While Loop

while b **do** P **od**

While Loop

$W \Leftarrow$ **while** b **do** P **od**

While Loop

$$W \Leftarrow \textbf{while } b \textbf{ do } P \textbf{ od}$$

means

$$W \Leftarrow \textbf{if } b \textbf{ then } P.\ W \textbf{ else } ok \textbf{ fi}$$

While Loop

$W \;\Leftarrow\;$ **while** b **do** P **od**

means

$W \;\Leftarrow\;$ **if** b **then** $P.\ W$ **else** ok **fi**

while $n \neq \#L$ **do** $s{:=}\ s + L\,n.\ \ n{:=}\ n{+}1.\ \ t{:=}\ t{+}1$ **od**

While Loop

$W \;\Leftarrow\;$ **while** b **do** P **od**

means

$W \;\Leftarrow\;$ **if** b **then** $P.\; W$ **else** ok **fi**

$s' = s + \Sigma\, L\, [n;..\#L] \;\wedge\; t' = t + \#L - n \;\Leftarrow$

while $n \neq \#L$ **do** $s := s + L\, n.\;\; n := n+1.\;\; t := t+1$ **od**

While Loop

$$W \Leftarrow \textbf{while } b \textbf{ do } P \textbf{ od}$$

means

$$W \Leftarrow \textbf{if } b \textbf{ then } P.\ W \textbf{ else } ok \textbf{ fi}$$

to prove

$$s' = s + \Sigma\, L\, [n;..\#L] \ \wedge\ t' = t + \#L - n \ \Leftarrow$$

$$\textbf{while } n{\ne}\#L \textbf{ do } s{:=}\ s + L\,n.\ n{:=}\ n{+}1.\ t{:=}\ t{+}1 \textbf{ od}$$

prove instead

$$s' = s + \Sigma\, L\, [n;..\#L] \ \wedge\ t' = t + \#L - n \ \Leftarrow$$

$$\textbf{if } n{\ne}\#L \textbf{ then } \quad s{:=}\ s + L\,n.\ n{:=}\ n{+}1.\ t{:=}\ t{+}1.$$

$$s' = s + \Sigma\, L\, [n;..\#L] \ \wedge\ t' = t + \#L - n$$

$$\textbf{else } ok \textbf{ fi}$$

Exit Loop

do

A.

exit when b.

C

od

Exit Loop

$L \Leftarrow$ **do**

$A.$

exit when $b.$

C

od

Exit Loop

$$L \Leftarrow \quad \textbf{do}$$

$$A.$$

$$\textbf{exit when } b.$$

$$C$$

$$\textbf{od}$$

means

$$L \Leftarrow A.\ \textbf{if } b \textbf{ then } ok \textbf{ else } C.\ L \textbf{ fi}$$

Exit Loop

$L \Leftarrow$ **do**

A.

exit when b.

C

od

means

$L \Leftarrow A$. **if** b **then** ok **else** C. L **fi**

Exit Loop

$L \Leftarrow$ **do**

$A.$

exit when $b.$ ←

C

od

means

$L \Leftarrow A.$ **if** b **then** ok **else** $C.$ L **fi**

Exit Loop

$L \Leftarrow$ **do**

$A.$

exit when $b.$

C

od

means

$$L \Leftarrow A.\ \textbf{if}\ b\ \textbf{then}\ ok\ \textbf{else}\ C.\ L\ \textbf{fi}$$

Exit Loop

$L \Leftarrow$ **do**

$A.$

exit when $b.$

C

od ←

means

$L \Leftarrow A.$ **if** b **then** ok **else** $C.$ L **fi**

↑

Deep Exit

do

  A.

  do

    B.

    exit 2 **when** c.

    D

  od.

  E

od

Deep Exit

$P \Leftarrow$ **do**

    $A.$

    do

        $B.$

        exit 2 **when** $c.$

        D

    od.

    E

od

Deep Exit

$P \Leftarrow$ **do**

A.

do

B.

exit 2 **when** c.

D

od.

E

od

means

$P \Leftarrow$

Deep Exit

$$
\begin{aligned}
P \;\Leftarrow\;\; & \textbf{do} \\
& \quad A. \;\; \color{red}{\longleftarrow} \\
& \quad \textbf{do} \\
& \quad\quad B. \\
& \quad\quad \textbf{exit}\ 2\ \textbf{when}\ c. \\
& \quad\quad D \\
& \quad \textbf{od}. \\
& \quad E \\
& \textbf{od}
\end{aligned}
$$

means

$$P \;\Leftarrow\; A.$$

Deep Exit

$P \Leftarrow$ **do**

A.

do $\leftarrow$

B.

exit 2 **when** c.

D

od.

E

od

means

$P \Leftarrow A.\ Q$

$Q \Leftarrow$

Deep Exit

$$P \Leftarrow \quad \textbf{do}$$

$$\qquad A.$$

$$\qquad \textbf{do}$$

$$\qquad\qquad B. \quad \leftarrow$$

$$\qquad\qquad \textbf{exit } 2 \textbf{ when } c.$$

$$\qquad\qquad D$$

$$\qquad \textbf{od}.$$

$$\qquad E$$

$$\textbf{od}$$

means

$$P \Leftarrow A.\ Q$$

$$Q \Leftarrow B.$$

Deep Exit

$P \Leftarrow$ **do**

A.

do

B.

exit 2 **when** c. ←

D

od.

E

od

means

$P \Leftarrow A.\ Q$

$Q \Leftarrow B.$ **if** c **then** ok

Deep Exit

$$
\begin{aligned}
P \;\Leftarrow\; & \textbf{do} \\
& \quad A. \\
& \quad \textbf{do} \\
& \qquad B. \\
& \qquad \textbf{exit}\ 2\ \textbf{when}\ c. \\
& \qquad D \;\longleftarrow \\
& \quad \textbf{od}. \\
& \quad E \\
& \textbf{od}
\end{aligned}
$$

means

$$P \;\Leftarrow\; A.\ Q$$

$$Q \;\Leftarrow\; B.\ \textbf{if}\ c\ \textbf{then}\ ok\ \textbf{else}\ D.$$

Deep Exit

$P \Leftarrow$ **do**

$A.$

do

$B.$

exit 2 **when** $c.$

D

od. ←

E

od

means

$P \Leftarrow A.\ Q$

$Q \Leftarrow B.$ **if** c **then** ok **else** $D.\ Q$ **fi**

Deep Exit

$P \Leftarrow$ **do**

  $A.$

  do

    $B.$

    exit 2 **when** $c.$

    D

  od.

