A NOTE ON PASCAL SCOPES

T. P. Baker and A. C. Fleck
Department of Computer Science
The University of Iowa
Iowa City, Iowa 52242

In response to the recent efforts toward development of a PASCAL standard [1], we would like to point out a peculiarity we have observed in the PASCAL notion of scopes as exemplified in the E.T.H. compilers, and to suggest how a "cleaner" alternative notion might be implemented.

Beginning with ALGOL60, "block structured" languages have followed the convention that scopes of local declarations correspond to the boundaries of the blocks in which they occur. Since PASCAL superficially appears to follow this convention, a programmer is likely to go along for some time before he stumbles upon a case where PASCAL scopes do not correspond to block boundaries. When he does, it is likely to be a source of confusion. For example, consider the programs and output below (from Version 3 of the PASCAL 6000 compiler).

```
1 PROGRAM P1(OUtPUT);
2   PROCEDURE Q; BEGIN WRITELN(1) END;
3   PROCEDURE R;
4     PROCEDURE S; BEGIN Q END;
5     PROCEDURE Q; BEGIN WRITELN(2) END;
6   BEGIN S; Q END;
7 BEGIN R END.

1
2
```

```
1 PROGRAM P2(OUTPUT);
2 TYPE A = CHAR;
3   PROCEDURE Q;
4   TYPE B = ^A;
5          A = RECORD L,R: B END;
6    VAR X: B;
7   BEGIN NEW(X); X^ := 'A' END;
8 BEGIN Q END.
```

```
1 PROGRAM P3(OUTPUT);
2 VAR F: INTEGER;
3   PROCEDURE Q;
4     PROCEDURE R; BEGIN WRITELN(F) END;
5     FUNCTION F: INTEGER; BEGIN F := 2 END;
6   BEGIN R; WRITELN(F) END;
7 BEGIN F := 1; Q END.

1
2
```

Note that according to current and proposed scope rules [1], this is the "correct" program behavior in each case.

We propose that PASCAL can be standardized to follow the ALGOL60 conventions, with the added restriction that (except in recursive pointer type declarations) no use of an identifier may precede its declaration (this appears to be the approach taken in ADA [2]). Thus, program P1 above would be considered incorrect, since the use of Q in procedure S precedes a local definition of Q. P3 would be incorrect for a similar reason, because the use of F in procedure R precedes a local declaration of F. Program P2 would be considered incorrect, but for a different reason. The variable X would be interpreted as a pointer to a record, so that the assignment "X^ := 'A'" would be a type conflict. This is exactly what would have happened if the outer declaration "A = CHAR" had not been present. In this case, the convention followed by the compiler not only makes the interpretation of the procedure Q dependent in an unobvious way on its global environment, but also effectively blocks the possibility of defining a pointer type for the local record type A.

A single pass compiler can enforce these conventions. On first encountering a use of an identifier X that is not yet declared in the local block, the compiler attempts to resolve the reference to a previously processed nonlocal declaration, say D, in one of the surrounding blocks. If this search is successful, the processor creates new "dummy" entries for X in the symbol table for the local block and all surrounding blocks, out to the block where D appeared. These dummy entries will contain a pointer to the entry corresponding to D and will serve the purpose of insuring that any subsequent declaration of X locally will be deleted and treated as an error.

PASCAL already provides means for handling the few cases where forward references are unavoidable. For procedures, functions, and labels, there are forward declarations. For recursively defined pointer types, processing can be deferred until it can be determined whether a type identifier should be resolved as a local or nonlocal reference. For example, processing of "B = ^A" in P2 would be deferred until the local declaration of A was encountered (or until the end of the TYPE section).

We believe that the proposed conventions are an improvement in the direction of simplicity and conformity to established practice, Furthermore, as exemplified best in program P2, they improve program modularity, by permitting reliable local resolution of references, which under present rules is impossible.

[1] A.M. Addyman er al. "A draft description of PASCAL", *Software Pract. & Exper.* 9, 5(1979), 381-424; also PASCAL News 14, 6(1979), 7-54.

[2] Preliminary ADA Reference Manual, SIGPLAN Notices 14, 6(1979).