Regular Languages

Mark Greenstreet, CpSc 421, Term 1, 2006/07

Lecture Outline

Regular Languages

- Definition of regular languages
- Closure properties

Lecture Outline

Regular Languages

- Definition of regular languages
  - Regular languages are recognized by finite automata
  - Examples
- Closure properties

Languages (review)

A language is a set of strings.

- Let Σ be a finite set, called an alphabet.
- Σ^* is the set of all strings of Σ, i.e. sequences of zero or more symbols from Σ.
- A language is a subset of Σ^*. Examples:
  - Example, $\Sigma = \{\mathsf{a}, \mathsf{b}\}$, and L_1 is the set of all strings that of length at most two:

$$L_1 \quad = \quad \{\epsilon, \mathsf{a}, \mathsf{b}, \mathsf{aa}, \mathsf{ab}, \mathsf{ba}, \mathsf{bb}\}$$

  - With Σ as above, let L_2 be the set of all strings where every a is followed immediately by a b:

$$L_2 \quad = \quad \{\epsilon, \mathsf{b}, \mathsf{ab}, \mathsf{bb}, \mathsf{abb}, \mathsf{bab}, \mathsf{bbb}, \ldots\}$$

  - With Σ as above, let L_3 be the set of all strings that have more a's than b's:

$$L_3 \quad = \quad \{\mathsf{a}, \mathsf{aa}, \mathsf{aaa}, \mathsf{aab}, \mathsf{aba}, \mathsf{aab}, \ldots\}$$

Deterministic Finite Automata (review)

- A deterministic finite automaton (DFA) is a 5-tuple, $(Q, \Sigma, \delta, q_0, F)$ where:

  Q is a finite set of states.

  Σ is a finite set of symbols.

  $\delta : Q \times \Sigma \rightarrow Q$ is the next state function.

  q_0 is the initial state.

  F is the set of accepting states.

- Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA.

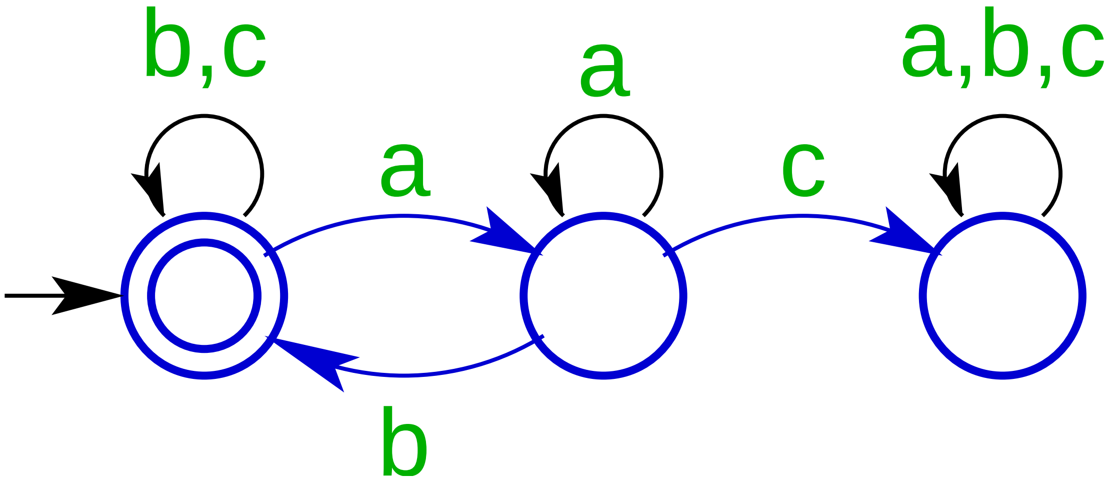
  For $s \in \Sigma^*$,

$$\begin{array}{rcll} \delta(q, s) & = & q, & \text{if } s = \epsilon \\ & = & \delta(\delta(q, x), c), & \text{if } s = x \cdot c \text{ for } c \in \Sigma \end{array}$$