  $E \;\; \leftarrow \; ?$

od

means

$P \Leftarrow A.\ Q$

$Q \Leftarrow B.$ **if** c **then** ok **else** $D.\ Q$ **fi**

Deep Exit

$P \Leftarrow$ **do**

  A.

  exit 1 **when** b.

  C.

  do

    D.

    exit 2 **when** e.

    F.

    exit 1 **when** g.

    H

  od.

  I

od

Deep Exit

$$
\begin{aligned}
P \quad \Leftarrow \quad & \textbf{do} \\
& \quad A. \\
& \quad \textbf{exit } 1 \textbf{ when } b. \\
& \quad C. \\
& \quad \textbf{do} \\
& \quad\quad D. \\
& \quad\quad \textbf{exit } 2 \textbf{ when } e. \\
& \quad\quad F. \\
& \quad\quad \textbf{exit } 1 \textbf{ when } g. \\
& \quad\quad H \\
& \quad \textbf{od}. \\
& \quad I \\
& \textbf{od}
\end{aligned}
$$

means

$P \quad \Leftarrow$

Deep Exit

$P \Leftarrow$ **do**

  $A.$ $\longleftarrow$

  exit 1 **when** $b.$

  $C.$

  do

    $D.$

    exit 2 **when** $e.$

    $F.$

    exit 1 **when** $g.$

    H

  od.

  I

od

means

$P \Leftarrow A.$

Deep Exit

$P \Leftarrow$ **do**

A.

exit 1 **when** b. $\leftarrow$

C.

do

D.

exit 2 **when** e.

F.

exit 1 **when** g.

H

od.

I

od

means

$P \Leftarrow A$. **if** b **then** ok

Deep Exit

$P \Leftarrow$ **do**

$A.$

exit 1 **when** $b.$

$C.$ ←

do

$D.$

exit 2 **when** $e.$

$F.$

exit 1 **when** $g.$

H

od.

I

od

means

$P \Leftarrow A.$ **if** b **then** ok **else** $C.$

Deep Exit

$P \Leftarrow$ **do**

A.

exit 1 **when** b.

C.

do $\leftarrow$

D.

exit 2 **when** e.

F.

exit 1 **when** g.

H

od.

I

od

means

$P \Leftarrow A$. **if** b **then** ok **else** C. Q **fi**

$Q \Leftarrow$

Deep Exit

$P \Leftarrow$ **do**

A.

exit 1 **when** b.

C.

do

D. $\leftarrow$

exit 2 **when** e.

F.

exit 1 **when** g.

H

od.

I

od

means

$P \Leftarrow A$. **if** b **then** ok **else** C. Q **fi**

$Q \Leftarrow D$.

Deep Exit

$P \;\Leftarrow\;$ **do**

  A.

  exit 1 **when** b.

  C.

  do

    D.

    exit 2 **when** e. $\longleftarrow$

    F.

    exit 1 **when** g.

    H

  od.

  I

od

means

$P \;\Leftarrow\; A.$ **if** b **then** ok **else** $C.\ Q$ **fi**

$Q \;\Leftarrow\; D.$ **if** e **then** ok

Deep Exit

$P \Leftarrow$ **do**

$\quad A.$

$\quad$ **exit** 1 **when** $b.$

$\quad C.$

$\quad$ **do**

$\qquad D.$

$\qquad$ **exit** 2 **when** $e.$

$\qquad F. \quad \longleftarrow$

$\qquad$ **exit** 1 **when** $g.$

$\qquad H$

$\quad$ **od**.

$\quad I$

od

means

$P \Leftarrow A.$ **if** b **then** ok **else** $C.\ Q$ **fi**

$Q \Leftarrow D.$ **if** e **then** ok **else** $F.$

Deep Exit

$P \Leftarrow$ **do**

  A.
  exit 1 **when** b.
  C.
  do
    D.
    exit 2 **when** e.
    F.
    exit 1 **when** g. ←
    H
  od.
  I
od

means

$P \Leftarrow A$. **if** b **then** *ok* **else** C. Q **fi**

$Q \Leftarrow D$. **if** e **then** *ok* **else** F. **if** g **then**

Deep Exit

$P \Leftarrow$ **do**

A.

exit 1 **when** b.

C.

do

D.

exit 2 **when** e.

F.

exit 1 **when** g.

H

od.

I ⟵

od

means

$P \Leftarrow A$. **if** b **then** ok **else** C. Q **fi**

$Q \Leftarrow D$. **if** e **then** ok **else** F. **if** g **then** I.

Deep Exit

$P \Leftarrow$ **do**

A.

exit 1 **when** b.

C.

do

D.

exit 2 **when** e.

F.

exit 1 **when** g.

H

od.

I

od ←

means

$P \Leftarrow A$. **if** b **then** *ok* **else** C. Q **fi**

$Q \Leftarrow D$. **if** e **then** *ok* **else** F. **if** g **then** I. P

Deep Exit

$P \Leftarrow$ **do**

A.

exit 1 **when** b.

C.

do

D.

exit 2 **when** e.

F.

exit 1 **when** g.

H ←

od.

I

od

means

$P \Leftarrow A.$ **if** b **then** ok **else** $C.\ Q$ **fi**

$Q \Leftarrow D.$ **if** e **then** ok **else** $F.$ **if** g **then** $I.\ P$ **else** $H.$

Deep Exit

$$
\begin{aligned}
P \;\Leftarrow\; & \textbf{do} \\
& \quad A. \\
& \quad \textbf{exit } 1 \textbf{ when } b. \\
& \quad C. \\
& \quad \textbf{do} \\
& \quad\quad D. \\
& \quad\quad \textbf{exit } 2 \textbf{ when } e. \\
& \quad\quad F. \\
& \quad\quad \textbf{exit } 1 \textbf{ when } g. \\
& \quad\quad H \\
& \quad \textbf{od}. \;\leftarrow \\
& \quad I \\
& \textbf{od}
\end{aligned}
$$

means

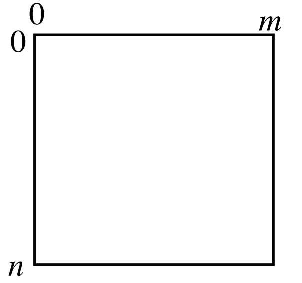
$$P \;\Leftarrow\; A.\ \textbf{if } b \textbf{ then } ok \textbf{ else } C.\ Q \textbf{ fi}$$

$$Q \;\Leftarrow\; D.\ \textbf{if } e \textbf{ then } ok \textbf{ else } F.\ \textbf{if } g \textbf{ then } I.\ P \textbf{ else } H.\ Q \textbf{ fi fi}$$

