Solutions to Assignment 2

April 2, 2001

Some of these solutions have ideas taken from the solutions which I received. So don't be surprised if your solution is here.

Describe the languages denoted by the following expressions: The description should be simple and in plain English. So a solution like "one or more zeroes followed by a 1 followed by one or more zeroes followed by ....." will not considered as a correct solution.

a) $0(0|1)^*0$: All strings beginning and ending with a zero of length at least 2. (Need to exclude the 0 string)

b) $((\epsilon|0)1^*)^*$: $((\epsilon|0)1^*)^* = (1^*|01^*)^* = (1^*|0)^* = (1|0)^*$. Language accepted is the set of all strings.

c) $(0|1)^*0(0|1)(0|1)$: The set of all strings where the third digit from the right is a 0.

d) $0^*10^*10^*10^*$: Set of all strings with exactly three 1's.

e) $(00|11)^*((01|10)(00|11)^*(01|10)(00|11)^*)^*$: Set of all strings with even number of 0's and even number of 1's

Write regular definitions for the following languages: Some of you have given context free grammars for these languages. The main difference between regular definitions and context free grammars, is that recursion is not allowed in regular definitions. Suppose you are defining a regular definition like:

$$\begin{aligned} a &\leftarrow \alpha \\ b &\leftarrow \beta \\ &\vdots \\ z &\leftarrow \zeta \end{aligned}$$

Every right hand side can use only the basic symbols and any others you have defined before. So you cannot use z or even b to define β. However, you can use a and any symbols of the base alphabet to define β.

Another common mistake was to give regular expressions which do not encompass the whole language or allow strings outside the language. One solution which I got for part (e) was "$d|d00d|\ldots|d99d$", where d is the solution to part (d). This on the face of it may seem correct. But it allows for strings like 011110, which have too many repetitions.

a) All strings of letters that contain the five vowels in order.

- $\Sigma \leftarrow (a|\ldots|z)$
- $\alpha_1 \leftarrow \Sigma^* a \Sigma^* e \Sigma^* i \Sigma^* o \Sigma^* u \Sigma^*$

b) All strings of letters in which the letters are in ascending order.

- $\alpha_2 \leftarrow a^* b^* \ldots z^*$

c) Comments consisting of a string surrounded by /* and */ without an intervening */ unless it appears inside the quotes " and ". This allows for weird strings like /*abc/*/, /*"abc/"/ab/*/. Anything inside the /* and */ can be broken into a sequence of blocks. Each block is either a normal block or a quoted block. A quoted block is any string whatsoever enclosed in quotes. A normal block is any string which does not have a quote and also does not have the substring */.

- $\Sigma \leftarrow$ all printable letters
- q $\leftarrow$ the double quote character
- oc $\leftarrow$ /*
- cc $\leftarrow$ */
- $A \leftarrow \Sigma - *, q$ i.e. every letter except '*' and the double quote
- $B \leftarrow \Sigma - /$ i.e. every letter except '/'
- $C \leftarrow *B$ i.e. all two letter strings starting with '*' and ending with a letter from B
- $\beta \leftarrow (A|C)^*$ i.e. strings over Σ without */ as a substring
- $\alpha_3 \leftarrow$ oc $(\beta | q\Sigma^* q)^*$ cc

d) All digit strings with no two equal consecutive digits. Let β_i denote the regular expression for all non-empty digit strings with no equal consecutive digits, if the digits were only from the set $\{0, 1, \ldots, i\}$. Then

- $\beta_1 \leftarrow (01)^*(0|010) \,|\, (10)^*(1|101)$
- $\beta_{k+1} \leftarrow \beta_k | [\beta_k ? (d\beta_k)^* d \beta_k ?]$, where d = k+1'st digit.
- $\alpha_4 \leftarrow \beta_9 ?$

e) All strings with at most one repeated digit. Note that every such string if broken at the repeating digit, will break into two pieces without any repeating digit. Conversely the concatenation of two strings without any repeating digit, can have at most one repetition (Since the empty string is part of α_4, one letter strings do not cause problems).

- $\alpha_5 \leftarrow \alpha_4 \alpha_4$

f) Set of all strings with even number of 0's and odd number of 1's.

- $a \leftarrow (00|11)$
- $b \leftarrow (01|10)$
- $\delta \leftarrow a^*(ba^*ba^*)^*$, even number of b's, hence even number of 0's and 1's (Why?).
- $\beta \leftarrow a^* b \delta$, odd number of b's, hence odd number of 0's and odd number of 1's. Also need to check that it captures all strings with odd number of 0's and odd number of 1's. Hint: Consider the shortest prefix of the string which has an odd number of 1's and odd number of 0's, and argue that this prefix must end with either a 01 or a 10.

- $\alpha_6 \leftarrow 1\delta|0\beta$

h) All binary strings which do not contain the substring 011. Equivalently but for the first block of 1's every 1 must be preceded by a 0, i.e. all strings over the alphabet 0,01.

- $\alpha_8 \leftarrow 1^*(0|01)^* = 1^*(01?)^* = 1^*(0^+1?)^*$

i) All binary strings that do not contain the subsequence 011. Equivalently all strings which have at most one 1 after the first 0.

- $\alpha_9 \leftarrow 1^*0^*1?0^* = 1^*0^*(10^*|\epsilon)$

Prove that L = $\{a^n b^n : n \geq 0\}$ is not regular Proof: If r is a regular expression which does not make use of the "*" operator, then it can be expanded out in full, and then it becomes a bunch of strings seperated by "|". In this case the language accepted by r is finite. But L is infinite. So if L were regular and r is the regular expression for it, then r must contain a subterm s^* for some regular expression s. Now let w be a string, accepted by r and break it up into three pieces x, y, z so that y corresponds to s^* part, and y is not the empty string. Now, since xyz is satisfied by the regular expression, so is $xyyz$. Since every string of L has all its a's before its b's, y cannot have both a and b in it. So y consists of only a's or only b's. In either case $xyyz$ does not have equal number of a's and b's, since $w = xyz$ had an equal number of a's and b's.