Computer Science 351

Regular Operations and Regular Expressions

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #7

Goal for Today

Goals for Today:

- Introduce the ***regular operations:*** Operations that can be used to produce new languages (subsets of $\Sigma^\star$, for some alphabet Σ) from old ones.
- Show that the set of "regular languages" is ***closed*** under each of these operations. This gives us another way to show that a given language is a "regular language" (that is, the language of some DFA).
- Introduce ***regular expressions*** — which provide another way to specify regular languages, and which have a variety of applications.

The "Union" of Two Languages

Definition: Let Σ be an alphabet and let $A, B \subseteq \Sigma^\star$. The ***union*** of the languages A and B is the language

$$A \cup B = \{\omega \in \Sigma^\star \mid \omega \in A \text{ or } \omega \in B \text{ (or both)}\}.$$

- ***Example:*** Suppose that $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $A = \{\mathtt{a}\}$, and $B = \{\mathtt{bb}, \mathtt{c}\}$. Then

$$A \cup B = B \cup A = \{\mathtt{a}, \mathtt{bb}, \mathtt{c}\}.$$

- This should not be a surprise... since the union of *languages* is the same as the union of *sets* (of strings).

The "Concatenation" of Two Languages

Definition: Let Σ be an alphabet and let $A, B \subseteq \Sigma^\star$. The ***concatenation*** of the languages A and B is the language

$$A \circ B = \{\omega_1 \cdot \omega_2 \mid \omega_1 \in A \text{ and } \omega_2 \in B\}.$$

- ***Example:*** Suppose that $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $A = \{\mathtt{a}\}$, and $B = \{\mathtt{bb}, \mathtt{c}\}$. Then

$$A \circ B = \{\mathtt{abb}, \mathtt{ac}\} \quad \text{and} \quad B \circ A = \{\mathtt{bba}, \mathtt{ca}\}.$$

The Kleene Star of a Language

Definition: Let Σ be an alphabet and let $A \subseteq \Sigma^\star$. The ***Kleene star*** of the language A is the language

$$A^\star = \{\omega_1 \cdot \omega_2 \ldots \omega_k \mid k \geq 0 \text{ and } \omega_i \in A \text{ for } 1 \leq i \leq k\}$$

This language is also, sometimes called the ***Kleene closure*** of A — or the ***star*** of A.

- ***Example:*** Suppose that $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $A = \{\mathtt{a}\}$, and $B = \{\mathtt{bb}, \mathtt{c}\}$. Then

$$\begin{aligned} A^\star &= \{\lambda, \mathtt{a}, \mathtt{aa}, \mathtt{aaa}, \mathtt{aaaa}, \ldots\} \\ &= \{\omega \in \Sigma^\star \mid \omega \text{ only includes } \mathtt{a}\text{'s}\}. \end{aligned}$$

The Kleene Star of a Language

On the other hand, $B^\star$ includes the following strings (along with lots more):

- λ (obtained by setting $k = 0$)
- bb (obtained by setting $k = 1$ and $x_1 = \texttt{bb}$)
- c (obtained by setting $k = 1$ and $x_1 = \texttt{c}$)
- bbbb (obtained by setting $k = 2$ and $x_1 = x_2 = \texttt{bb}$)
- bbc (obtained by setting $k = 2$, $x_1 = \texttt{bb}$, and $x_2 = \texttt{c}$)
- cbb (obtained by setting $k = 2$, $x_1 = \texttt{c}$, and $x_2 = \texttt{bb}$)
- cc (obtained by setting $k = 2$ and $x_1 = x_2 = \texttt{c}$)

Regular Operations

- Each of ***union***, ***concatenation*** and ***Kleene star*** of can be thought of as ***operations*** on the set of languages over an alphabet Σ.
- Each of ***union*** and ***concatenation*** can be thought as ***binary operations*** on the set of languages over Σ. That is, each can be used, with a *pair* of languages over Σ, to produce another language over Σ.
- ***Kleene star*** can be thought of as a ***unary operation***. That is, it can be used, with a *single* language over Σ, to produce another language over Σ.

