

Substitutiva de Construção de Compiladores.  
Turma A. Terça Feira.  
Prof. José de Oliveira Guimarães.

1 (5.0) O comando for do compilador 10 tem o seguinte formato:

```
for variable = expression to expression do statement
```

Um analisador sintático incompleto para este comando é dado abaixo. Copie este analisador para a folha de respostas acrescentando o restante da análise sintática, o código para a criação do objeto da ASA para o for e a análise semântica. Faça também a classe da ASA ForStatement com o método genC para gerar código. Assuma que apenas inteiros podem ser variáveis de comando for.

Você talvez usará a classe Type. Esta possui subclasses IntegerType, CharType e BooleanType. Type possui variáveis estáticas integerType, charType e booleanType. Cada uma delas aponta para o único objeto do programa de cada subclasse de Type. A tabela de símbolos é referenciada pela variável st e possui métodos put(Object name, Object value) e Object get(Object name). O primeiro insere um símbolo e o segundo faz a busca por símbolo.

```
ForStatement forStatement() {
    nextToken();
    if ( lexer.token != Symbol.IDENT )
        error();
    // nome do identificador
    String name = lexer.getValue();
    nextToken();

    nextToken();
    Expr e2 = expr();
    if ( lexer.token != Symbol.DO )
        error();
    nextToken();
    Statement s = statement();
}
```

4. (2.5) Dado o trecho de código

```
for i = 1 to n do
    write(i);
```

faça:

- a) a representação gráfica da ASA deste trecho;
- b) o código em Java que produz código equivalente.

3.(2.5) Copie o analisador léxico abaixo na folha de respostas e o modifique para que ele reconheça comentários como os de C++, delimitados por /\* e \*/. Não se esqueça de sinalizar o erro “comentário aberto e não fechado”. Lembre-se de que após reconhecer o comentário, nextToken() deve ser recursivamente chamado.

```
public void nextToken() {
    char ch;
```

```
while ( (ch = input[tokenPos]) == ' ' || ch == '\r' ||
        ch == '\t' || ch == '\n')
    tokenPos++;
if ( ch == '\0') token = new Symbol(Symbol.EOF);
else // coloque o seu código aqui.
    if ( Character.isLetter( ch ) ) {
        // get an identifier or keyword
        ...
    }
    else {
        ...
    }
}
```