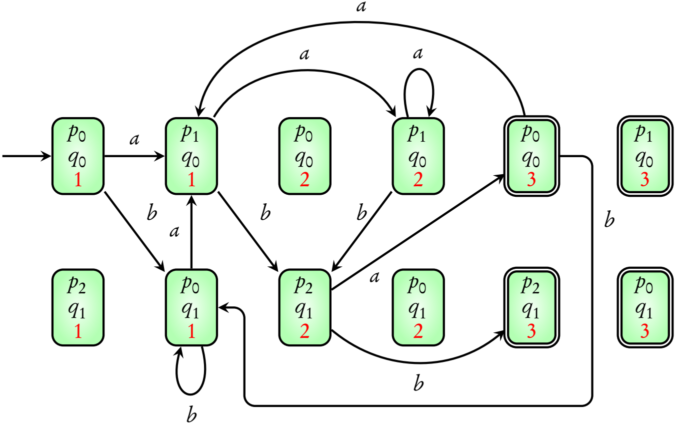

b
a
b
a
b
a
b
p_0
p_1
p_2
a
b
a
b
q_0
q_1

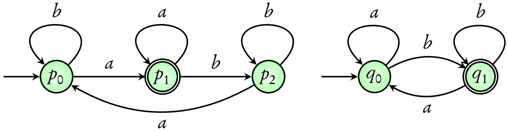

a
a
a
b
a
b
b
b
p_0 q_0 1
p_1 q_0 1
p_0 q_0 2
p_1 q_0 2
p_0 q_0 3
p_1 q_0 3
p_2 q_1 1
p_0 q_1 1
p_2 q_1 2
p_0 q_1 2
p_2 q_1 3
p_0 q_1 3


b
a
b
a
b
b
a
b
p_0
p_1
p_2
q_0
q_1
a
a

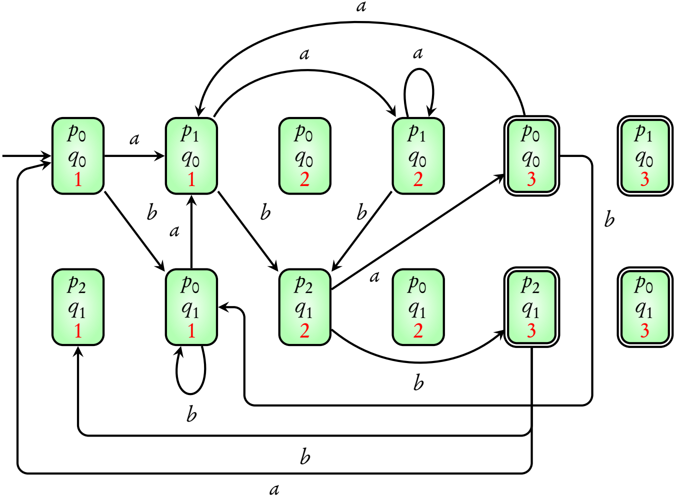

a
a
a
p_0 q_0 1
p_1 q_0 1
p_0 q_0 2
p_1 q_0 2
p_0 q_0 3
p_1 q_0 3
b
a
b
b
a
p_2 q_1 1
p_0 q_1 1
p_2 q_1 2
p_0 q_1 2
p_2 q_1 3
p_0 q_1 3
b
b


