CSE 135: Introduction to Theory of Computation
Closure Properties of Regular Languages

Sungjin Im

University of California, Merced

02-19-2015
Closure Properties

- Recall that we can carry out operations on one or more languages to obtain a new language.
Closure Properties

- Recall that we can carry out operations on one or more languages to obtain a new language.
- Very useful in studying the properties of one language by relating it to other (better understood) languages.
Closure Properties

- Recall that we can carry out operations on one or more languages to obtain a new language.
- Very useful in studying the properties of one language by relating it to other (better understood) languages.
- Most useful when the operations are sophisticated, yet are guaranteed to preserve interesting properties of the language.
Closure Properties

- Recall that we can carry out operations on one or more languages to obtain a new language.
- Very useful in studying the properties of one language by relating it to other (better understood) languages.
- Most useful when the operations are sophisticated, yet are guaranteed to preserve interesting properties of the language.
- Today: A variety of operations which preserve regularity.
Closure Properties

- Recall that we can carry out operations on one or more languages to obtain a new language.
- Very useful in studying the properties of one language by relating it to other (better understood) languages.
- Most useful when the operations are sophisticated, yet are guaranteed to preserve interesting properties of the language.
- Today: A variety of operations which preserve regularity.
  - i.e., the universe of regular languages is closed under these operations.
Closure Properties

Definition
Regular Languages are closed under an operation \( \text{op} \) on languages if

\[
L_1, L_2, \ldots L_n \text{ regular} \implies L = \text{op}(L_1, L_2, \ldots L_n) \text{ is regular}
\]
Closure Properties

Definition
Regular Languages are closed under an operation \( \text{op} \) on languages if

\[
L_1, L_2, \ldots, L_n \text{ regular} \implies L = \text{op}(L_1, L_2, \ldots, L_n) \text{ is regular}
\]

Example
Regular languages are closed under

- “halving”, i.e., \( L \) regular \( \implies \frac{1}{2}L \) regular.
Closure Properties

Definition
Regular Languages are closed under an operation $\text{op}$ on languages if

$$L_1, L_2, \ldots L_n \text{ regular } \implies L = \text{op}(L_1, L_2, \ldots L_n) \text{ is regular}$$

Example
Regular languages are closed under

- “halving”, i.e., $L$ regular $\implies \frac{1}{2} L$ regular.
- “reversing”, i.e., $L$ regular $\implies L^{\text{rev}}$ regular.
Operations from Regular Expressions

Proposition

Regular Languages are closed under $\cup$, $\circ$ and $\ast$. 

Proof.

$(\text{Summarizing previous arguments.})$

$\Rightarrow L_1, L_2 \text{ regular } \Rightarrow \exists \text{ regexes } R_1, R_2 \text{ s.t. } L_1 = L(R_1) \text{ and } L_2 = L(R_2)$. 

$= \Rightarrow L_1 \cup L_2 = L(R_1 \cup R_2) = \Rightarrow L_1 \cup L_2 \text{ regular}$. 

$= \Rightarrow L_1 \circ L_2 = L(R_1 \circ R_2) = \Rightarrow L_1 \circ L_2 \text{ regular}$. 

$= \Rightarrow L_1^* = L(R_1^*) = \Rightarrow L_1^* \text{ regular}$. 

$\blacksquare$
Proposition

*Regular Languages are closed under $\cup$, $\circ$ and $\ast$.*

Proof.

(Summarizing previous arguments.)

- $L_1, L_2$ regular $\implies \exists$ regexes $R_1, R_2$ s.t. $L_1 = L(R_1)$ and $L_2 = L(R_2)$.
  - $\implies L_1 \cup L_2 = L(R_1 \cup R_2) \implies L_1 \cup L_2$ regular.
  - $\implies L_1 \circ L_2 = L(R_1 \circ R_2) \implies L_1 \circ L_2$ regular.
  - $\implies L_1^* = L(R_1^*) \implies L_1^*$ regular.
Proposition

Regular Languages are closed under $\cup$, $\circ$ and $\ast$.

Proof.

(Summarizing previous arguments.)

