CSC236 fall 2012
regular expressions

Danny Heap
eath@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

regular expressions

product, non-deterministic FSAs

regular languages

notes
another way to define languages
In addition to the language accepted by DFSA $L(M)$
and set description $L = \ldots$.

Definition: The regular expressions (regexps or REs) over
alphabet $\Sigma$ is the smallest set such that:

1. $\{\}$, $\epsilon$, and $a$, for every $a \in \Sigma$ are REs over $\Sigma$
2. if $T$ and $S$ are REs over $\Sigma$, then so are:
   - $T + S$ (union) — lowest precedence operator
   - $TS$ (concatenation) — middle precedence operator
   - $T^*$ (star) — highest precedence
**regular expression to language:**

The $L(R)$, the language denoted (or described) by $R$ is defined by structural induction:

**Basis:** If $R$ is a regular expression by the basis of the definition of regular expressions, then define $L(R)$:

- $L(\emptyset) = \emptyset$ (the empty language)
- $L(\varepsilon) = \{\varepsilon\}$ (the language consisting of just the empty string)
- $L(a) = \{a\}$ (the language consisting of the one-symbol string)

**Induction step:** If $R$ is a regular expression by the induction step of the definition, then define $L(R)$:

- $L(S + T) = L(S) \cup L(T)$
- $L(ST) = L(S)L(T)$
- $L(T^*) = L(T)^*$
regexp examples

- $L(0 + 1) = \{0, 1\}$

- $L((0 + 1)^*)$ All binary strings over $\{0, 1\}$

- $L((01)^*) = \{\epsilon, 01, 0101, 010101, \ldots\}$

- $L(0^*1^*)$ 0 or more 0s followed by 0 or more 1s.

- $L(0^* + 1^*)$ 0 or more 0s or 0 or more 1s.

- $L((0 + 1)(0 + 1)^*)$ Non-empty binary strings over $\{0, 1\}$. 
example

$L = \{ x \in \{0,1\}^* \mid x \text{ begins and ends with a different bit} \}$
RE identities
some of these follow from set properties…
others require some proof (see 7.2.5 example)

- communitativity of union: $R + S \equiv S + R$
- associativity of union: $(R + S) + T \equiv R + (S + T)$
- associativity of concatenation: $(RS)T \equiv R(ST)$
- left distributivity: $R(S + T) \equiv RS + RT$
- right distributivity: $(S + T)R \equiv SR + TR$
- identity for union: $R + \emptyset \equiv R$
- identity for concatenation: $R\epsilon \equiv R \equiv \epsilon R$
- annihilator for concatenation: $\emptyset R \equiv \emptyset \equiv R\emptyset$
- idempotence of Kleene star: $(R^*)^* \equiv R^*$
product construction

$L$ is the language of binary strings over $\{0, 1\}^*$ with two $1$s in a row and an even number of $0$s.

Idea: $\delta((q_i, q_j), a) = (\delta(q_i, a), \delta(q_j, a))$
non-deterministic FSA (NFSA) example

FSA that accepts $L((010 + 01)^*)$
they’re equivalent:

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