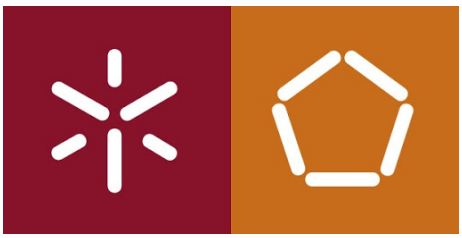
Gramática de Atributos

TPC2

Grupo 10 de GCS



Mestrado em Engenharia Informática
Universidade do Minho

1 Gramática Independente do Contexto

As Gramáticas Independente do Contexto, ou **GIC**, é uma gramática formal onde são utilizadas para descrever a estrutura das frases e palavras em linguagem natural. Como um **analisador sintático**, analisam uma sequência de entradas para verificar a sua estrutura gramatical segundo uma determinada gramática formal.

As GIC podem ser definidas como um conjunto de **Produções** onde todas as regras são da forma:

$$P : A \rightarrow \alpha$$

Onde o A corresponde ao lado esquerdo da produção P, ou *left-hand side*, que está associado a um símbolo **Não Terminal**. Enquanto que o α, corresponde ao conjunto de os símbolos **Terminais e Não Terminais**, também chamado *right-hand side* da produção P. A terminalidade do símbolo significa a sua "capacidade"de derivar esse mesmo em outros símbolos.

As GIC também tem um axioma S associado que representa a inicialização das regras gramaticais, estruturalmente correta.

Então podemos definir uma Gramática Independente do Contexto como:

$$GIC = \{S, NT, T, P\}$$

- S: Axioma da Gramática.
- NT: Conjunto de Símbolos Não Terminais da Gramática.
- T: Conjunto de Símbolos Terminais da Gramática.
- P: Conjunto de Produções da Gramática.

Estas gramáticas surgem como complementares do **analisador léxico**, que trata do processo de *tokenização*, processo este de converter uma sequência de caracteres numa sequência de **Tokens**, que são retornados ao analisador sintático.

A gramática e o *Lexer* estudados foram: ***Yacc*** + ***Flex***. Inicialmente exploramos a parte léxica, onde manipulávamos ficheiros de entrada e as suas sequências de caracteres.

O *Yacc* "chama"o *Lexer* para o reconhecimento dos tokens, e este retorna-o ao *Yacc* num par "Tipo", Valor do Token. Os valores dos tokens são retornados através de uma variável global partilhada entre os analisadores, de nome *yylval*. Comunicam através de 2 *Stacks* partilhadas, uma para os "tipos"dos Tokens (***Parse Stack***) e uma para os seus valores (***Value Stack***).

Este valor do Token passado no *yylval* é importante para definir o valor semântico do Token na gramática, para ser usado, se necessário, na atribuição do valor semântico da produção que está inserido (normalmente definido como *$$*). Este valor semântico de uma produção P pode ser usado pelas produções que derivam o **Não Terminal** definido no *left-hand side* da produção P. Neste

sentido, apenas existem valores sintetizados, ou seja, valores que "sobem"para as produções "acima"de uma produção P. Assim, é comum recorrer a variáveis globais para que seja possível "simular"valores herdados pelas produções abaixo da produção P.

$P0 : S \rightarrow elems\{\$\$ = \$1\}$
$P1 : elems \rightarrow INT' +' INT\{\$\$ = \$1 + \$2\}$

Valor semântico de *elems* sintetizado na Produção *P0*

Na secção a seguir, vai ser discutido o ganho ao usar Gramática de Atributos relativamente a este assunto dos "valores"semânticos herdados e sintetizados.

2 Gramáticas de Atributos

A melhor forma de introduzir este conceito de Gramáticas de Atributos é que, estas surgiram como uma extensão às gramáticas discutidas acima, GIC, permitindo a definição local (sem a utilização) do significado de cada símbolo num estilo declarativo.

Estas gramáticas também são definidos pelos símbolos terminais e não terminais. Além disso, esta gramática por ser uma extensão da GIC, também é definida pela GIC. Mas a abordagem é diferente pois, como dito em cima a atribuição do valor de cada símbolo é local.

Os símbolos terminais têm atributos intrínsecos, que descrevem o valor léxico associado a cada um deles. Por outro lado, os símbolos não-terminais são associados com atributos genéricos, através dos quais se poderá **sintetizar informação semântica**, ou **herdar informação semântica**. Aqui, percebe-se facilmente o ganho relativamente às GIC, pois pode-se "passar"(sintetizar) e herdar atributos, ou valores semânticos, entre produções, enquanto que na GIC apenas se podia sintetizar um atributo correspondente ao valor semântico da produção, o atributo $\$\$$.

Tanto as GIC como as gramáticas de atributos geram uma árvore de sintaxe em memória quando processadas, para interligar produções e os conjuntos de não terminais e terminais. No entanto, a árvore associada à gramática de atributos é **abstrata**, no sentido em que apenas é representado os símbolos com carga semântica, e são descartados os nodos que correspondem a palavras reservadas.

Este conceito de árvore é importante para perceber o conceito dos atributos sintetizados e herdados. Sintetizar informação semântica corresponde à subida na árvore de sintaxe abstrata, das folhas para a raiz. Herdar informação semântica corresponde à descida na árvore de sintaxe abstrata, do topo para as folhas.

Podemos concluir então que, uma gramática de atributos (GA) é um tuplo,

$$GA =< G, A, R, C >$$

onde:

- G: Corresponde à GIC.
- A: representa o conjunto de todos os atributos. O conjunto dos respetivos atributos $A(X)$ divide-se em 2 subconjuntos: os atributos herdados $AH(X)$ e os atributos sintetizados $AS(X)$.
- R: É a união dos R_p, o conjunto das regras de cálculo dos valores de atributos para cada produção $p \in P$.
- C: É a união dos C_p, o conjunto das condições de contexto para cada produção $p \in P$.