Compilers — Homework 6
Regular expressions and scanning

Due Monday, 5 Nov 2012, 20:00

This is a written assignment. For this assignment, each person should submit their own solution. You are free to talk to other students, but (1) you must acknowledge any discussions and (2) you must write up your solutions on your own.

1. [80 pts] For each of the following language descriptions, write a regular expression that matches exactly the set of strings in the language.

   (a) [10 pts] Strings of the letters 'a' and 'b', beginning and ending with 'a'.

   (b) [10 pts] Identifiers containing two lowercase words separated by '2'. For example:

   ```
   string2int
   int2string
   list2map
   ```

   (c) [15 pts] The set of all phone numbers in Switzerland, formatted like any of the following examples:

   ```
   058 666 43 10
   41 58 666 43 10
   41 (0)58 666 43 10
   ```

   Only the `41` country code and the leading (optional) `0` should be hard-coded into the regular expression. The *national destination code* (the `58` above) cannot start with a `0`.

   (d) [15 pts] Hexadecimal integers in Java: `0`, then either `x` or `X`, then one or more hexadecimal digits (`0` through `9`, `a` through `f`, and `A` through `F`).

   (e) [15 pts] Comments in the programming language SML: comments begin with `(*` and end with `*)`. Comments cannot be nested; that is, in the following example, the comment ends at the first `*)`, not at the second.

   ```
   (* this is a (* comment *) and this is not *)
   ```

   (f) [15 pts] Decimal floating point numbers. Your regular expression should handle all of the following cases (with different digits, obviously):

   ```
   123.456
   123.
   123.0
   0.123
   .123
   123.456e7
   123.E7
   123.0e+7
   0.123E-7
   .123e-78
   ```

   Floats cannot start with a `0` followed immediately by another digit. For instance `01.0` is not a legal float.

2. [10 pts] In Fortran, a Hollerith constant is a string literal that begins with a positive integer n, an 'H', and then a sequence of n characters. For instance, the string "abcdefghijklm" can be written using the Hollerith constant 13Habcdefghijklm. Can you write a regular expression for Hollerith constants? If not, why not?

3. [10 pts] Keywords and identifiers overlap in the syntax of many languages. For instance, C identifiers are defined as strings that start with a letter (including '_') and are followed by zero or more letters or digits. All keywords in C (**if**, **while**, etc.) fit this definition but are *reserved*—they are not allowed to be used as ordinary identifiers. Two common ways to implement keywords in a scanner are:

   - Encode each keyword directly into the scanner's state machine (i.e., the implementation of the DFA).
   - Encode only identifiers into the state machine. When an identifier is recognized, lookup the string in a *keyword table* (usually implemented as a hash table or a binary search tree. If the string is in the table, the scanner returns the appropriate keyword token; otherwise, it returns an identifier token.

   Is one approach better than the other? Which is easier to implement if implementing a scanner by hand? When using a scanner generator like PLY? Which approach do you think results in a faster scanner and why? Think about how a DFA is implemented.