# Exercise V

# Implementation of refinements

The language MINI has to be extended with refinements. Secondly, we will allow declarations in all paragraphs. Example:

```
kwadrateren:
   WHILE er is invoer
   REP
      kwadrateer invoer;
      druk kwadraat af
   ENDREP.

er is invoer:
   INT VAR invoer;
   put ("Geef een getal: ");
   get (invoer);
   invoer <> 0.

kwadrateer invoer:
   INT kwadraat :: invoer * invoer.

druk kwadraat af:
   put ("Het kwadraat van ");
   put (text (invoer));
   put (" is ");
   put (text (kwadraat));
   put ("\n").
```

The syntax of the MINI language is therefore changed and augmented by the following syntax rules.

## Programs and refinements

### Syntax

a) `program:`
   `refinement sequence.`

b) `refinement:`
   `refinement name, colon token, paragraph, point token.`

**Context conditions**

a) The first refinement occurring in a program has type VOID.

b) The refinement has the same type and access as the paragraph. It is not allowed to define a refinement with the same name as an object or a procedure. The scope of refinement definitions is that of the program.

**Semantics**

a) The execution of a program consists of the execution of its first refinement.

## Paragraphs and units

The rules for `paragraph without declarations` and `proper paragraph without declarations` are deleted from the grammar.

## Object declarations

The scope of object declarations is that of the program.

## Control constructs

The following rules for control constructs are changed:

a) `loop body:`
```
    rep token, paragraph, endrep token.
```

b) `then part:`
```
    then token, paragraph.
```

c) `else part:`
```
    else token, paragraph;
    elif token, condition, then part, else part option.
```

## Primaries

a) `primary:`
```
    repetition;
    conditional choice;
    call;
    denoter;
```

```
identifier application;
refinement application;
open token, expression, close token.
```

b) `refinement application:`

```
    refinement name.
```

**Context conditions**

a) The type and access of the primary are the same as those of the refinement application.

b) Using the name of the refinement, the corresponding refinement definition is identified among the declared refinements. The refinement application has the same type and access as the corresponding refinement.

**Semantics**

b) A refinement application is evaluated by evaluating the paragraph of the corresponding refinement definition.

A small set of test programs can be obtained by unpacking the file `ftp.cs.kun.nl://pub/cdl3/ex5.tgz` in your MINI working directory.