CS 536

Midterm Exam

Thursday, April 2, 1998

12:45 — 2:15 PM

1325 CSST

Instructions

Answer question #1 and any three others. (If you answer more, only the first four will count.) Point values are as indicated. Please try to make your answers neat and coherent. Remember, if we can't read it, it's wrong. Partial credit will be given, so try to put something down for each question (a blank answer always gets 0 points!).

1. (1 point)
   Easter eggs, Easter bunnies and Easter parades are associated with what spring holiday?

2. (33 points)
   Some programming languages support *compilable comments*. A compilable comment is an ordinary multi-line comment that contains a special marker in the comment text. For example, in CSX ordinary multi-line comments begin and end with a `##`. Assume a CSX compilable comment contains the character `!` just after the initial `##` and just before the terminating `##`. (For example, `##! debug = true; !##`.) Ordinary (non-compilable) comments contain a non-`!` just after the initial `##` and just before the terminating `##`.

   When a compiler is called, a special option can be set that tells the compiler to compile the contents of compilable comments (after stripping the comment delimiters). As usual, non-compilable comments are ignored after scanning. If the compiler option is not set, all comments (including compilable ones) are ignored after scanning.

   Assume the predicate `compilableCommentsOn()` can be called to determine if the option that requests processing of compilable comments is currently set. How would you change your CSX scanner to support compilable comments?

3. Write regular expression definitions for the following token classes:

   (a) (11 points)
   Unsigned integer literals that contain no unnecessary leading zeroes. That is, `0`, `10`, `123`, and `1000001` are allowed, but `01`, `007` and `000` are not allowed.

   (b) (11 points)
   A CSX-style single line comment that does not contain the character pair `##` anywhere within its text. Thus
   `// We're #1!!`
   is OK, but

```
//  ##  This is really a comment!
```
is not allowed.

(c) (11 points)
A CSX-style string literal that contains exactly one newline in its text (and thus spans exactly two lines). For example
```
 println("Hello
            World!");
```

4. (33 points)
Let s be a string. Define delete1(s) to be a function that deletes a single character from s. If s is n characters long, delete1 returns a set of n (or fewer) strings since there are n characters that might be deleted in a string of length n.

For example, delete1(abc) = { bc, ac, ab }. delete1 applied to a set of strings is the union of delete1 applied to members of the set. Hence delete1(ab, de) = { b, a ,e, d }.

Let R be any regular set (my choice). Show that delete1(R) is a regular set.

5. (a) (17 points)
Let S_1 and S_2 be sets of strings. Define $S_1 - S_2$ to be the set of all strings in S_1 that aren't in S_2. Assume R_1 and R_2 are regular sets. Show that $R_1 - R_2$ is also a regular set.

(b) (16 points)
Let FA be any deterministic finite automaton (my choice). Explain how to test whether all the strings FA accepts are exactly two characters in length.

6. (33 points)
Let H1 be a hash table of size 2×3×5×7 = 210. Let H2 be a hash table of size 211 (which is prime).

A hash function is said to be *uniform* if given a large number of randomly selected identifiers, *on average* each position in the table is chosen with equal probability. A hash function that is not uniform is said to be *non-uniform*.

Give three hash functions, f1, f2, and f3, with the following properties:

f1 is non-uniform when used with either H1 or H2.

f2 is non-uniform when used with H1, but is uniform when used with H2.

f3 is uniform when used with either H1 or H2.