Two-Dimensional Search

Two-Dimensional Search

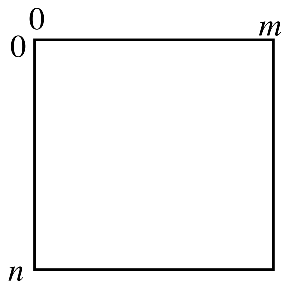
$P \;=\; \textbf{if } x\text{: } A\ (0,..n)\ (0,..m) \textbf{ then } x = A\ i'\ j' \textbf{ else } i'{=}n \wedge j'{=}m \textbf{ fi}$



Two-Dimensional Search

$P \;=\; \textbf{if } x\text{: } A\ (0,..n)\ (0,..m) \textbf{ then } x = A\ i'\ j' \textbf{ else } i'{=}n \land j'{=}m \textbf{ fi}$

$P \;\Leftarrow\; i{:=}\,0.\;\; i{\le}n \Rightarrow Q$



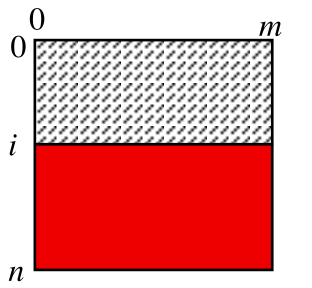
Two-Dimensional Search

P = **if** x: A $(0,..n)$ $(0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

↓

Q = **if** x: A $(i,..n)$ $(0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$P \Leftarrow i:=0.\ i \le n \Rightarrow Q$

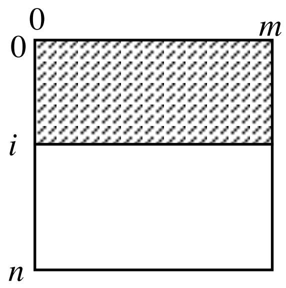


Two-Dimensional Search

$P \;=\;$ **if** x: $A\ (0,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$Q \;=\;$ **if** x: $A\ (i,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$P \;\Leftarrow\; i:=0.\ \ i\le n \Rightarrow Q$



$i\le n \Rightarrow Q \;\;\Leftarrow$

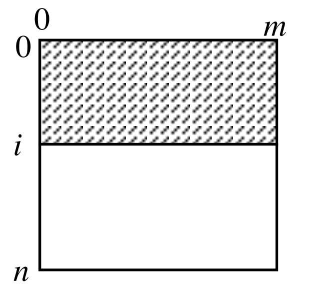
Two-Dimensional Search

$P \;=\; \textbf{if}\ x{:}\ A\ (0,..n)\ (0,..m)\ \textbf{then}\ x = A\ i'\ j'\ \textbf{else}\ i'{=}n \land j'{=}m\ \textbf{fi}$

$Q \;=\; \textbf{if}\ x{:}\ A\ (i,..n)\ (0,..m)\ \textbf{then}\ x = A\ i'\ j'\ \textbf{else}\ i'{=}n \land j'{=}m\ \textbf{fi}$

$P \;\Leftarrow\; i{:=}\,0.\ \ i{\le}n \Rightarrow Q$

$i{\le}n \Rightarrow Q \;\Leftarrow\; \textbf{if}\ i{=}n\ \textbf{then}\ j{:=}\ m$



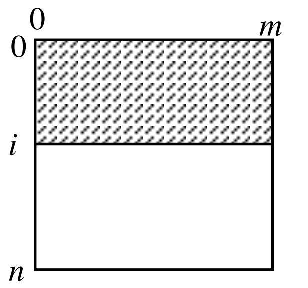
Two-Dimensional Search

$P \;=\; \textbf{if } x: A\,(0,..n)\,(0,..m) \textbf{ then } x = A\, i'\, j' \textbf{ else } i'=n \wedge j'=m \textbf{ fi}$

$Q \;=\; \textbf{if } x: A\,(i,..n)\,(0,..m) \textbf{ then } x = A\, i'\, j' \textbf{ else } i'=n \wedge j'=m \textbf{ fi}$

$P \;\Leftarrow\; i:= 0.\;\; i\le n \Rightarrow Q$

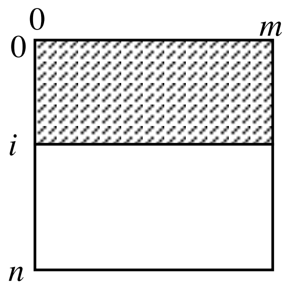
$i\le n \Rightarrow Q \;\Leftarrow\; \textbf{if } i=n \textbf{ then } j:= m \textbf{ else } i<n \Rightarrow Q \textbf{ fi}$



Two-Dimensional Search

$P \;=\; \textbf{if } x\text{: } A\,(0,..n)\,(0,..m) \textbf{ then } x = A\,i'\,j' \textbf{ else } i'=n \wedge j'=m \textbf{ fi}$

$Q \;=\; \textbf{if } x\text{: } A\,(i,..n)\,(0,..m) \textbf{ then } x = A\,i'\,j' \textbf{ else } i'=n \wedge j'=m \textbf{ fi}$



$P \;\Leftarrow\; i:= 0.\;\; i{\le}n \Rightarrow Q$

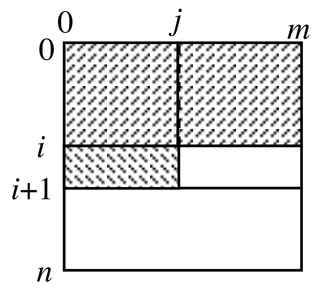
$i{\le}n \Rightarrow Q \;\Leftarrow\; \textbf{if } i=n \textbf{ then } j:= m \textbf{ else } i{<}n \Rightarrow Q \textbf{ fi}$

