Some questions

(a) Is there an algorithm which, given a string u and a regular expression r, computes whether or not u matches r?

(b) In formulating the definition of regular expressions, have we missed out some practically useful notions of pattern?

(c) Is there an algorithm which, given two regular expressions r and s, computes whether or not they are **equivalent**, in the sense that $L(r)$ and $L(s)$ are equal sets?

(d) Is every language (subset of Σ^*) of the form $L(r)$ for some r?

Equivalent regular expressions

Definition. Two regular expressions r and s are said to be **equivalent** if $L(r) = L(s)$, that is, they determine exactly the same sets of strings via matching.

For example, are $b^*a(b^*a)^*$ and $(a|b)^*a$ equivalent?

Equivalent regular expressions

Definition. Two regular expressions r and s are said to be **equivalent** if $L(r) = L(s)$, that is, they determine exactly the same sets of strings via matching.

For example, are $b^* a (b^* a)^*$ and $(a|b)^* a$ equivalent?

Answer: yes (Exercise 2.3)

How can we decide all such questions?

Note that $L(r) = L(s)$

iff $L(r) \subseteq L(s)$ and $L(s) \subseteq L(r)$

Note that $L(r) = L(s)$

iff $L(r) \subseteq L(s)$ and $L(s) \subseteq L(r)$
iff $(\Sigma^* \setminus L(r)) \cap L(s) = \emptyset = (\Sigma^* \setminus L(s)) \cap L(r)$

Note that $L(r) = L(s)$

iff $L(r) \subseteq L(s)$ and $L(s) \subseteq L(r)$
iff $(\Sigma^* \setminus L(r)) \cap L(s) = \emptyset = (\Sigma^* \setminus L(s)) \cap L(r)$
iff $L((\sim r) \,\&\, s) = \emptyset = L((\sim s) \,\&\, r)$

Note that $L(r) = L(s)$

$$\begin{aligned}
&\text{iff } L(r) \subseteq L(s) \text{ and } L(s) \subseteq L(r) \\
&\text{iff } (\Sigma^* \setminus L(r)) \cap L(s) = \emptyset = (\Sigma^* \setminus L(s)) \cap L(r) \\
&\text{iff } L((\sim r) \,\&\, s) = \emptyset = L((\sim s) \,\&\, r) \\
&\text{iff } L(M) = \emptyset = L(N)
\end{aligned}$$

where M and N are DFAs accepting the sets of strings matched by the regular expressions $(\sim r) \,\&\, s$ and $(\sim s) \,\&\, r$ respectively.

Note that $L(r) = L(s)$

$$\begin{aligned}
&\text{iff } L(r) \subseteq L(s) \text{ and } L(s) \subseteq L(r) \\
&\text{iff } (\Sigma^* \setminus L(r)) \cap L(s) = \emptyset = (\Sigma^* \setminus L(s)) \cap L(r) \\
&\text{iff } L((\sim r) \,\&\, s) = \emptyset = L((\sim s) \,\&\, r) \\
&\text{iff } L(M) = \emptyset = L(N)
\end{aligned}$$

where M and N are DFAs accepting the sets of strings matched by the regular expressions $(\sim r) \,\&\, s$ and $(\sim s) \,\&\, r$ respectively.

So to decide equivalence for regular expressions it suffices to

check, given any given DFA M, whether or not it accepts some string.

Note that the number of transitions needed to reach an accepting state in a finite automaton is bounded by the number of states (we can remove loops from longer paths). So we only have to check finitely many strings to see whether or not $L(M)$ is empty.

The Pumping Lemma

Some questions

(a) Is there an algorithm which, given a string u and a regular expression r, computes whether or not u matches r?

(b) In formulating the definition of regular expressions, have we missed out some practically useful notions of pattern?

(c) Is there an algorithm which, given two regular expressions r and s, computes whether or not they are **equivalent**, in the sense that $L(r)$ and $L(s)$ are equal sets?

(d) Is every language (subset of Σ^*) of the form $L(r)$ for some r?

Examples of languages that are not regular

- The set of strings over $\{(,),a,b,\ldots,z\}$ in which the parentheses '$($' and '$)$' occur well-nested.
- The set of strings over $\{a,b,\ldots,z\}$ which are palindromes, i.e. which read the same backwards as forwards.
- $\{a^n b^n \mid n \geq 0\}$

The Pumping Lemma

For every regular language L, there is a number $\ell \geq 1$ satisfying the **pumping lemma property**:

All $w \in L$ with $|w| \geq \ell$ can be expressed as a concatenation of three strings, $w = u_1 v u_2$, where u_1, v and u_2 satisfy:

- $|v| \geq 1$
  (i.e. $v \neq \varepsilon$)
- $|u_1 v| \leq \ell$
- for all $n \geq 0$, $u_1 v^n u_2 \in L$
  (i.e. $u_1 u_2 \in L$, $u_1 v u_2 \in L$ [but we knew that anyway], $u_1 v v u_2 \in L$, $u_1 v v v u_2 \in L$, etc.)

