CS 606 Compiler Constructions
Solution of Assignment # 01
Spring 2011

Total Marks = 20

Deadline

Your assignment must be uploaded / submitted before or on **April 18, 2011**

Upload Instructions
Please view the assignment submission process document provided to you by the Virtual University.

Rules for Marking
Please note that your assignment will not be graded if:
• It is submitted after due date
• The file you uploaded does not open
• The file you uploaded is copied from someone else or from internet
• It is in some format other than **.doc**

Note: Material that is an exact copy from handouts or internet would be graded zero marks. Your solution should consist of the material found through different sources and written in your own words.

Assignment Statements:

Question 1:
Draw the FA for a language which consists of the tokens:
<, <=, =, >=, >, <>, (,), +, -, *, /,:=, ;, identifier, keywords, constants and literals (which are enclosed in apostrophes). Comments, which begin with /* and end with a */, and blanks are ignored by the scanner \9that is blanks are treated as blank separator only).

Solution:
Let us take assumption that "not" preceding a character indicates that all input characters other than one listed can cause the indicated state transition. The arc labeled "error" indicates that an invalid character has been found in the source text.

The FA is shown below as follows:

$M = (Q, \Sigma, \delta, q_0, F)$

$Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22\}$
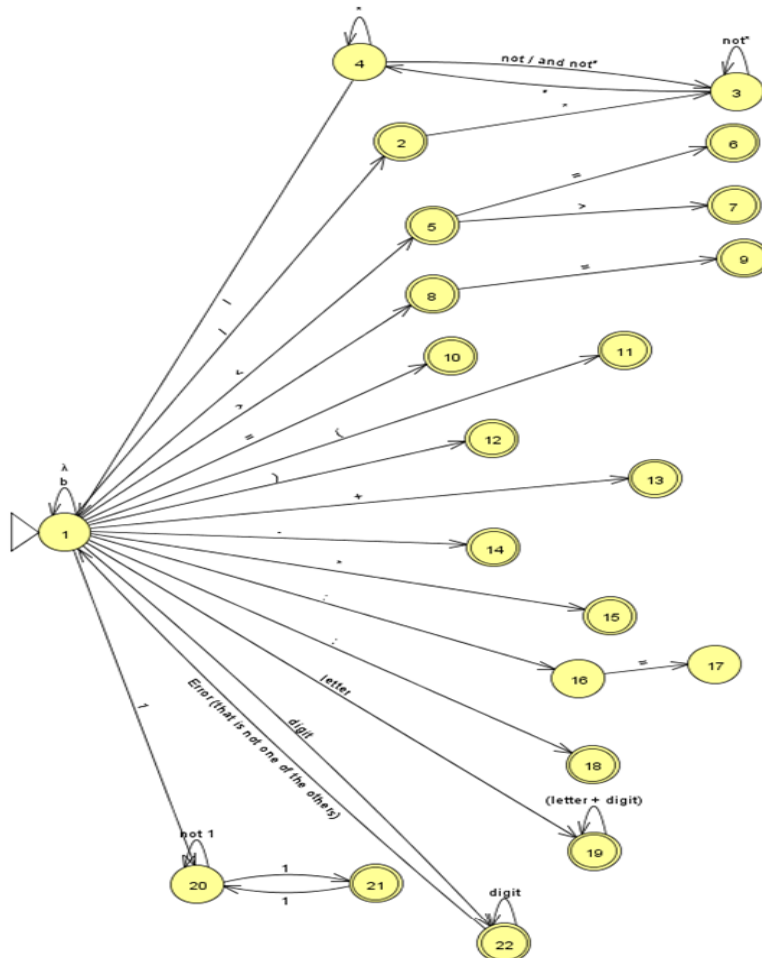
$\Sigma = \{<, <, =, >, (,), +, -, *, /, :, =, ;,$ letter, digit, not, blank$\}$

(where letter = $\{a, b, \ldots, z\}$, digit = $\{0, \ldots, 9\}$, blank = $\{b\}$)

Note: Literals begin and end with an apostrophe (').

$q_0 = \{1\}$

F = {2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22}

δ is transition function.

Question 2:

Design a context-free grammar which generates the mathematical expressions for '+' and '-' operators. Also apply this grammar on algebraic expression "a – b + c" and construct the parse tree for this string

Solution:

Let CFG be G = {V$_n$, V$_t$, S, P}

Where   V$_n$ = {<goal>, <expression>, <term>, factor>}

                V$_t$ = {a, b, c, -, +}

                S = <goal>

And productions are defined as follows:

                <goal> → <expression>

                <expression> → <term>/<expression> - <term>

<term> → <factor> / <term> + <factor>

<factor> → a/b/c

Let us apply above grammar on algebric expression a – b + c.

<goal>→ <expression>

→ <expression> - <term>

→ <term> - <term>

→ <factor> - <term>

→ a - <term>

→ a - <term> + <factor>

→ a - <factor> + <factor>

→ a – b + <factor>

→ a – b + c

The parse tree for above string as follows: