Lab Session 4 – Solutions
Context-free Grammars and LL(1) Parsing

Interpretation and Compilation of Languages

Nova School of Science and Technology
Mário Pereira mjp.pereira@fct.unl.pt

Version of April 7, 2025

1 Context-Free Grammars

Exercise 1. Choose the correct answer: "The output of the lexical analyzer is

A a set of string characters".

B an abstract syntax tree".

C a set of regular expressions".

D a set of tokens".

Solution

□

Exercise 2. Consider the following grammar

$$\mathsf{S} \quad \rightarrow \quad \texttt{int} \mid \texttt{bool} \mid \texttt{if}\ \mathsf{S}\ \texttt{then}\ \mathsf{S} \mid \texttt{if}\ \mathsf{S}\ \texttt{then}\ \mathsf{S}\ \texttt{else}\ \mathsf{S}$$

where `if`, `then`, `else`, `int`, and `bool` are terminal symbols and S is the axiom of the grammar. Show that this grammar is ambiguous (so called "dangling else" problem) by giving two distinct parse trees for the word "`if bool then if bool then int else int`".

Solution

□

Exercise 3. Consider three simple grammars below that all recognize the same language of balanced parentheses, *i.e.* they recognize words such as ϵ, `()`, `(())`, `(()(()))`, and don't recognize words such as `)`, `)(`, `(()(`.

Grammar G_1	Grammar G_2	Grammar G_3
$\mathsf{P} \rightarrow \epsilon \mid (\mathsf{P}) \mid \mathsf{PP}$	$\mathsf{P} \rightarrow \epsilon \mid (\mathsf{P})\mathsf{P}$	$\mathsf{P} \rightarrow \epsilon \mid \mathsf{P}(\mathsf{P})$

Only one grammar is LL(1). For the other two, explain why these are not LL(1). In particular, for an ambiguous grammar show the ambiguity by providing two distinct derivation trees for the same word `()()`.

Solution □

Exercise 4. For the grammar that is LL(1) assume that we add a non-terminal symbol A as a new axiom and we add a rule (where # is the end-of-file terminal symbol)

$$\mathsf{A} \rightarrow \mathsf{P}\#$$

Give NULL, FIRST, FOLLOW, and the expansion tables for that grammar.

Solution □

2 LL(1) Parsing for Roman Numbers

Consider the following grammar for roman numbers from 0 to 9:

$$
\begin{array}{rcl}
\mathsf{S} & \rightarrow & \mathsf{N}\ \mathtt{EOF} \\
\mathsf{N} & \rightarrow & \mathtt{i}A \mid \mathcal{I}_3 \mid \mathtt{v}\mathcal{I}_3 \mid \epsilon \\
A & \rightarrow & \mathtt{v} \mid \mathtt{x} \\
\mathcal{I}_3 & \rightarrow & \mathcal{I}_2\mathtt{i} \mid \epsilon \\
\mathcal{I}_2 & \rightarrow & \mathcal{I}_1\mathtt{i} \mid \epsilon \\
\mathcal{I}_1 & \rightarrow & \mathtt{i} \mid \epsilon
\end{array}
$$

Exercise 5. Write the derivation of the number `viii`.

Solution □

Exercise 6. Is this grammar ambiguous? If no, explain why. If yes, give two derivation trees for the same word recognized by this grammar.

Solution □

Exercise 7. Is this grammar LL(1)? Motivate your answer by either explaining why it is not LL(1) or build an LL(1) parsing table.

Solution □

Exercise 8. Consider another variant of the grammar:

$$
\begin{array}{rcl}
\mathsf{S} & \rightarrow & \mathsf{N}\ \mathtt{EOF} \\
\mathsf{N} & \rightarrow & \mathtt{i}A \mid \mathtt{v}\mathcal{I}_3 \mid \epsilon \\
A & \rightarrow & \mathcal{I}_2 \mid \mathtt{v} \mid \mathtt{x} \\
\mathcal{I}_3 & \rightarrow & \mathtt{i}\mathcal{I}_2 \mid \epsilon \\
\mathcal{I}_2 & \rightarrow & \mathtt{i}\mathcal{I}_1 \mid \epsilon \\
\mathcal{I}_1 & \rightarrow & \mathtt{i} \mid \epsilon
\end{array}
$$

Give NULL, FIRST, FOLLOW, and the expansion table for that grammar.