Suppose $L = L(M)$ for a DFA $M = (Q, \Sigma, \delta, s, F)$. Taking ℓ to be the number of elements in Q, if $n \geq \ell$, then in

$$s = \underbrace{q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \cdots \xrightarrow{a_\ell} q_\ell}_{\ell+1 \text{ states}} \cdots \xrightarrow{a_n} q_n \in F$$

$q_0, \ldots, q_\ell$ can't all be distinct states. So $q_i = q_j$ for some $0 \leq i < j \leq \ell$. So the above transition sequence looks like

$$s \;=\; q_0 \xrightarrow{u_1}{}^{*} q_i \;=\; q_j \xrightarrow{u_2}{}^{*} q_n \;\in\; F \qquad (q_i \xrightarrow{v}{}^{*} q_j)$$

where

$$u_1 \triangleq a_1 \ldots a_i \qquad v \triangleq a_{i+1} \ldots a_j \qquad u_2 \triangleq a_{j+1} \ldots a_n$$

How to use the Pumping Lemma to prove that a language L is *not* regular

For each $\ell \geq 1$, find some $w \in L$ of length $\geq \ell$ so that

$$\left.\begin{array}{l}\text{no matter how } w \text{ is split into three, } w = u_1 v u_2,\\ \text{with } |u_1 v| \leq \ell \text{ and } |v| \geq 1, \text{ there is some } n \geq 0\\ \text{for which } u_1 v^n u_2 \text{ is } \textit{not} \text{ in } L\end{array}\right\} \quad (\dagger)$$

Examples

None of the following three languages are regular:

(i) $L_1 \triangleq \{a^n b^n \mid n \geq 0\}$

[For each $\ell \geq 1$, $a^\ell b^\ell \in L_1$ is of length $\geq \ell$ and has property (†) on Slide 104.]

(ii) $L_2 \triangleq \{w \in \{a, b\}^* \mid w \text{ a palindrome}\}$

[For each $\ell \geq 1$, $a^\ell b a^\ell \in L_1$ is of length $\geq \ell$ and has property (†).]

(iii) $L_3 \triangleq \{a^p \mid p \text{ prime}\}$

[For each $\ell \geq 1$, we can find a prime p with $p > 2\ell$ and then $a^p \in L_3$ has length $\geq \ell$ and has property (†).]

Example (i) on p 104

$\boxed{L_1 = \{a^n b^n \mid n \geq 0\}}$

For each $\ell \geq 1$, take $w = a^\ell b^\ell \in L_1$.

Example (i) on p 104

$L_1 = \{a^n b^n \mid n \geq 0\}$

For each $\ell \geq 1$, take $w = a^\ell b^\ell \in L_1$.

If $w = u_1 v u_2$ with $|u_1 v| \leq \ell$ & $|v| \geq 1$

then
$$\begin{cases} u_1 = a^r \\ v = a^s \\ u_2 = a^{\ell - r - s} b^\ell \end{cases}$$

Example (i) on p 104

$L_1 = \{a^n b^n \mid n \geq 0\}$

For each $\ell \geq 1$ take $w = a^\ell b^\ell \in L_1$

If $w = u_1 v u_2$ with $|u_1 v| \leq \ell$ & $|v| \geq 1$

then
$$\begin{cases} u_1 = a^r \\ v = a^s \\ u_2 = a^{\ell-r-s} b^\ell \end{cases}$$
for some r, s with $r+s \leq \ell$ & $s \geq 1$

Example (i) on p 104

$L_1 = \{a^n b^n \mid n \geq 0\}$

For each $\ell \geq 1$ take $w = a^\ell b^\ell \in L_1$

If $w = u_1 v u_2$ with $|u_1 v| \leq \ell$ & $|v| \geq 1$

then
$$\begin{cases} u_1 = a^r \\ v = a^s \\ u_2 = a^{\ell-r-s} b^\ell \end{cases}$$
for some r, s with $r+s \leq \ell$ & $s \geq 1$

