any other activation that B is within.

A procedure statement or function designator that is contained in an algorithm and that specifies the activation of a block is called the *activation–point* of that activation.

11. Procedures and Functions

Procedures and functions are named program parts that are activated by procedure statements (Section 9.1.2) and function designators (Section 8.1), respectively. The programmer can declare new procedures and functions as needed. Procedure declarations and function declarations are collected into procedure and function declaration parts.

ProcedureAndFunctionDeclarationPart =

[(*ProcedureDeclaration* | *FunctionDeclaration*) ";"] .

In addition, each implementation is required to provide numerous "predeclared" procedures and functions. Since these, as all such entities, are assumed to be declared in a scope surrounding the program, no conflict arises from a declaration redefining the same identifier within the program.

11.1. Procedure Declarations

A procedure declaration serves to introduce a procedure identifier, and to associate the identifier with a block and possibly with a formal parameter list. The procedure heading of a procedure declaration introduces the procedure identifier and the formal parameter list.

A procedure may be declared by a single procedure declaration consisting of the procedure heading and the block. This is the most common form.

Alternatively, it may be declared with a "forward declaration": one procedure declaration consists of the procedure heading and the directive `forward`, and a second declaration in the same procedure and function declaration part consists of a procedure identification and the block. The procedure identifier in the procedure identification must be the identifier introduced by the first declaration. Note that the formal parameter list, if any, is not specified in the second declaration.