PDF Lexical Analysis

Boujamaa ATRMOUH

June 3, 2024

1 Introduction

The objective of this project is to develop a series of tools that, given a PDF file, can extract information about its structure. To do this, we will develop different parsers with the aim of analyzing the content of a file. Ideally, these tools should be able to serve as a basis for the development of code that can repair, or even edit, a PDF file.

2 PDF Structure

Addressing all the subtleties of the PDF format would be far too ambitious, and we will therefore make the following assumptions about the argument file:

- The document is not protected by encryption;
- The file contains only one reference table;
- This reference table is not compressed;
- PDF objects are defined directly in the body of the document (and not within a stream);
- Streams never contain the sequence of characters ”endstream”. This string will therefore always represent the end of stream marker;
- Parentheses within PDF character strings are always preceded by the character.

In practice, this means that our code will be able to handle almost all PDF files in version 1.4 (or lower), unencrypted and without incremental modification.

I

Files `parser1.y` and `lexer1.l`. The main program takes as its only argument the path to the file to be analyzed. If this file is not valid, an error message will be displayed. If it is valid, the version of the PDF format and the address of the reference table will be displayed on the standard output.

- `VERSION` is a lexeme to represent comments of the form %PDF-x,y,
- `LINE` is a lexeme to represent any other line of the file,
- `lines` is a non-terminal symbol, representing a sequence of lines from which the last line must be extracted, verify that it contains only a positive integer, and display this integer.

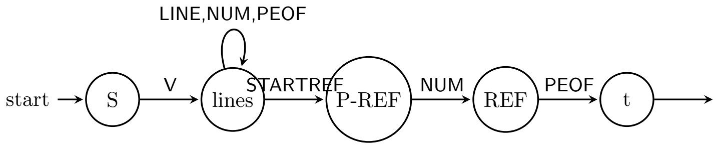

Figure 1: automata I

S → VERSION lines PEOF

lines → STARTREF NUM
| LINE lines
| NUM lines
| PEOF lines

Figure 2: I: Grammar

S → VERSION lines

lines → LINE lines
| NUM lines
| PEOF lines
| STARTREF ref
ref → NUM end_line
end_line → PEOF

Figure 3: I: Right-linear grammar

VERSION = `%PDF-[0-9]+\.[0-9]+`
LINE = `[^\n]+`
NUM = `[0-9]+`
PEOF = `%%EOF`
STARTREF = `startxref`