5. Gaia: Programen eratorpen formala

7. Ariketa-orria

1. Proposatutako errekurrentzia erlazioa erabiliz, hurrengo programa errekurtsiboak iteratibo bihurtu.

1.1. *agerkop* funtzioak x digitua ($0 \leq x \leq 9$) y zenbaki arruntean zenbat aldiz agertzen den kalkulatzen du.

$$agerkop(x,y) = \begin{cases} 1 & \texttt{baldin } x = y \\ 0 & \texttt{baldin } 0 \leq y \leq 9 \wedge x \neq y \\ agerkop(x, y/10) + 1 & \texttt{baldin } y \geq 10 \wedge y \ mod \ 10 = x \\ agerkop(x, y/10) & \texttt{baldin } y \geq 10 \wedge y \ mod \ 10 \neq x \end{cases}$$

Errekurrentzia erlazioa:

$$agerkop(x,y) = agerkop(x,u) + v$$

Soluzioa:

Hasieraketa:

```
u := y;
v := 0;
```

Emaitza. Bi kasu nabari daude.

Lehenengo kasu nabarian ($x = u$):

$$(\ x = u\ ) \rightarrow agerkop(x,y) = agerkop(x,u) + v = agerkop(x,x) + v = 1 + v$$

Beraz:

```
return 1 + v;
```

Bigarren kasu nabarian ($0 \leq u \leq 9 \ \wedge \ x \neq u$):

$$(\ 0 \leq u \leq 9 \ \wedge \ x \neq u\ ) \rightarrow agerkop(x,y) = agerkop(x,u) + v = v$$

Beraz:

```
return v;
```

Bukaera baldintza. Bi kasu nabari daudenez, baldintza konposatua da:

$$bukbal(u,v) \equiv x = u \ \vee \ (\ 0 \leq u \leq 9 \ \wedge \ x \neq u\ ) \equiv x = u \ \vee \ 0 \leq u \leq 9$$

Iterazioaren gorputza. Bi kasu induktibo daude.

Lehenengo kasu induktiboan ($u \geq 10 \ \wedge \ u \ mod \ 10 = x$):

$$
\begin{aligned}
& agerkop(x,y) \\
& \qquad = \underbrace{agerkop(x,u)}_{destolestu} + v \\
& \qquad = \underbrace{(agerkop(x, \frac{u}{10}) + 1) + v}_{tolestu} \\
& \qquad = agerkop(x, \underbrace{\frac{u}{10}}) + \underbrace{1 + v} \\
& \qquad = agerkop(x, \ u' \ ) + \ \ v'
\end{aligned}
$$

Aldibereko asignazioa:

```
(u,v) := (u/10,1+v);
```

Aldibereko asignazioaren inplementazioa:

```
u := u/10;
v := 1+v;
```

Bigarren kasu induktiboan ($u \geq 10 \ \wedge \ u \ mod \ 10 \neq x$):

$$
\begin{aligned}
& agerkop(x,y) \\
& \qquad = \underbrace{agerkop(x,u)}_{destolestu/tolestu} + v \\
& \qquad = agerkop(x, \underbrace{\frac{u}{10}}) + \underbrace{v} \\
& \qquad = agerkop(x, \ u' \ ) + \ v'
\end{aligned}
$$

Aldibereko asignazioa:

```
(u,v) := (u/10,v);
```

Aldibereko asignazioaren inplementazioa:

```
u := u/10;
```

Funtzio iteratiboa:

```
function agerkop_it(x, y :  nat) return nat is
{ 0 ≤ x ≤ 9 ∧ y ≥ 0 }
   u, v :  nat;
begin
   u := y;
   v := 0;
         INB ≡ { agerkop(x, y) = agerkop(x, u) + v }
   while x /= u and ( u < 0 or u > 9 ) loop
      if u >= 10 & u mod 10 = x then
         u := u/10;
         v := 1+v;
      else
         u := u/10;
      end if;
   end loop;
   if x = u then
      return 1+v;
   else
      return v;
   end if;
end agerkop_it;
{ agerkop(x, y) = agerkop_it(x, y) }
```

1.2. $HurBainoHandi : sekuentzia(nat) \rightarrow nat$ funtzioak kalkulatzen du zenbat elementu dauden sekuentzia batean hurrengo elementua baino handigoak.

