Lecture 11

Regular Expressions

CS 241: Foundations of Sequential Programs
Fall 2009

Troy Vasiga et al
University of Waterloo

Review

- An example comparing DFAs and NFAs
- Can convert between an NFA and DFA using the subset construction
    - define each set of states that can be occupied at the same step
    - one state in the DFA for each unique set of states in the NFA

ϵ-NFAs

- allows transition between states on "no input"
- can be used as "glue" for joining machines together
- example: $L = \{$ card, cab, calf $\}$
- it is not surprising that ϵ-NFAs can be converted to NFAs

Regular Expressions

Defined recursively: a regular expression (RE) is

- $\emptyset$, or
- ϵ, or
- a, where $a \in \Sigma$
- $E_1 E_2$ where E_1 and E_2 are REs
- $E_1 | E_2$ where E_1 and E_2 are REs
- E^* where E is a REs

RE examples

- L $=$ {cab, car, card}
- $\Sigma = \{a\}$, L $=$ {even #s of a's}
- $\Sigma = \{a, b\}$, L $=$ {even #s of a's}

(Unix) Shorthands and Extensions

- $[a - z]$
- E+
- others may occur in your future readings

Circle of Life

Practical Applications

- Most real-world examples do not care about recognizers (DNA match may be the exception)
- Mostly, DFAs are used for:
  - transforming/transducing input
  - searching in text
  - scanning/translating

Transducers

Example 1: Remove stutters from $\Sigma = \{a, b\}$.

Example 2: Binary integers from $\Sigma = \{0, 1\}$.

Searching

Goal: Find the first occurrence of a string p inside a (larger) string T.
This turns out to be difficult, but does really on DFAs (see CS240).

Scanning

See slides