Definition of CMG

1. CMG(00) ::= 0.
2. Let t = PF, DECL or ELSE. Then CMG(T $\circ$ t) ::= NMG(T $\circ$ t) − UOI.
3. Let t = RECORD, LPAREN, REPEAT, DO, CASE, THEN or COLON. Then CMG(T $\circ$ t) ::= NMG(T).
4. For all t not covered above, CMG(T $\circ$ t) ::= NMG(T $\circ$ t).

It follows from the above that CMG(T) = CMG(RED(T)).

Definition of MG

1. MG(0ss) ::= 0.
2. MG(z ! x) ::= CMG(T $\circ$ t), where t is the first token of LEX(T, x), and T = TKNSEQ(z).

Note that MG maps segment sequences to margins in contrast to NMG and CMG which map token sequences to margins.

5.5. Final specification

Let *zi*|*e* be the input file to indenting program, and let *zo* be the corresponding output file of an indenting program. Then

zo = INDENT(*zi*)|*e*

is the relation between them, where INDENT(*zi*) ::= IND(SEGSEQ(*zi*)), and IND is given below.

Defintion of IND

IND maps segment sequences to sequences of lines. Let z be a sequence of segments, and x a segment.

1. IND(0ss) ::= "", the empty string.
2. IND(z ! x) ::= IND(z)|*b* ** MG(z ! x)|*ps*TRIM(x)|*n*, where *ps*TRIM(x) trims x by removing all its prefix and suffix white space.

Note that *ps*TRIM(*i*th *segment of input file*) = *ps*TRIM(*i*th *output line*).

6. THE EQUIVALENCE OF THE TWO SPECIFICATIONS

The low level specifications coincide with the high level specifications in the following sense: Let *zi*|*e* be the text input to an indenting program satisfying our low level specifications. Clearly its output *zo* = INDENT(*zi*)|*e*. Then we say that the two specifications are *coincident* if PLOT(*nt*, *n*|*zo*, 0) = true, whenever *nt* →* *zi*. Note that if *zi* were not a valid construct, PLOT(*nt*, *n*|*zo*, 0) would be false for all *nt*. As Figure 2 completely ignores comments, PLOT does not say how comments should be laid out, and we therefore give INDENT complete freedom in this regard.

A proof that the two specifications are coincident proceeds by induction on the syntactic structure of the input *zi*. As the base step, we show that if *zi* can be generated by one application of a production rule then *zo* = INDENT(*zi*) would satisfy the high level specifications. If, for $1 \leqslant j \leqslant k$, zu_j = INDENT(zi_j), and N = $N_1 N_2 \ldots N_k$ is a production of Pascal grammar such that $N_j \rightarrow^* zi_j$, then the induction hypothesis is that PLOT(N_j, *n*|zu_j, 0) = true. We need to show that PLOT(N, *n*|*zo*, 0) = true, where *zo* = INDENT(*zi*) and $zi = zi_1 | zi_2 | \ldots | zi_k$.