(1) $HurBainoHandi(\langle\ \rangle) = 0$
(2) $HurBainoHandi(\langle x \rangle) = 0$
(3) $HurBainoHandi(x \bullet (y \bullet S)) =$
$$\begin{cases} 1 + HurBainoHandi(y \bullet S) & \texttt{baldin } x > y \\ \quad HurBainoHandi(y \bullet S) & \texttt{bestela} \end{cases}$$

Errekurrentzia erlazioa:

$$HurBainoHandi(S) = HurBainoHandi(U) + v$$

Soluzioa:

Hasieraketa:

```
U := S;
v := 0;
```

Emaitza. Bi kasu nabari daude.

Lehenengo kasu nabarian ($hutsa\_da?(U)$):

$$\begin{aligned} hutsa\_da?(U) \ \rightarrow \ HurBainoHandi(S) &= HurBainoHandi(U) + v \\ &= HurBainoHandi(\langle\ \rangle) + v \\ &= 0 + v \ = \ v \end{aligned}$$

Beraz:

```
return v;
```

Bigarren kasu nabarian ($luzera(U) = 1$):

$$\begin{aligned}(\ luzera(U) = 1\ ) \ \rightarrow \ HurBainoHandi(S) &= HurBainoHandi(U) + v \\ &= 0 + v \ = \ v\end{aligned}$$

Beraz:

```
return v;
```

Kontuan hartu bi kasu nabarietan emaitza bera bueltatzen dela. Honen ondorioz, bukaeran kasu nabariak ez dira bereiztuko.

Bukaera baldintza. Bi kasu nabari daudenez, baldintza konposatua da, baina sinplifika daiteke:

$$bukbal(U, v) \ \equiv \ hutsa\_da?(U) \ \vee \ luzera(U) = 1 \ \equiv \ luzera(U) \leq 1$$

Iterazioaren gorputza. Bi kasu induktibo daude.
Lehenengo kasu induktiboan ($lehena(U) > lehena(hondarra(U))$):

$$\begin{aligned}&HurBainoHandi(S)\\ &\quad = \underbrace{HurBainoHandi(U)}_{destolestu} + v\\ &\quad = \underbrace{(1 + HurBainoHandi(hondarra(U))) + v}_{tolestu}\\ &\quad = HurBainoHandi(\underbrace{hondarra(U)}) + \underbrace{1 + v}\\ &\quad = HurBainoHandi(\quad U' \quad) + \quad v'\end{aligned}$$

Aldibereko asignazioa:

```
(U,v) := (hondarra(U),1+v);
```

Aldibereko asignazioaren inplementazioa:

```
U := hondarra(U);
v := 1+v;
```

Bigarren kasu induktiboan $lehena(U) \leq lehena(hondarra(U))$):

$$\begin{aligned}&HurBainoHandi(S)\\ &\quad = \underbrace{HurBainoHandi(U)}_{destolestu/tolestu} + v\\ &\quad = HurBainoHandi(\underbrace{hondarra(U)}) + \underbrace{v}\\ &\quad = HurBainoHandi(\quad U' \quad) + \ v'\end{aligned}$$

Aldibereko asignazioa:

```
(U,v) := (hondarra(U),v);
```

Aldibereko asignazioaren inplementazioa:

```
U := hondarra(U);
```

Funtzio iteratiboa:

```
function HurBainoHandi_it(S : sekuentzia(nat)) return nat is
{ true }
   U : sekuentzia(nat);
   v :  nat;
begin
   U := S;
   v := 0;
         INB ≡ { HurBainoHandi(S) = HurBainoHandi(U) + v }
   while luzera(U) <= 1 loop
      if lehena(U) > lehena(hondarra(U)) then
         U := hondarra(U);
         v := 1+v;
      else
         U := hondarra(U);
      end if;
   end loop;
   return v;
end HurBainoHandi_it;
{ HurBainoHandi(S) = HurBainoHandi_it(S) }
```

1.3. f funtzio, $1 \leq n \leq m$ aurrebaldintza hartuta:

$$f(m,n) = \begin{cases} 1 & \texttt{baldin } n = m \\ \frac{(n+1)\times f(m,n+1)}{(m-n)} & \texttt{baldin } n < m \end{cases}$$