$\quad L_1, L_2$ regular $\implies \exists$ regexes $R_1, R_2$ s.t. $L_1 = L(R_1)$ and $L_2 = L(R_2)$.

$\quad \implies L_1 \cup L_2 = L(R_1 \cup R_2) \implies L_1 \cup L_2$ regular.

$\quad \implies L_1 \circ L_2 = L(R_1 \circ R_2) \implies L_1 \circ L_2$ regular.

$\quad \implies L_1^* = L(R_1^*) \implies L_1^*$ regular.

□
Closure Under Complementation

Proposition

Regular Languages are closed under complementation, i.e., if $L$ is regular then $\overline{L} = \Sigma^* \setminus L$ is also regular.

Proof.

If $L$ is regular, then there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $L(M) = L$. Then, $M = (Q, \Sigma, \delta, q_0, Q \setminus F)$ (i.e., switch accept and non-accept states) accepts $\overline{L}$.

□
Closure Under Complementation

Proposition

Regular Languages are closed under complementation, i.e., if \( L \) is regular then \( \overline{L} = \Sigma^* \setminus L \) is also regular.

Proof.

- If \( L \) is regular, then there is a DFA \( M = (Q, \Sigma, \delta, q_0, F) \) such that \( L = L(M) \).
Closure Under Complementation

Proposition

Regular Languages are closed under complementation, i.e., if \( L \) is regular then \( \overline{L} = \Sigma^* \setminus L \) is also regular.

Proof.

- If \( L \) is regular, then there is a DFA \( M = (Q, \Sigma, \delta, q_0, F) \) such that \( L = L(M) \).
- Then, \( \overline{M} = (Q, \Sigma, \delta, q_0, Q \setminus F) \) (i.e., switch accept and non-accept states) accepts \( \overline{L} \).
Closure Under Complementation

Proposition

Regular Languages are closed under complementation, i.e., if \( L \) is regular then \( \overline{L} = \Sigma^* \setminus L \) is also regular.

Proof.

- If \( L \) is regular, then there is a DFA \( M = (Q, \Sigma, \delta, q_0, F) \) such that \( L = L(M) \).
- Then, \( \overline{M} = (Q, \Sigma, \delta, q_0, Q \setminus F) \) (i.e., switch accept and non-accept states) accepts \( \overline{L} \). □

What happens if \( M \) (above) was an NFA?
Closure under ∩

Proposition

Regular Languages are closed under intersection, i.e., if \( L_1 \) and \( L_2 \) are regular then \( L_1 \cap L_2 \) is also regular.
Closure under $\cap$

Proposition

Regular Languages are closed under intersection, i.e., if $L_1$ and $L_2$ are regular then $L_1 \cap L_2$ is also regular.

Proof.
Observe that $L_1 \cap L_2 = \overline{L_1 \cup L_2}$. 

Is there a direct proof for intersection (yielding a smaller DFA)?
Proposition

Regular Languages are closed under intersection, i.e., if \( L_1 \) and \( L_2 \) are regular then \( L_1 \cap L_2 \) is also regular.

Proof.

Observe that \( L_1 \cap L_2 = \overline{L_1} \cup \overline{L_2} \). Since regular languages are closed under union and complementation, we have

- \( \overline{L_1} \) and \( \overline{L_2} \) are regular
Closure under $\cap$

Proposition

Regular Languages are closed under intersection, i.e., if $L_1$ and $L_2$ are regular then $L_1 \cap L_2$ is also regular.

Proof.

Observe that $L_1 \cap L_2 = \overline{L_1} \cup \overline{L_2}$. Since regular languages are closed under union and complementation, we have

- $\overline{L_1}$ and $\overline{L_2}$ are regular
- $\overline{L_1} \cup \overline{L_2}$ is regular
Closure under $\cap$

Proposition

Regular Languages are closed under intersection, i.e., if $L_1$ and $L_2$ are regular then $L_1 \cap L_2$ is also regular.

Proof.