  The language accepted by M is

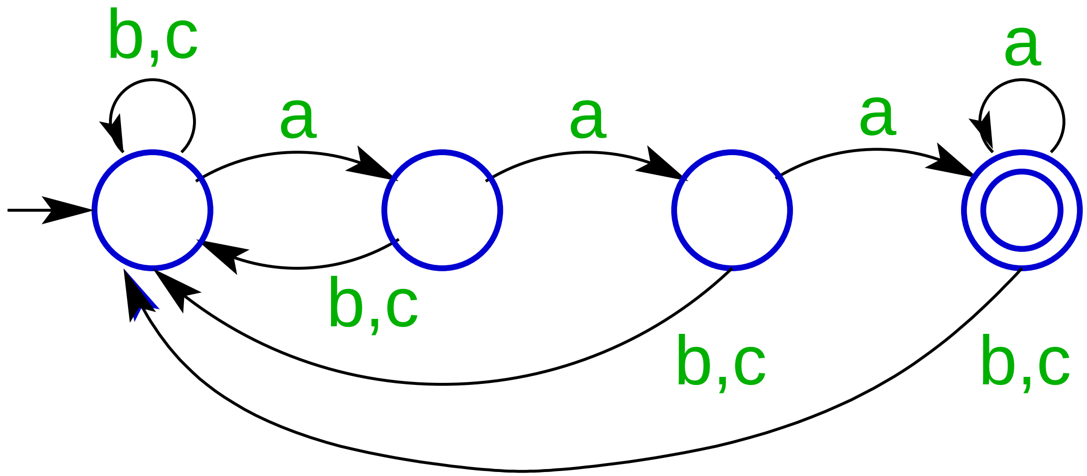
$$L(M) \quad = \quad \{s \in \Sigma^* \mid \delta(q_0, s) \in F\}$$

DFA examples



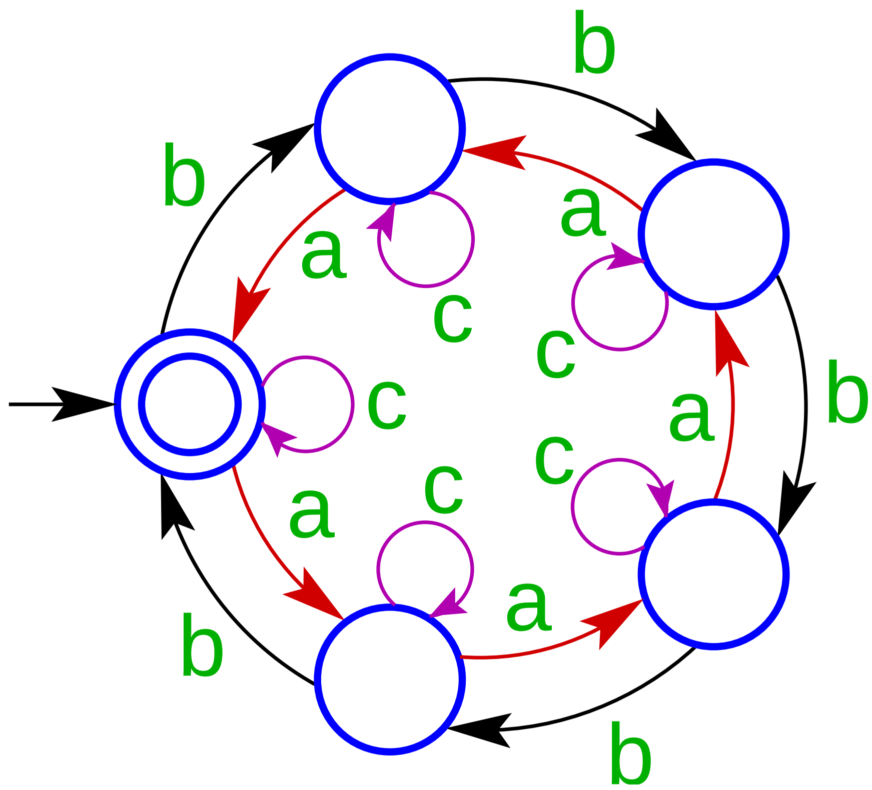
$$L(M_1) = \left\{ s \in \Sigma^* \;\middle|\; \text{Every a in } s \text{ is followed by a b without an intervening c.} \right\}$$

DFA examples



$$L(M_2) = \{s \in \Sigma^* \mid s \text{ ends with three consecutive a's.}\}$$

DFA examples



$$L(M_3) = \left\{ s \in \Sigma^* \;\middle|\; \begin{array}{l}\text{the difference between the number of}\\ \text{a's in } s \text{ and the number of b's is a mul-}\\ \text{tiple of 5.}\end{array} \right\}$$

Regular Languages (Definition)

A language, B, is a regular language iff there is some DFA M such that $L(M) = B$.

In other words, the regular languages are the languages that are recognized by DFAs.

- To show that a language is regular, we can construct a DFA that recognizes is.
- Conversely, we can show that a language is not regular by proving that there can be no DFA that accepts it.