Errekurrentzia erlazioa:

$$f(m,n) = \frac{u \times f(m,v)}{w}$$

Soluzioa:

Hasieraketa:

```
u := 1;
v := n;
w := 1;
```

Emaitza. Kasu nabari bakar bat dago:

$$m = v \;\rightarrow\; f(x,y) \;=\; \frac{u \times f(m,v)}{w} \;=\; \frac{u \times f(m,m)}{w} \;=\; \frac{u \times 1}{w} \;=\; \frac{u}{w}$$

Beraz:

```
return u/w;
```

Bukaera baldintza. Kasu nabari bakat bat dago:

$$bukbal(u,v,w) \ \equiv \ m = v$$

Iterazioaren gorputza. Kasu induktibo bakar bat dago:

$$\begin{aligned}
&f(m,n)\\
&\quad = u \times \underbrace{f(m,v)}_{destolestu} /w\\
&\qquad = \underbrace{u \times ((v+1) \times f(m,v+1)/(m-v))/w}_{tolestu}\\
&\quad = \underbrace{u \times (v+1)} \times f(m, \underbrace{v+1})/\underbrace{((m-v) \times w)}\\
&\quad = \quad u' \quad \times f(m, \quad v' \ )/ \quad w'
\end{aligned}$$

Aldibereko asignazioa:

```
(u,v,w) := (u*(v+1),v+1,(m-v)*w);
```

Aldibereko asignazioaren inplementazioa:

```
u := u*(v+1);
w := (m-v)*w;
v := v+1;
```

Funtzio iteratiboa:

```
function f_it(m, n : nat) return nat is
{ 1 ≤ n ≤ m }
   u, v, w : nat;
begin
   u := 1;
   v := n;
   w := 1;
   while m /= v loop        INB ≡ { f(m,n) = u × f(m,v)/w }
      u := u*(v+1);
      w := (m-v)*w;
      v := v+1;
   end loop;
   return u/w;
end f_it;
{ f(m,n) = f_it(m,n) }
```

1.4. $f : nat^{+} \rightarrow nat^{+}$ funtzioa errekurtsibitate bikoitzekoa da:

$$f(n) = \begin{cases} n & \texttt{baldin } 1 \leq n \leq 2 \\ f(n-3) & \texttt{baldin } n \geq 3 \wedge bikoiti(n) \\ f(n-1)+f(n-2) & \texttt{baldin } n \geq 3 \wedge bakoiti(n) \end{cases}$$

Errekurrentzia erlazioa:

$$f(n) = x \times f(2 \times z) + y \times f(2 \times z - 1)$$

Laguntza! Hasieraketan baikoitiak eta bikoitiak bereizi egin behar dira.

Soluzioa:

<u>Hasieraketa</u>. Bi kasu kontuan hartuko dugu.

n bikoitia denean:

```
x := 1;
y := 0;
z := n/2;
```

n bakoitia denean:

```
x := 0;
y := 1;
z := n/2+1;
```

<u>Emaitza</u>. Kasu nabari bakar bat dago:

$$\begin{aligned} z = 1 \;\rightarrow\; f(n) \;&=\; x \times f(2 \times z) + y \times f(2 \times z - 1) \;=\; x \times f(2) + y \times f(1) \\ &=\; x \times 2 + y \times 1 \;=\; x \times 2 + y \end{aligned}$$

Beraz:

```
return x*2+y;
```

<u>Bukaera baldintza</u>. Kasu nabari bakar bat dago:

$$bukbal(x, y, z) \;\equiv\; z = 1$$

<u>Iterazioaren gorputza</u>:

$$\begin{aligned} f(n) &= x \times \underbrace{f(2 \times z)}_{destolestu} + y \times f(2 \times z - 1) \\ &= x \times f(2 \times z - 3) + y \times \underbrace{f(2 \times z - 1)}_{destolestu} \\ &= x \times f(2 \times z - 3) + y \times (f(2 \times z - 1 - 1) + f(2 \times z - 1 - 2)) \\ &= \underbrace{x \times f(2 \times (z-1) - 1) + y \times (f(2 \times (z-1)) + f(2 \times (z-1) - 1))}_{tolestu} \\ &= \underbrace{y} \times f(2 \times \underbrace{(z-1)}) + \underbrace{(x+y)} \times f(2 \times \underbrace{(z-1)} - 1) \\ &= x' \times f(2 \times z') + y' \times f(2 \times z' - 1) \end{aligned}$$

