A2 - posted by section (Lila, Glin, ...).

T2 spoile: CSC236 fall 2012

B average: automata and languages

back Friday - F5A

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

$$o$$
, 1 , $1\varepsilon \stackrel{?}{=} 1$

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

string: finite (including empty) sequence of symbols over an alphabet: abba is a string over $\{a, b\}$. $\not = a + b$ Convention: $[\varepsilon]$ is the empty string, never an allowed symbol, Σ^* is set of all strings over Σ .

$$Z = \{0,1\}$$
, $Z^* = \{\xi, 6, 1, 0\}$

language: Subset of Σ^* for some alphabet Σ . Possibly empty, possibly infinite subset. E.g. $\{\}$, $\{aa, \beta's\}$.

N.B.:
$$\{\} \neq \{\epsilon\}$$
. $\mathcal{L} = \mathbb{L}^n$

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

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

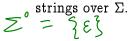
$$s = t$$
: if and only if $|s| = |t|$, and $s_i = t_i$ for $1 \le i \le |s|$.

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

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

 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





$$\sim$$

 \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

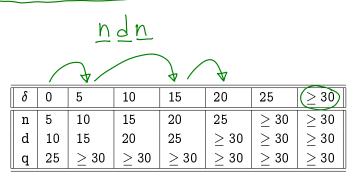
 $L \cap L'$: intersection

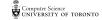
$$L-L'$$
: difference $L \setminus L'$



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)







build an automaton with formalities... transitions quintuple: $(Q, \Sigma, q_0, F, \delta)$ Levis and Lorit Q is set of states, Σ is finite, non-empty alphabet, q_0 is start state commute

We can extend $\delta: Q \times \Sigma \mapsto Q$ to a transition function that

tells us what state a string takes the automaton to:
$$\begin{cases} \begin{pmatrix} 1 \\ 1 \\ 2 \\ 3 \end{pmatrix} \end{pmatrix} = \begin{cases} \delta \left(\delta^* \left(5, n \right)_1 \right) \\ \delta \left(\delta \left(\delta^* \left(5, n \right)_2 \right) \right) \\ \delta \left(\delta^* \left(\delta^* \left(5, n \right)_2 \right) \right) \\ \delta^* : Q \times \Sigma^* \mapsto Q \end{cases} \qquad \begin{cases} \delta^* \left(q, s \right) \\ \delta \left(\delta^* \left(q, s' \right), a \right) \\ \delta \left(\delta^* \left(q, s' \right), a \right) \end{cases} \text{ if } s' \in \Sigma^*, a \in \Sigma \\ \delta \left(\delta^* \left(q, s' \right), a \right) \end{cases} \end{cases}$$

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



example — an odd machine

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

Formal proof requires inductive proof of invariant:

$$\delta^*(E,s) = egin{cases} E & ext{if s has even number of $as} \ O & ext{if s has odd number of $as} \end{cases}$$



notes