Regular Operations

Definition: For any alphabet Σ, the set of operations

- union,
- concatenation, and
- Kleene star

are the ***regular operations*** over Σ. (These are sometimes just called the "regular operations", without saying what the alphabet is.)

Closure Properties

Theorem 1. Let Σ be an alphabet, and let $A, B \subseteq \Sigma^\star$.

(a) If A and B are regular languages then $A \cup B$ is a regular language, as well.

(b) If A and B are regular languages, then $A \circ B$ is a regular language, as well.

(c) If A is a regular language then $A^\star$ is a regular language as well.

Closure Properties — About the Proof

- The proof of Theorem #1 is a ***constructive proof***. That is, it includes ***algorithms*** (or "constructions") that receive nondeterministic finite automata for any pair of languages $A, B \subseteq \Sigma$ as input, and return nondeterministic finite automata for $A \cup B$, $A \circ B$ and $A^\star$ as output — along with proofs that these algorithms are correct.
- This proof is given — for interested students — in a supplement for this lecture.

Meaning of "Closure Property"

- A ***closure property*** for a set S of languages over an alphabet Σ, is a property stating — for an operation on languages over Σ — that if the operation is applied to languages that all belong to the set S, then the result is a language that belongs to the set S, as well.
- Parts (a), (b), and (c) of Theorem #1 are examples of closure properties for the set of regular languages over an alphabet Σ.

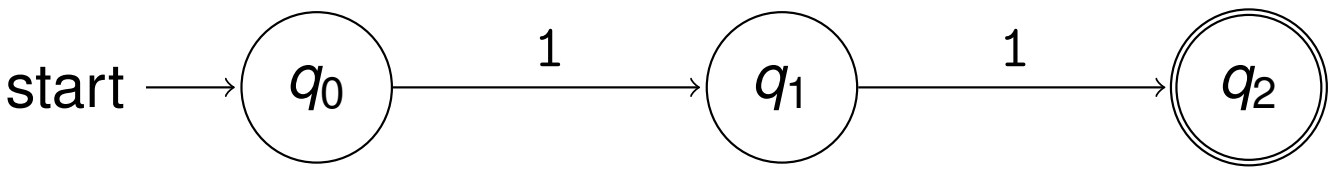
Closure Properties — An Application

This gives us ***another way to prove that a language is regular:*** Show that it is the union, concatenation or star of other regular language(s).

Example: Suppose that $\Sigma = \{0, 1\}$ and consider the language

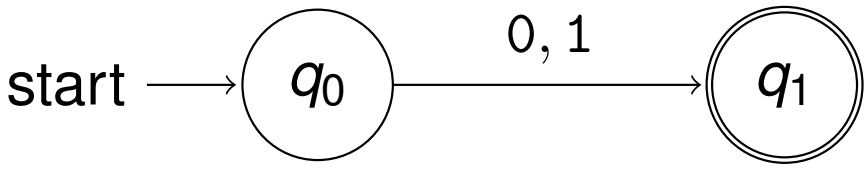
$$L = \{\omega \in \Sigma^\star \mid \text{either } 11 \text{ or } 101 \text{ (or both) is a substring of } \omega\}.$$

1. Let $L_1 = \{11\}$. Then L_1 is a regular language, because it is the language of the following NFA:



Closure Properties — An Application

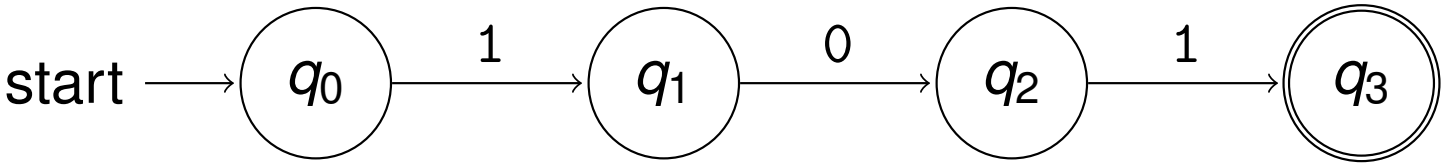
2. Let $L_2 = \Sigma = \{0, 1\}$. Then L_2 is a regular language because it is the language of the following NFA:



3. Let $L_3 = \Sigma^\star$. Then L_3 is a regular language — by ***closure under Kleene star*** — because L_2 is a regular language and $L_3 = L_2^\star$.
4. Let $L_4 = \{\omega \in \Sigma^\star \mid \omega \text{ ends with } 11\}$. Then L_4 is a regular language — by ***closure under concatenation*** — because L_1 and L_3 are regular languages, and $L_4 = L_3 \circ L_1$.