Aldibereko asignazioa:

```
(x,y,z) := (y,x+y,z-1);
```

Aldibereko asignazioaren inplementazioa:

```
y := x+y;
x := y-x;
z := z-1;
```

Funtzio iteratiboa:

```
function f_it(n : nat) return nat is
{ n ≥ 1 }
   x, y, z : nat;
begin
   if bikoiti(n) then
      x := 1;
      y := 0;
      z := n/2;
   else
      x := 0;
      y := 1;
      z := n/2+1;
   end if;
         INB ≡ { f(n) = x × f(2 × z) + y × f(2 × z − 1) }
   while z /= 1 loop
      y := x+y;
      x := y-x;
      z := z-1;
   end loop;
   return x*2+y;
end f_it;
{ f(n) = f_it(n) }
```

2. Hurrengo funtzio errekurtsiboak emanda, errekurrentzia erlazioa atera.

2.1. Aurre $\equiv \{ x \geq 0 \}$
Post $\equiv \{ emaitza = f(x,y) \}$

$$f(x,y) = \begin{cases} y & \texttt{baldin } x = 0 \\ x * y + f(x-1, y+1) & \texttt{bestela } x > 0 \end{cases}$$

Soluzioa:

Destolesketa/tolesketa teknika erabiliko dugu $f(4,2)$ adibidearen gainean:

$$\begin{aligned}
&\underbrace{f(4,2)}_{destolestu/tolestu} \\
&= 8 + \underbrace{f(3,3)}_{destolestu} \\
&\qquad = \underbrace{8 + (9 + f(2,4))}_{tolestu} \\
&= 17 + \underbrace{f(2,4)}_{destolestu} \\
&\qquad = \underbrace{17 + (8 + f(1,5))}_{tolestu} \\
&= 25 + \underbrace{f(1,5)}_{destolestu} \\
&\qquad = \underbrace{25 + (5 + f(0,6))}_{tolestu} \\
&= 30 + \underbrace{f(0,6)}_{destolestu} \\
&= \underbrace{30 + 6}_{tolestu} \\
&= 36
\end{aligned}$$

Beraz, betetzen den errekurrentzia erlazioa hurrengoa da:

$$f(x,y) = u + f(v,w)$$

2.2. $partitu : T \times sekuentzia(T) \rightarrow sekuentzia(T)$ funtzioak sekuentzia baten partizioa lortzen du, emandako elementu bat baino txikiagoak edo berdinak direnak hasieran jarriz eta handiagoak bukaeran.

$$
\begin{aligned}
&(1)\ partitu(x, \langle\ \rangle) = \langle\ \rangle \\
&(2)\ partitu(x, y \bullet S) = \begin{cases} y \bullet partitu(x,S) & \texttt{baldin}\ y \leq x \\ partitu(x,S)@\langle y\rangle & \texttt{bestela} \end{cases}
\end{aligned}
$$

Soluzioa:

Destolesketa/tolesketa teknika erabiliko dugu $partitu(3, \langle 1,4,2,5\rangle)$ adibidearen gainean:

$$
\begin{aligned}
&\underbrace{partitu(3, \langle 1,4,2,5\rangle)}_{destolestu/tolestu} \\
&\quad = 1 \bullet \underbrace{partitu(3, \langle 4,2,5\rangle)}_{destolestu} \qquad (= \langle 1\rangle@partitu(3, \langle 4,2,5\rangle)\ ) \\
&\qquad\quad = \underbrace{1 \bullet (partitu(3, \langle 2,5\rangle)@\langle 4\rangle)}_{tolestu} \quad (= \langle 1\rangle@(partitu(3, \langle 2,5\rangle)@\langle 4\rangle)\ ) \\
&\quad = \langle 1\rangle@\underbrace{partitu(3, \langle 2,5\rangle)}_{destolestu}@\langle 4\rangle \\
&\qquad\quad = \underbrace{\langle 1\rangle@(2 \bullet partitu(3, \langle 5\rangle))@\langle 4\rangle}_{tolestu} \\
&\quad = \langle 1,2\rangle@\underbrace{partitu(3, \langle 5\rangle)}_{destolestu}@\langle 4\rangle \\
&\qquad\quad = \underbrace{\langle 1,2\rangle@(partitu(3, \langle\ \rangle)@\langle 5\rangle)@\langle 4\rangle}_{tolestu} \\
&\quad = \langle 1,2\rangle@\underbrace{partitu(3, \langle\ \rangle)}_{destolestu}@\langle 5,4\rangle \\
&\qquad\quad = \underbrace{\langle 1,2\rangle@\langle\ \rangle@\langle 5,4\rangle}_{tolestu} \\
&\quad = \langle 1,2,5,4\rangle
\end{aligned}
$$