Regular Languages (Properties)

The regular languages are closed under:

Complement: If B is a regular language, then so is $\overline{B}$.

- A string is in $\overline{B}$ iff it is not in B.

Intersection: If B_1 and B_2 are regular languages, then so is $B_1 \cap B_2$.

- A string is in $B_1 \cap B_2$ iff it is in both B_1 and B_2.
- Because we have complement and intersection, we can conclude that the union, difference, symmetric difference, etc. of regular langauges is regular.

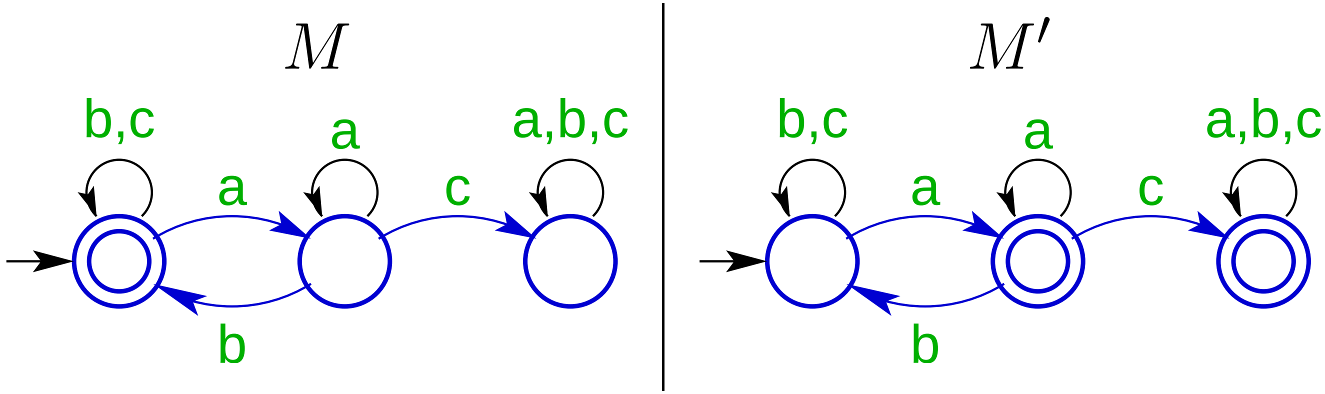
Concatenation: If B_1 and B_2 are regular languages, then so is $B_1 \cdot B_2$.

- A string, s, is in $B_1 \cdot B_2$ iff there are strings x and y such that $x \in B_1$, $y \in B_2$, and $s = x \cdot y$. Note that x and/or y may be ϵ.

Kleene star: B is a regular language, then so is B^*.

- A string, s, is in B^* iff $s = \epsilon$ or there are strings x and y such that $x \in B^*$, $y \in B$, and $s = x \cdot y$.
- Note that even if $B = \emptyset$, $\epsilon \in B^*$. Thus, for any language B, $B^* \neq \emptyset$.

Complement example



$$L(M') = \left\{ s \in \Sigma^* \;\middle|\; \begin{array}{l} s \text{ ends with an a or has an a followed} \\ \text{immediately by a } c. \end{array} \right\}$$

Closure under Complement

Let $B \subseteq \Sigma^*$ be a regular language.

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that recognizes B.

Let $M' = (Q, \Sigma, \delta, q_0, \overline{F})$. M' recognizes $\overline{B}$.

Proof: let $s \in \Sigma^*$ be a string.

- If $s \in B$, then $\delta(q_0, s) \in F$.
  Thus, $\delta(q_0, s) \notin \overline{F}$.
  Thus $s \notin L(M')$.
- If $s \notin B$, then $\delta(q_0, s) \notin F$.
  Thus, $\delta(q_0, s) \in \overline{F}$.
  Thus $s \in L(M')$.

$\overline{B}$ is recognized by a DFA;
therefore, $\overline{B}$ is regular.

$\square$

Closure under Intersection

- Let $B_1, B_2 \subseteq \Sigma^*$ be regular languages.
- Let $M_1 = (Q_1, \Sigma, \delta_1, q_{1,0}, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_{2,0}, F_2)$ be DFAs that recognize B_1 and B_2 respectively.
- Let $M^\cap = (Q_1 \times Q_2, \Sigma, \delta, q_0, F_1 \times F_2)$ where

$$\begin{aligned} q_0 &= (q_{1,0}, q_{2,0}) \\ \delta((q_1, q_2), c) &= (\delta_1(q_1, c), \delta_2(q_2, c)) \end{aligned}$$

  for any $q_1 \in Q_1$, $q_2 \in Q_2$ and $c \in \Sigma$.
  $M^\cap$ recognizes $B^1 \cap B^2$.
  Proof on next slide.