So $u_1 v^0 u_2 = a^r \varepsilon a^{\ell-r-s} b^\ell = a^{\ell-s} b^\ell$

Example (i) on p 104

$L_1 = \{a^n b^n \mid n \geq 0\}$

for each $\ell \geq 1$ take $w = a^\ell b^\ell \in L_1$

If $w = u_1 v u_2$ with $|u_1 v| \leq \ell$ & $|v| \geq 1$

then $\begin{cases} u_1 = a^r \\ v = a^s \\ u_2 = a^{\ell - r - s} b^\ell \end{cases}$ for some r, s with $r + s \leq \ell$ & $s \geq 1$

so $u_1 v^0 u_2 = a^{\ell - s} b^\ell \notin L_1$ 'cos $\ell - s \neq \ell$

Example (iii) on p 104

$$\boxed{L_2 = \{ a^p \mid p \text{ prime} \}}$$

For each $\ell \geq 1$, take $w = a^p \in L_2$

where p prime & $p > 2\ell$

Example (iii) on p 104

$\boxed{L_2 = \{ a^p \mid p \text{ prime} \}}$

For each $\ell \geq 1$, take $w = a^p \in L_2$

where p prime & $p > 2\ell$

If $w = u_1 v u_2$ with ...

then $u_1 = a^r$ $\quad v = a^s$ $\quad u_2 = a^{p-r-s}$

with $s \geq 1$ & $r + s \leq \ell$

Example (iii) on p 104

$\boxed{L_2 = \{ a^p \mid p \text{ prime} \}}$

For each $\ell \geq 1$, take $w = a^p \in L_2$

where p prime & $p > 2\ell$

If $w = u_1 v u_2$ with ...

then $u_1 = a^r$ $\quad v = a^s$ $\quad u_2 = a^{p-r-s}$

with $s \geq 1$ & $r+s \leq \ell$

Then $u_1 v^{p-s} u_2 = a^r a^{s(p-s)} a^{p-r-s}$

Example (iii) on p 104

$$\boxed{L_2 = \{a^p \mid p \text{ prime}\}}$$

For each $\ell \geq 1$, take $w = a^p \in L_2$

where p prime & $p > 2\ell$

If $w = u_1 v u_2$ with ...

then $u_1 = a^r$ $\quad v = a^s$ $\quad u_2 = a^{p-r-s}$

with $s \geq 1$ & $r + s \leq \ell$

Then $u_1 v^{p-s} u_2 = a^{(p-s)(s+1)}$

Example (iii) on p 104

$\boxed{L_2 = \{ a^p \mid p \text{ prime} \}}$

For each $\ell \geq 1$, take $w = a^p \in L_2$

where p prime & $p > 2\ell$

If $w = u_1 v u_2$ with ...

then $u_1 = a^r$ $\quad v = a^s$ $\quad u_2 = a^{p-r-s}$

with $s \geq 1$ & $r + s \leq \ell$

Then $u_1 v^{p-s} u_2 = a^{(p-s)(s+1)} \notin L_2$

'cos $s+1 \geq 2$ & $p - s > 2\ell - \ell \geq 1$

Example of a non-regular language with the pumping lemma property

$$L \triangleq \{c^m a^n b^n \mid m \geq 1 \,\&\, n \geq 0\} \cup \{a^m b^n \mid m, n \geq 0\}$$

satisfies the pumping lemma property on Slide 101 with $\ell = 1$.

[For any $w \in L$ of length ≥ 1, can take $u_1 = \varepsilon$, $v =$ first letter of w, $u_2 =$ rest of w.]

But L is not regular – see Exercise 5.1.

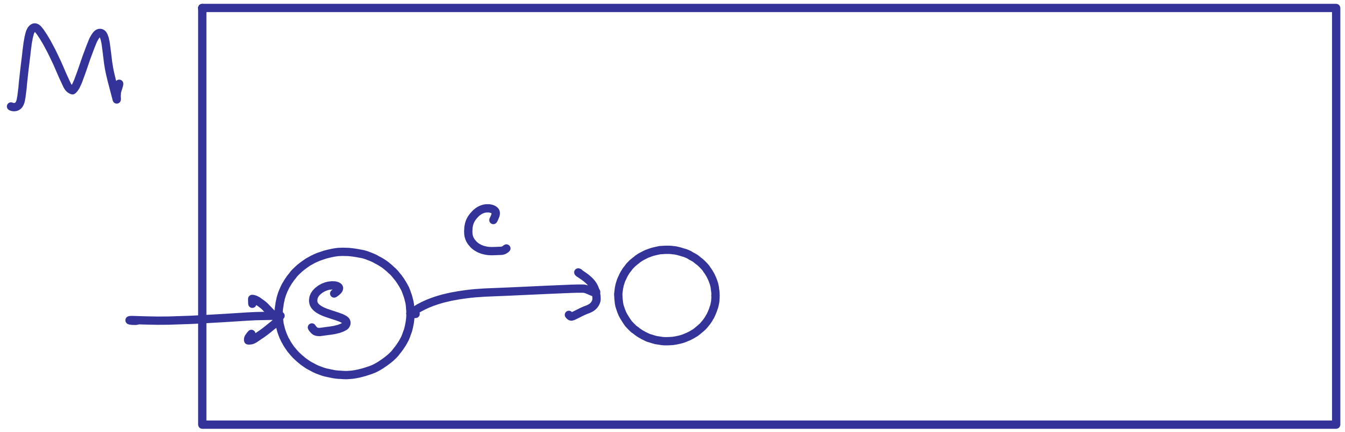
$\boxed{L \text{ on p107 is } \underline{\text{not}} \text{ regular.}}$

