The first segment function FIRSTSEG may be more complex for other indentation schemes. For instance, if we had required that the BEGIN of the code body of a procedure start on a new line, but other BEGINs need not, decomposing an input line into segments can no longer be done on the basis of a set like LO.

Definition of SEGSEQ

Let z be a sequence of lines, and w a prefix of a line.

1. SEGSEQ(" ") ::= 0ss.
2. SEGSEQ($z \mid w$) ::= SEGSEQ(z) ! w, if w is all white;
   SEGSEQ($z \mid w$) :: = SEGSEQ(z) ! DS(z, w), otherwise.
3. DS(z, w) ::= 0ss, if w is all white. Otherwise, DS(z, w) :: = u ! DS($z \mid u, v$) where u = FIRSTSEG(TKNSEQ(z), w), and $w = u \mid v$.

As an example, consider the following string, z, where the end of lines are shown explicitly.

```
"if b1 then (* loop *) while b2 do begin"      |\n|
"  "                                           |\n|
"   x := f(x); (* c1 *) {invariant} g(x)"      |\n|
"end; (* of while and if *)"
```

The segments of SEGSEQ(z) are given below.

```
"if b1 then (* loop *)"                        !
"while b2 do begin "|\n                        !
"  "|\n                                         !
"   x := f(x); (* c1 *) {invariant}"           !
"g(x)"|\n                                       !
"end; (* of while and if *)"
```

5.4. Indentation

Most often the indentation (i.e. the width of the left margin) of a given output line depends on the indentation of the previous line and on the reserved words occurring in it. On rarer occasions, the indentation depends also on the reserved words appearing in that line itself. An example of this is the "until". NMG(T) gives the margin the next line should have if the last token of T corresponds to the last word of the current line; CMG(T) gives the margin the current line should have *if* the last token of T corresponds to the first word of the current line. The function MG gives the actual margin of each output line.

Definition of NMG

1. NMG(00) ::= 0.
2. NMG(T) :: = NMG(RED(T)). Thus we assume below that the argument of NMG is reduced.
3. Let t = PF, or BEGIN. Then NMG(T ∘ t) :: = NMG(T).
4. Let t = DECL, RECORD, LPAREN, REPEAT, DO, CASE, THEN, ELSE or COLON. Then NMG(T ∘ t) :: = NMG(T) + UOI.