distinct characters occurring as string elements must denote different values of type Char. The string element consisting of two apostrophes denotes the apostrophe character.

CharacterString = "′" *StringElement* { *StringElement* } "′" .

StringElement = "′ ′" | *AnyCharacterExceptApostrophe* .

A character string is a constant of type Char if it has one string element; otherwise it is a constant of a string type (see Section 6.2.1) that has as many components as there are string elements.

Note: A character string must be written on just one line of program text.

Examples of character strings:

```
'A'          ';'
'Pascal'     ''''
'This is a character string'
```

Symbol separators may be placed between any two adjacent symbols or before the first symbol of a program. At least one symbol separator must occur between two adjacent identifiers, directives, word symbols, labels, or numbers. A separator is a space, the end of a line of program text, or a comment. The meaning of a program is unaltered if a comment is replaced with a space.

Comment = ("{" | "(*") [*CommentElement*] ("}" | "*)") .

A *CommentElement* is either an end of line or any sequence of characters not containing "]" or "*)".

Notes: { ... *) and (* ... } are valid comments. The comment {(*) is equivalent to the comment {(}.

5. Constants

A constant definition introduces a constant identifier to denote the value that is specified by the constant in the definition; the constant identifier being defined must not occur in the constant part of the definition. Constant definitions are collected into constant definition parts.