

CSC236 fall 2016

regular expressions

Danny Heap

heap@cs.toronto.edu / BA4270 (behind elevators)

<http://www.cdf.toronto.edu/~csc236h/fall/>

416-978-5899

Using Introduction to the Theory of Computation,
Chapter 7



Outline

regular expressions

NFSAs

regular languages

notes



another way to define languages

In addition to the language accepted by DFSA $L(M)$
and set description $L = \{\dots\}$.

Definition: The regular expressions (regexps or REs) over alphabet Σ is the **smallest** set such that:

1. $\{\}$, ϵ , and x , for every $x \in \Sigma$ are REs over Σ
2. if T and S are REs over Σ , 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 — no strings!)
- ▶ $L(\varepsilon) = \{\varepsilon\}$ (the language consisting of just the empty string)
- ▶ $L(x) = \{x\}$ (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

- ▶ java REs
- ▶ command-line REs
- ▶ $L(0 + 1) = \{0, 1\}$
- ▶ $L((0 + 1)^*)$ All binary strings over $\{0, 1\}$
- ▶ $L((01)^*) = \{\varepsilon, 01, 0101, 010101, \dots\}$
- ▶ $L(0^*1^*)$ 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: $R\varepsilon \equiv R \equiv \varepsilon R$
- ▶ annihilator for concatenation: $\emptyset R \equiv \emptyset \equiv R\emptyset$
- ▶ idempotence of Kleene star: $(R^*)^* \equiv R^*$



non-deterministic FSA (NFSA) example

FSA that accepts $L((010 + 01)^*)$

convenient!



NFSAs are real

...you can always convert them to DFSA

Use **subset construction**, notes page 217



they're equivalent:

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



notes