p_0
p_1
p_2
q_0
q_1
a
b

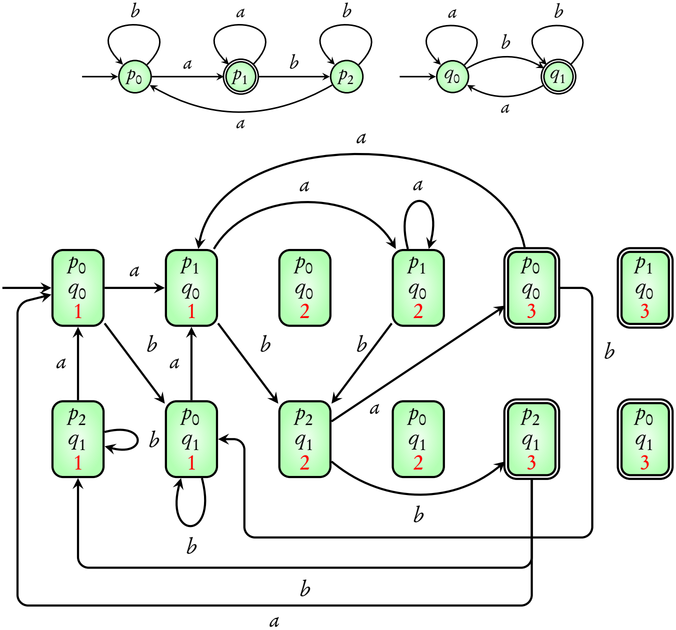

p_0 q_0 1
p_1 q_0 1
p_0 q_0 2
p_1 q_0 2
p_0 q_0 3
p_1 q_0 3
p_2 q_1 1
p_0 q_1 1
p_2 q_1 2
p_0 q_1 2
p_2 q_1 3
p_0 q_1 3
a
b



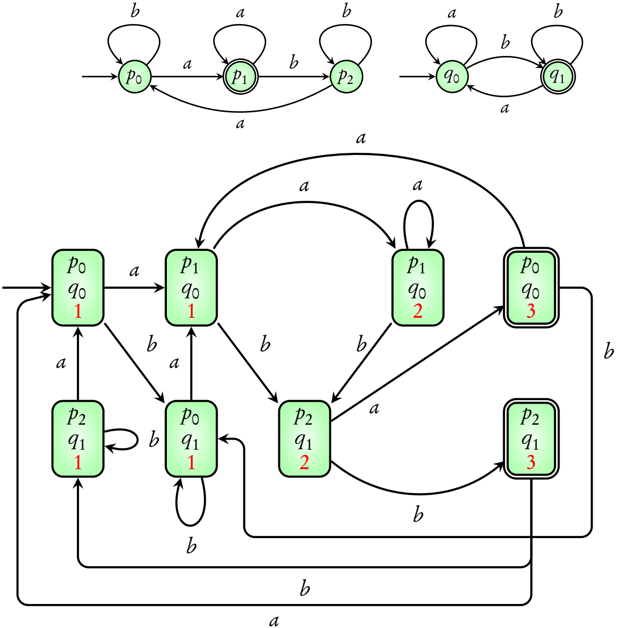
p_0
p_1
p_2
q_0
q_1
a
b



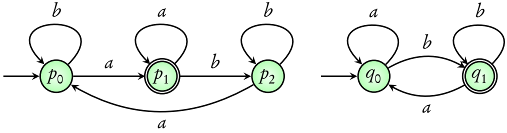
p_0 q_0 1
p_1 q_0 1
p_0 q_0 2
p_1 q_0 2
p_0 q_0 3
p_1 q_0 3
p_2 q_1 1
p_0 q_1 1
p_2 q_1 2
p_0 q_1 2
p_2 q_1 3
p_0 q_1 3
a
b

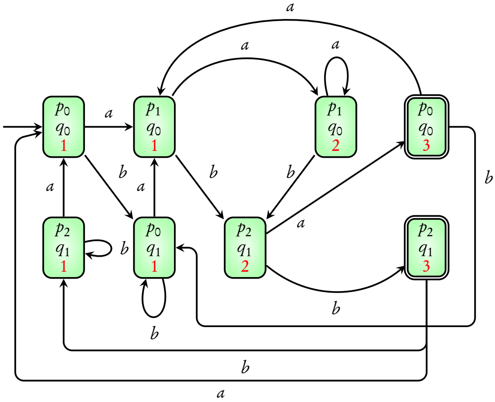


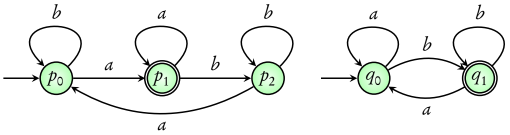
p_0
p_1
p_2
q_0
q_1
a
b
1
2
3

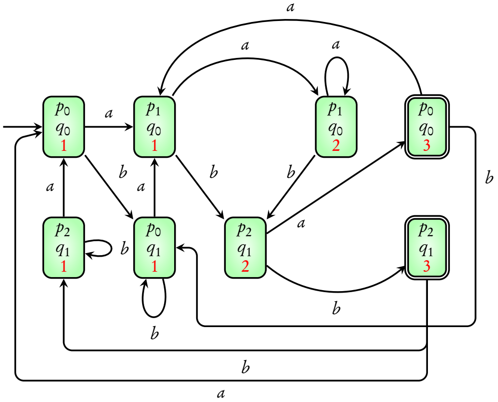


p_0 p_1 p_2
q_0 q_1
a b
p_0 q_0 1
p_1 q_0 1
p_1 q_0 2
p_0 q_0 3
p_2 q_1 1
p_0 q_1 1
p_2 q_1 2
p_2 q_1 3

