Perfil de PLC - Processamento de Linguagens e de Conhecimento (MiEI-MEI)

Resolução do Exercício 1 da Ficha de Exercícios 1 de
GCS – Gramáticas na Compreensão de Software

Ano Letivo 2018/19

1 Descrição de Ações de Formação

1.1 Definição Sintática

Pretende-se definir uma nova Linguagem que permita descrever ações de formação (uma o mais) conforme se explica abaixo.

Cada Ação de Formação (identificada por uma sigla) é organizada em torno de um Tema (definido por uma Descrição), o qual pode ser teórico ou prático. Se for Teórico é caracterizado por Tópicos (cada um definido também por uma descrição) que serão apresentados e por a Bibliografia de suporte (definida pelo título da obra) que deve ser consultada; se for Prático requer Recursos (cada qual definido por uma Descrição) que devem ser disponibilizados. Além disso qualquer acção de formação tem uma duração e um horário (dia/hora), bem como um custo. A Formação envolve um Formador que terá de ter um Diploma compatível com o tema em causa (técnico, bacharel, licenciado ou mestre). Por fim a Formação tem Alunos inscritos que, tal como o Formador, são Pessoas (definidas pelo nome, morada e cartão de cidadão).

a) Desenhe uma GIC a seu gosto para definir a linguagem pretendida.

Resolução:

Para resolver este exercício, relembremos o conceito de GIC:

Uma ***Gramática Independente de Contexto (GIC)*** *define-se como sendo um tuplo*

$$GIC = <T,N,S,P>$$

onde
T é o conjunto dos ***símbolos terminais*** *da linguagem (o alfabeto ou vocabulário).*
N é o conjunto dos ***símbolos não-terminais*** *da gramática.*
$S \in N$ é o ***símbolo inicial ou axioma*** *da gramática.*
P é o conjunto de ***produções ou regras de derivação*** *da gramática.*

Cada produção $p \in P$ é uma regra da forma

$$p : X0 \rightarrow X1 \ldots Xi \ldots Xn$$

em que p é o identificador da regra, $\rightarrow$ é o operador derivação, $X0 \in N$ e $Xi \in (N \cup T)$, $0 \leq i \leq n$.

Numa produção com etiqueta p o lado esquerdo do operador de derivação, sempre um não-terminal, denotase por ***LHS****(p)*[a] *e o lado direito do operador de derivação, uma sequência de símbolos terminais ou não-terminais, denota-se por* ***RHS****(p)*[b].
O conjunto T dos símbolos terminais divide-se em 3 subconjuntos disjuntos — T = PR∪Sin∪TV — das ***Palavras-Reservadas****, dos* ***Sinais*** *e dos* ***Terminais-Variáveis****.* [Ped13]

[a]Do inglês, Left Hand Side.
[b]Do inglês, Right Hand Side.

Como o objetivo deste ano letivo é usarmos Gramáticas de Atributos (GA) e o gerador AnTLR[1] (ANother Tool for Language Recognition), vamos ter escrever a GIC na Notação BNF-estendido.

Para o fazer relembremos o conceito de BNF:

A notação `BNF (de Backus-Naur Form)` *é uma notação textual, formal, para representar gramáticas independentes de contexto.*
Em BNF cada produção ou regra de derivação da gramática é vista como um triplo cujo elemento central é o operador de derivação. O operando do lado esquerdo desse operador é um símbolo não-terminal; o do lado direito é sua expansão, que pode conter zero ou mais símbolos terminais e não-terminais.[Ric12]
Os meta-símbolos utilizados na notação BNF são [Ric12]:

- *::= – representa o operador "deriva em" ou "definido como";*
- *| – indica um operando direito alternativo para o mesmo operando esquerdo;*
- *< > – delimita o identificador de cada símbolo gramatical.*

A notação `EBNF` *extende a notação BNF com os seguinte meta-símbolos [Ric12]:*

- ***** *(ou* **{ }***) – indica uma parte que se pode repetir 0 ou mais vezes;*
- *+ – indica uma parte que se pode repetir 1 ou mais vezes;*
- *?* *(ou [ ]) – indica uma parte opcional;*
- *( ) – indica precedências dentro da regra;*
- *" " – indica um carácter a tratar como terminal e.g., "<".*

