Formal Languages

Thursday, September 27, 2012
Reading: Sipser pp 13-14, 44-45
Stoughton 2.1, 2.2, end of 2.3, beginning of 3.1

CS235 Languages and Automata

Department of Computer Science
Wellesley College

Alphabets, Strings, and Languages

An **alphabet** is a set of symbols.
E.g.: $\Sigma_1 = \{0,1\}$; $\Sigma_2 = \{-,0,+\}$ $\Sigma_3 = \{a,b, \ldots, y, z\}$; $\Sigma_4 = \{\text{☺}, \Rightarrow, \boxed{a}, \boxed{aa}\}$

A **string over Σ** is a sequence of symbols from Σ.
The empty string is ofte written ε (Sipser) or λ; Stoughton uses %.
Σ^* denotes all strings over Σ. E.g.:

- Σ_1^* contains ε, 0, 1, 00, 01, 10, 11, 000, …
- Σ_2^* contains ε, -, 0, +, --, -0, -+, 0-, 00, 0+, +-, +0, ++, ---, …
- Σ_3^* contains ε, a, b, …, aa, ab, …, bar, baz, foo, wellesley, …
- Σ_4^* contains ε, ☺, ⇒, $\boxed{a}$, $\boxed{aa}$, …, $\boxed{a}$⇒☺, …, $\boxed{a}\boxed{aa}$, $\boxed{aa}\boxed{a}$,…

A **language over Σ** (Stoughton's Σ-language) is any subset of Σ^*.
I.e., it's a set of strings over Σ. E.g.:

- L_1 over Σ_1 is all sequences of 1s and all sequences of 10s.
- L_2 over Σ_2 is all strings with equal numbers of -, 0, and +.
- L_3 over Σ_3 is all lowercase words in the OED.
- L_4 over Σ_4 is {☺, ☺⇒☺, $\boxed{a}\boxed{aa}$ }.

Languages over Finite Alphabets are Countable

A language = a set of strings over an alphabet Σ = a subset of Σ^*.

Suppose Σ is finite. Then Σ^* (and any subset thereof) is countable.

Why? Can enumerate all strings in lexicographic (dictionary) order!

- $\Sigma_1 = \{0,1\}$

  $\Sigma_1^* = \{\varepsilon,$
  0, 1
  00, 01, 10, 11
  000, 001, 010, 011, 100, 101, 110, 111,
  …}
- for $\Sigma_3 = \{a,b, \ldots, y, z\}$, can enumerate all elements of Σ_3^* in lexicographic order -- we'll eventually get to any given element.
- The following are countable: all English books; all Java programs.

String Operations

Length: $|s|$ is the length of a string s. E.g.:
$|\%| = 0$, $|foo| = 3$, $|\text{☺} \Rightarrow \boxed{a}\boxed{aa}| = 4$

Concatenation: If x, y in Σ^*, then xy in Σ^* is the string consisting of all symbols in x followed by all symbols in y. Concatenation is also written $x@y$ (Stoughton) and $x \cdot y$. E.g. baz@quux = bazquux

Concatenation Properties:

- $(x@y)@z = xyz = x@(y@z)$ (*Associativity*)
- $x@\varepsilon = x = \varepsilon@x$ (*Identity*)
- $|x@y| = |x| + |y|$

Monoid

Other Definitions:

- x is a **prefix** of y iff $y = xv$ for some v
- x is a **suffix** of y iff $y = ux$ for some u
- x is a **substring** of y iff $y = uxv$ for some u and v

There are **proper** versions of these, too.

What are all prefixes, suffixes, substrings of bar?

More String Operations

String Powers: Suppose x is a string.

- $x^0 = \varepsilon$
- $x^n = x@x^{n-1}$, abbreviated $xx^{n-1} = x(x^{n-1})$

Power Properties:

- $x^{a+b} = x^a@x^b$
- $|x^n| = n \cdot |x|$

String Reversal: Suppose a is a symbol and x is a string.

- $\varepsilon^R = \varepsilon$
- $(a@x)^R = x^R@a$

Reversal Properties:

- $(x@y)^R = y^R@x^R$
- $(x^R)^R = x$
- $|x^R| = |x|$

String Induction

Suppose P(w) is a property of strings w in Σ^*. Can prove P(w) by *natural induction (or strong induction) on $|w|$*. Equivalently:

Right String Induction:

Suppose that

1. (basis step) P(ε) holds.
2. (inductive step) For all $a \in \Sigma$ and x in Σ^*, P(x) $\Rightarrow$ P(ax).

the inductive hypthesis (IH)

Then P(w) holds for all $w \in \Sigma^*$.