$i{<}n \Rightarrow Q \;\Leftarrow$

Two-Dimensional Search

$P \;=\; \textbf{if } x\text{: } A\ (0,..n)\ (0,..m) \textbf{ then } x = A\ i'\ j' \textbf{ else } i'{=}n \wedge j'{=}m \textbf{ fi}$

$Q \;=\; \textbf{if } x\text{: } A\ (i,..n)\ (0,..m) \textbf{ then } x = A\ i'\ j' \textbf{ else } i'{=}n \wedge j'{=}m \textbf{ fi}$



$P \;\Leftarrow\; i{:=}\,0.\ \ i{\le}n \Rightarrow Q$

$i{\le}n \Rightarrow Q \;\Leftarrow\; \textbf{if } i{=}n \textbf{ then } j{:=}\ m \textbf{ else } i{<}n \Rightarrow Q \textbf{ fi}$

$i{<}n \Rightarrow Q \;\Leftarrow\; j{:=}\,0.\ \ i{<}n \wedge j{\le}m \Rightarrow R$

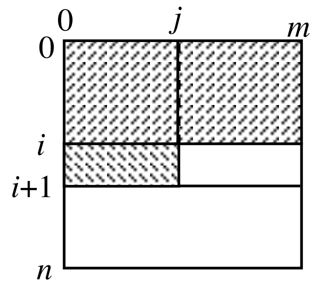
Two-Dimensional Search

$P \; = \;$ **if** x: $A\ (0,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$Q \; = \;$ **if** x: $A\ (i,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$R \; = \;$ **if** x: $A\ i\ (j,..m),\ A\ (i+1,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$P \;\Leftarrow\; i:=0.\ \ i\le n \Rightarrow Q$



$i\le n \Rightarrow Q \;\Leftarrow\;$ **if** $i=n$ **then** $j:=m$ **else** $i<n \Rightarrow Q$ **fi**

$i<n \Rightarrow Q \;\Leftarrow\; j:=0.\ \ i<n \wedge j\le m \Rightarrow R$

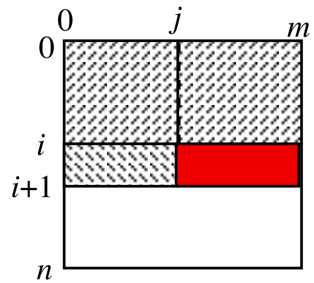
Two-Dimensional Search

$P \;=\;$ **if** x: $A\ (0,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$Q \;=\;$ **if** x: $A\ (i,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$R \;=\;$ **if** x: $\underline{A\ i\ (j,..m)}$, $A\ (i+1,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$P \;\Leftarrow\; i:= 0.\ \ i\le n \Rightarrow Q$



$i\le n \Rightarrow Q \;\Leftarrow\;$ **if** $i=n$ **then** $j:= m$ **else** $i<n \Rightarrow Q$ **fi**

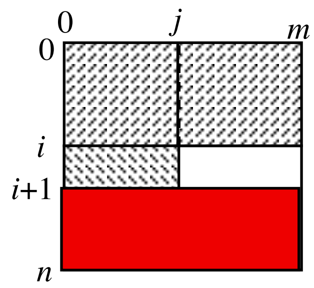
$i<n \Rightarrow Q \;\Leftarrow\; j:= 0.\ \ i<n \wedge j\le m \Rightarrow R$

Two-Dimensional Search

$P \;=\;$ **if** $x{:}\ A\ (0,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**

$Q \;=\;$ **if** $x{:}\ A\ (i,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**

$R \;=\;$ **if** $x{:}\ A\ i\ (j,..m),\ \underline{A\ (i{+}1,..n)\ (0,..m)}$ **then** $x = A\ i'\ j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**



$P \;\Leftarrow\; i{:=}\ 0.\ \ i{\le}n \Rightarrow Q$

$i{\le}n \Rightarrow Q \;\Leftarrow\;$ **if** $i{=}n$ **then** $j{:=}\ m$ **else** $i{<}n \Rightarrow Q$ **fi**

$i{<}n \Rightarrow Q \;\Leftarrow\; j{:=}\ 0.\ \ i{<}n \wedge j{\le}m \Rightarrow R$

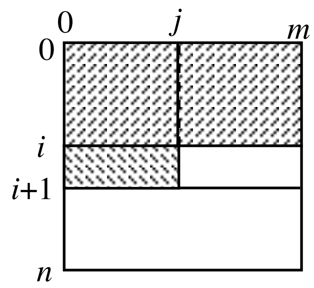
Two-Dimensional Search

$P \;=\;$ **if** x: A (0,..n) (0,..m) **then** $x = A\ i'\ j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**

$Q \;=\;$ **if** x: A (i,..n) (0,..m) **then** $x = A\ i'\ j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**

$R \;=\;$ **if** x: $A\ i$ (j,..m), A (i+1,..n) (0,..m) **then** $x = A\ i'\ j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**

$P \;\Leftarrow\; i{:=}\,0.\;\; i{\le}n \Rightarrow Q$



$i{\le}n \Rightarrow Q \;\Leftarrow\;$ **if** $i{=}n$ **then** $j{:=}\,m$ **else** $i{<}n \Rightarrow Q$ **fi**

$i{<}n \Rightarrow Q \;\Leftarrow\; j{:=}\,0.\;\; i{<}n \wedge j{\le}m \Rightarrow R$

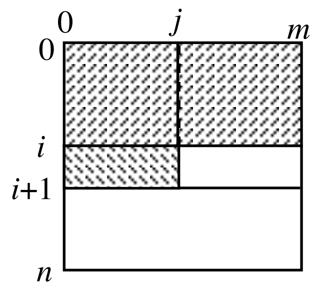
$i{<}n \wedge j{\le}m \Rightarrow R \;\Leftarrow$

Two-Dimensional Search

$P \;=\;$ **if** $x\!: A\ (0,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**

$Q \;=\;$ **if** $x\!: A\ (i,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**

$R \;=\;$ **if** $x\!: A\ i\ (j,..m),\ A\ (i{+}1,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**



$P \;\Leftarrow\; i{:=}\,0.\ \ i{\le}n \Rightarrow Q$

$i{\le}n \Rightarrow Q \;\Leftarrow\;$ **if** $i{=}n$ **then** $j{:=}\ m$ **else** $i{<}n \Rightarrow Q$ **fi**