Desta forma vamos obter uma Gramática Independente de Contexto escrita em notação BNF-estendido do AnTLR (Listing 1).

```
1  /*
2   * Linguagem: "Acoes de Formacao"
3   * Processador: Gramatica Independente de Contexto que permite descrever acoes de formacao
4   * PRH 2018.09.24
5   */
6
7  grammar gcs18F1Ex1_GIC;
8
9
10 acoes      : (acao '.')+
11           ;
12
13 acao       : cabec  tema  duracao  horario  custo formador alunos
14           ;
15
16
```

[1]AnTLR é um poderoso gerador de compiladores para reconhecimento (parsing) e processamento de frases da linguagem definida pela gramática que lhe é fornecida como entrada. A partir de uma gramática independente de contexto, tradutora ou de atributos, o ANTLR gera um parser, um construtor/ navegador na árvore de parsing e um tradutor. In: `http://www.antlr.org/`

```
cabec      : 'FORMACAO:' sigla '-' descricao
           ;

sigla      : IDENT
           ;

tema       : 'TEMA:' descricao   tipo
           ;

descricao  : TEXTO
           ;

tipo       : 'TIPO:' teor
           | 'TIPO:' prat
           ;

teor       : 'TEORICO'  topicos   bibliografia
           ;

topicos    :    descricaoTopico (';'  descricaoTopico)*
           ;

descricaoTopico : 'TOPICOS:' TEXTO
                ;

bibliografia    :'BIBLIOGRAFIA:' (titulo    obra)+
                ;

titulo          : 'TITULO:' TEXTO
                ;

obra            : 'OBRA:' TEXTO
                ;

prat            : 'PRATICO'   recursos
                ;

recursos        : descricaoRecurso (';'  descricaoRecurso)*
                ;


descricaoRecurso : 'RECURSOS:' TEXTO
                 ;

duracao          : 'DURACAO:' NUMERO 'h'
                 ;

horario          : 'HORARIO:' dia ','  HORA '--' HORA
                 ;

dia              : '2f'| '3f'| '4f'| '5f'| '6f'| 'sab'
                 ;


custo            : 'CUSTO:' NUMERO
                 ;

alunos           : aluno  (';' aluno)*
                 ;

aluno            : 'ALUNO:' pessoa
                 ;

formador         : 'FORMADOR:'  pessoa ','  diploma
                 ;
pessoa           : nome ','  morada ',' cartaoC
                 ;

```

```
85 nome                : TEXTO
86                     ;
87
88 morada              : TEXTO
89                     ;
90
91 cartaoC             : TEXTO
92                     ;
93
94 diploma            : 'tecnico'|'bacharel'|'licenciado'|'mestre'
95                     ;
96
97
98
99
100 /* Definicao do Analisador LEXICO */
101 IDENT : LETRA(LETRA|[0-9-_/])* ;
102
103 fragment LETRA : [a-zA-Z] ;
104
105 TEXTO:     (('\''|'\"') ~('\''|'\"')* ('\''|'\"'));
106
107 NUMERO: ('0'..'9')+ ; // [0-9]+
108
109 HORA: [0-9]?[0-9] ':' [0-9][0-9];
110
111 Separador: ('\r'? '\n' | ' ' | '\t')+ -> skip;
112
113 COMENT: '%'~('\r'|'\n')*[\r\n] -> skip;
```

Listing 1: Notação BNF-estendido do AnTLR – Gramática Independente de Contexto (GIC)

Note no exemplo da Listing 1 que, em AnTLR, toda a gramática abre com um preâmbulo que contém uma ou mais secções com informações gerais para o gerador; essas secções são, em geral, auto-explicativas: a primeira, que deve estar sempre presente, é o tipo de gramática e seu nome único (neste caso, '`grammar gcs18F1Ex1_GIC`'). Apenas um detalhe deve ser marcado: o nome da gramática tem de ser precisamente o nome do ficheiro que contém a gramática. O nome do ficheiro também deve ter uma extensão '.g4'; assim sendo, neste exemplo o ficheiro de entrada tem de ser denominado '`gcs18F1Ex1_GIC.g4`'.