Closure Properties — An Application

5. Let $L_5 = \{\omega \in \Sigma^\star \mid \texttt{11} \text{ is a substring of } \omega\}$. Then L_5 is a regular language — by ***closure under concatenation*** — because L_3 and L_4 are regular languages, and $L_5 = L_4 \circ L_3$.
6. Let $L_6 = \{\texttt{101}\}$. Then L_6 is a regular language because it is the language of the following NFA:



7. Let $L_7 = \{\omega \in \Sigma^\star \mid \omega \text{ ends with } \texttt{101}\}$. Then L_7 is a regular language — by ***closure under concatenation*** — because L_3 and L_6 are regular languages, and $L_7 = L_3 \circ L_6$.

Closure Properties — An Application

8. Let $L_8 = \{\omega \in \Sigma^\star \mid \texttt{101}$ is a substring of $\omega\}$. Then L_8 is a regular language — by ***closure under concatenation*** — because L_7 and L_3 are regular languages, and $L_8 = L_7 \circ L_3$.
9. Finally, consider (once again) the language

   $L = \{\omega \in \Sigma^\star \mid$ either `11` or `101` (or both) is a substring of $\omega\}$.

   L is a regular language — by ***closure under union*** — since L_5 and L_8 are regular languages, and $L = L_5 \cup L_8$.

Regular Expressions and Their Languages

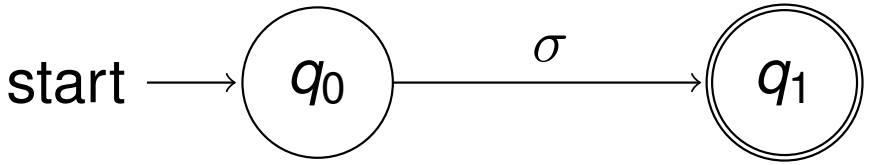
Regular Expressions over an alphabet Σ are ***strings of symbols*** — where the symbols that can appear in these regular expressions can include every symbol in Σ, along with a small set of additional symbols. These — and their languages (all subsets of $\Sigma^\star$ to which they correspond — will now be described.

- Regular expressions are used for searching and updating information in text editors, word processors, data base software, and internet access. Supplemental material, concerning the use of regular expressions in software applications, is available.

Regular Expressions and Their Languages: Informal Introduction

If $\sigma \in \Sigma$ then (the string) σ is also a regular expression (over the alphabet Σ).

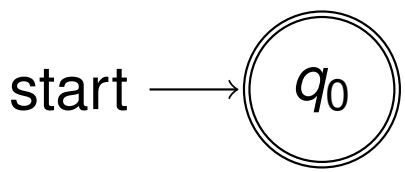
- ***Examples:*** If $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ then each of the following is a regular expression over the alphabet Σ:
  - $\mathtt{a}$
  - $\mathtt{b}$
  - $\mathtt{c}$
- The ***language*** of the regular expression σ (for $\sigma \in \Sigma$) is the set $\{\sigma\}$. This is a regular language since it is the language of the NFA



Regular Expressions and Their Languages: Informal Introduction

λ is a regular expression (over *every* alphabet Σ).

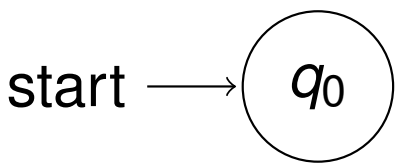
- The ***language*** of the regular expression λ is the set $\{\lambda\}$. This is a regular language since it is the language of the NFA



Regular Expressions and Their Languages: Informal Introduction

$\emptyset$ is a regular expression (over *every* alphabet Σ).

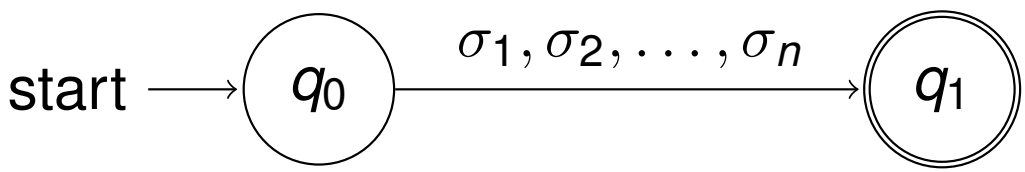
- The ***language*** of the regular expression $\emptyset$ is the empty set $\emptyset$. This is a regular language since it is the language of the NFA



Regular Expressions and Their Languages: Informal Introduction

Σ is a regular expression (over the alphabet Σ).

- The ***language*** of the regular expression Σ is the set Σ — that is, the set of all strings over this alphabet with length one. This is a regular language since it is the language of the NFA that looks like the following, if $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$.



Regular Expressions and Their Languages: Informal Introduction

If R_1 and R_2 are regular expressions over the alphabet Σ then the string

$$(R_1 \cup R_2)$$

is a regular expression over the alphabet Σ as well.

- If $\omega = (R_1 \cup R_2)$ then the language $L(\omega)$ of ω is the union of the languages of R_1 and R_2 — that is,

$$L(\omega) = L(R_1) \cup L(R_2).$$

- Recall that the set of regular languages is closed under union (by Theorem #1(a), above). Thus, if $\omega = (R_1 \cup R_2)$ and the languages $L(R_1)$ and $L(R_2)$ are both regular languages, then $L(\omega)$ is a regular language too.

Regular Expressions and Their Languages: Informal Introduction

If R_1 and R_2 are regular expressions over the alphabet Σ then the string

$$(R_1 \circ R_2)$$

is a regular expression over the alphabet Σ as well.

- If $\omega = (R_1 \circ R_2)$ then the language $L(\omega)$ of ω is the concatenation of the languages of R_1 and R_2 — that is,

  $$L(\omega) = L(R_1) \circ L(R_2) = \{\omega_1 \cdot \omega_2 \mid \omega_1 \in L(R_1) \text{ and } \omega_2 \in L(R_2)\}.$$

- Recall that the set of regular languages is closed under concatenation (by Theorem #1(b), above). Thus, if $\omega = (R_1 \circ R_2)$ and the languages $L(R_1)$ and $L(R_2)$ are both regular languages, then $L(\omega)$ is a regular language too.

Regular Expressions and Their Languages: Informal Introduction

If R_1 is a regular expression over the alphabet Σ then the string

$$(R_1)^\star$$

is a regular expression over the alphabet Σ as well.

- If $\omega = (R_1)^\star$ then the language $L(\omega)$ of ω is the star of the language of R_1 — that is

$$L(\omega) = (L(R_1))^\star = \{\omega_1 \cdot \omega_2 \ldots \omega_k \mid k \geq 0 \text{ and } \omega_1, \omega_2, \ldots, \omega_k \in L(R_1)\}.$$

- Recall that the set of regular languages is closed under star (by Theorem #1(c), above). Thus, if $\omega = (R_1)^\star$ and the language $L(R_1)$ of R_1 is a regular language, then $L(\omega)$ is a regular language too.

Regular Expressions and Their Languages: Making Things More Formal

- To keep things simple we will generally restrict attention to regular expressions over alphabets, Σ, that *do not* include any of the symbols

$$\lambda, \emptyset, \Sigma, (,), \cup, \circ, ^\star$$

- For any such alphabet Σ, a "regular expression over Σ" can be defined as a string over $\widehat{\Sigma}^\star$, for

$$\widehat{\Sigma} = \Sigma \cup \{\lambda, \emptyset, \Sigma, (,), \cup, \circ, ^\star\}.$$

Regular Expressions: Recursive Definition

Regular expressions are defined more formally, using a ***recursive definition***, as follows.