$i{<}n \Rightarrow Q \;\Leftarrow\; j{:=}\,0.\ \ i{<}n \wedge j{\le}m \Rightarrow R$

$i{<}n \wedge j{\le}m \Rightarrow R \;\Leftarrow\;$ **if** $j{=}m$ **then** $i{:=}\ i{+}1.\ \ i{\le}n \Rightarrow Q$

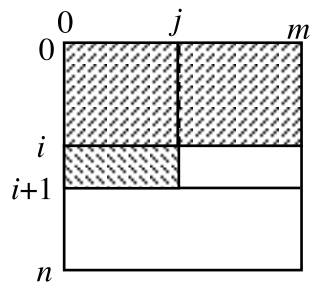
Two-Dimensional Search

$P \;=\;$ **if** $x\colon A\,(0,..n)\,(0,..m)$ **then** $x = A\,i'\,j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**

$Q \;=\;$ **if** $x\colon A\,(i,..n)\,(0,..m)$ **then** $x = A\,i'\,j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**

$R \;=\;$ **if** $x\colon A\,i\,(j,..m),\, A\,(i{+}1,..n)\,(0,..m)$ **then** $x = A\,i'\,j'$ **else** $i'{=}n \wedge j'{=}m$ **fi**

$P \;\Leftarrow\; i{:=}\,0.\;\; i{\le}n \Rightarrow Q$



$i{\le}n \Rightarrow Q \;\Leftarrow\;$ **if** $i{=}n$ **then** $j{:=}\,m$ **else** $i{<}n \Rightarrow Q$ **fi**

$i{<}n \Rightarrow Q \;\Leftarrow\; j{:=}\,0.\;\; i{<}n \wedge j{\le}m \Rightarrow R$

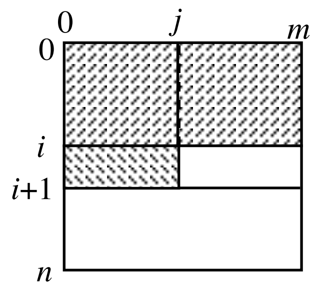
$i{<}n \wedge j{\le}m \Rightarrow R \;\Leftarrow\;$ **if** $j{=}m$ **then** $i{:=}\,i{+}1.\;\; i{\le}n \Rightarrow Q$ **else** $i{<}n \wedge j{<}m \Rightarrow R$ **fi**

Two-Dimensional Search

$P \;=\; \textbf{if } x\!: A\ (0,..n)\ (0,..m) \textbf{ then } x = A\ i'\ j' \textbf{ else } i'=n \wedge j'=m \textbf{ fi}$

$Q \;=\; \textbf{if } x\!: A\ (i,..n)\ (0,..m) \textbf{ then } x = A\ i'\ j' \textbf{ else } i'=n \wedge j'=m \textbf{ fi}$

$R \;=\; \textbf{if } x\!: A\ i\ (j,..m),\ A\ (i+1,..n)\ (0,..m) \textbf{ then } x = A\ i'\ j' \textbf{ else } i'=n \wedge j'=m \textbf{ fi}$



$P \;\Leftarrow\; i{:=}\,0.\ \ i{\le}n \Rightarrow Q$

$i{\le}n \Rightarrow Q \;\Leftarrow\; \textbf{if } i{=}n \textbf{ then } j{:=}\ m \textbf{ else } i{<}n \Rightarrow Q \textbf{ fi}$

$i{<}n \Rightarrow Q \;\Leftarrow\; j{:=}\,0.\ \ i{<}n \wedge j{\le}m \Rightarrow R$

$i{<}n \wedge j{\le}m \Rightarrow R \;\Leftarrow\; \textbf{if } j{=}m \textbf{ then } i{:=}\ i{+}1.\ \ i{\le}n \Rightarrow Q \textbf{ else } i{<}n \wedge j{<}m \Rightarrow R \textbf{ fi}$

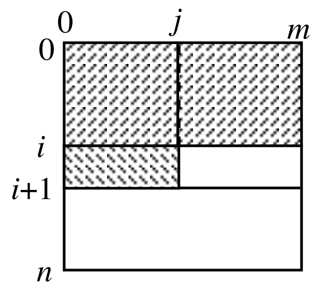
$i{<}n \wedge j{<}m \Rightarrow R \;\Leftarrow$

Two-Dimensional Search

$P \;=\;$ **if** x: $A\ (0,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$Q \;=\;$ **if** x: $A\ (i,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$R \;=\;$ **if** x: $A\ i\ (j,..m),\ A\ (i+1,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**



$P \;\Leftarrow\; i:=0.\ \ i\le n \Rightarrow Q$

$i\le n \Rightarrow Q \;\Leftarrow\;$ **if** $i=n$ **then** $j:=m$ **else** $i<n \Rightarrow Q$ **fi**

$i<n \Rightarrow Q \;\Leftarrow\; j:=0.\ \ i<n \wedge j\le m \Rightarrow R$

$i<n \wedge j\le m \Rightarrow R \;\Leftarrow\;$ **if** $j=m$ **then** $i:=i+1.\ \ i\le n \Rightarrow Q$ **else** $i<n \wedge j<m \Rightarrow R$ **fi**

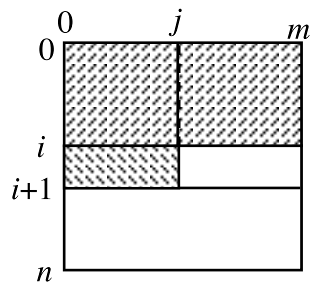
$i<n \wedge j<m \Rightarrow R \;\Leftarrow\;$ **if** $A\ i\ j = x$ **then** ok

Two-Dimensional Search

$P \;=\;$ **if** x: $A\ (0,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$Q \;=\;$ **if** x: $A\ (i,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**

$R \;=\;$ **if** x: $A\ i\ (j,..m),\ A\ (i+1,..n)\ (0,..m)$ **then** $x = A\ i'\ j'$ **else** $i'=n \wedge j'=m$ **fi**



$P \;\Leftarrow\; i:=0.\ \ i\le n \Rightarrow Q$

$i\le n \Rightarrow Q \;\Leftarrow\;$ **if** $i=n$ **then** $j:=m$ **else** $i<n \Rightarrow Q$ **fi**

