A proposal of a formal grammar for a subset of PDF

Guillaume Endignoux Olivier Levillain Jean-Yves Migeon

This document describes a grammar for a restricted subset of the Portable Document Format, which excludes – among others – incremental updates, object streams and linearization. The grammar is presented in BNF, in a top-down order – from the file structure to the character set.

1 File structure

The following rules describe the overall file structure.

⟨*PDF*⟩ ::= ⟨*header*⟩ ⟨*body*⟩ ⟨*xref*⟩ ⟨*trailer*⟩ ⟨*eof-marker*⟩

⟨*header*⟩ ::= ⟨*version*⟩ ⟨*non-ascii-marker*⟩? ⟨*spaces*⟩

⟨*version*⟩ ::= '`%PDF-1.`' ⟨*version-digit*⟩ ⟨*eol*⟩

⟨*non-ascii-marker*⟩ ::= '`%`' ⟨*non-ascii-char*⟩ ⟨*non-ascii-char*⟩ ⟨*non-ascii-char*⟩ ⟨*non-ascii-char*⟩ ⟨*eol*⟩

⟨*body*⟩ ::= ⟨*empty*⟩
  | ⟨*body*⟩ ⟨*indirect-object*⟩ ⟨*spaces*⟩

⟨*xref*⟩ ::= ⟨*xref-header*⟩ ⟨*xref-section*⟩

⟨*xref-header*⟩ ::= '`xref`' ⟨*eol*⟩ ⟨*unsigned-int*⟩ ⟨*space*⟩ ⟨*unsigned-int*⟩ ⟨*eol*⟩

⟨*xref-section*⟩ ::= ⟨*empty*⟩
  | ⟨*xref-section*⟩ ⟨*unsigned-int-10*⟩ ⟨*space*⟩ ⟨*unsigned-int-5*⟩ ⟨*space*⟩ ('`f`' | '`n`') ⟨*xref-eol*⟩

⟨*trailer*⟩ ::= '`trailer`' ⟨*spaces*⟩ ⟨*dictionary*⟩ ⟨*eol*⟩ '`startxref`' ⟨*eol*⟩ ⟨*unsigned-int*⟩ ⟨*eol*⟩

⟨*eof-marker*⟩ ::= '`%%EOF`' ⟨*eol*⟩?

2 Objects

The following rules describe the syntax of direct and indirect objects.

⟨*indirect-object*⟩ ::= ⟨*indirect-object-header*⟩ ⟨*direct-object*⟩ ⟨*spaces*⟩ '`endobj`'
  | ⟨*indirect-object-header*⟩ ⟨*dictionary*⟩ ⟨*spaces*⟩ ⟨*stream*⟩ ⟨*spaces*⟩ '`endobj`'

⟨*indirect-object-header*⟩ ::= ⟨*unsigned-int*⟩ ⟨*spaces*⟩ ⟨*unsigned-int*⟩ ⟨*spaces*⟩ '`obj`' ⟨*spaces*⟩

⟨*direct-object*⟩ ::= ⟨*null*⟩
  | ⟨*bool*⟩
  | ⟨*int*⟩
  | ⟨*real*⟩
  | ⟨*string*⟩
  | ⟨*name*⟩
  | ⟨*reference*⟩
  | ⟨*array*⟩
  | ⟨*dictionary*⟩

⟨*reference*⟩ ::= ⟨*unsigned-int*⟩ ⟨*spaces*⟩ ⟨*unsigned-int*⟩ ⟨*spaces*⟩ 'R'

⟨*array*⟩ ::= '[' ⟨*spaces*⟩ ⟨*array-content*⟩ ']'

⟨*array-content*⟩ ::= ⟨*empty*⟩
  | ⟨*direct-object*⟩ ⟨*spaces*⟩ ⟨*array-content*⟩

⟨*dictionary*⟩ ::= '<<' ⟨*spaces*⟩ ⟨*dictionary-content*⟩ '>>'

⟨*dictionary-content*⟩ ::= ⟨*empty*⟩
  | ⟨*dictionary-content*⟩ ⟨*key-value*⟩

⟨*key-value*⟩ ::= ⟨*name*⟩ ⟨*spaces*⟩ ⟨*direct-object*⟩ ⟨*spaces*⟩

3 Complex tokens

The following rules describe the syntax of complex tokens such as strings, names and streams.

