

Exercise session 1

LEIA Architecture, Lexical Analysis and Grammars

1.1 The LEIA architecture

We give you the “LEIA cheat sheet”. The objective is to recall concepts from the architecture course and manipulate the assembly code of the architecture you will compile to.

Your teaching assistant will make a demo of the LEIA simulator during this session.

EXERCISE #1 ► TD

On paper, write (in LEIA assembly language) a program which initializes the $R0$ register to 1 and increments it until it becomes equal to 8.

Then, write a similar program that increments it until it becomes equal to 4242, using only two registers.

EXERCISE #2 ► Hand disassembling

In Figure 1.1 we depicted a toy example with its corresponding assembly code. Disassemble the two first instructions in the table.

Fill the first two row of the table, read the rest of the solution, and answer the following questions:

- Which instruction is used to load data from memory?
- How is the pointer jumping done to create the loop?
- What happens to the labels in the disassembled program?

Address	Content	Binary	Instructions	pseudo-code
0000	4800			
0001	1901			
0002	c209	1011 0010 0000 1001	letl r2 9	$R_2 \leftarrow 0009$ (label data)
0003	f302	1111 0011 0000 0010	rmem r3 [r2]	$R_3 \leftarrow mem[R_2]$ (content at label data)
loop:0004	1912	0001 1 001 0001 0010	add r1 r1 2	$R_1 \leftarrow R_1 + 2$
0005	2b31	0010 1 011 0011 0001	sub r3, r3 1	$R_3 \leftarrow R_3 - 1$
0006	3f30	0011 1 111 0011 0000	snif r3 le 0	if $R_3 \leq 0$ skip next statement
0007	bffd	1101 1111 1111 1101	jump -3	jump to loop label
0008	b000	1101 0000 0000 0000	jump 0	HALT
data:0009	0006	data	-	

Figure 1.1: A binary/hexadecimal program (tp1-1.s)

EXERCISE #3 ► C to LEIA- Skip if you are late

Translate into LEIA code the following C-code:

```
x=5;
if (x>12) y=70; else y=x+12
```

1.2 Lexical Analysis

A bit of ANTLR4 syntax is given as companion material.

EXERCISE #4 ► **Regular expressions for lexing**

Use the ANTLR4 syntax to define ANTLR4 macros to define:

1. Identifiers : any sequence of letters, digits and `_` that do not begin by a digit nor `_`.
2. Floats like `-3.96` (the sign is optional, but the dot is not).
3. Scientific notation like `-1.6E-12`.

EXERCISE #5 ► **Romans numbers**

Write an ANTLR4 lexical file that reads and interprets Roman numerals : $IV \rightarrow 4 \dots$ You can use the fact that the lexical analysis always takes the rule to match the longest subchain.

1.3 Grammars

All grammars will use the ANTLR4 syntax.

EXERCISE #6 ► **Well-founded parenthesis**

Write a grammar and files to accept any text with well-formed parenthesis `)` and `'[`.