$i<n \Rightarrow Q \;\Leftarrow\; j:=0.\ \ i<n \wedge j\le m \Rightarrow R$

$i<n \wedge j\le m \Rightarrow R \;\Leftarrow\;$ **if** $j=m$ **then** $i:=i+1.\ \ i\le n \Rightarrow Q$ **else** $i<n \wedge j<m \Rightarrow R$ **fi**

$i<n \wedge j<m \Rightarrow R \;\Leftarrow\;$ **if** $A\ i\ j = x$ **then** ok **else** $j:=j+1.\ \ i<n \wedge j\le m \Rightarrow R$ **fi**

Two-Dimensional Search

$t' \le t + n{\times}m \;\Leftarrow\; i{:=}\,0.\;\; i{\le}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m$

$i{\le}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m \;\Leftarrow\;$ **if** $i{=}n$ **then** $j{:=}\,m$ **else** $i{<}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m$ **fi**

$i{<}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m \;\Leftarrow\; j{:=}\,0.\;\; i{<}n \wedge j{\le}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j$

$i{<}n \wedge j{\le}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j \;\Leftarrow$

$t{:=}\,t{+}1.$

if $j{=}m$ **then** $i{:=}\,i{+}1.\;\; i{\le}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m$

else $i{<}n \wedge j{<}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j$ **fi**

$i{<}n \wedge j{<}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j \;\Leftarrow$

if $A\;i\;j = x$ **then** ok **else** $j{:=}\,j{+}1.\;\; i{<}n \wedge j{\le}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j$ **fi**

Two-Dimensional Search

$t' \le t + n{\times}m \;\Leftarrow\; i{:=}\,0.\;\; i{\le}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m$

$i{\le}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m \;\Leftarrow\;$ **if** $i{=}n$ **then** $j{:=}\,m$ **else** $i{<}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m$ **fi**

$i{<}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m \;\Leftarrow\; j{:=}\,0.\;\; i{<}n \wedge j{\le}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j$

$i{<}n \wedge j{\le}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j \;\Leftarrow$

$t{:=}\,t{+}1.$ ←

if $j{=}m$ **then** $i{:=}\,i{+}1.\;\; i{\le}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m$

else $i{<}n \wedge j{<}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j$ **fi**

$i{<}n \wedge j{<}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j \;\Leftarrow$

if $A\;i\;j = x$ **then** ok **else** $j{:=}\,j{+}1.\;\; i{<}n \wedge j{\le}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j$ **fi**

Two-Dimensional Search

$t' \le t + n{\times}m \quad \Leftarrow \quad i{:=}\, 0.\; i{\le}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m$

$\uparrow$

$i{\le}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m \quad \Leftarrow \quad$ **if** $i{=}n$ **then** $j{:=}\, m$ **else** $i{<}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m$ **fi**

$i{<}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m \quad \Leftarrow \quad j{:=}\, 0.\; i{<}n \wedge j{\le}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j$

$i{<}n \wedge j{\le}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j \quad \Leftarrow$

$t{:=}\, t{+}1.$

if $j{=}m$ **then** $i{:=}\, i{+}1.\; i{\le}n \;\Rightarrow\; t' \le t + (n{-}i){\times}m$

else $i{<}n \wedge j{<}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j$ **fi**

$i{<}n \wedge j{<}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j \quad \Leftarrow$

if $A\ i\ j = x$ **then** ok **else** $j{:=}\, j{+}1.\; i{<}n \wedge j{\le}m \;\Rightarrow\; t' \le t + (n{-}i){\times}m - j$ **fi**

Two-Dimensional Search

$t' \le t + n{\times}m \quad \Leftarrow \quad i:= 0.\ \ i{\le}n \ \Rightarrow\ t' \le t + (n{-}i){\times}m$

$i{\le}n \ \Rightarrow\ t' \le t + (n{-}i){\times}m \quad \Leftarrow \quad$ **if** $i{=}n$ **then** $j:= m$ **else** $i{<}n \ \Rightarrow\ t' \le t + (n{-}i){\times}m$ **fi**

$i{<}n \ \Rightarrow\ t' \le t + (n{-}i){\times}m \quad \Leftarrow \quad j:= 0.\ \ i{<}n \wedge j{\le}m \ \Rightarrow\ t' \le t + (n{-}i){\times}m - j$

$i{<}n \wedge j{\le}m \ \Rightarrow\ t' \le t + (n{-}i){\times}m - j \quad \Leftarrow$

$t:= t{+}1.$

if $j{=}m$ **then** $i:= i{+}1.\ \ i{\le}n \ \Rightarrow\ t' \le t + (n{-}i){\times}m$

else $i{<}n \wedge j{<}m \ \Rightarrow\ t' \le t + (n{-}i){\times}m - j$ **fi**

$i{<}n \wedge j{<}m \ \Rightarrow\ t' \le t + (n{-}i){\times}m - j \quad \Leftarrow$

if $A\ i\ j = x$ **then** ok **else** $j:= j{+}1.\ \ i{<}n \wedge j{\le}m \ \Rightarrow\ t' \le t + (n{-}i){\times}m - j$ **fi**

Two-Dimensional Search

$t' \le t + n{\times}m \quad \Leftarrow \quad i{:=}\,0.\ \ i{\le}n \ \Rightarrow\ t' \le t + (n{-}i){\times}m$

$i{\le}n \ \Rightarrow\ t' \le t + (n{-}i){\times}m \quad \Leftarrow \quad$ **if** $i{=}n$ **then** $j{:=}\ m$ **else** $i{<}n \ \Rightarrow\ t' \le t + (n{-}i){\times}m$ **fi**

$i{<}n \ \Rightarrow\ t' \le t + (n{-}i){\times}m \quad \Leftarrow \quad j{:=}\,0.\ \ i{<}n \wedge j{\le}m \ \Rightarrow\ t' \le t + (n{-}i){\times}m - j$

$i{<}n \wedge j{\le}m \ \Rightarrow\ t' \le t + (n{-}i){\times}m - j \quad \Leftarrow$

$t{:=}\ t{+}1.$

if $j{=}m$ **then** $i{:=}\ i{+}1.\ \ i{\le}n \ \Rightarrow\ t' \le t + (n{-}i){\times}m$

else $i{<}n \wedge j{<}m \ \Rightarrow\ t' \le t + (n{-}i){\times}m - j$ **fi**

$i{<}n \wedge j{<}m \ \Rightarrow\ t' \le t + (n{-}i){\times}m - j \quad \Leftarrow$

if $A\ i\ j = x$ **then** ok **else** $j{:=}\ j{+}1.\ \ i{<}n \wedge j{\le}m \ \Rightarrow\ t' \le t + (n{-}i){\times}m - j$ **fi**

