COMP 2049 (AE2LAC) Languages and Computation

Coursework: Floating-Point Literals and Simple Arithmetic Expressions

---

Release date: Monday, April 18th, 2022

***Deadline*: Tuesday, May 3rd, 2022, 16:00**
Cut-off Date: Friday, May 6th, 2022, 16:00

Total Mark: 100
Weight: 15% of the module mark
How to submit: Via Moodle

---

1 Floating-Point Literals

Design a right-linear grammar G_1 that generates the language of binary floating-point literals according to the following rules:

- Each number may be signed or unsigned.
  - unsigned as in `1.01`, signed as in `+1.01` or `−1.01`
- The numerical part (also called the value field) must be non-empty and may optionally include a decimal point '.', in which case it must be followed by some other digits. For instance:
  - In the number `+110.011`, the value field is `110.011`
  - `1` and `.01` and `−.001` are acceptable, but `1.` is not acceptable.
- There may be an optional exponent field, in which case, it must contain the letter 'e', followed by a signed or unsigned integer.
  - For instance, `101e+11` or `−1.11e101` are acceptable, but `1.01e` and `1.01e-1.1` are not acceptable.

Task 1. *Implement the grammar G_1 in JFLAP, and test it on some input strings of your choice.*

A screenshot of the result of parsing of some sample input strings for grammar G_1 in JFLAP is provided in Figure 1 on the following page.

Remark 1.1. *In all of the tasks of this coursework, the default parsing method should be the "brute force" parsing. Hence, to test your grammars in JFLAP on several input strings, choose the tab `"Input"` and then the item `"Multiple Brute Force Parse"`.*

2 Arithmetic Expressions

For the second task, you are required to design a context-free grammar (CFG) G_2 that generates the language of arithmetic expressions over natural numbers in binary format. Each arithmetic expression is constructed from the following:

- Binary unsigned integer literals, with leading zeros accepted;

Figure 1 Some sample input values for the right-linear grammar G_1.

Input	Result
0.11	Accept
+0	Accept
-.001e1	Accept
+00100.0010	Accept
100.0e011	Accept
100e1	Accept
-0	Accept
111e	Reject
1.0e1.0	Reject
e11	Reject
0.001.e	Reject
1.1e.1	Reject

- Arithmetic operators $+$, $-$, $*$, and $/$;
- Properly nested parentheses.

For instance, an expression such as $(11 + 0101)/001$ must be accepted, whereas $((11 - 01)$ must be rejected because the parentheses do not match.

Task 2. *Implement the grammar G_2 in JFLAP and test it on some input expressions of your choice.*

Check all the production rules of the grammar G_2 to see if there are any λ-productions or unit-productions. If there are any such productions, you may notice that for more complicated input strings, it takes a long time for JFLAP to parse the string. In fact, at times it may enter into a non-terminating loop.

Task 3. *Use JFLAP to remove the λ-productions and unit-productions of the grammar G_2 to obtain the grammar G_2'. Then, try to parse the same strings as before and notice that it takes a shorter time to parse them, and the parser does not enter into non-terminating loops.*

In JFLAP, to remove λ-productions and unit-productions, you may first choose the tab `"Convert"`, and then the item `"Transform Grammar"`.

A screenshot of the result of parsing of some sample input strings for grammar G_2' in JFLAP is provided in Figure 2 on the next page.

3 Submission

You must submit three files, named according to the following templates:

(1) A JFLAP file for grammar G_1 of Task 1 named:

`ID-Surname_FirstName-01-Right_Linear.jff`

(2) A JFLAP file for grammar G_2 of Task 2 named:

`ID-Surname_FirstName-02-CFG.jff`

(3) A JFLAP file for grammar G_2' of Task 3 named:

`ID-Surname_FirstName-03-CFG_no_unit.jff`

Figure 2 Some sample input values for the CFG G_2'. Note that `-00*11` is rejected because `-00` may only be interpreted as a signed literal, while in G_2 and G_2' only unsigned literals are allowed.

Input	Result
11+11	Accept
(11+0101)/001	Accept
((1-0)/11)	Accept
1/11/111	Accept
1-(00/00)	Accept
11-01	Accept
((11-01)	Reject
-00*11	Reject
1*(1/01))	Reject

4 Marking scheme

Correctness: (80%) Correct answers for the three tasks contribute to 80% of the total mark, as follows:

Task 1: 40%

Task 2: 20%

Task 3: 20%

Format: (20%)

(a) While the grammar G_2' of Task 3 is generated by JFLAP, the grammars for Tasks 1 and 2 must be written by you. For grammars G_1 and G_2, all productions with the same left-hand-side variable must appear in one block one after another. (15%)

(b) The three files must be named according to the templates given above. (5%)

Late Submissions: The standard University penalty for late submission is applied, i. e., 5% absolute standard University scale per day, until the mark reaches zero. For example, an original mark of 67% would be successively reduced to 62%, 57%, 52%, 47%, etc.