Observe that $L_1 \cap L_2 = \overline{L_1 \cup L_2}$. Since regular languages are closed under union and complementation, we have

- $\overline{L_1}$ and $\overline{L_2}$ are regular
- $\overline{L_1 \cup L_2}$ is regular
- Hence, $L_1 \cap L_2 = \overline{L_1 \cup L_2}$ is regular. □
Closure under $\cap$

Proposition

Regular Languages are closed under intersection, i.e., if $L_1$ and $L_2$ are regular then $L_1 \cap L_2$ is also regular.

Proof.

Observe that $L_1 \cap L_2 = \overline{L_1} \cup \overline{L_2}$. Since regular languages are closed under union and complementation, we have

- $\overline{L_1}$ and $\overline{L_2}$ are regular
- $\overline{L_1} \cup \overline{L_2}$ is regular
- Hence, $L_1 \cap L_2 = \overline{L_1} \cup \overline{L_2}$ is regular.

Is there a direct proof for intersection (yielding a smaller DFA)?
Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be DFAs recognizing $L_1$ and $L_2$, respectively.

Idea: Run $M_1$ and $M_2$ in parallel on the same input and accept if both $M_1$ and $M_2$ accept.
Cross-Product Construction

Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be DFAs recognizing $L_1$ and $L_2$, respectively.

**Idea:** Run $M_1$ and $M_2$ in parallel on the same input and accept if both $M_1$ and $M_2$ accept.

Consider $M = (Q, \Sigma, \delta, q_0, F)$ defined as follows

- $Q = Q_1 \times Q_2$
- $q_0 = \langle q_1, q_2 \rangle$
- $\delta(\langle p_1, p_2 \rangle, a) = \langle \delta_1(p_1, a), \delta_2(p_2, a) \rangle$
- $F = F_1 \times F_2$
Cross-Product Construction

Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be DFAs recognizing $L_1$ and $L_2$, respectively.

Idea: Run $M_1$ and $M_2$ in parallel on the same input and accept if both $M_1$ and $M_2$ accept.

Consider $M = (Q, \Sigma, \delta, q_0, F)$ defined as follows

- $Q = Q_1 \times Q_2$
- $q_0 = \langle q_1, q_2 \rangle$
- $\delta(\langle p_1, p_2 \rangle, a) = \langle \delta_1(p_1, a), \delta_2(p_2, a) \rangle$
- $F = F_1 \times F_2$

$M$ accepts $L_1 \cap L_2$ (exercise)
Cross-Product Construction

Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be DFAs recognizing $L_1$ and $L_2$, respectively.

**Idea:** Run $M_1$ and $M_2$ in parallel on the same input and accept if both $M_1$ and $M_2$ accept.

Consider $M = (Q, \Sigma, \delta, q_0, F)$ defined as follows

- $Q = Q_1 \times Q_2$
- $q_0 = \langle q_1, q_2 \rangle$
- $\delta(\langle p_1, p_2 \rangle, a) = \langle \delta_1(p_1, a), \delta_2(p_2, a) \rangle$
- $F = F_1 \times F_2$

$M$ accepts $L_1 \cap L_2$ (exercise)

What happens if $M_1$ and $M_2$ where NFAs?
Cross-Product Construction

Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be DFAs recognizing $L_1$ and $L_2$, respectively.

**Idea:** Run $M_1$ and $M_2$ in parallel on the same input and accept if both $M_1$ and $M_2$ accept.

Consider $M = (Q, \Sigma, \delta, q_0, F)$ defined as follows

- $Q = Q_1 \times Q_2$
- $q_0 = \langle q_1, q_2 \rangle$
- $\delta(\langle p_1, p_2 \rangle, a) = \langle \delta_1(p_1, a), \delta_2(p_2, a) \rangle$
- $F = F_1 \times F_2$

$M$ accepts $L_1 \cap L_2$ (exercise)

