Apendice A) Características de Lenguaje Dot

A Graph File Grammar

The following is an abstract grammar for the *DOT* language. Terminals are shown in bold font and nonterminals in italics. Literal characters are given in single quotes. Parentheses (and) indicate grouping when needed. Square brackets [and] enclose optional items. Vertical bars j separate alternatives.

graph ! [**strict**] (**digraph** j **graph**) *id* 'f' *stmt-list* 'g'
stmt-list ! [*stmt* [';'] [*stmt-list*]]
stmt ! *attr-stmt* j *node-stmt* j *edge-stmt* j *subgraph* j *id* '=' *id*
attr-stmt ! (**graph** j **node** j **edge**) *attr-list*
attr-list ! '[' [*a-list*] ']' [*attr-list*]
a-list ! *id* '=' *id* [','] [*attr-list*]
node-stmt ! *node-id* [*attr-list*]
node-id ! *id* [*port*]
port ! *port-location* [*port-angle*] j *port-angle* [*port-location*]
port-location ! ':' *id* j ':' '(' *id* ',' *id* ')'
port-angle ! '@' *id*
edge-stmt ! (*node-id* j *subgraph*) *edgeRHS* [*attr-list*]
edgeRHS ! *edgeop* (*node-id* j *subgraph*) [*edgeRHS*]
subgraph ! [**subgraph** *id*] 'f' *stmt-list* 'g' j **subgraph** *id*

An *id* is any alphanumeric string not beginning with a digit, but possibly including underscores; or a number; or any quoted string possibly containing escaped quotes.
An *edgeop* is -> in directed graphs and -- in undirected graphs.
The language supports C++-style comments: /* */ and //.
Semicolons aid readability but are not required except in the rare case that a named subgraph with no body immediate precedes an anonymous subgraph, because
under precedence rules this sequence is parsed as a subgraph with a heading and a body.
Complex attribute values may contain characters, such as commas and white space, which are used in parsing the *DOT* language. To avoid getting a parsing error, such values need to be enclosed in double quotes.