Beraz, betetzen den errekurrentzia erlazioa hurrengoa da:

$$partitu(x,S) = U@partitu(x,V)@W$$

2.3. *irauli* : *pila*(T) $\times$ *pila*(T) $\rightarrow$ *pila*(T) funtzioak pila bat beste batean iraultzen du.

(1) $irauli(hutsa, Q) = Q$
(2) $irauli(pilaratu(K, x), Q) = irauli(K, pilaratu(Q, x))$

Soluzioa:

Destolesketa/tolesketa teknika erabiliko dugu $irauli(\left[\!\left[\begin{array}{c}1\\2\\3\end{array}\right]\!\right], \left[\!\left[\begin{array}{c}4\\5\end{array}\right]\!\right])$ adibidearen gainean:

$$\underbrace{irauli(\left[\!\left[\begin{array}{c}1\\2\\3\end{array}\right]\!\right], \left[\!\left[\begin{array}{c}4\\5\end{array}\right]\!\right])}_{destolestu/tolestu}$$

$$= \underbrace{irauli(\left[\!\left[\begin{array}{c}2\\3\end{array}\right]\!\right], \left[\!\left[\begin{array}{c}1\\4\\5\end{array}\right]\!\right])}_{destolestu/tolestu}$$

$$= \underbrace{irauli(\left[\!\left[\begin{array}{c}3\end{array}\right]\!\right], \left[\!\left[\begin{array}{c}2\\1\\4\\5\end{array}\right]\!\right])}_{destolestu/tolestu}$$

$$= \underbrace{irauli([\![]\!], \left[\!\left[\begin{array}{c}3\\2\\1\\4\\5\end{array}\right]\!\right])}_{destolestu}$$

$$= \left[\!\left[\begin{array}{c}3\\2\\1\\4\\5\end{array}\right]\!\right]$$

Beraz, betetzen den errekurrentzia erlazioa hurrengoa da:

$$irauli(P, Q) = irauli(U, V)$$

3. Burstall-en metodoa erabiliz, bihurtu iteratibo honako funtzio errekurtsiboak (ekuazionalki definituak).

3.1. *mugkop* funtzioak, $n \geq 1$ (disko-kopurua) emanda, Hanoiko dorreen problema n diskotarako ebazteko egin behar diren mugimenduaren kopurua kalkulatzen du.

$$mugkop(n) = \begin{cases} 1 & \texttt{baldin } n = 1 \\ 2 * mugkop(n-1) + 1 & \texttt{bestela} \end{cases}$$

Soluzioa:

<u>Errekurrentzia erlazioa</u> ateratzeko destolesketa/tolesketa teknika erabiliko dugu $mugkop(3)$ adibidearen gainean:

$$\begin{aligned}
&\underbrace{mugkop(4)}_{destolestu/tolestu} \\
&= 2 \times \underbrace{mugkop(3)}_{destolestu} +1 \\
&\qquad = \underbrace{2 \times (2 \times mugkop(2) + 1) + 1}_{tolestu} \\
&= 4 \times \underbrace{mugkop(2)}_{destolestu} +3 \\
&\qquad = \underbrace{4 \times (2 + mugkop(1) + 1) + 3}_{tolestu} \\
&= 8 \times \underbrace{mugkop(1)}_{destolestu} +7 \\
&\qquad = \underbrace{8 \times 1 + 7}_{tolestu} \\
&= 15
\end{aligned}$$