What happens if $M_1$ and $M_2$ where NFAs? Still works! Set $\delta(\langle p_1, p_2 \rangle, a) = \delta_1(p_1, a) \times \delta_2(p_2, a)$. 
An Example

\[
\begin{array}{c}
q_0^0 \\
\quad 1 \\
\quad 0 \\
\quad q_1^0 \\
\quad 1
\end{array}
\quad \times 
\quad \begin{array}{c}
\quad 0 \\
\quad 1 \\
q_0^1 \\
\quad 0 \\
q_1^1
\end{array}
= 
\begin{array}{c}
q_{00} \\
\quad 1 \\
\quad 0 \\
\quad q_{10} \\
\quad q_{11}
\end{array}
\quad \begin{array}{c}
\quad 1 \\
\quad 1 \\
\quad 0 \\
\quad 0 \\
\quad 0
\end{array}
\]
Example (1)

\[ \Sigma = \{0, 1\}. \text{ Say } w \in L \text{ iff } |w| \text{ is divisible by 7, but not 3, and } w \text{ contains 1100 as a substring. Is } L \text{ regular or not?} \]

Answer: \( L \) is regular. Define

- \( L_1 \) to be the strings where \( |w| \) is divisible by 7.
Example (1)

$\Sigma = \{0, 1\}$. Say $w \in L$ iff $|w|$ is divisible by 7, but not 3, and $w$ contains 1100 as a substring. Is $L$ regular or not?

Answer: $L$ is regular. Define

- $L_1$ to be the strings where $|w|$ is divisible by 7. ($\left(\{0, 1\}^7\right)^* \text{ is for } L_1.$)
- $L_2$ to be the strings where $|w|$ is divisible by 3.
Example (1)

\[\Sigma = \{0, 1\}.\] Say \(w \in L\) iff \(|w|\) is divisible by 7, but not 3, and \(w\) contains 1100 as a substring. Is \(L\) regular or not?

Answer: \(L\) is regular. Define

- \(L_1\) to be the strings where \(|w|\) is divisible by 7. \(((\{0, 1\}^7)^*\) is for \(L_1\).)
- \(L_2\) to be the strings where \(|w|\) is divisible by 3. \(((\{0, 1\}^3)^*\) is for \(L_2\).)
- \(L_3\) to be the strings having 1100 as a substring.
Example (1)

\[ \Sigma = \{0, 1\} \]. Say \( w \in L \) iff \( |w| \) is divisible by 7, but not 3, and \( w \) contains 1100 as a substring. Is \( L \) regular or not? Answer: \( L \) is regular. Define

\- \( L_1 \) to be the strings where \( |w| \) is divisible by 7. ((\( \{0, 1\}^7 \))^* is for \( L_1 \).)

\- \( L_2 \) to be the strings where \( |w| \) is divisible by 3. ((\( \{0, 1\}^3 \))^* is for \( L_2 \).)

\- \( L_3 \) to be the strings having 1100 as a substring. ((\( \{0, 1\}^*1100\{0, 1\}^* \))^* is for \( L_3 \).)

\[ L = (L_1 \cap L_2) \cap L_3. \]
Example (1)

$$\Sigma = \{0, 1\}$$. Say \(w \in L\) iff \(|w|\) is divisible by 7, but not 3, and \(w\) contains 1100 as a substring. Is \(L\) regular or not?

Answer: \(L\) is regular. Define

- \(L_1\) to be the strings where \(|w|\) is divisible by 7.
  \(((\{0, 1\}^7)^* \text{ is for } L_1.\))
- \(L_2\) to be the strings where \(|w|\) is divisible by 3.
  \(((\{0, 1\}^3)^* \text{ is for } L_2.\))
- \(L_3\) to be the strings having 1100 as a substring.
  \((\{0, 1\}^*1100\{0, 1\}^* \text{ is for } L_3.\))

\(L = (L_1 \cap \overline{L_2}) \cap L_3.\)
Example (1)

\( \Sigma = \{0, 1\} \). Say \( w \in L \) iff \( |w| \) is divisible by 7, but not 3, and \( w \) contains 1100 as a substring. Is \( L \) regular or not? 

Answer: \( L \) is regular. Define

- \( L_1 \) to be the strings where \( |w| \) is divisible by 7.
  \( (\{0, 1\}^7)^* \) is for \( L_1 \).

- \( L_2 \) to be the strings where \( |w| \) is divisible by 3.
  \( (\{0, 1\}^3)^* \) is for \( L_2 \).

- \( L_3 \) to be the strings having 1100 as a substring.
  \( \{0, 1\}^*1100\{0, 1\}^* \) is for \( L_3 \).