Caso pretenda aceitar nos textos de entrada carateres em *UTF-8* então em vez de
`fragment LETRA : [a-zA-Z] ;`
use no Analisador Léxico a seguinte definição de `LETRA`
`fragment LETRA : [a-zA-ZáéíóúÁÉÍÓÚÃãÕõâêôÂÊÔÀÈÌÒÙàèìòùÇç];`

b) Usando a GIC especificada em cima apresente uma frase exemplo e a respetiva AD (árvore de derivação). Depois, analise a qualidade da sua GIC e da linguagem definida.

No Listing 2 são exibidos dois exemplos de frases distintas. A primeira frase ilustra um exemplo de uma ação de formação do tipo PRATICO e a segunda frase uma ação de formação do tipo TEORICO.

```
1 % Accao tipo TEORICO
2 FORMACAO:
3 GCS - 'gramaticas na compreensao de SW'
4
5 TEMA: 'um texto para descrever'
6 TIPO: PRATICO
7 RECURSOS: 'Recurso1 = .....'
8 DURACAO: 24h
9 HORARIO: 2f, 9:30 -- 11:30
10 CUSTO: 10
11 FORMADOR: 'João', 'Rua da Universidade', 'cc22248', bacharel
12 ALUNO: 'Pedro', 'Rua da Torre',  'e222225';
13 ALUNO: 'Cristiana', 'Rua da Maré',  'pg887225';
```

```
14 ALUNO: 'Ana', 'Rua de Cima',  'al8822225'.
15
16
17 % Accao tipo TEORICO
18 FORMACAO:
19 PLC - 'Processamento de Linguagens e compiladores'
20
21 TEMA: 'Um compilador e...'
22 TIPO: TEORICO
23 TOPICOS: 'Topico 1 = ..... '
24 BIBLIOGRAFIA:
25     TITULO: 'COMPILADORES'
26     OBRA: 'antLr'
27     TITULO: 'Sintaxe antlr'
28     OBRA: 'As  sintaxes'
29 DURACAO: 40h
30 HORARIO: sab, 15:30 -- 19:30
31 CUSTO: 25
32 FORMADOR: 'Ricardo', 'Rua da Outeiro', 'cc33248', mestre
33
34 ALUNO: 'Tiago', 'Rua da Torre', 'e222225';
35 ALUNO: 'Joana', 'Rua da Mare', 'pg887225';
36 ALUNO: 'Ana', 'Rua de Cima', 'al8822225'.
```

Listing 2: Exemplo de frases válidas

Para ilustrar as diversas saídas produzidas pelo ambiente de desenvolvimento ANTLRWorks, quando se executa o processador gerado pelo ANTLR sobre um ficheiro de entrada com a frase válida acima, suponha que foi usada a opção 'Run in TestRig' disponível no menu 'Run' desse ambiente de desenvolvimento.

A primeira saída que nos aparece numa janela popup é a imagem da árvore sintática que se mostra na Figura 1.

