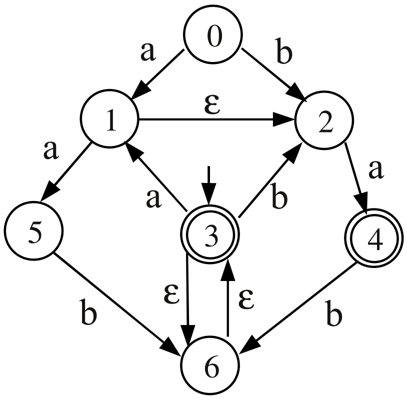
July 2, 2012

Compilers

Surname, Name	
Email	

1. After generating the DFA equivalent to the following NFA, specify the BNF expressing the regular language relevant to the DFA.



2. Given the following table of operators, with increasing precedence from top to bottom,

Operators	*Associativity*
`&&`, `||`	left
`==`, `!=`	nonassoc
`+`, `-`	right

specify the BNF of a corresponding language for expressions based on the following requirements:

- parentheses can be used for arithmetic operators (`+`, `-`);
- parentheses cannot be used for logical operators (`&&`, `||`);
- atomic elements of the expression are either constants or identifiers.

3. After constructing the complete parsing automaton, check whether the following grammar is LR(1):

A → *B* | **id** | **num**
B → *C* **num** | **id**
C → **id** | **num** | **ε**

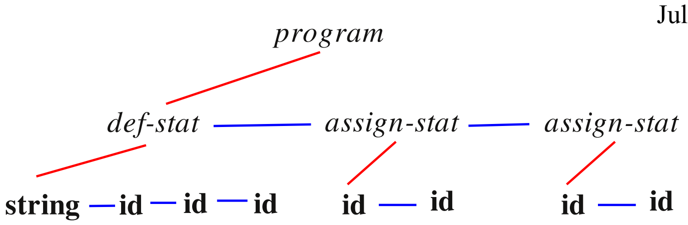
4. Codify in Yacc the generator of the binary abstract trees relevant to the language defined by the following BNF:

program → *stat-list*
stat-list → *stat* **;** *stat-list* | *stat* **;**
stat → *def-stat* | *assign-stat*
def-stat → *id-list* **:** *type*
id-list → **id** **,** *id-list* | **id**
type → **int** | **string** | **bool**
assign-stat → **id** **=** **id**

```
x, y, z: string;
x = y;
y = z;
```

based on the following abstract EBNF:

program → { *def-stat* | *assign-stat* }$^+$
def-stat → (**int** | **string** | **bool**) { **id** }$^+$
assign-stat → **id id**



5. With reference to the BNF of point 4, specify the attribute grammar based on the following semantic constraints:

- all definitions shall precede all assignments;
- variable names are unique;
- the two variables involved in assignment shall exist and be of same type;
- a variable cannot be assigned with itself.

Notes:

- the lexical value of identifiers is stored in the `lexval` field of the tree node;
- a symbol table is used to catalog variables by means of the following functions:
  `void insert(name, type)`
  `Type lookup(name)`: returns the type of variable `name` (`INT`, `STRING`, `BOOL`) if cataloged, otherwise `NULL`;
- no other global variables can be used;
- in case of semantic error, function `error(string message)` is called, which prints the relevant error `message` before terminating the analysis.

6. A language is defined by the following EBNF:

program → { *stat* }$^+$
stat → *assign* | *loop* | **break**
assign → (*indexpr* | **id**) = *expr*
indexpr → **id [** *expr* **]**
expr → *indexpr* | **num** | **id**
loop → **while** *expr* **do** { *stat* }$^+$

Assuming that the abstract tree is binary (pointers: `child`, `brother`) and irrelevant syntax sugar is not stored, we ask to codify a procedure for P-code generation based on the following requirements:

a) the **break** statement breaks the loop in which it appears;

b) the language of the P-machine includes the following set of instructions:

- `LDA` *id*: load address of variable *id*;
- `LOD` *id*: load value of variable *id*;
- `LDI` *value*: load integer *value*;
- `LAB` *label*: create *label*;
- `GOF` *label:* conditional jump (to false);
- `GOT` *label*: unconditional jump;
- `IND` *offset*: indirect load;
- `IXA` *scale*: indexed address;
- `STO`: store.

c) the size of array elements is 4;

d) the (overloaded) auxiliary function `emit(string operator [, string operand])` is used to print an instruction of the P-machine.