Left String Induction:

(inductive step) For all $a \in \Sigma$ and x in Σ^*, P(x) $\Rightarrow$ P(xa).

Strong String Induction:

(inductive step) For all $w \in \Sigma^*$, $(\bigwedge_{x \in \Sigma^* \text{ s.t. } |x| < |w|} P(x)) \Rightarrow P(w)$.

String Induction Example: Reversal

Prove that $(x@y)^R = y^R@x^R$

Hold y constant, and perform induction on x.

(basis step) $x = \varepsilon$

(inductive step) $x = a@w$ (for symbol a and string w)

What is IH?

Set Operations on Languages

Suppose L_1 and L_2 are Σ-languages.

The following are all Σ-languages:

$L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 - L_2$, $\overline{L_1}$ ($= \Sigma^* - L_1$)

E.g. , what are all elts with size ≤ 3 for the following sets?

Even0s = all binary strings with even # of 0s.

Odd1s = all binary strings with odd # of 1s.

Give English descriptions of the following *and* list all elts with size ≤ 3

Even0s $\cup$ Odd1s =

Even0s $\cap$ Odd1s =

Even0s - Odd1s =

$\overline{\text{Even0s}}$ =

Language Concatenation:

Suppose L_1 and L_2 are Σ-languages.

Definition:

$L_1 @ L_2 = \{x @ y \mid x \text{ in } L_1 \text{ and } y \text{ in } L_2\}$ (also written $L_1 \circ L_2$, L_1L_2)

E.g. {CS, PHYS} @ {110, 111, 115} =

What are all elts with size ≤ 3 for Even0s @ Odd1s?

Concatenation Properties:

- $(L_1 @ L_2) @ L_3 = L_1 @ (L_2 @ L_3)$ (Associativity)
- $\{\varepsilon\} @ L = L = L @ \{\varepsilon\}$ (Identity)
- $\varnothing @ L = \varnothing = L @ \varnothing$ (Zero)
- $|L_1 @ L_2| = |L_1| \cdot |L_2|$ for finite L_1, L_2

Language Powers (L^n)

Definition:

- $L^0 = \{\varepsilon\}$
- $L^n = L @ L^{n-1}$

E.g., $\{0,1\}^2$ =

Odd1s^2 =

Properties:

- $L^{a+b} = L^a @ L^b$
- $|L^n| = |L|^n$ for finite L
- $\{x\}^n = \{x^n\}$
- $\{\varepsilon\}^n = \{\varepsilon\}$

Kleene Star/Kleene Closure (L*)

Definition:

$L^* = \{L^n \mid n \text{ in Nat}\}$

Examples:

- $\{0,1\}^*$ =

  (This is consistent with notation Σ^*)

- Which of the following are in {10, 011, 101, 110}*?

  101011

  1011010

  1011011

* Kleene is pronounced ("clay knee").

Where are We Headed?

Want to explore/relate the following:

- English descriptions of formal languages.
- Machines (automata) that determine language membership.
- Programs that determine language membership.
- Grammars that describe how to generate all strings in a language.
- Programs that enumerate strings in a language (or list all strings in the language up to a certain length).

Classifying Languages in a Hierarchy

Reg = Regular Languages
- Deterministic Finite Automaton
- Nondeterministic Finite Automaton
- Regular Expression
- Right-Linear Grammar

CFL = Context-Free Language
- Nondeterministic Pushdown Automaton
- Context-Free Grammar

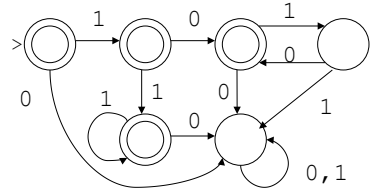
Dec = Recursive (Turing-Decidable) Language
- Turing Machine
- Unrestricted Grammar

RE = Recursively Enumerable (Turing-Recognizable/Acceptable) Language

Lan = All Languages

We've seen DFAs; Next stop: NFAs

Deterministic Finite Automaton



Nondeterministic Finite Automaton