Universidade Federal de Viçosa
Centro de Ciências Exatas e Tecnológicas
Departamento de Informática
INF340 – Linguagens de Programação
11/05/2018

Trabalho Prático 2

Entregar até 29/05/18

O objetivo deste trabalho é exercitar os métodos formais de descrição sintática e semântica usando uma pequena linguagem de programação como exemplo que, sob a inspiração de Gordon (1979)[*], chamaremos de TINY. A sintaxe primária de TINY é apresentada abaixo em EBNF. Os metassímbolos do EBNF que aparecem no lado direito de uma produção estão escritos na fonte negrito de máquina de datilografia. As palavras chaves de TINY estão na fonte romana em negrito. O aluno será responsável então por apresentar uma gramática de atributos para TINY, de acordo com as regras complementares que definiremos. Após, o aluno descreverá a semântica (dinâmica) de TINY por meio da semântica denotacional.

Sintaxe de TINY

<programa> → **program** id **begin** **[** <decl> **]** **{** <comando> **}** **end** id
<decl> → **{** **(** **const** id = <expr> **{** , id = <expr> **}** ;
| **var** id = <expr> **{** , id = <expr> **}** ; **)** **}**
<comando> → <atribuição> **|** <composição> **|** <seleção> **|** <iteração> **|** **skip**
<atribuição> → <var> **:=** <expr>
<composição> → <comando> **{** ; <comando **}**
<seleção> → **if** <expr> **then** <comando> **else** <comando> **end**
<iteração> → **while** <expr> **do** <comando> **end**
<var> → id
<expr> → const **|** id **|** <expr_binária> **|** <expr_unária> **|** (<expr>)
<expr_binária> → <expr_esq> <op_binário> <expr_dir>
<expr_esq> → const **|** id
<expr_dir> → const **|** id
<op_binário> → + **|** * **|** / **|** **mod** **|** **or** **|** **and** **|** = **|** <
<expr_unária> → <op_unário> <expr>
<op_unário> → – **|** **not**

[*] GORDON, Michael J. C. **The denotational description of programming languages: an introduction.** New York, USA: Springer-Verlag, 1979.

Características extras de TINY

Os *tokens* id e const são, respectivamente, nomes e constantes literais. Os nomes são cadeias de letras, dígitos, ou sublinhas em que o primeiro caractere tem que ser letra. As constantes literais são os números decimais inteiros ou os valores-verdade **true** ou **false**. Os tipos das constantes nomeadas e das variáveis são estabelecidos na hora de suas declarações pelas expressões do lado direito do símbolo =. Todos os nomes têm que ser declarados antes de serem usados nos comandos. No comando de atribuição, o tipo da variável (do lado esquerdo do símbolo de atribuição) tem que ser o mesmo da expressão (do lado direito do símbolo de atribuição) e é esta que estabelece o tipo esperado da variável. O tipo efetivo da expressão em si é dado pelo tipo dos operadores: operadores aritméticos estabelecem o tipo inteiro para a expressão e operadores lógicos e relacionais estabelecem o tipo booleano. Os operandos de uma expressão relacional têm que ser do mesmo tipo. Os operadores binários têm a mesma precedência e os operadores unários têm maior precedência que os binários. O nome do programa (após a palavra chave **program**) tem que ser o mesmo que aparece após a palavra chave **end**.

O comando *skip* não faz nada, isto é, não muda o estado da máquina durante sua execução. Os outros comandos têm a semântica usual que se espera deles. O tipo das expressões que aparecem na seleção e na iteração tem que ser booleano.

Exemplo 1 de programa em TINY:

```
program mdc
begin
    var a = 770, b = 441, t = 0;
    while not (b = 0) do
        if (b < a) or (b = a) then a ≔ a mod b
        else skip end;
        if 0 < a then b ≔ b mod a
        else
            t ≔ a; a ≔ b; b ≔ t
        end
    end
end mdc
```

Exemplo 2 de programa em TINY:

```
program fatorial
begin
    const lim_sup = 2147483647;
    var n = 10;
    var cont = 0, fat = 1, overflow = false;
    while (cont < n) and (not overflow) do
        cont ≔ cont + 1;
        if (0 < fat) and (fat < lim_sup) then fat ≔ fat * cont
        else overflow ≔ true end
    end
end fatorial
```

Exercícios a ser entregues:

1. Escreva uma gramática de atributos para TINY, estabelecendo os atributos sintetizados e herdados necessários e os predicados necessários para a verificação do que for preciso. Os atributos intrínsecos serão inseridos ou buscados na tabela de símbolos por meio das funções *insere*(*x*) e *pesquisa*(*x*), respectivamente. O argumento *x* é uma cadeia contendo a chave de pesquisa/inserção.
2. Escreva as funções de mapeamento denotacional para os constructos da linguagem.