Two-Dimensional Search

$P \Leftarrow i{:=}\,0.\ L0$

$L0 \Leftarrow$ **if** $i{=}n$ **then** $j{:=}\ m$ **else** $j{:=}\,0.\ L1$ **fi**

$L1 \Leftarrow$ **if** $j{=}m$ **then** $i{:=}\ i{+}1.\ L0$

else if $A\ i\ j = x$ **then** ok

else $j{:=}\ j{+}1.\ L1$ **fi fi**

Two-Dimensional Search

$P \Leftarrow i{:=}\,0.\ L0$

$L0 \Leftarrow$ **if** $i{=}n$ **then** $j{:=}\ m$ **else** $j{:=}\,0.\ L1$ **fi**

$L1 \Leftarrow$ **if** $j{=}m$ **then** $i{:=}\ i{+}1.\ L0$

else if $A\ i\ j = x$ **then** ok

else $j{:=}\ j{+}1.\ L1$ **fi fi**

in C:

```
P:   i = 0;
L0:  if (i==n) j = m;
     else {     j = 0;
            L1: if (j==m) {i = i+1;  goto L0;}
                else if (A[i][j]==x);
                     else {j = j+1;  goto L1;}}
```

For Loop

for i:= m;..n **do** P **od**

For Loop

for i:= m;..n **do** P **od**

i is a fresh name (a local constant)

For Loop

for i:= m;..n **do** P **od**

i is a fresh name (a local constant)

m and n are integer expressions such that $m \le n$

For Loop

for i:= m;..n **do** P **od**

i is a fresh name (a local constant)

m and n are integer expressions such that $m \le n$

the number of iterations is $n-m$

For Loop

for i:= m;..n **do** P **od**

i is a fresh name (a local constant)

m and n are integer expressions such that $m \le n$

the number of iterations is $n-m$

P is a specification

For Loop

for i:= m;..n **do** P **od**

For Loop

Let $F\ i$ describe the computation from from index i to the end.

for i:= m;..n **do** P **od**

For Loop

Let $F\ i$ describe the computation from from index i to the end.

for i:= m;..n **do** P **od**

$$F\ i \ \Leftarrow\ i\text{: } m\text{,..}n \wedge (P.\ \ F(i+1))$$

For Loop

Let $F\ i$ describe the computation from from index i to the end.

for i:= m;..n **do** P **od**

$F\ i \ \Leftarrow\ i\colon m,..n \land (P.\ \ F(i+1))$

$F\ n \ \Leftarrow\ ok$

For Loop

Let $F\ i$ describe the computation from from index i to the end.

$$F\ m \ \Leftarrow\ \textbf{for}\ i{:=}\ m;..n\ \textbf{do}\ P\ \textbf{od}$$

means

$$F\ i \ \Leftarrow\ i{:}\ m,..n \wedge (P.\ \ F(i{+}1))$$

$$F\ n \ \Leftarrow\ ok$$

For Loop

example: $x' = 2^n$

For Loop

example: $x' = 2^n$

define: $F\ i \ = \ x' = x \times 2^{n-i}$

For Loop

example: $x'=2^n$

define: $F\ i \ = \ x' = x \times 2^{n-i}$

refine: $x'=2^n \ \Leftarrow \ x:= 1.\ F\ 0$

For Loop

example: $x'=2^n$

define: $F\ i\ \ =\ \ x' = x\times 2^{n-i}$

refine: $x'=2^n \ \ \Leftarrow\ \ x{:=}\ 1.\ \ F\ 0$

proof: $x{:=}\ 1.\ \ F\ 0$ expand $F\ 0$

$=$ $x{:=}\ 1.\ \ x' = x\times 2^{n-0}$ Substitution Law and simplify

$=$ $x'=2^n$

For Loop

example: $x'=2^n$

define: $F\ i \ = \ x' = x\times 2^{n-i}$

refine: $x'=2^n \ \Leftarrow \ x:= 1.\ F\ 0$

$F\ 0 \ \Leftarrow$ **for** $i:= 0;..n$ **do** $x:= 2\times x$ **od**

For Loop

example: $x'=2^n$

define: $F\,i \;=\; x' = x\times 2^{n-i}$

refine: $x'=2^n \;\Leftarrow\; x:= 1.\; F\,0$

$F\,0 \;\Leftarrow\;$ **for** $i:= 0;..n$ **do** $x:= 2\times x$ **od**

proof: $F\,i \;\Leftarrow\; i: m,..n \wedge (P.\; F(i+1))$

$F\,n \;\Leftarrow\; ok$

For Loop

example: $x'=2^n$

define: $F\ i \ = \ x' = x \times 2^{n-i}$

refine: $x'=2^n \ \Leftarrow \ x:= 1.\ \ F\ 0$

$F\ 0 \ \Leftarrow$ **for** $i:= 0;..n$ **do** $x:= 2 \times x$ **od**

proof: $F\ i \ \Leftarrow \ i\!: m,..n \land (P.\ \ F(i+1))$

For Loop

example: $x'=2^n$

define: $F\ i \ = \ x' = x \times 2^{n-i}$

refine: $x'=2^n \ \Leftarrow \ x:= 1.\ F\ 0$

$F\ 0 \ \Leftarrow$ **for** $i:= 0;..n$ **do** $x:= 2\times x$ **od**

proof: $F\ i \ \Leftarrow \ i\text{: } m,..n \wedge (P.\ F(i+1))$

$i\text{: } m,..n \wedge (P.\ F(i+1))$ replace P and $F(i+1)$

$=$ $i\text{: } m,..n \wedge (x:= 2\times x.\ x' = x\times 2^{n-(i+1)})$ substitution law