Beraz, betetzen den errekurrentzia erlazioa hurrengoa da:

$$mugkop(n) = u \times mugkop(v) + w$$

<u>Hasieraketa</u>:

```
u := 1;
v := n;
w := 0;
```

<u>Emaitza</u>. Kasu nabari bakar bat dago:

$$\begin{aligned}
v = 1 \ \rightarrow \ mugkop(n) &= u \times mugkop(v) + w \ = \ u \times mugkop(1) + w \\
&= u \times 1 + w \ = \ u + w
\end{aligned}$$

Beraz:

```
return u+w;
```

Bukaera baldintza. Kasu nabari bakat bat dago:

$$bukbal(u,v,w) \;\equiv\; v=1$$

Iterazioaren gorputza. Kasu induktibo bakar bat dago:

$$\begin{aligned}
&mugkop(n)\\
&\qquad = u \times \underbrace{mugkop(v)}_{destolestu} + w\\
&\qquad\qquad = \underbrace{u \times (2 \times mugkop(v-1)+1)+w}_{tolestu}\\
&\qquad = \underbrace{u \times 2}\times mugkop(\underbrace{v-1})+\underbrace{w+u}\\
&\qquad = \quad u' \quad \times mugkop(\quad v' \quad)+ \quad w'
\end{aligned}$$

Aldibereko asignazioa:

```
(u,v,w) := (u*2,v-1,w+u);
```

Aldibereko asignazioaren inplementazioa:

```
w := w+u;
u := u*2;
v := v-1;
```

Funtzio iteratiboa:

```
function mugkop_it(n : nat) return nat is
{ n ≥ 1 }
   u, v, w : nat;
begin
   u := 1;
   v := n;
   w := 0;
         INB ≡ { mugkop(n) = u × mugkop(v) + w }
   while n /= 1 loop
      w := w+u;
      u := u*2;
      v := v-1;
   end loop;
   return u+w;
end mugkop_it;
{ mugkop(n) = mugkop_it(n) }
```

3.2. *bitar* funtzioak x zenbaki arruntaren balio bitarra kalkulatzen du.

$$bitar(x) = \begin{cases} x & \texttt{baldin } 0 \le x \le 1 \\ 10 * bitar(x/2) & \texttt{baldin } bikoiti(x) \wedge x \ge 2 \\ 10 * bitar(x/2) + 1 & \texttt{baldin } bakoiti(x) \wedge x \ge 2 \end{cases}$$

Soluzioa:

<u>Errekurrentzia erlazioa</u> ateratzeko destolesketa/tolesketa teknika erabiliko dugu $bitar(11)$ adibidearen gainean:

$$\underbrace{bitar(11)}_{destolestu/tolestu}$$

$$= 10 \times \underbrace{bitar(5)}_{destolestu} +1$$

$$= \underbrace{10 \times (10 \times bitar(2) + 1) + 1}_{tolestu}$$

$$= 100 \times \underbrace{bitar(2)}_{destolestu} +11$$

$$= \underbrace{100 \times (10 \times bitar(1)) + 11}_{tolestu}$$

$$= 1000 \times \underbrace{bitar(1)}_{destolestu} +11$$

$$= \underbrace{1000 \times 1 + 11}_{tolestu}$$

$$= 1011$$

Beraz, betetzen den errekurrentzia erlazioa hurrengoa da:

$$bitar(x) = u \times bitar(v) + w$$

<u>Hasieraketa</u>:

```
u := 1;
v := x;
w := 0;
```

<u>Emaitza</u>. Kasu nabari bakar bat dago:

$$0 \le v \le 1 \ \rightarrow \ bitar(x) \ = \ u \times bitar(v) + w \ = \ u \times v + w$$

Beraz:

```
return u*v+w;
```

<u>Bukaera baldintza</u>. Kasu nabari bakat bat dago:

$$bukbal(u, v, w) \ \equiv \ 0 \le v \le 1$$

Iterazioaren gorputza. Bi kasu induktibo daude.

