UNIP – Universidade Paulista	
Curso.....:	Bach. em Ciência da Computação
Disciplina..:	Compiladores e Computabilidade
Professor...:	Leandro Carlos Fernandes

-:: Lista de Exercícios #3 ::-

Tópico: Análise Semântica; Gramáticas de Atributos e Tab. de Símbolos

1) A respeito da análise semântica responda cada uma das questões a seguir:
   a) Quais verificações e/ou tarefas ficam sob sua responsabilidade?
   b) Em linhas gerais, qual a diferença entre semântica estática e semântica de tempo de execução?
   c) Qual o papel da Tabela de Símbolos no processo de análise?

2) Dada a Gramática de Atributos abaixo, responda:

G = ({N, I, B}, {0, 1, . }, P, N)

P:

$N \rightarrow I_1 . I_2$	$N.v := I_1.v + I_2.v$ $I_1.p := 0$ $I_2.p := -I_2.l$
$N \rightarrow I$	$N.v := I.v$ $I.p := 0$
$I \rightarrow B$	$I.v := B.v$ $I.l := 1$ $B.p := I.p$
$I_0 \rightarrow I_1 B$	$I_0.v := I_1.v + B.v$ $I_0.l := I_1.l + 1$ $I_1.p := I_0.p + 1$ $B.p := I_0.p$
$B \rightarrow 0$	$B.v := 0$
$B \rightarrow 1$	$B.v := 2^{-B.p}$

   a) Para que serve uma gramática deste tipo?
   b) Para qual subfase do processo de análise ela é importante?
   c) O que são os chamados *atributos* e quais deles são s*intetizados* e quais são *herdados*?
   d) Qual o grafo de dependência para esses atributos e em que ordem eles devem ser computados?
   e) Calcule os valores dos atributos para a cadeia: `10.01`

3) Dada a gramática a seguir:

G = ({S,E,F,X}, {0,1,2,3,4,5,6,7,8,9,+}, P, S)

P:

$S \rightarrow E$	$S.v := E.v$
$E \rightarrow FX$	$E.v := X.r$ $X.d := F.v$
$X_1 \rightarrow +FX_2$	$X_1.r := X_2.r$ $X_2.d := X_1.d + F.v$
$X \rightarrow \varepsilon$	$X.r := X.d$
$F \rightarrow 0$	$F.v := 0$
$F \rightarrow 1$	$F.v := 1$
$F \rightarrow 2$	$F.v := 2$
$F \rightarrow 3$	$F.v := 3$
$F \rightarrow 4$	$F.v := 4$
$F \rightarrow 5$	$F.v := 5$
$F \rightarrow 6$	$F.v := 6$
$F \rightarrow 7$	$F.v := 7$
$F \rightarrow 8$	$F.v := 8$
$F \rightarrow 9$	$F.v := 9$

   a) Quais atributos podem ser classificados como sintetizados e quais como herdados?
   b) De acordo com a gramática dada monte a árvore sintática para a cadeia `3+2+9`
   c) Calcule o valor dos atributos de cada nó da árvore dada no item anterior.

4) A gramática de atributos a seguir permite a verificação de tipos para programas similares aos de **Pascal:**

$P \rightarrow$ **program** D **begin** C **end.**	C.env := D.t P.ck := C.ck

D → **var id :** T	D.t := *SetOf*(**id**.txt, T.t)
D → D **;** D	D_0.t := D_1.t ∪ D_2.t
T → **integer**	T.t := *inteiro*
T → **char**	T.t := *caractere*
T → **bool**	T.t := *booleano*
T → **array [num] of** T	T_0.t := *array*(**num**.val, T_1.t)
C → **id** := E	E.env := C.env C.ck := (lookup(**id**, E.env) = E.tp)
C → **if** E **then** C **else** C	E.env := C_0.env C_1.env := C_0.env C_2.env := C_0.env C_0.ck := (E.tp = *booleano*) ∧ C_1.ck ∧ C_2.ck
C → **while** E **do** C	E.env := C_0.env C_1.env := C_0.env C_0.ck := (E.tp = *booleano*) ∧ C_1.ck
C → **read(id)**	C.ck := *verdadeiro*
C → **write(** E **)**	E.env := C.env C.ck := (E.tp = *booleano* ∨ E.tp = *caractere* ∨ E.tp = *inteiro*)

E → **id**
E → **id[** E **]**
	E.tp = lookup(**id**, E.env)
	E_1.env := E_0.env
	if E_1.tp = *inteiro*
	  then E_0.tp := lookup(**id**, E.env)
	  else E_0.tp := *erro*

E → **num**	E.tp := *inteiro*
E → **truth-value**	E.tp := *booleano*
E → **character**	E.tp := *caractere*
E → **(** E **)**	E_1.env := E_0.env E_0.tp := E_1.tp

E → E **+** E	E_1.env := E_0.env
	E_2.env := E_0.env
	if E_1.tp = *inteiro* ∧ E_2.tp = *inteiro*
	  then E_0.tp := *inteiro*
	  else E_0.tp := *erro*

E → E **or** E	E_1.env := E_0.env
	E_2.env := E_0.env
	if E_1.tp = *booleano* ∧ E_2.tp = *booleano*
	  then E_0.tp := *booleano*
	  else E_0.tp := *erro*

E → **not** E	E_1.env := E_0.env ***if*** E_1.tp = *booleano*

then E_0.tp := *booleano*
else E_0.tp := *erro*

a) Faça um programa simples e válido nesta linguagem.
b) Calcule os atributos do programa dado no item anterior.

5) Supondo o trecho de código em C dado abaixo, responda:

```
int f(int a, float b) {
   float r = a + b;
   return r;
}
void main() {
   int a = 2, b = 3, ri;
   float rf;
   ri = f(a,3);
   rf = f(2,b);
}
```

a) Qual a tabela de símbolos resultante de sua análise?
b) Em quais pontos do código são realizadas as operações de *consulta* a tabela de símbolos?
c) Em quais pontos do código são realizadas as operações de *inserção* na tabela de símbolos?
d) Haveriam erros semânticos a serem reportados? Se sim, quais e por quê?

6) Qual seria o conteúdo da tabela de símbolos resultante da análise do seguinte trecho de código:

```
public class C {

   private int x;

   class I {
      public static float square() {
          return x * x;
      }
   }

   public void setX(int valor) {
      this.x = valor;
   }

   public int getX() {
      return this.x;
   }

   public float getXQuadrado() {
      return I.square();
   }
}
```

7) Explique por que empregar gramáticas S-atribuídas é conveniente com parsers ascendentes.