Z Synopsis

A Z specification document consists of interleaved passages of formal specification and informal narrative. The formal specification consists of a sequence of *paragraphs* that gradually introduce the elements of the formal specification, each paragraph building on the ones that come before. Z specifications are primarily intended for human consumption not computer processing. Z involves an elaborate notation, including many special symbols, that makes the formal components particularly terse, so the informal narrative is regarded as an essential component of a specification. Recursion is generally not permitted (one exception is described later).

Each paragraph may define one or more names for schemas, basic types, global variables or global constants. It may use the names defined in preceding paragraphs, and the names it defines are available in the paragraphs that follow. The scope of each global name extends from its definition to the end of the specification. The gradual building-up of the vocabulary of a specification is called the principle of *definition before use*. However, exceptions to this rule may occur when it is convenient.

There are two presentation styles, vertical (or graphical) and horizontal — the vertical style is generally used in presentations, and the horizontal style is used with computer tools.

There are several kinds of paragraph — basic type definitions, axiomatic descriptions, constraints, schema definitions, abbreviations definitions, generic schema, and constant definitions. We give a brief list of the horizontal formats for the various kinds of paragraph — details of the syntax and semantics will be elaborated little by little.

B**asic type definition** — [Ident, … , Ident]

Axiomatic description — | Declarations | Predicate; … ; Predicate

Constraint — Predicate

Schema definition — Schema-Name $\widehat{=}$ Schema-Exp, or

Schema-Name $\widehat{=}$ $D_1; \ldots ; D_m \mid P_1; \ldots ; P_n$

D_i declaration, P_i predicate

Abbreviation definition — Ident == Expression

Generic schema — Schema-Name[Ident, … , Ident] $\widehat{=}$ $[D_1; \ldots ; D_m \mid P_1; \ldots ; P_n]$,

D_i declaration, P_i predicate

Generic constant — [Ident, … , Ident] | Declarations | Predicate; … ; Predicate