Lehenengo kasu induktiboan ($bikoiti(v) \wedge v \geq 2$):

$$
\begin{aligned}
&bitar(x)\\
&\quad = u \times \underbrace{bitar(v)}_{destolestu} + w\\
&\qquad = \underbrace{u \times (10 \times bitar(\tfrac{v}{2})) + w}_{tolestu}\\
&\quad = \underbrace{u \times 10} \times bitar(\underbrace{\tfrac{v}{2}}) + \underbrace{w}\\
&\quad = \quad u' \quad \times bitar(\ v'\ ) + \ w'
\end{aligned}
$$

Aldibereko asignazioa:

```
(u,v,w) := (u*10,v/2,w);
```

Aldibereko asignazioaren inplementazioa:

```
u := u*10;
v := v/2;
```

Bigarren kasu induktiboan ($bakoiti(v) \wedge v \geq 2$):

$$
\begin{aligned}
&bitar(x)\\
&\quad = u \times \underbrace{bitar(v)}_{destolestu} + w\\
&\qquad = \underbrace{u \times (10 \times bitar(\tfrac{v}{2}) + 1) + w}_{tolestu}\\
&\quad = \underbrace{u \times 10} \times bitar(\underbrace{\tfrac{v}{2}}) + \underbrace{w + u}\\
&\quad = \quad u' \quad \times bitar(\ v'\ ) + \ w'
\end{aligned}
$$

Aldibereko asignazioa:

```
(u,v,w) := (u*10,v/2,w+u);
```

Aldibereko asignazioaren inplementazioa:

```
w := w+u;
u := u*10;
v := v/2;
```

Funtzio iteratiboa:

```
function bitar_it(x : nat) return nat is
{ x ≥ 0 }
   u, v, w : nat;
begin
   u := 1;
   v := x;
   w := 0;
   while x > 1 loop        INB ≡ { bitar(x) = u × bitar(v) + w }
      if bikoiti(v) then
         u := u*10;
         v := v/2;
      else
         w := w+u;
         u := u*10;
         v := v/2;
      end if;
   end loop;
   return u*v+w;
end bitar_it;
{ bitar(x) = bitar_it(x) }
```

3.3. $azkenin : arbit(T) \to T$ funtzioak arbola bitar bateko azkeneko nodoa lortzen du, inordenan korrituta.

$$(1)\ azkenin(hutsa) = errore$$
$$(2)\ azkenin(errotu(x, Ezker, Ezkuin) = \begin{cases} x & \texttt{baldin } hutsa\_da(Eskuin) \\ azkenin(Eskuin) & \texttt{bestela} \end{cases}$$

Soluzioa:

Errekurrentzia erlazioa ateratzeko destolesketa/tolesketa teknika erabiliko dugu $azkenin(errotu(1, hutsa, errotu(2, errotu(3, hutsa, hutsa), hutsa)))$ adibidearen gainean:

$$\underbrace{azkenin(errotu(1, hutsa, errotu(2, errotu(3, hutsa, hutsa), hutsa)))}_{destolestu/tolestu}$$
$$= \underbrace{azkenin(errotu(2, errotu(3, hutsa, hutsa), hutsa))}_{destolestu/tolestu}$$
$$= 2$$

Beraz, betetzen den errekurrentzia erlazioa hurrengoa da:

$$azkenin(A) = azkenin(U)$$

Hasieraketa:

```
U := A;
```

Emaitza. Kasu nabari bakar bat dago:

$$hutsa\_da?(eskuin(U)) \;\rightarrow\; azkenin(A) \;=\; azkenin(U) \;=\; erroa(U)$$

Beraz:

```
return erroa(U);
```

Bukaera baldintza. Kasu nabari bakat bat dago:

$$bukbal(U) \;\equiv\; hutsa\_da?(eskuin(U))$$

Iterazioaren gorputza. Kasu induktibo bakar bat dago:

$$\begin{aligned} & azkenin(A) \\ & \quad = \underbrace{azkenin(U)}_{destolestu/tolestu} \\ & \quad = azkenin(\underbrace{eskuin(U)}) \\ & \quad = azkenin(\quad U' \quad) \end{aligned}$$

Aldibereko asignazioa:

```
U := eskuin(U);
```

Funtzio iteratiboa:

```
function azkenin_it(A : arbit(T)) return T is
{ ¬hutsa_da?(A) }
   U : arbit(T);
begin
   U := A;
         INB ≡ { azkenin(A) = azkenin(U) }
   while not hutsa_da?(eskuin(U)) loop
      U := eskuin(U);
   end loop;
   return erroa(U);
end azkenin_it;
{ azkenin(A) = azkenin_it(U) }
```

3.4. $elkartu : sekuentzia(T) \times sekuentzia(T) \rightarrow sekuentzia(T)$ funtzioak bi sekuentzia elkartu egiten ditu.