Proof that $L(M^{\cap}) = B_1 \cap B_2$

Let $s \in \Sigma^*$ be a string.

First, we note that for any string $s \in \Sigma^*$, $\delta((q_1, q_2), s) = (\delta(q_1, s), \delta(q_2, s))$.
This can be proven by induction (see slide 15).

If $s \in B_1 \cap B_1$, then $s \in B_1$ and $s \in B_2$.
Thus, $\delta_1(q_{0,1}, s) \in F_1$ and $\delta_2(q_{0,2}, s) \in F_2$.
Thus,

$$\begin{array}{rrll} & \delta(q_0, s) & & \\ & = & \delta((q_{0,1}, q_{0,2}), s), & \text{def. } q_0 \\ & = & (\delta_1(q_{0,1}, s), \delta_2(q_{0,2}), s)), & \text{def. } \delta \\ & \in & F_1 \times F_2, & (s \in B_1) \Rightarrow \delta_1(q_{0,1}, s) \in F_1 \\ & & & (s \in B_2) \Rightarrow \delta_1(q_{0,2}, s) \in F_2 \\ \therefore \quad s & = & L(M^{\cap}) & \end{array}$$

If $s \notin B_1$, then ...

Proof that $L(M^{\cap}) = B_1 \cap B_2$

Let $s \in \Sigma^*$ be a string.

First, we note that for any string $s \in \Sigma^*$, $\delta((q_1, q_2), s) = (\delta(q_1, s), \delta(q_2, s))$.
This can be proven by induction (see slide 15).

If $s \in B_1 \cap B_1$, then $s \in B_1$ and $s \in B_2$.
Thus, $\delta_1(q_{0,1}, s) \in F_1$ and $\delta_2(q_{0,2}, s) \in F_2$.
Thus, $s \in L(M^{\cap})$.

If $s \notin B_1$, then $\delta(q_0, s) = (q_1, q_2)$ with $q_1 \notin F_1$ – just work out $\delta(q_0, s)$ as above.
Thus, $(q_1, q_2) \notin F_1$ and $s \notin L(M^{\cap})$.

If $s \notin B_2$, then $s \notin L(M^{\cap})$ by an argument equivalent to the one for $s \notin B_1$. Thus,
$\delta(q_0, s) \in \overline{F}$.
Thus $s \in L(M')$.

$\therefore s \in L(M^{\cap}$ iff $s \in B_1 \cap B_2$.