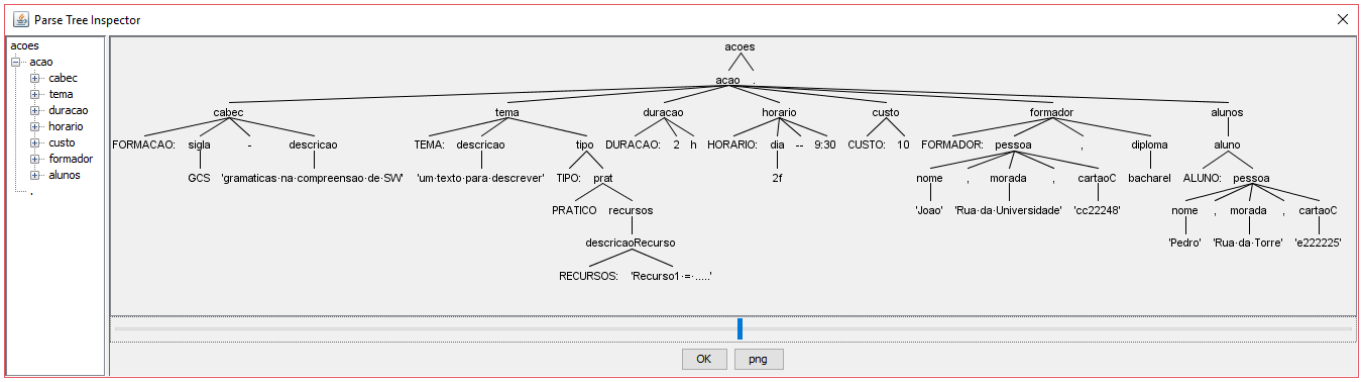

Figura 1: Diagrama da Árvore de Sintaxe para a frase exemplo

Repare-se que além do desenho da árvore esquematizada, através de um diagrama hierárquico que aparece na subjanela da direita, ainda surge na subjanela da esquerda um esquema vertical da árvore para facilitar a navegação na árvore da direita.

Passando agora a discutir a qualidade pode-se começar por analisar a **qualidade da linguagem** definida pela GIC do Listing1.

Para estruturar esta análise é importante recordar as **Características para avaliar a Qualidade de uma Linguagem** propostas em [Ped13] e que se enumeram abaixo:

- (CL1) **expressividade** (inclui a *abstracção* e a *clareza léxico-sintáctica*[a])
- (CL2) **documentação**
- (CL3) **unicidade**
- (CL4) **consistência** e **ortogonalidade** (inclui a *fidelidade ao paradigma*)
- (CL5) **extensibilidade**
- (CL6) **escalabilidade**
- (CL7) **fiabilidade** ou confiança (inclui a *clareza semântica*, mas distingue-se da *segurança* cf. referido abaixo)
- (CL8) **modularidade** (interfere na *reutilização*)

[a] Que outros autores consideram separadamente.

De uma rápida análise à GIC supra e tomando em consideração a frase exemplo do Listing2, podem-se tirar as ilações seguintes.

É uma linguagem verbosa que prima pela **clareza** devido à quantidade e especificidade das *palavras reservadas*, sendo naturalmente mais fatigante na escrita. De acordo com a informação fornecida sobre o domínio a que se destina, pode também observar-se que é **expressiva** pois permite de forma simples descrever as várias partes que é necessário descrever e além disso é **completa** visto que contém construções para definir as várias componentes do domínio.

Constata-se também que é **consistente** (ou coerente), quer porque as diversas partes a descrever seguem uma estrutura ou padrão análogo, quer porque o mesmo conceito em contextos diferentes é descrito da mesma forma (por exemplo, a descrição de formador e aluno segue o padrão de descrição de uma pessoa); manifesta também por isso a caraterística de **unicidade**.

A linguagem não é **extensível** (não tem mecanismos para acrescentar novos construtores) nem inclui nenhuma facilidade de **documentação** ou de **modularidade**, mas **escala** bem (a sua legibilidade mantém-se quando a número de ações de formação descritas aumenta).

Neste caso, em que se analisa a sintaxe, não faz sentido discutir a **fiabilidade**.

A seguir devemos analisar a gramática em si mesma. Para isso enumeram-se abaixo as **Características para avaliar a qualidade de uma Gramática** de acordo com o que é dito em [Ped13].

- (**CG1**, como *geradora de linguagens*) **usabilidade** da gramática enquanto instrumento para derivar frases de uma linguagem:
  - (CG1.1) facilidade de compreensão
  - (CG1.2) facilidade de derivação
  - (CG1.3) facilidade de manutenção
- (**CG2**, como *geradora de programas*) **eficiência** da gramática enquanto instrumento para derivar processadores para uma linguagem:
  - (CG2.1) eficiência no reconhecimento (processamento) das frases da linguagem gerada.
  - (CG2.2) eficiência na geração automática do processador.

Definição 1 (Qualidade de uma Gramática) *: A **qualidade de uma gramática, enquanto especificação que gera uma linguagem** afere-se em termos da facilidade com que se aprende (lê e compreende o que ela descreve), se usa para derivar frases e se mantém (correctiva ou evolutivamente). Neste sentido, diz-se, então, que uma* gramática tem qualidade *se facilita a usabilidade.*
*A **qualidade de uma gramática, enquanto especificação que gera um processador** afere-se em termos da eficiência do programa que dela deriva e da eficiência do próprio processo de geração. Neste sentido, diz-se, então, que uma* gramática tem qualidade *se permite gerar processadores de linguagens eficientes sem degradar a facilidade de geração automática.*

A tabela 1 mostra a influência dos vários elementos de uma GIC sobre as características que afectam a qualidade de uma gramática à luz da definição 1.