Suppose $L = L(M)$ for some DFA $M = (Q, \Sigma, \delta, s, F)$ & derive a contradiction ...

$\boxed{L \text{ on p107 is } \underline{\text{not}} \text{ regular.}}$

Suppose $L = L(M)$ for some DFA $M = (Q, \Sigma, \delta, s, F)$ & derive a contradiction ...

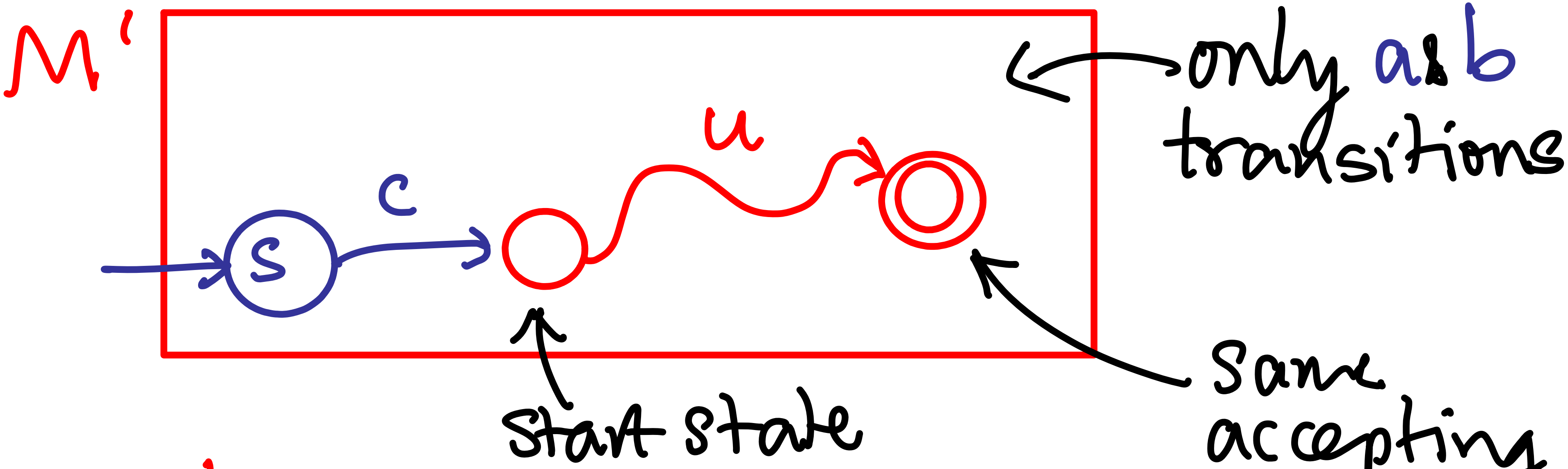
Define an NFA M' from M



$\boxed{L \text{ on p107 is } \underline{\text{not}} \text{ regular.}}$

Suppose $L = L(M)$ for some DFA $M = (Q, \Sigma, \delta, s, F)$
& derive a contradiction ...

