T2: back before 8 pm, average B
A2: some sections done released soon.
A3: up tomorrow night.

tutorials changed polarity due to fall break ... so suddenly evening section leads off on formal languages... it'll be okay.

CSC236 fall 2012

automata and languages

Danny Heap
heap@cs.toronto.edu
BA4270 (behind elevators)
http://www.cdf.toronto.edu/~heap/236/F12/
416-978-5899

Using Introduction to the Theory of Computation, Chapter 7

Outline

formal languages

FSAs

notes

some definitions

alphabet: finite, non-empty set of symbols, e.g. $\{a, b\}$ or $\{0, 1, -1\}$. Conventionally denoted Σ.

$\{0,1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, \dots\}$

string: finite (including empty) sequence of symbols over an alphabet: abba is a string over $\{a, b\}$. Σ

Λ ϵ Convention: ε is the empty string, never an allowed symbol, Σ^* is set of all strings over Σ.

language: Subset of Σ^* for some alphabet Σ. Possibly empty, possibly infinite subset. E.g. $\{\}$, $\emptyset$
$\{aa, aaa, aaaa, \overset{\epsilon}{\dots}\}$.

N.B.: $\{\} \neq \{\varepsilon\}$.

Many problems can be reduced to languages: logical formulas, identifiers for compilation, natural language processing. Key question is recognition:

Given language L and string s, is $s \in L$?

Languages may be described either by descriptive generators (for example, regular expressions) or procedurally (e.g. finite state automata)

more notation

$\{1, 11\}$ 11

string length: denoted $|s|$, is the number of symbols in s, e.g. $|bba| = 3$.

$s = t$: if and only if $|s| = |t|$, and $s_i = t_i$ for $1 \leq i \leq |s|$.

s^R: reversal of s is obtained by reversing symbols of s, e.g. $1011^R = 1101$.

st or $s \circ t$: contcatenation of s and t — all characters of s followed by all those of t, e.g. $bba \circ bb = bbabb$.

s^k: denotes s concatenated with itself k times. E.g., $ab^3 = ababab$, $101^0 = \varepsilon$.

Σ^n: all strings of length n over Σ, Σ^* denotes all strings over Σ.

language operations

$\overline{L}$: Complement of L, i.e. $\Sigma^* - L$. If L is language of strings over $\{0,1\}$ that start with 0, then $\overline{L}$ is the language of strings that begin with 1 plus the empty string.

$L \cup L'$: union

$L \cap L'$: intersection

$L = \{a, ab\}$
$L' \{0, 1\}$

$L - L'$: difference
$L \setminus L'$

$LL' = \{a0, ab0, a1, ab1\}$

$\mathrm{Rev}(L)$: $= \{s^R : s \in L\}$
$L' \circ L$

concatenation: LL' or $L \cdot L' = \{rt | r \in L, t \in L'\}$. Special cases $L\{\varepsilon\} = L = \{\varepsilon\}L$, and $L\{\} = \{\} = \{\}L$.

more language operations

$$\{\}^0 = \{\varepsilon\}$$

exponentiation: L^k is concatenation of L k times. Special case, $L^0 = \{\varepsilon\}$, including $L = \{\}$!

$$\Sigma^* = \{a\}^*$$

Kleene star: $L^* = L^0 \cup L^1 \cup L^2 \cup \ldots$

$$\{\varepsilon\}^2 = \{\varepsilon\}$$

states needed to classify a string

what state is a stingy vending machine in based on coins?
accepts only nickles (a), dimes (b), and quarters (c),
no change given, and everything costs 30 cents
useful toy (you'll need JRE)

$\Sigma = \{n, d, q\}$

$\underline{n}\,\underline{d}\,\underline{n}$

δ	0	5	10	15	20	25	≥ 30
n	5	10	15	20	25	≥ 30	≥ 30
d	10	15	20	25	≥ 30	≥ 30	≥ 30
q	25	≥ 30	≥ 30	≥ 30	≥ 30	≥ 30	≥ 30

build an automaton with formalities...

quintuple: $(Q, \Sigma, q_0, F, \delta)$

Q is set of states, Σ is finite, non-empty alphabet, q_0 is start state

F is set of accepting states, and $\delta : \underline{Q \times \Sigma} \mapsto Q$ is transition function

We can extend $\delta : Q \times \Sigma \mapsto Q$ to a transition function that tells us what state a **string** s takes the automaton to:

ndn

$\delta^*(q_0, ndn) = \delta(\delta^*(q_0, nd), n)$

$= \delta(\delta(\delta^*(q_0, n), d), n)$

$\delta(\delta(\delta(\underbrace{\delta^*(q_0, \varepsilon)}_{q_0}, n), d), n)$

q_5 15 20

$$\delta^* : Q \times \Sigma^* \mapsto Q \qquad \delta^*(q, s) = \begin{cases} q & \text{if } s = \varepsilon \\ \delta(\delta^*(q, s'), a) & \text{if } s' \in \Sigma^*, a \in \Sigma, s = \end{cases}$$

$s = s'a$

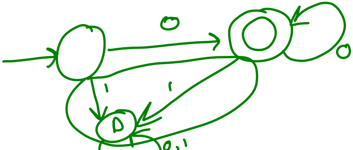
$a \in \Sigma \quad s' \in \Sigma^*$

String s is accepted if and only iff $\underline{\delta^*(q_0, s)} \in F$, it is rejected otherwise.

example — an odd machine

devise a machine that accepts strings over $\{a, b\}$ with an odd number of as

$\Sigma = \{0, 1\}$



L has even # of as + odd length

Formal proof requires inductive proof of invariant:

E + O are states

$$\delta^*(E, s) = \begin{cases} E & \text{if } s \text{ has even number of } a\text{s} \\ O & \text{if } s \text{ has odd number of } a\text{s} \end{cases}$$

for all $k \in \mathbb{N}$, $s \in \Sigma^k$, then $\delta^*(E, S)$ is true. want to conclude that M accept S iff s has ~~even~~ odd # of as.

float machine

$L_1 = \{0, \ldots, 9\}$

$L_2 = \{+, -\}, L_3 = \{.\}$

$L_F = \{s \in L_2^j L_1^m L_3^k L_1^n \mid j, k \leq 1, m, n \geq 1\}$

Devise a machine that accepts L_F