Definition: Let Σ be an alphabet that does not include any of the symbols "λ", "$\emptyset$", "Σ", "$($", "$)$", "$\cup$", "$\circ$" or "*", and let

$$\widehat{\Sigma} = \Sigma \cup \{\lambda, \emptyset, \Sigma, (,), \cup, \circ, ^\star\}.$$

Then a string of symbols $\omega \in \widehat{\Sigma}^\star$ is a ***regular expression*** over Σ if and only if it can be formed using a finite number of applications of the rules shown on the next two slides.

Regular Expressions: Recursive Definition

Base Cases:

1. $\omega = \sigma$, for some symbol $\sigma \in \Sigma$.
2. $\omega = \lambda$.
3. $\omega = \emptyset$.
4. $\omega = \Sigma$.

Regular Expressions: Recursive Definition

Additional (Recursive) Cases:

5. If R_1 and R_2 are both regular expressions over the alphabet Σ — and ω is the string $(R_1 \cup R_2)$ — then ω is regular expression over the alphabet Σ as well.
6. If R_1 and R_2 are both regular expressions over the alphabet Σ — and ω is the string $(R_1 \circ R_2)$ — then ω is regular expression over the alphabet Σ as well.
7. If R_1 is a regular expression over the alphabet Σ — and ω is the string $(R_1)^\star$ — then ω is a regular expression over the alphabet Σ as well.

The Language of a Regular Expression: Recursive Definition

Definition: If ω is a regular expression for the alphabet Σ then the ***language*** $L(\omega)$ of ω is as shown below, and on the following slide.

Base Cases:

1. If $\omega = \sigma$, for $\sigma \in \Sigma$, then $L(\omega) = L(\sigma) = \{\sigma\}$.
2. If $\omega = \lambda$ then $L(\omega) = L(\lambda) = \{\lambda\}$.
3. If $\omega = \emptyset$ then $L(\omega) = L(\emptyset) = \emptyset$.
4. If $\omega = \Sigma$ then $L(\omega) = L(\Sigma) = \Sigma$ (the set of all strings in $\Sigma^\star$ with length one).

The Language of a Regular Expression: Recursive Definition

Additional (Recursive) Cases:

5. If ω is the string $(R_1 \cup R_2)$ — where R_1 and R_2 are regular expressions over Σ — then the language $L(\omega)$ of ω is the set $L(R_1) \cup L(R_2)$.
6. If ω is the string $(R_1 \circ R_2)$ — where R_1 and R_2 are regular expressions over Σ — then the language $L(\omega)$ of ω is the set $L(R_1) \circ L(R_2)$.
7. If ω is the string $(R_1)^\star$ — where R_1 is a regular expression over Σ — then the language $L(\omega)$ of ω is the set $(L(R_1))^\star$.

Regular Expressions and Their Languages: Problems To Be Solved

Two problems concerning regular expressions, which students might be asked to solve — for reasonably *short* and *simple* regular expressions over an alphabet Σ, are as follows.

1. Given a string $\omega \in \widehat{\Sigma}^\star$, for $\widehat{\Sigma}$ as above, decide whether ω *is* a regular expression over Σ — and, if it is, describe how it can be produced using the rules that are given in the *recursive definition* of a "regular expression over Σ".
2. Given a regular expression ω over Σ, describe the *language* of ω — including enough information for a reader to be able to confirm that your answer is correct.

Regular Expressions and Their Languages: Problems To Be Solved

- One way to solve the first problem — in the case that the given string, ω, *is* a regular expression over Σ — is to give a sequence of applications of the rules, included in the *definition* of a "regular expression", to show how ω can be obtained.
- The second problem can then be solved, as well, by describing the languages of the regular expressions that are generated, along the way.

Regular Expressions and Their Languages: Problems To Be Solved

Example: Let $\Sigma = \{0, 1\}$ and consider the following string ω:

$$((((((\Sigma)^\star \circ 1) \circ (\lambda \cup 0)) \circ 1) \circ (\Sigma)^\star)$$

1. By rule #4, Σ is a regular expression over Σ whose language is $\Sigma = \{0, 1\}$.
2. By rule #7 — and considering the regular expression at line 1 — $(\Sigma)^\star$ is a regular expression over Σ whose language is $(\Sigma)^\star = \Sigma^\star$ — the set of all strings over the alphabet Σ.
3. Since $1 \in \Sigma$, by rule #1, 1 is a regular expression over Σ whose language is $\{1\}$.

