

line. Good style for visual appeal demand that reserved words and comments be flanked by white spaces and we do not see the need to rectify these 'anomalies'.

5.2. Reduced token sequences

We introduce the notion of 'reduced' token sequences which makes it easy to define the functions that give the left margin width of output lines. Intuitively speaking, a token sequence $\tau_1 \circ \tau_3$ is a reduced token sequence of $\tau_1 \circ \tau_2 \circ \tau_3$ if τ_2 is the token sequence of a syntactically 'sensible' Pascal statement. One might insist that τ_2 correspond to a syntactically correct statement. This, we believe, is overburdening the indenting programs; guaranteeing syntactic correctness is the function of a compiler, not indenting programs. What is syntactically sensible is made clear in the way the mathematical function RED maps a given token sequence to its reduction.

The P, Q, R, S and T below denote token sequences, and s and t denote single tokens. Expressions of the kind "if $T = R \circ \text{DECL} \circ S$ for some R and S" are abbreviated as 'if $T = R \circ \text{DECL} \circ S$ '. The special tokens DECL and PF are devised for the purposes of the RED function below and do not have corresponding words.

Definition of RED (see notes below)

1. $\text{RED}(T \circ t) ::= \text{RED}(\text{RED}(T) \circ t)$
Thus we assume below that the sequence denoted by T is not reducible any further.
2. Let $t = \text{PROCEDURE, FUNCTION or PROGRAM}$. Then $\text{RED}(T \circ t) ::= R \circ \text{DECL} \circ \text{PF}$, where $R = S$ if $T = S \circ \text{DECL}$, $R = T$ if T does not end with either DECL or LPAREN; $::= T$, if T does end with LPAREN.
3. Let $t = \text{LABEL, CONST, TYPE or VAR}$. Then $\text{RED}(T \circ t) ::= R \circ \text{DECL}$, where $R = S$ if $T = S \circ \text{DECL}$, $R = T$ if T does not end with either DECL or LPAREN; $::= T$, if T does end with LPAREN.
4. Let $t = \text{FORWARD, or EXTERN}$. Then $\text{RED}(T \circ t) ::= S$, if $T = S \circ \text{PF}$; $::= T$, otherwise.
5. $\text{RED}(T \circ \text{BEGIN}) ::= S \circ \text{BEGIN}$, if $T = S \circ \text{PF}$, or if $T = S \circ \text{PF} \circ \text{DECL}$; $::= T \circ \text{BEGIN}$, otherwise.
6. Let $t = \text{RECORD, LPAREN, REPEAT, CASE, DO, THEN or COLON}$. Then $\text{RED}(T \circ t) ::= T \circ t$.
7. $\text{RED}(T \circ \text{OF}) ::= S \circ \text{CASE}$ if $T = S \circ \text{CASE} \circ \text{COLON}$; $::= T$, otherwise.
8. Let the pair $\langle t, s \rangle$ be one of $\langle \text{RPAREN, LPAREN} \rangle$, $\langle \text{UNTIL, REPEAT} \rangle$. Then $\text{RED}(T \circ t) ::= R$, if $T = R \circ s \circ S$ where S does not have any tokens s ; $::= \text{00}$, otherwise.
9. $\text{RED}(T \circ \text{END}) ::= R$, if $T = R \circ \text{RECORD} \circ S$ where S is free of RECORDS; $::= P$, if T is free of RECORDS and $T = P \circ s \circ Q$ where s is either a BEGIN, or a CASE and Q does not have any of these tokens; $::= \text{00}$, otherwise.
10. $\text{RED}(T \circ \text{ELSE}) ::= R \circ \text{ELSE}$, if $T = R \circ \text{THEN} \circ S$ where S is free of THENS; $::= \text{ELSE}$, otherwise.
11. $\text{RED}(T \circ \text{SEMICOLON}) ::= R \circ s$, if $T = R \circ s \circ S$ where s is any token but THEN, ELSE, DO, or COLON and S is a sequence of these tokens only; $::= \text{00}$, otherwise.
12. $\text{RED}(T \circ t) ::= T$, for any t not covered above.

The many cases in the definition reflect the syntax of the language. It should be clear that many illegal Pascal constructs would result in valid reduced sequences. As