A short example: "Little garden"

Daniela da Cruz Pedro Rangel Henriques

{danieladacruz,prh}@di.uminho.pt
Departamento de Informática, CCTC
Universidade do Minho

1 Braga Gardener

A gardener of this city makes proposals to supply plants (trees and shrubs) to build, or to rebuild, interior or exterior gardens. In this context it is intended to develop a simple language processor to implement some operations associated to the management of the Gardener enterprise.
Analyze the following context free grammar, that is a simplified version of the real one, and compute LL(1) parsing table.

```
p1:  Flores    -->  FsExt  FsInt
p2:  FsExt     -->  FEXTERIOR  Fs
p3:  FsInt     -->  &
p4:               |  FINTERIOR
p5:  Fs        -->  Flor  MaisFs
p6:  MaisFs    -->  &
p7:               |  ","  Fs
p8:  Flor      -->  Cod  NomVulgar  Preco
p9:  NomVulgar-->  str
p10: Preco     -->  num
p11: Cod       -->  pal
```

Note: Initial symbol is `Flores` and Terminal Symbols are written in lowercase (pseudo-terminals), or in uppercase (reserved-words), or between apostrophes (punctuation marks). Null string is denoted by `&`, and `$` represents end-of-file (input text).

Resolution

To build the LL(1) Parsing Table, it is necessary to start computing the lookahead of each production in P[1].
To compute those sets it is mandatory to begin identifying the *nullable* symbols; in this example just FsInt and MaisFs are *nullable*.

- p1

$$\begin{aligned} lookahead(Flores \rightarrow FsExt\ FsInt) &= First(FsExt) \\ &= First(FEXTERIOR) \\ &= \{FEXTERIOR\} \end{aligned}$$

- p2

$$\begin{aligned} lookahead(FsExt \rightarrow FEXTERIOR\ Fs) &= First(FEXTERIOR) \\ &= \{FEXTERIOR\} \end{aligned}$$

- p3

$$\begin{aligned} lookahead(FsInt \rightarrow \epsilon) &= First(\epsilon) \bigcup Follow(FsInt) \\ &= \emptyset \bigcup Follow(FsInt) \\ &= First(\epsilon) \bigcup Follow(Flores) \\ &= \{\$\} \end{aligned}$$

- p4

$$\begin{aligned} lookahead(FsInt \rightarrow FINTERIOR) &= First(FINTERIOR) \\ &= \{FINTERIOR\} \end{aligned}$$

- p5

$$\begin{aligned} lookahead(Fs \rightarrow FlorMaisFs) &= First(Flor) \\ &= First(Cod) \\ &= \{pal\} \end{aligned}$$

[1]Remember formulas in the document with the formal definitions — `www.di.uminho.pt/~gepl/LP/docs/AlgorithmLL1.pdf`

- p6

$$\begin{aligned} lookahead(\text{MaisFs} \rightarrow \epsilon) &= First(\epsilon) \bigcup Follow(MaisFs) \\ &= \emptyset \bigcup Follow(Fs) \\ &= Follow(FsExt) \bigcup Follow(MaisFs) \\ &= First(FsInt) \bigcup Follow(Flores) \\ &= \{FINTERIOR, \$\} \end{aligned}$$

- p7

$$\begin{aligned} lookahead(MaisFs \rightarrow \text{","} \text{ Fs}) &= First(\text{","}) \\ &= \{\text{","}\} \end{aligned}$$

- p8

$$\begin{aligned} lookahead(Flor \rightarrow Cod\ NomVulgar\ Preco) &= First(Cod) \\ &= \{pal\} \end{aligned}$$

- p9

$$\begin{aligned} lookahead(NomVulgar \rightarrow str) &= First(str) \\ &= \{str\} \end{aligned}$$

- p10

$$\begin{aligned} lookahead(Cod \rightarrow pal) &= First(pal) \\ &= \{pal\} \end{aligned}$$

- p11

$$\begin{aligned} lookahead(Preco \rightarrow \text{num}) &= First(num) \\ &= \{num\} \end{aligned}$$

We can conclude that **it is a LL(1) grammar**, because:

$$\forall_{A \rightarrow \alpha_1, A \rightarrow \alpha_2} : lookahead(A \rightarrow \alpha_1) \bigcap lookahead(A \rightarrow \alpha_2) = \emptyset$$

Then it is possible to proceed with our objective, the construction of the LL(1) table. That table derives directly from the lookahead sets computed above applying the algorithm listed in `www.di.uminho.pt/~gepl/LP/docs/AlgorithmLL1.pdf`. After the derivation the following table is obtained:

	FEXTERIOR	FINTERIOR	","	str	pal	num	$
Flores	p1						
FsExt	p2						
FsInt		p4					p3
Fs					p5		
MaisFs		p6	p7				p6
Flor					p8		
NomVulgar				p9			
Cod					p10		
Preco						p11	

As the grammar is LL(1), the parsing table does not exhibit any entry with **conflicts** (more that one production in the same *line* × *column*).