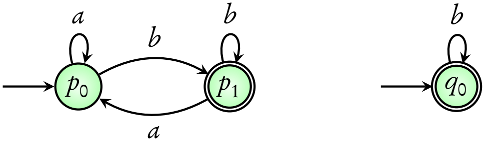

a
b
p0
p1
p2
q0
q1

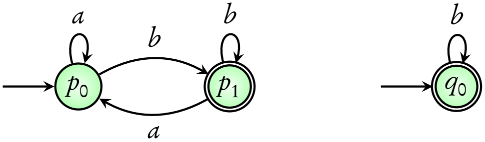

a
b
p0 q0 1
p1 q0 1
p1 q0 2
p0 q0 3
p2 q1 1
p0 q1 1
p2 q1 2
p2 q1 3

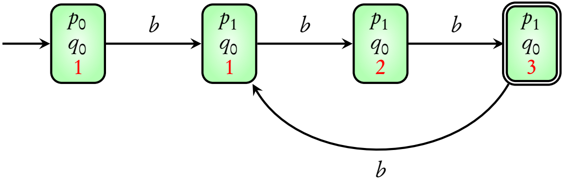




Word is accepted by product ⟷ it is accepted by both component automata


a
b
b
a
p_0
p_1
b
q_0


a
b
b
p_0
p_1
a
b
q_0


p_0 q_0 1
b
p_1 q_0 1
b
p_1 q_0 2
b
p_1 q_0 3
b

Determinization

DBA less powerful than NBA

Product construction

Language intersection

Emptiness

Complementation

Union

Determinization

DBA less powerful than NBA

Product construction

Language intersection

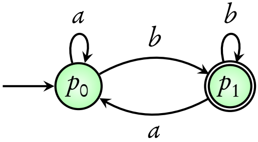
Emptiness

Next unit ...

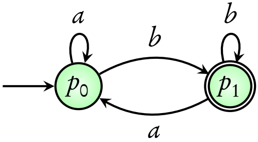
Complementation

Union

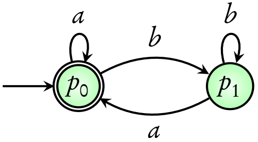
Language: b occurs infinitely often



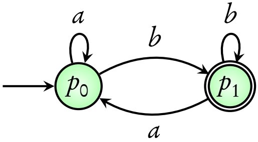
Language: b occurs infinitely often



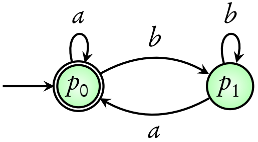
Language: a occurs infinitely often



Language: b occurs infinitely often



Language: a occurs infinitely often



Not the complement!

$(ab)^\omega$ present in both

Challenges

- Mere interchange of accepting states does not work

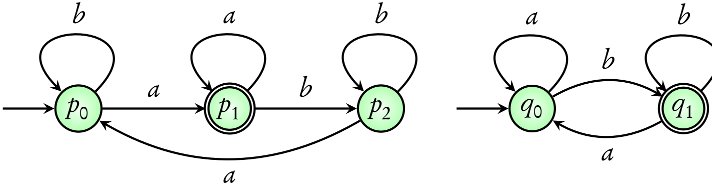
- Moreoever, NBA are more expressive than DBA

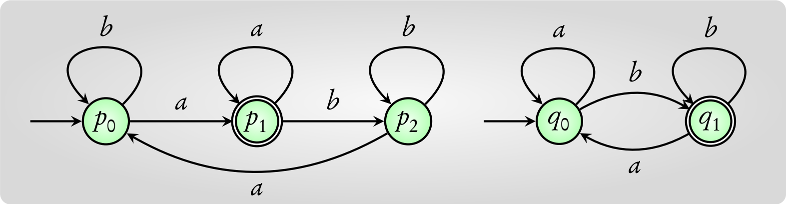
Complementation

Theorem

Given an NBA $\mathscr{A}$, there is an algorithm to compute the NBA accepting the complement language $\mathscr{L}(\mathscr{A})^c$

Proof out of scope of this course


b
a
b
a
b
p0
p1
p2
a
a
b
b
a
q0
q1
a



For **union**, take the disjoint union of the two NBA

Determinization

DBA less powerful than NBA

Product construction

Language intersection

Emptiness

Next unit ...

Complementation

Union