CSC236 fall 2012

regular languages, regular expressions

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

Computer Science
UNIVERSITY OF TORONTO

Outline

regular expressions, regular languages

notes

they're equivalent:

$L = L(M)$ for some DFSA $M \Leftrightarrow L = L(M')$ for some NFSA $M' \Leftrightarrow$
$L = L(R)$ for some regular expression R

step 1: convert $L(M)$ to $L(R)$, eliminate states

equivalence. . .

state elimination recipe for state q

1. $s_1 \ldots s_m$ are states with transitions **to** q, with labels $S_1 \ldots S_m$
2. $t_1 \ldots t_n$ are states with transitions **from** q, with labels $T_1 \ldots T_n$
3. Q is any self-loop on q
4. Eliminate q, and add (union) transition label $S_i Q^* T_j$ from s_i to t_j.

equivalence:

step 2: convert $L(R)$ to $L(M)$:

start with $\emptyset$, ε, $a \in \Sigma$

equivalence...

step 2.5: convert $L(R)$ to $L(M)$:
union, concatenation, stars

consequences of regularity

How to represent $L = \{\text{string with a 0 in fourth-last place}\}$

How about $L = \{1^n 0^n \,|\, n \in \mathbb{N}\}$

notes