\( L = (L_1 \cap \overline{L_2}) \cap L_3 \). So \( L \) is regular.
Example (2)

For any two languages $L_1$ and $L_2$, if $L_1 \cup L_2$ and $L_1$ are both regular, then $L_2$ must be regular. Is this claim true? Answer: This is not true. Consider the case $L_1 = \Sigma^*$. Clearly, $L_1 \cup L_2 = L_1$ is regular. But we can pick any non-regular language $L_2$. 
Example (3)

\[ \Sigma = \{0, 1\} \]. Say \( w \in L \) iff \( w \) starts with 1101 and ends with 010 or 101, and contains 101010 as a substring.

Answer: \( L \) is regular. Define

- \( L_1 \) to be the strings starting with 1101
- \( L_2 \) to be the strings ending with 010 or 101
- \( L_3 \) to be the strings having 101010 as a substring
Example (3)

\[ \Sigma = \{0, 1\}. \] Say \( w \in L \) iff \( w \) starts with 1101 and ends with 010 or 101, and contains 101010 as a substring.

Answer: \( L \) is regular. Define

- \( L_1 \) to be the strings starting with 1101
  \((1101\{0, 1\}^* \) is for \( L_1 \).)
- \( L_2 \) to be the strings ending with 010 or 101.

\[ L = L_1 \cap L_2 \cap L_3. \] So \( L \) is regular.
Example (3)

Σ = \{0, 1\}. Say \( w \in L \) iff \( w \) starts with 1101 and ends with 010 or 101, and contains 101010 as a substring.

Answer: \( L \) is regular. Define

- \( L_1 \) to be the strings starting with 1101
  
  \( (1101\{0, 1\}^* \) is for \( L_1 \).)

- \( L_2 \) to be the strings ending with 010 or 101.
  
  \( \{0, 1\}^*(010 \cup 101) \) is for \( L_2 \).)

- \( L_3 \) to be the strings having 101010 as a substring.
Example (3)

$\Sigma = \{0, 1\}$. Say $w \in L$ iff $w$ starts with 1101 and ends with 010 or 101, and contains 101010 as a substring.

Answer: $L$ is regular. Define

- $L_1$ to be the strings starting with 1101
  ($1101\{0, 1\}^*$ is for $L_1$.)
- $L_2$ to be the strings ending with 010 or 101.
  ($\{0, 1\}^*(010 \cup 101)$ is for $L_2$.)
- $L_3$ to be the strings having 101010 as a substring.
  ($\{0, 1\}^*101010\{0, 1\}^*$ is for $L_3$.)

$L = L_1 \cap L_2 \cap L_3$.

So $L$ is regular.
Example (3)

$\Sigma = \{0, 1\}$. Say $w \in L$ iff $w$ starts with 1101 and ends with 010 or 101, and contains 101010 as a substring.

Answer: $L$ is regular. Define

- $L_1$ to be the strings starting with 1101
  ($1101\{0, 1\}^* \text{ is for } L_1$.)

- $L_2$ to be the strings ending with 010 or 101.
  ($\{0, 1\}^*(010 \cup 101) \text{ is for } L_2$.)

- $L_3$ to be the strings having 101010 as a substring.
  ($\{0, 1\}^*101010\{0, 1\}^* \text{ is for } L_3$.)

$L = L_1 \cap L_2 \cap L_3$. 

Example (3)

$\Sigma = \{0, 1\}$. Say $w \in L$ iff $w$ starts with 1101 and ends with 010 or 101, and contains 101010 as a substring.

Answer: $L$ is regular. Define

- $L_1$ to be the strings starting with 1101
  ($1101\{0, 1\}^*$ is for $L_1$.)

- $L_2$ to be the strings ending with 010 or 101.
  ($\{0, 1\}^*(010 \cup 101)$ is for $L_2$.)

- $L_3$ to be the strings having 101010 as a substring.
  ($\{0, 1\}^*101010\{0, 1\}^*$ is for $L_3$.)

$L = L_1 \cap L_2 \cap L_3$. So $L$ is regular.