

UNIVERSIDADE DO MINHO

GT & GA

MESTRADO INTEGRADO EM ENGENHARIA
INFORMÁTICA

GRAMÁTICAS NA COMPREENSÃO DE SOFTWARE

1º SEMESTRE 2020/21

Aluno:

Etienne Costa A76089

Docente:

Pedro Rangel Henriques

8 de novembro de 2020

1 Resumo

O presente trabalho tem como principal objectivo esmiuçar as principais diferenças entre as gramáticas de atributos e gramáticas tradutoras de modo a sintetizar os conhecimentos adquiridos na unidade curricular de **Gramáticas na Compreensão de Software**.

Conteúdo

1	Resumo	1
2	Introdução	3
3	Preliminares	4
4	Gramática	5
4.1	Gramática Tradutora	6
4.2	Gramtica de Atributos	7

2 Introdução

O relatório apresentado diz respeito ao segundo trabalho proposto no âmbito da unidade curricular de gramáticas na compreensão de software. O universo de discurso recai sobre gramáticas, concretamente sobre gramáticas de atributos e gramáticas tradutoras.

3 Preliminares

Para o desenvolvimento deste trabalho foi necessário alguns conhecimentos previamente adquiridos sendo que este conhecimento foi absorvido durante as aulas de Processamento de Linguagens, e também com alguma pesquisa individual. De seguida, apresenta-se alguns conceitos fundamentais para a compreensão e realização deste trabalho.

4 Gramática

Em teoria das linguagens formais, uma gramática formal ou simplesmente gramática é um conjunto finito de regras que quando aplicadas sucessivamente , geram **palavras** . O conjunto de todas as palavras geradas por sua vez define a **linguagem**.

Portanto formalmente podemos definir um gramática como $G = (T, N, P, S)$ onde:

- T - Conjunto finito de símbolos terminais da linguagem (alfabeto).
- N - Conjunto finito dos símbolos não terminais da gramática.
- P - Conjunto de produções ou regras de derivação da gramática.
- S - Axioma da gramática.

É Importante salientar as seguintes características que este tuplo apresenta :

- $T \cap N = \{\}$
- $S \in N$
- Para cada $p \in P \mid p : X_0 \rightarrow X_1 \dots X_i \dots X_n$

Sendo que X_0 aparece no que é chamado **Left Hand Side** da produção aonde só se encontram presentes símbolos não terminais(N).

E os restantes elementos representados acima estão posicionados no que é chamado **Right Hand Side** da produção aonde encontram-se elementos que pertencem a união de símbolos terminais e não terminais ($T \cup N$).

De seguida é apresentado um pequeno exemplo que especifica uma GIC para listas de inteiros em Haskell :

```
ListaHaskell : ListaHaskell Haskell
| Haskell
;
Haskell : '['Lists']'
| '['']'
;
Lists : Lists ',' Valor
| Valor
;
Valor : num
| num dots num
| Haskell
;
```

4.1 Gramática Tradutora

Na tradução dirigida pela sintaxe a tradução é feita através de reconhecedores do tipo descendente (**Top-Down**), ou ascendente (**Bottom-Up**), baseados em gramáticas independentes do contexto na qual se incluem acções semânticas. Esse tipo de gramáticas são designadas por **gramáticas tradutoras**. Com a utilização deste tipo de gramáticas efectua-se acções semânticas a medida que se faz o reconhecimento.

Portanto formalmente podemos definir uma gramática tradutora como $GT = (T, N, S, A, P)$ onde:

- T - Conjunto finito de símbolos terminais da linguagem (alfabeto).
- N - Conjunto finito dos símbolos não terminais da gramática.
- S - Axioma da gramática.
- A - Conjunto de acções semânticas.
- P - Conjunto de produções ou regras de derivação da gramática.

No que concerne as gramáticas tradutoras as produções têm a si associadas um conjunto de acções semânticas, então:

Para cada $p \in P$ | $p : X_0 \rightarrow X_1 \dots X_i \dots X_n$ a

Em que a pertence ao conjunto de A e especifica a acção semântica a ser executada após o reconhecimento dos símbolos pertencentes ao **Right Hand Side**, isto é, o conjunto de símbolos terminais e não terminais.

De seguida é apresentado um pequeno exemplo que exemplifica uma gramática tradutora:

```
I : Lista {printf("somatorio:%d",somatorio);}
;
Lista : Elem
| Elem ',' Lista
;
Elem : num {somatorio+= $1}
| palavra
;
```

4.2 Gramática de Atributos

As gramáticas de atributos são uma extensão das gramáticas independentes de contexto, sendo que nestas surge a possibilidade de definir localmente variáveis nas produções.

Portanto formalmente podemos definir uma gramática de atributos como $GA = (GIC, A, RC, CC, RT)$ onde :

- GIC - Gramática Independente de Contexto.
- A - Conjunto dos atributos de todos os símbolos da gramática.
- RC - Conjunto das regras de cálculo dos atributos em todas as produções das gramáticas.
- CC - Conjunto das condições de contexto em todas as produções da gramática.
- RT - Conjunto das regras de tradução em todas as produções da gramática.

A é a união dos $A(X)$ para cada x pertencente $(T \cup N)$ e representa o conjunto de todos os atributos. Sendo que os Atributos $A(X)$ de cada símbolo dividem-se em dois subconjuntos disjuntos :

- $AH(X)$ - Conjunto dos atributos herdados do símbolo X , sendo estes propagados para um nível abaixo.
- $AS(X)$ - Conjunto dos atributos sintetizados do símbolo X , sendo estes propagados para um nível acima.

De seguida é apresentado um exemplo que especifica uma gramática tradutora :

```
list : '[' content ']' { System.out.println("Somatorio: " + $content.somatorio);}
;
content returns [int somatorio=0]:
| item {$somatorio += $item.num;} (',' item {$somatorio += $item.num;})*
;
item returns [int num]: NUM {$num = $NUM.int;}
| WRD {$num = 0;}
;
```