Define an NFA M' from M

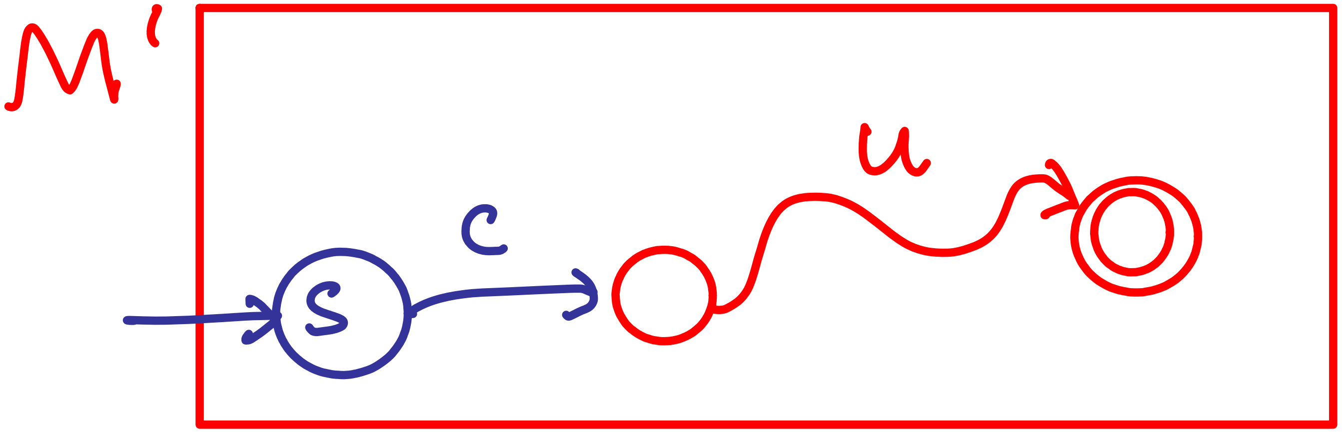


$u \in L(M') \Leftrightarrow u = a^n b^n$ some n

$\boxed{L \text{ on p107 is } \underline{\text{not}} \text{ regular.}}$

Suppose $L = L(M)$ for some DFA $M = (Q, \Sigma, \delta, s, F)$
& derive a contradiction ...

Define an NFA M' from M



$L(M') = \{a^n b^n \mid n \geq 0\}$ contradicting Pumping Lemma

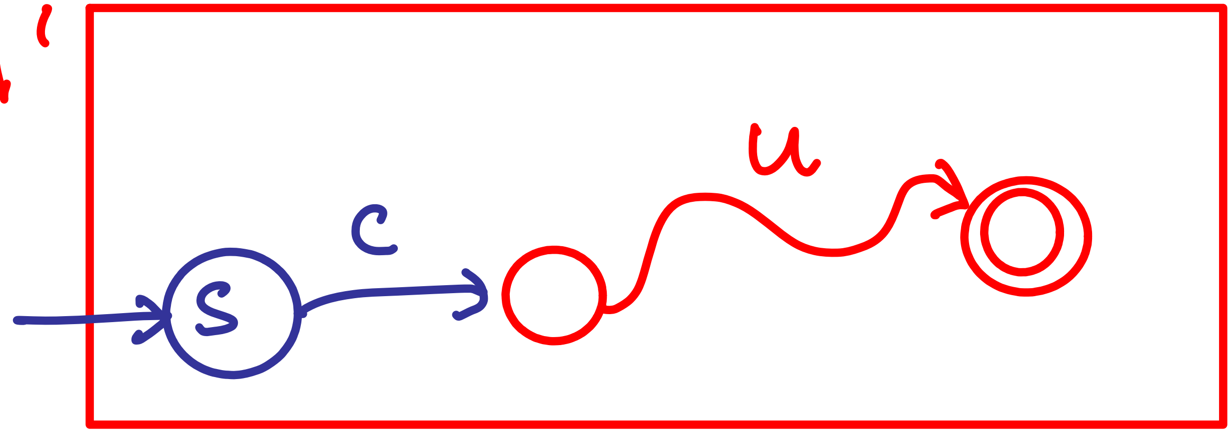
$\boxed{\text{L on p107 is \underline{not} regular.}}$

Suppose $L = L(M)$ for some DFA $M = (Q, \Sigma, \delta, s, F)$
& derive a contradiction ...

Define an NFA M' from M

So no such M can exist

M'



$L(M') = \{a^n b^n \mid n \geq 0\}$ contradicting Pumping Lemma

The way ahead, in THEORY

- What does it mean for a function to be computable?
  [IB Computation Theory]

The way ahead, in THEORY

- What does it mean for a function to be computable?
  [IB Computation Theory]
- Are some computational tasks intrinsically unfeasible?
  [IB Complexity Theory]

The way ahead, 'in THEORY'

- What does it mean for a function to be computable?
[IB Computation Theory]
- Are some computational tasks intrinsically unfeasible?
[IB Complexity Theory]
- How to rigorously specify & reason about program behaviour?
[IB Logic & Proof ; IB Semantics of PLs]