IEEE Standards Interpretations for IEEE Std 1003.2™-1992 IEEE Standard for Information Technology--Portable Operating System Interfaces (POSIX®)--Part 2: Shell and Utilities

IEEE Standards Department, Copyrights and Permissions, 445 Hoes Lane, Piscataway,
New Jersey 08855-1331, USA

Interpretation Request #112

Topic: more on yacc **Relevant Clauses:** A.3.7.4

Lines 1134--1135 state that "If duplicate token numbers cause conflicts in the parser generation, yacc shall report an error...." Historically, yacc has not kept track of the token numbers assigned to individual tokens, thus, it was impossible for yacc to tell if there was an ambiguity in the grammar which was caused by duplicated token numbers. For example, the grammar %% start: a | b; a: A; b: B; is, as written, clearly unambiguous. However, by adding the two lines %token A 300 %token B 300 it is now impossible for the parser to determine, at runtime, whether a "300" value returned from the lexical analyser should be taken as an "A", or as a "B". The behaviour of historical yacc in this context is unspecified. My question is, do the lines quoted above require POSIX yacc to maintain a table of token-to-number mappings, and ensure that the table generated does not have any such ambiguities in it?

Interpretation Response

We believe that the last sentence on page 715 line 1129-1136 requires yacc to report errors if it detects them, but is not required to go looking for them. Application use of duplicate token numbers and token numbers < 256 is non-portable and implementations are allowed to report errors if these conditions are seen. The standard clearly states the behavior for duplicate token numbers in yacc, and conforming implementations must conform to this.

Rationale for Interpretation

None.