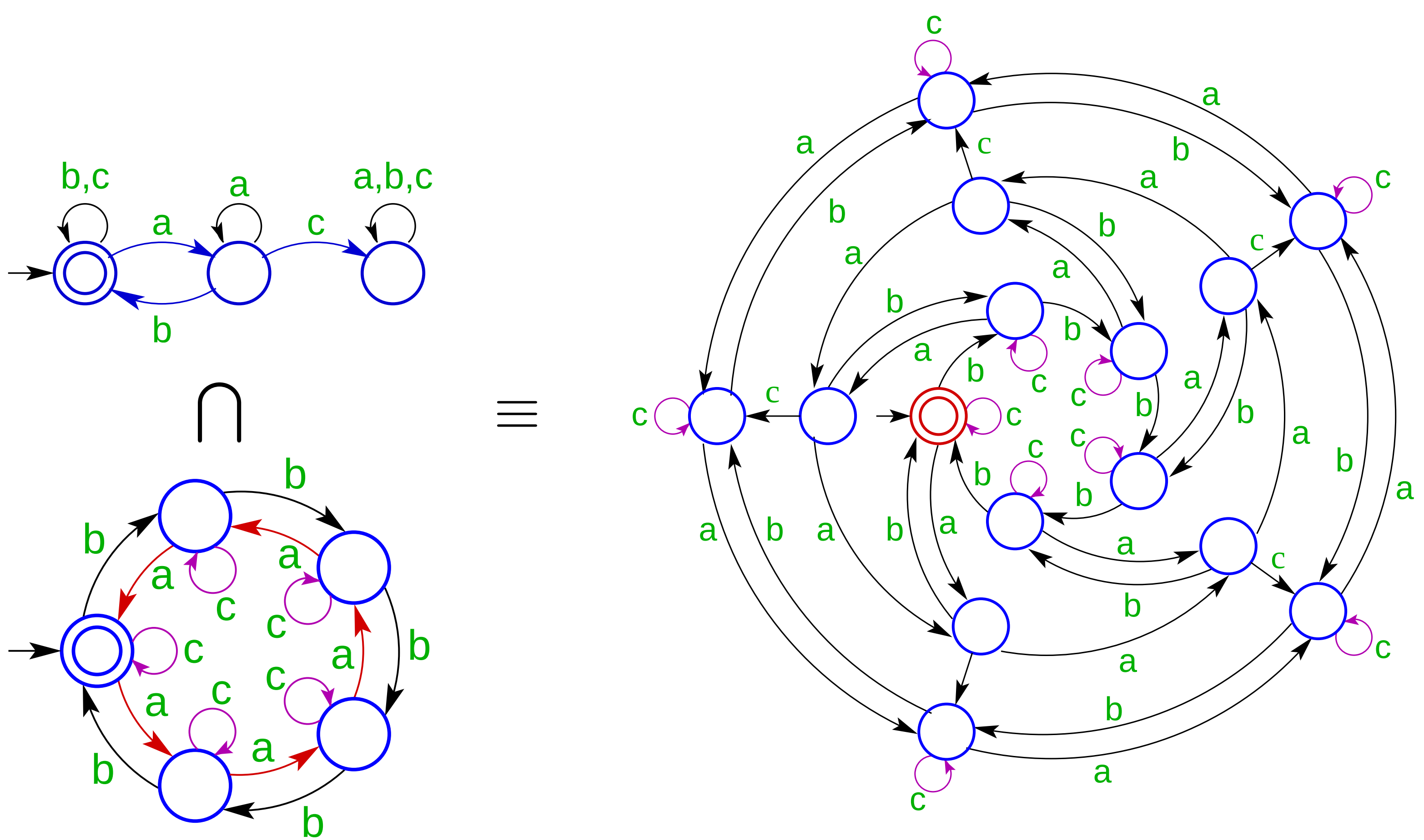
Closure under Intersection (cont.)

- Let $B_1, B_2 \subseteq \Sigma^*$ be a regular language.
- Let $M_1 = (Q_1, \Sigma, \delta_1, q_{1,0}, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_{2,0}, F_2)$ be DFAs that recognize B_1 and B_2 respectively.
- Let $M^\cap = (Q_1 \times Q_2, \Sigma, \delta, q_0 F_1 \times F_2)$ where

$$\begin{array}{rcl} q_0 & = & (q_{1,0}, q_{2,0}) \\ \delta((q_1, q_2), c) & = & (\delta_1(q_1, c), \delta_2(q_2, c)) \end{array}$$

  for any $q_1 \in Q_1$, $q_2 \in Q_2$ and $c \in \Sigma$.
  $M^\cap$ recognizes $B^1 \cap B^2$.
- $B_1 \cap B_2$ is recognized by a DFA;
  therefore, $\overline{B}$ is regular.
- $\square$
- Note: $M^\cap$ is called a product machine because of the use of cartesian cross-product to define the set of states.

Intersection Example



This week

Reading:

September 10 (Today): *Sipser* 1.1 (continued).
Lecture will cover the rest of the section (i.e. pages 40–47).

September 12 (Friday): *Sipser* 1.2.
Lecture will cover through Example 1.35 (i.e. pages 47–52).

Homework:

September 12 (Friday): Homework 0 due. Homework 1 goes out (due Sept. 19).

Proof that $\delta((q_1, q_2), s) = \ldots$

By induction on s:

case $s = \epsilon$:

$$
\begin{array}{lll}
\delta((q_1, q_2), \epsilon) & & \\
\quad = (q_1, q_2), & \text{def. } \delta \text{ for strings} & \\
\quad = (\delta_1(q_1, \epsilon)\delta_2(q_2, \epsilon)), & \text{"} & \checkmark
\end{array}
$$

case $s = x \cdot c$:

$$
\begin{array}{lll}
\delta((q_1, q_2), x \cdot c) & & \\
\quad = \delta(\delta((q_1, q_2), x), c), & \text{def. } \delta \text{ for strings} & \\
\quad = \delta((\delta_1(q_1, x), \delta_2(q_2, x)), c) & \text{ind. hyp.} & \\
\quad = (\delta_1(\delta_1(q_1, x), c), \delta_2(\delta_2(q_2, x), c)) & \text{def. } \delta & \\
\quad = (\delta_1(q_1, s), \delta_2(q_2, s)), & \text{def. } \delta_1, \delta_2 \text{ for strings} & \checkmark
\end{array}
$$