Regular Expressions and Their Languages: Problems To Be Solved

4. By rule #6 — and considering the regular expressions at lines 2 and 3 — $((\Sigma)^\star \circ 1)$ is a regular expression over Σ whose language is

$$\Sigma^\star \circ \{1\} = \{\omega \in \Sigma^\star \mid \omega \text{ ends with } 1\}.$$

5. By rule #2, λ is a regular expression over Σ whose language is $\{\lambda\}$.
6. Since $0 \in \Sigma$, by rule #1, 0 is a regular expression over Σ whose language is $\{0\}$.
7. By rule #5 — and considering the regular expressions at lines #5 and #6 — $(\lambda \cup 0)$ is a regular expression over Σ whose language is $\{\lambda\} \cup \{0\} = \{\lambda, 0\}$.

Regular Expressions and Their Languages: Problems To Be Solved

8. By rule #6 – and considering the regular expressions at lines #4 and #7 —

$$(((\Sigma)^\star \circ 1) \circ (\lambda \cup 0))$$

is a regular expression over Σ whose language is

$$\{\omega \in \Sigma^\star \mid \omega \text{ ends with } 1\} \circ \{\lambda, 0\}$$
$$= \{\omega \in \Sigma^\star \mid \text{either } \omega \text{ ends with } 1 \text{ or } \omega \text{ ends with } 10\}.$$

9. By rule #6 — and considering the regular expressions at lines #8 and #3 —

$$((((\Sigma)^\star \circ 1) \circ (\lambda \cup 0)) \circ 1)$$

is a regular expression over Σ whose language is

$$\{\omega \in \Sigma^\star \mid \text{either } \omega \text{ ends with } 1 \text{ or } \omega \text{ ends with } 10\} \circ \{1\}$$
$$= \{\omega \in \Sigma^\star \mid \text{either } \omega \text{ ends with } 11 \text{ or } \omega \text{ ends with } 101\}.$$

Regular Expressions and Their Languages: Problems To Be Solved

10. By rule #6 — and considering the regular expressions at lines #9 and #2 —

$$(((((\Sigma)^\star \circ 1) \circ (\lambda \cup 0)) \circ 1) \circ (\Sigma)^\star)$$

is a regular expression over Σ. With a *little* bit of work its language can be shown to be

$$\{\omega \in \Sigma^\star \mid \text{either } 11 \text{ or } 101 \text{ (or both) is a substring of } \omega\}$$

Regular Expressions and Their Languages: Problems To Be Solved

Another problem, concerning regular expression, is as follows.

3. Given a languges $L \subseteq \Sigma^\star$, find a regular expression, ω, over Σ whose language is L.

This problem can sometimes be solved by working from the top down — expression the given language as the union, concatenation or Kleene star of other, simpler languages, and then showing that each of *these* languages is the language of a regular expression.

The result described next —- and its proof — gives another way to try to solve this kind of problem.

Regular Expressions and Regular Languages: Equivalence

Theorem 2. Let Σ be an alphabet. Then a language $L \subseteq \Sigma^\star$ is a regular language if and only if L is the language of a regular expression for Σ.

- The proof of Theorem #2 (provided for this course) is also a ***constructive proof***. That is, provides an algorithm that receives a regular expression for Σ and produces a nondeterministic finite automaton, with alphabet Σ, that has the same language. It also provides an algorithm that receives a nondeterministic finite automaton with alphabet Σ, as input, and produces a regular expression for Σ with the same language.
- The proof of this result is given — for interested students — in a supplemental document.

Regular Expressions and Regular Languages: There is Lots More!

- Regular expressions have a variety of applications of applications in text processing and data science.
- *Software that processes regular expressions* is, therefore, important.
- *Versions of regular expressions that can be used in computer programs* are important too.
- Additional supplemental documents, with more information about this, will also be available.