$$
\begin{aligned}
&(1)\ elkartu(\langle\ \rangle, S) = S\\
&(2)\ elkartu(R, \langle\ \rangle) = R\\
&(3)\ elkartu(x \bullet R, y \bullet S) = \langle x, y\rangle @ elkartu(R, S)
\end{aligned}
$$

Soluzioa:

<u>Errekurrentzia erlazioa</u> ateratzeko destolesketa/tolesketa teknika erabiliko dugu $elkartu(\langle 1,2,3\rangle, \langle 4,5\rangle)$ adibidearen gainean:

$$
\begin{aligned}
&\underbrace{elkartu(\langle 1,2,3\rangle, \langle 4,5\rangle)}_{destolestu/tolestu}\\
&\quad = \langle 1,4\rangle @ \underbrace{elkartu(\langle 2,3\rangle, \langle 5\rangle)}_{destolestu}\\
&\qquad = \underbrace{\langle 1,4\rangle @ (\langle 2,5\rangle) @ elkartu(\langle 3\rangle, \langle\ \rangle)}_{tolestu}\\
&\quad = \langle 1,4,2,5\rangle @ \underbrace{elkartu(\langle 3\rangle, \langle\ \rangle)}_{destolestu}\\
&\qquad = \underbrace{\langle 1,4,2,5\rangle @ \langle 3\rangle)}_{tolestu}\\
&\quad = \langle 1,4,2,5,3\rangle
\end{aligned}
$$

Beraz, betetzen den errekurrentzia erlazioa hurrengoa da:

$$elkartu(R, S) = U @ elkartu(V, W)$$

<u>Hasieraketa</u>:

```
U := <>;
V := R;
W := S;
```

<u>Emaitza</u>. Bi kasu nabari daude.

Lehenengo kasu nabarian (hutsa_da?(V)):

$$hutsa\_da?(V) \;\rightarrow\; elkartu(R, S) \;=\; U @ (V, W) \;=\; U @ elkartu(\langle\ \rangle, W) \;=\; U @ W$$

Beraz:

```
return U @ W;
```

Bigarren kasu nabarian (hutsa_da?(W)):

$$hutsa\_da?(W) \;\rightarrow\; elkartu(R, S) \;=\; U @ (V, W) \;=\; U @ elkartu(V, \langle\ \rangle) \;=\; U @ V$$

Beraz:

```
return U @ V;
```

Bukaera baldintza. Bi kasu nabari daudenez, baldintza konposatua da:

$$bukbal(U, V, W) \;\equiv\; hutsa\_da?(V) \;\vee hutsa\_da?(W)$$

Iterazioaren gorputza. Kasu induktibo bakar bat dago:

$$\begin{aligned}
&elkartu(R, S)\\
&\quad = U@\underbrace{elkartu(V, W)}_{destolestu}\\
&\quad = \underbrace{U@(\langle lehena(V), lehena(W)\rangle @elkartu(hondarra(V), hondarra(W)))}_{tolestu}\\
&\quad = \underbrace{U@\langle lehena(V), lehena(W)\rangle}_{U'} @elkartu(\underbrace{hondarra(V)}_{V'}, \underbrace{hondarra(W)}_{W'})\\
&\quad = U' @elkartu(V', W')
\end{aligned}$$

Aldibereko asignazioa:

```
(U,V,W) := (U @ <lehena(V),lehena(W)>,hondarra(V),hondarra(W));
```

Aldibereko asignazioaren inplementazioa:

```
U := U @ <lehena(V),lehena(W)>;
V := hondarra(V);
W := hondarra(W);
```

Funtzio iteratiboa:

```
function elkartu_it(R, S : sekuentzia(T))
                                        return sekuentzia(T) is
{ true }
   U, V, W : sekuentzia(T);
begin
   U := <>;
   V := R;
   W := S;
      INB ≡ { elkartu(R,S) = U@elkartu(V,W) }
   while not hutsa_da?(V) & not hutsa_da?(W) loop
      U := U @ <lehena(V),lehena(W)>;
      V := hondarra(V);
      W := hondarra(W);
   end loop;
   return V;
end elkartu_it;
{ elkartu(R,S) = elkartu_it(R,S) }
```