Elems x Caracts	**(CG1)Usabilidade**			**(CG2)Eficiência**	
	Compreensão	**Derivação**	**Manutenção**	**Reconh-L**	**Geraç-Rec**
Ids Símbolos claros	+	+	+	*X*	+Te,Ta
Prods Unitárias	+	−	+	+Te,Ta	+Te,Ta
Comprimento RHS	−	+	−	*x*Te,Ta	*X*
Notação	+/−	−p, +ex	−p, +/−ex	*X*	*X*
Esquema Recursivo	+/−	+d, −e	+/−	−Te,Ta	*X*
Modularidade	−	−	+	*X*	+Te
Complexidade Sint	*X*	*X*	−	*X*	*X*

Tabela 1: Influência dos Elementos de uma GIC nas Características do Critério de Qualidade

(Legenda) A Tabela 1 foi preenchida de acordo com o seguinte critério:

\+ **(influência positiva)** – contribui para facilitar o factor em causa

− **(influência negativa)** – contribui para dificultar o factor em causa

+/− **(influência ambivalente)** – tanto pode contribuir para facilitar como para dificultar o factor em causa

X **(não tem influência)** – não interfere com o factor em causa

x **(influência mínima)** – contribui de forma pouco significativa para dificultar o factor em causa

Te – Tempo de processamento/geração

Ta – Tamanho das Estruturas de Dados internas de suporte ao processamento/geração

p – pure-BNF

ex – extended-BNF

d – recursividade à direita

e – recursividade à esquerda

Mais tarde serão introduzidos vários conjuntos de métricas para avaliar quantitativamente a qualidade gramatical, mas por agora iremos avaliar os elementos da GIC do Listing 1 que podem determinar a qualidade da gramática.

Os identificadores dos símbolos Terminais e Não-terminais são longos e designam claramente os conceitos que denotam, por isso a GIC é fácil de compreender, derivar e manter embora possa requerer mais algum tempo e memória durante a fase de geração do processador.

Existem várias produções unitárias o que aumenta a facilidade de compreensão e manutenção, embora possa requerer mais algum tempo no processo de derivação; é também sabido que este fator tem impacto negativo no processamento e na geração (mais tempo e mais memória).

Como é sabido, lados direitos da produções muito longos dificultam a compreensão e a manutenção e praticamente não impactam na eficiência do processador/gerador. Neste caso concreto os lados direitos são quase sempre muito curtos (2 ou 3 símbolos) exceto no caso da produção que define `ação` que é bastante longa dada a complexidade do objeto que se quer descrever; este é sem dúvida um elemento que poderia ser melhorado criando 1 ou 2 novos símbolos não-terminais.

Por imposição da ferramenta escolhida (ANTLR) não se usa recursividade para descrever listas de elementos, recorrendo-se sim à notação extended-BNF e aos respetivos operadores iterativos para permitir ter essas listas. Em geral essa notação é atualmente considerada mais fácil de compreender, derivar e manter. Além disso o gerador em causa (ANTLR) tira partido da notação extended-BNF para ultrapassar a limitação dos conflitos LL(1) na gramática e para produzir um processador final bastante eficiente.

Quanto à modularidade, não existe nesta gramática; ela é toda apresentada num só ficheiro, mas a sua dimensão (cerca de 2 ou 3 dezenas de produções) não sofre com esse facto — provavelmente até é mais fácil de usar em um só módulo.

A complexidade sintática, que se avalia em termos do grau de dependência entre os símbolos, será mais tarde estudada com outro rigor (através de métricas apropriadas) mas à primeira vista não parece ser elevada pois a maioria dos símbolos só é usado em 1 ou 2 produções, o que facilita claramente a manutenção, sem ter grande impacto nas demais nas restantes caraterísticas.

Referências

[Ped13] Pedro Manuel Rangel Santos Henriques. Brincando às Linguagens com rigor: Engenharia Gramatical. Technical report, Universidade do Minho, November 2013.

[Ric12] Ivan Ricarte. *Introdução à Compilação*. Elsevier Brasil, 2012.