Chapter 15 : Being careful with the base cases of functions.

In which we apply the techniques we learned in Chapters 13 and 14, but learn to be careful when looking at the base cases of functions.

Suppose we are given the function f, defined as follows.

f : natural → natural

* (0) f.0 = 13

* (1) f.1 = 37

* (2) f.(2*n) = f.n + 2 , 0 < n

* (3) f.(2*n+1) = f.n + 3*f.(n+1) + 6, , 0 < n

We are given a natural number N and asked to write a program to compute f.N

Choosing invariants.

As in our previous examples we generalise from the most complicate shape of the definition of f and propose as invariants

P0 : α * fusc.n + β * fusc.(n+1) + γ = fusc.N
P1 : $0 \le n \le N$

Note that we have included the expression + γ in P0. This is because in equations (2) and (3) we have an extra term on the right hand side.

Establishing the Invariants.

The following assignment establishes the invariants

n, α, β, γ := N, 1, 0, 0

A useful observation.

Note that equations (2) and (3) tell us how we can rewrite f in the cases where the arguments are of the form 2*n and 2*n+1 where 0 < n. This means that the argument must be 2 or higher. This is important. Suppose we allowed n = 0, then (2) could be used to give us a value for f.0 which would be different from the value specified by (0), and (3) could be used to give a value for f.1 which is different from the value specified by (1). We can only appeal to (2) and (3) when we know that the argument for f is 2 or more. In cases where the argument is less than 2 we must appeal to (0) and (1). This suggests that our program will have to deal with 3 cases, $n = 0$, $n = 1$ and $1 < n$. We will deal with these in an [if.fi](if.fi) statement.

The first 2 of these cases are trivial and will not require us to appeal to the invariant, we simply appeal to (0) and (1). In the final case we will use our invariant where $1<n$. We expect that the invariant will allow us to calculate the value of f.N when $n = 1$. We investigate.

Termination.

We note the following.

$$
\begin{array}{ll}
 & P0 \wedge P1 \wedge n=1 \\
= & \quad \{\text{definitions of } P0, P1\} \\
 & \alpha * f.n + \beta * f.(n+1) + \gamma = f.N \wedge 1 \le n \le N \wedge n=1 \\
\Rightarrow & \quad \{\text{leibniz}\} \\
 & \alpha * f.1 + \beta * f.2 + \gamma = f.N \wedge \text{true} \\
= & \quad \{\text{predicate calculus}\} \\
 & \alpha * f.1 + \beta * f.2 + \gamma = f.N \\
= & \quad \{(0), (1)\} \\
 & \alpha * 37 + \beta * f.2 + \gamma = f.N \\
= & \quad \{ (2) \text{ with } n = 1 \} \\
 & \alpha * 37 + \beta * (37 + 2) + \gamma = f.N \\
= & \quad \{\text{arithmetic}\} \\
 & \alpha * 37 + \beta * 39 + \gamma = f.N
\end{array}
$$

So keeping the invariant true and bringing n down to 1 results in
$f.N = \alpha * 37 + \beta * 39 + \gamma$.

Guard.

We choose as our guard

$$n \neq 1$$

Variant.

As our variant we choose

$$n$$

Calculate the loop body.

$$
\begin{array}{ll}
 & P0 \\
= & \quad \{\text{definition } P0\} \\
 & \alpha*f.n + \beta*f.(n+1) + \gamma = f.N \\
= & \quad \{\text{case analysis, even.n i.e. } n = 2*p\} \\
 & \alpha*f.(2*p) + \beta*f.(2*p+1) + \gamma = f.N \\
= & \quad \{\text{definition f (2) (3)}\} \\
 & \alpha *(f.p + 2) + \beta*(f.p + 3*f.(p+1) + 6) + \gamma = f.N \\
= & \quad \{*/+\} \\
 & \alpha *f.p + \alpha*2 + \beta*f.p + \beta*3*f.(p+1) + \beta* 6 + \gamma = f.N \\
= & \quad \{\text{gather terms}\} \\
 & (\alpha+\beta)*f.p + \beta*3*f.(p+1) + \alpha*2 + \beta*6 + \gamma = f.N \\
= & \quad \{WP\} \\
 & (n, \alpha, \beta, \gamma := n \text{ div } 2, \alpha + \beta, \beta*3, \alpha*2 + \beta*6 + \gamma).P0
\end{array}
$$

Giving us the program fragment

$\text{if even.n} \rightarrow n, \alpha, \beta, \gamma := n \text{ div } 2, \alpha + \beta, \beta*3, \alpha*2 + \beta*6 + \gamma$

Let us analyse the other case

$$
\begin{array}{ll}
 & P0 \\
= & \quad \{\text{definition } P0\} \\
 & \alpha*f.n + \beta*f.(n+1) + \gamma = f.N \\
= & \quad \{\text{case analysis, even.n i.e. } n = 2*p+1\} \\
 & \alpha*f.(2*p + 1) + \beta*f.(2*p + 2) + \gamma = f.N \\
= & \quad \{(2), (3)\} \\
 & \alpha*(f.p + 3*f.(p+1) + 6) + \beta*(f.(p+1)+ 2) + \gamma = f.N \\
= & \quad \{*/+\} \\
 & \alpha*f.p + \alpha*3*f.(p+1) + \alpha*6 + \beta*f.(p+1) + \beta*2 + \gamma = f.N \\
= & \quad \{\text{gather terms}\} \\
 & \alpha*f.p + (\alpha*3 + \beta)*f.(p+1) + \alpha*6 + \beta*2 +\gamma = f.N \\
= & \quad \{WP\} \\
 & (n, \alpha, \beta , \gamma := (n-1) \text{ div } 2, \alpha , (\alpha*3 + \beta), \alpha*6 + \beta*2. + \gamma).P0
\end{array}
$$

Which gives us the program fragment

$\text{if odd.n} \rightarrow n, \alpha, \beta , \gamma := (n-1) \text{ div } 2, \alpha , (\alpha*3 + \beta), \alpha*6 + \beta*2 + \gamma$

Finished program.

And so the entire program is as follows.

```
if N = 0 → r := 13
[] N = 1 → r := 37
[] N > 1 →
        n, α, β := N, 1, 0, 0 {P0 ∧ P1}
        ;do n ≠ 1 → {P0 ∧ P1 ∧ n ≠ 1}

                if even.n → n, α, β, γ := n div 2, α + β, β*3, α*2 + β*6 + γ
                [] odd.n → n, α, β , γ := (n-1) div 2, α , (α*3 + β), α*6 + β*2 + γ
                fi

                        {P0 ∧ P1}
        od
        r := α * 37 + β *39 + γ
fi
```

We note that this solution has temporal complexity O(log.N)