⟨*string*⟩ ::= ⟨*string-literal*⟩ | ⟨*string-hexa*⟩

⟨*string-literal*⟩ ::= '(' ⟨*string-content*⟩ ')'

⟨*string-content*⟩ ::= ⟨*empty*⟩
  | ⟨*string-content*⟩ ⟨*string-char*⟩

⟨*string-char*⟩ ::= ⟨*string-regular*⟩
  | '\' ('n' | 'r' | 't' | 'b' | 'f' | '(' | ')' | '\' | ⟨*eol*⟩)
  | '\' ⟨*four-digit*⟩ ⟨*octal-digit*⟩ ⟨*octal-digit*⟩
  | ⟨*string-literal*⟩

⟨*string-hexa*⟩ ::= '<' ⟨*hexa-content*⟩ '>'

⟨*hexa-content*⟩ ::= ⟨*spaces*⟩
  | ⟨*hexa-content*⟩ ⟨*hexa-char*⟩

⟨*hexa-char*⟩ ::= ⟨*hexa-digit*⟩ ⟨*spaces*⟩ ⟨*hexa-digit*⟩ ⟨*spaces*⟩

⟨*name*⟩ ::= '/' ⟨*name-content*⟩

⟨*name-content*⟩ ::= ⟨*empty*⟩
  | ⟨*name-content*⟩ ⟨*name-char*⟩

⟨*name-char*⟩ ::= ⟨*name-regular*⟩
  | '`#`' ⟨*hexa-digit*⟩ ⟨*hexa-digit*⟩

⟨*stream*⟩ ::= '`stream`' ⟨*eol*⟩ ⟨*stream-content*⟩ '`endstream`'

⟨*stream-content*⟩ ::= ⟨*empty*⟩
  | ⟨*stream-content*⟩ ⟨*any-char*⟩

4 Simple tokens

The following rules describe the syntax of simple tokens such as numbers.

⟨*null*⟩ ::= '`null`'

⟨*bool*⟩ ::= '`true`' | '`false`'

⟨*int*⟩ ::= ⟨*sign*⟩? ⟨*unsigned-int*⟩

⟨*real*⟩ ::= ⟨*sign*⟩? (⟨*digits*⟩ '`.`' ⟨*digits*⟩? | '`.`' ⟨*digits*⟩)

⟨*unsigned-int*⟩ ::= ⟨*digits*⟩

⟨*digits*⟩ ::= ⟨*digit*⟩
  | ⟨*digits*⟩ ⟨*digit*⟩

⟨*unsigned-int-10*⟩ ::= ⟨*unsigned-int-5*⟩ ⟨*unsigned-int-5*⟩

⟨*unsigned-int-5*⟩ ::= ⟨*digit*⟩ ⟨*digit*⟩ ⟨*digit*⟩ ⟨*digit*⟩ ⟨*digit*⟩

⟨*xref-eol*⟩ ::= ⟨*space*⟩ ⟨*cr*⟩ | ⟨*space*⟩ ⟨*lf*⟩ | ⟨*cr*⟩ ⟨*lf*⟩

⟨*eol*⟩ ::= ⟨*cr*⟩ | ⟨*lf*⟩ | ⟨*cr*⟩ ⟨*lf*⟩

⟨*any-space*⟩ ::= ⟨*space*⟩ | ⟨*cr*⟩ | ⟨*lf*⟩ | ⟨*tab*⟩

⟨*spaces*⟩ ::= ⟨*empty*⟩
  | ⟨*spaces*⟩ ⟨*any-space*⟩

5 Character set

The following rules describe character classes.

⟨*sign*⟩ ::= '`+`' | '`-`'

⟨*version-digit*⟩ ::= ['`0`'-'`7`']

⟨*hexa-digit*⟩ ::= ['`0`'-'`9`'`a`'-'`f`'`A`'-'`F`']

⟨*octal-digit*⟩ ::= ['`0`'-'`7`']

⟨*four-digit*⟩ ::= ['`0`'-'`3`']

⟨*digit*⟩ ::= [‘0’-‘9’]

⟨*non-ascii-char*⟩ ::= [‘\x80’-‘\xFF’]

⟨*space*⟩ ::= ‘\x20’

⟨*cr*⟩ ::= ‘\x0D’

⟨*lf*⟩ ::= ‘\x0A’

⟨*tab*⟩ ::= ‘\x09’

⟨*any-char*⟩ ::= [‘\x00’-‘\xFF’]

⟨*string-regular*⟩ ::= ⟨*any-char*⟩ - (‘(’ | ‘)’ | ‘\’)

⟨*regular*⟩ ::= ⟨*any-char*⟩ - (‘\x00’ | ‘\x09’ | ‘\x0A’ | ‘\x0C’ | ‘\x0D’ | ‘\x20’ | ‘(’ | ‘)’ | ‘<’ | ‘>’ | ‘[’ | ‘]’
| ‘{’ | ‘}’ | ‘/’ | ‘%’)

⟨*name-regular*⟩ ::= ⟨*regular*⟩ - ‘#’