Solution □

3 Solutions

Exercise 1, page 1

The correct answer is **D**.

Exercise 2, page 1

See Figure 2.

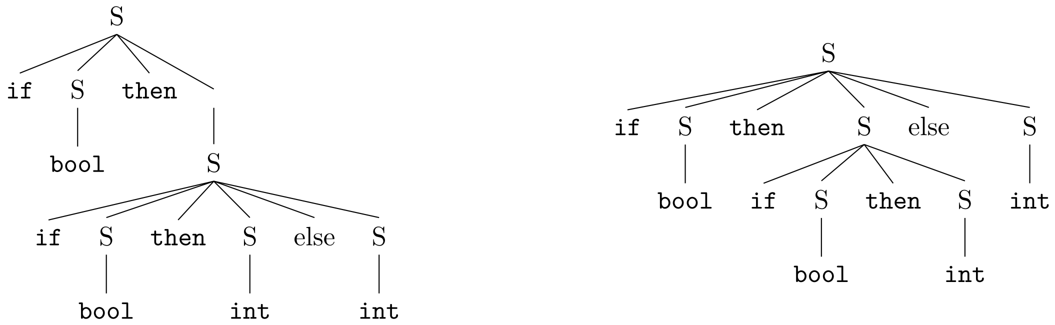


Figure 2: Solution for Exercise 3.

Exercise 3, page 1

Grammar G_1 is ambiguous and G_3 is left-recursive.
Using grammar G_1 to produce two different derivations for word `()()`:

Exercise 4, page 1

For the grammar G_2 extended with the new rule

$$\begin{array}{lcl} \mathsf{A} & \to & \mathsf{P}\ \# \\ \mathsf{P} & \to & \epsilon \mid \mathsf{(P)P} \end{array}$$

we have

NULL = { P }

FIRST(A) = { (}
FIRST(P) = { (}

FOLLOW(A) = $\emptyset$
FOLLOW(P) = { #,) }

Expansion table:

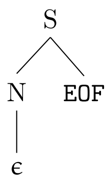
	(	)	#
A	P #		P #
P	(P) P	ϵ	ϵ

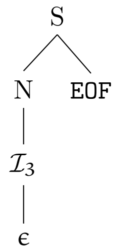
Exercise 5, page 2

$$\mathsf{S} \longrightarrow \mathsf{N}\,\texttt{EOF} \longrightarrow \texttt{v}\,\mathcal{I}_3\,\texttt{EOF} \longrightarrow \texttt{v}\,\mathcal{I}_2\texttt{i}\,\texttt{EOF} \longrightarrow \texttt{v}\,\mathcal{I}_1\texttt{ii}\,\texttt{EOF} \longrightarrow \texttt{viii}\,\texttt{EOF}$$

Exercise 6, page 2

Yes, the grammar is ambiguous, since there are two derivation trees for recognizing the empty word ϵ:





Exercise 7, page 2

The grammar is not LL(1). For instance, in the non-terminal `N`, the entry in the expansion table for terminal `i` will contain both productions $\texttt{i}A$ and $\mathcal{I}_3$, since both alternatives can start with i. Also, ambiguous grammars can never be LL(1).

Exercise 8, page 2

NULL $= \{\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3, \text{A}, \text{N}\}$

FIRST$(\mathcal{I}_1)$ = FIRST$(\mathcal{I}_2)$ = FIRST$(\mathcal{I}_3)$ = $\{\ \texttt{i}\ \}$
FIRST(A) = $\{\texttt{i}, \texttt{v}, \texttt{x}\}$
FIRST(N) = $\{\texttt{i}, \texttt{v}\}$
FIRST(S) = $\{\texttt{i}, \texttt{v}, \texttt{EOF}\}$

FOLLOW(S) = $\emptyset$
FOLLOW(X) = $\{\texttt{EOF}\}$ where $\mathsf{X} \in \{\mathsf{N}, \mathsf{A}, \mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3\}$.

Expansion table:

	i	v	c	#
S	N EOF	N EOF		N EOF
N	$\texttt{i}A$	$\texttt{v}\mathcal{I}_3$		ϵ
A	$\mathcal{I}_2$	v	x	$\mathcal{I}_2$
$\mathcal{I}_3$	$\texttt{i}\mathcal{I}_2$			ϵ
$\mathcal{I}_2$	$\texttt{i}\mathcal{I}_1$			ϵ
$\mathcal{I}_1$	i			ϵ