Programming Language Concepts, CS2104
Tutorial 2 (9 September 2011)
(All students must prepare/attempt in advance.)

Exercise 1

Consider the following Ruby regular expressions:

```
(a[bcd]*(aa)+){2,4}
(a|bc)(d?|e)*
(((a|bc)d+)e)+
```

For each of these regular expressions, give 3 strings that match it. Each string must be of length greater than 10 characters. Moreover, for each of these regular expressions, give a string that does not match it, and indicate the first character in the string where the matching procedure becomes aware of the fact that a match is not possible.

Test your answers at rubular.com.

Exercise 2

For each of the above regular expressions, define a DFA that accepts the language specified by it.

.

Exercise 3

For each of the automata defined in your previous solution, define a regular grammar that generates the same language.

Exercise 4

Write a C procedure that implements the automaton equivalent to the third regular expression of Exercise 1.

Exercise 5

Consider the following incomplete implementation of a DFA.

```
#define N … // fill in a value here
#define M … // fill in a value here
#define Final … // fill in a value here
int t[N][M] = { …/* fill in initial values for array here */ … } ;

int accept (char *s) {
  int state = 0 ;
  while ( *s != '\0' ) {
    state = t[state][*s] ;
    s ++ ;
    if (state == -1 ) return 0 ; // reject
  }
  if ( state == Final ) return 1 ; // accept
  return 0 ; // reject
}
```

Fill in the blanks so that the procedure accepts the language of the second regular expression of Exercise 1.