Programmes Corrects par Construction
Partie 2 : Exemples - Plus Long Segment Nul

Pierre-Edouard Portier

1 Plus long segment nul

```
  var N: int {N≥0}
; var f(0..N-1): array of int
; var r: int
r: R

R: r = (↑p,q : 0≤p≤q≤N ∧ Z.p.q : q-p)

Z.p.q ≙ (∀i : p≤i<q : f.i=0)
```

Les éléments du tableau doivent être vus au moins une fois. Ainsi, la stratégie de transformation de la constante `N` en une variable `n` est appliquée pour construire un invariant. D'où l'esquisse de programme suivante.

```
P0: r = (↑p,q : 0≤p≤q≤n ∧ Z.p.q : q-p)
P1: 0≤n≤N

  n,r := 0,0
  {P0 ∧ P1}
; do n≠N ->
     r: S
     {?P0[n\n+1]}
   ; n:=n+1
  od
```

Il faut réaliser `P0[n\n+1]` pour construire le programme `S` qui mettra à jour la variable `r` afin de maintenir l'invariant `P0`.

```
Re P0[n\n+1]

  (↑p,q : 0≤p≤q≤n+1 ∧ Z.p.q : q-p)
= { split sur q=n+1, correct car n≠N par la clause de garde et
    n≤N par P1,
    P0 }
  r ↑ (↑p : 0≤p≤n+1 ∧ Z.p.(n+1) : n+1-p)
= { + distribue sur ↑ }
  r ↑ (n+1 + (↑p : 0≤p≤n+1 ∧ Z.p.(n+1) : -p))
= { dualité min/max }
  r ↑ (n+1 - (↓p : 0≤p≤n+1 ∧ Z.p.(n+1) : p))
```

Introduction d'une nouvelle variable `s` et renforcement de l'invariant :

```
P2: s = (↓p : 0≤p≤n ∧ Z.p.n : p)
```

D'où l'esquisse de programme :

```
  n,r,s := 0,0,0
  {inv P0 ∧ P1 ∧ P2, bf N-n}
; do n≠N ->
     s: T
     {?P2[n\n+1]}
   ; r:= r↑(n+1-s)
   ; n:=n+1
  od {R}
```

Pour calculer le programme `T` qui met à jour la variable `s`, il faut réaliser la vérité de `P2[n\n+1]`.

```
Re P2[n\n+1]

  (↓p : 0≤p≤n+1 ∧ Z.p.(n+1) : p)
= { split sur p=n+1, correct car n≠N par la clause de garde et
    n≤N par P1,
    Z.(n+1).(n+1)=true }
  (n+1)↓(↓p : 0≤p≤n ∧ Z.p.(n+1) : p)
= { soit f.n≠0 et le domaine du min est vide
                donc sa valeur est ∞,
    soit f.n=0 et la valeur du min nous est donnée par P2 }
  (n+1)↓(if f.n=0 -> s
           |f.n≠0 -> ∞
         fi)
```

D'où le programme :

```
  var n,r,s: int
; var N: int {N≥0}
; var f(0..N-1): array of int
; n,r,s := 0,0,0
; do n≠N ->
     if f.n=0 -> skip
       |f.n≠0 -> s:=n+1
     fi
   ; r:= r↑(n+1-s)
   ; n:=n+1
  od
```