Describing Finite State Machines

Murray Cole

A Notation for FSMs?

- FSMs are a useful tool (reactive systems are everywhere)
- graphical notation gets cumbersome, so need a “better” notation
- Regular Expressions do the job for acceptors (with real applications in Unix commands, text processing, compilers)

Regular Expressions

7(0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9)(0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9)

(a|b)*ca*

1(0 | 1)*1

Each of these defines a set of strings corresponding to a language

Syntax and Semantics

- any symbol from the input alphabet stands for itself
  (means the set containing just that symbol as a string)
- the symbol ϵ stands for the empty string
  (means the set containing just the empty string)
- the symbol $\emptyset$ stands for no strings at all
  (means the empty set)
- if R and S are RE's, RS stands for a sequence of an R then an S
  (means the set of strings which are formed from a string from R followed by a string from S)

Syntax and Semantics

- if R and S are RE's, $R \mid S$ stands for the choice of R or an S (means the set of strings which are in the set union of R and S)
- if R is a RE, R^* stands for zero or more repeats Rs (means the set closure of the set of strings which are in R)
- the operators have precedences, repeat > sequence > choice, and we can use parentheses '(' and ')' to group expressions (like arithmetic expressions in Java)

Examples

Write down all the strings in the language defined by (b | c)a(b | c)d

Write some of the strings in the language defined by (0* | 1)1(1 | 0*) and some strings which aren't

Give an informal description of the language defined by (a | b)*b

Write down a regular expression defining the language of As and Bs in which every string contains an odd number of Bs

Java Floating Point Numbers

Use italics (S, D, N etc) to stand for sub-expressions (not symbols)

$$
\begin{aligned}
S &= \epsilon \mid + \mid - \\
D &= \texttt{0} \mid \texttt{1} \mid \texttt{2} \mid \texttt{3} \mid \texttt{4} \mid \texttt{5} \mid \texttt{6} \mid \texttt{7} \mid \texttt{8} \mid \texttt{9} \\
N &= D^* \\
M &= S(DN.N \mid .DN) \\
E &= \texttt{E}SDN \mid \epsilon \\
F &= ME
\end{aligned}
$$

So, `+12.01E-34` is a valid constant while `-12E+5` is'nt

Variable Names

$$
\begin{aligned}
L &= \texttt{A} \mid \ldots \texttt{Z} \mid \texttt{a} \mid \ldots \texttt{z} \\
D &= \texttt{0} \mid \texttt{1} \mid \texttt{2} \mid \texttt{3} \mid \texttt{4} \mid \texttt{5} \mid \texttt{6} \mid \texttt{7} \mid \texttt{8} \mid \texttt{9} \\
I &= L(L \mid D)^*
\end{aligned}
$$

Algebraic Laws

Regular expressions are a kind of algebra
Like other algebras, there are rules which let us manipulate expressions (usually with the goal of simplifying them)

$$\emptyset \mid R = R = R \mid \emptyset \quad (1)$$

$$R \mid R = R \quad (2)$$

$$R \mid S = S \mid R \quad (3)$$

$$(R \mid S) \mid T = R \mid (S \mid T) \quad (4)$$

$$\epsilon R = R = R\epsilon \quad (5)$$

$$\emptyset R = \emptyset = R\emptyset \quad (6)$$

More Algebraic Laws

$$(RS)T = R(ST) \quad (7)$$

$$R(S \mid T) = RS \mid RT \quad (8)$$

$$(R \mid S)T = RT \mid ST \quad (9)$$

$$\emptyset^* = \epsilon \quad (10)$$

$$RR^* = R^*R \quad (11)$$

$$RR^* \mid \epsilon = R^* \quad (12)$$

$$(R \mid S)^* = (R^*S^*)^* \quad (13)$$

$$(RS)^*R = R(SR)^* \quad (14)$$

An example

Show that $0(10)^*1 \mid (01)^* = (01)^*$

(This is in the lecture note - you add the rules used)

$$
\begin{aligned}
0(10)^*1|(01)^* &= 01(01)^*|(01)^* \\
&= 01(01)^*|01(01)^*|\epsilon \\
&= 01(01)^*|\epsilon \\
&= (01)^*
\end{aligned}
$$

Power of Regular Expressions

- For every regular expression there is an equivalent FSM acceptor
- For every FSM acceptor there is an equivalent regular expression
- Even true for acceptors with the "extra" compositions like interleaving and intersection.
- Regular expressions are widely used in file and text processing, and in the definition and compilation of programming languages

Summary

- Regular expressions define sets of strings (languages)
  - sequence, choice, repetition
- We can manipulate them with algebraic rules
- They have the same power as FSM acceptors