$=$ $i\text{: } m,..n \wedge x' = 2\times x\times 2^{n-(i+1)}$ simplify

$=$ $i\text{: } m,..n \wedge x' = x\times 2^{n-i}$ specialize

$\Rightarrow$ $F\ i$

For Loop

example: $x'=2^n$

define: $F\ i \ = \ x' = x \times 2^{n-i}$

refine: $x'=2^n \ \Leftarrow \ x:= 1.\ F\,0$

$F\,0 \ \Leftarrow$ **for** $i:= 0;..n$ **do** $x:= 2 \times x$ **od**

proof: $F\ i \ \Leftarrow \ i: m,..n \wedge (P.\ F(i+1))$

For Loop

example: $x'=2^n$

define: $F\ i \ = \ x' = x \times 2^{n-i}$

refine: $x'=2^n \ \Leftarrow \ x:= 1.\ F\ 0$

$F\ 0 \ \Leftarrow$ **for** $i:= 0;..n$ **do** $x:= 2 \times x$ **od**

proof: $F\ i \ \Leftarrow \ i: m,..n \wedge (P.\ F(i+1))$

$F\ n \ \Leftarrow \ ok$

For Loop

example: $x'=2^n$

define: $F\ i \ = \ x' = x{\times}2^{n-i}$

refine: $x'=2^n \ \Leftarrow \ x{:=}\ 1.\ F\ 0$

$F\ 0 \ \Leftarrow$ **for** $i{:=}\ 0;..n$ **do** $x{:=}\ 2{\times}x$ **od**

proof: $F\ i \ \Leftarrow \ i{:}\ m,..n \wedge (P.\ \ F(i+1))$

$F\ n \ \Leftarrow \ ok$

$F\ n$

$=\quad x' = x{\times}2^{n-n}$ simplify

$=\quad x'=x$

$=\quad ok$

For Loop

example: $t' = t + \Sigma j\colon m,..n\cdot\ G\ j$

For Loop

example: $t' = t + \Sigma j\colon m,..n\cdot\ G\ j$

define: $F\ i \ = \ t' = t + \Sigma j\colon i,..n\cdot\ G\ j$

For Loop

example: $t' = t + \Sigma j\colon m,..n\cdot\ G\ j$

define: $F\ i \quad = \quad t' = t + \Sigma j\colon i,..n\cdot\ G\ j$

refine: $F\ m \quad \Leftarrow \quad$ **for** $i{:=}\ m;..n$ **do** $t' = t + G\ i$ **od**

For Loop

example: $t' = t + \Sigma j \colon m,..n \cdot G\, j$

define: $F\, i \;=\; t' = t + \Sigma j \colon i,..n \cdot G\, j$

refine: $F\, m \;\Leftarrow\;$ **for** $i := m;..n$ **do** $t' = t + G\, i$ **od**

prove: $F\, i \;\Leftarrow\; i \colon m,..n \wedge (P.\;\; F(i+1))$

$F\, n \;\Leftarrow\; ok$

For Loop

example: $t' = t + \Sigma j\colon m,..n\cdot\ G\,j$

define: $F\,i \;=\; t' = t + \Sigma j\colon i,..n\cdot\ G\,j$

refine: $F\,m \;\Leftarrow\;$ **for** $i{:=}\ m;..n$ **do** $t' = t + G\,i$ **od**

prove: $F\,i \;\Leftarrow\; i\colon m,..n \wedge (P.\ \ F(i{+}1))$

$t' = t + \Sigma j\colon i,..n\cdot\ G\,j \;\Leftarrow\; i\colon m,..n \wedge (t' = t + G\,i.\ \ t' = t + \Sigma j\colon i{+}1,..n\cdot\ G\,j)$

$F\,n \;\Leftarrow\; ok$

For Loop

example: $t' = t + \Sigma j\colon m,..n\cdot G\, j$

define: $F\, i \;=\; t' = t + \Sigma j\colon i,..n\cdot G\, j$

refine: $F\, m \;\Leftarrow\;$ **for** $i{:=}\, m;..n$ **do** $t' = t + G\, i$ **od**

prove: $F\, i \;\Leftarrow\; i\colon m,..n \wedge (P.\;\; F(i{+}1))$

$t' = t + \Sigma j\colon i,..n\cdot G\, j \;\Leftarrow\; i\colon m,..n \wedge (t' = t + G\, i.\;\; t' = t + \Sigma j\colon i{+}1,..n\cdot G\, j)$

$F\, n \;\Leftarrow\; ok$

$t' = t + \Sigma j\colon n,..n\cdot G\, j \;\Leftarrow\; t'{=}t$

For Loop

example: $t' = t + \Sigma j\colon m,..n\cdot\ G\ j$

define: $F\ i\ \ =\ \ t' = t + \Sigma j\colon i,..n\cdot\ G\ j$

refine: $F\ m\ \ \Leftarrow\ \ $ **for** $i{:=}\ m;..n$ **do** $t' = t + G\ i$ **od**

prove: $F\ i\ \ \Leftarrow\ \ i\colon m,..n \land (P.\ \ F(i{+}1))$

$t' = t + \Sigma j\colon i,..n\cdot\ G\ j\ \ \Leftarrow\ \ i\colon m,..n \land (t' = t + G\ i.\ \ t' = t + \Sigma j\colon i{+}1,..n\cdot\ G\ j)$

$F\ n\ \ \Leftarrow\ \ ok$

$t' = t + \Sigma j\colon n,..n\cdot\ G\ j\ \ \Leftarrow\ \ t'{=}t$

If $G\ i = c$ (a constant) then

$t' = t + (n{-}m){\times}c\ \ \Leftarrow\ \ $ **for** $i{:=}\ m;..n$ **do** $t' = t{+}c$ **od**

For Loop

example: add 1 to each item in a list

For Loop

example: add 1 to each item in a list

$\#L'=\#L \;\wedge\; \forall n\colon \square L\cdot\; L'n = L\,n + 1$

For Loop

example: add 1 to each item in a list

$\#L'=\#L \;\land\; \forall n: \square L \cdot L'n = L\,n + 1$

define: $F\,i \;=\; \#L'=\#L$

$\land\; (\forall n: (0,..i)\cdot L'n = L\,n)$

$\land\; (\forall n: i,..\#L\cdot L'n = L\,n + 1)$

For Loop

example: add 1 to each item in a list

$\#L'=\#L \;\wedge\; \forall n: \square L\cdot\; L'n = L\,n + 1$

define: $F\,i \;=\; \#L'=\#L$

$\wedge\; (\forall n: (0,..i)\cdot\; L'n = L\,n)$

$\wedge\; (\forall n: i,..\#L\cdot\; L'n = L\,n + 1)$

refine: $F\,0 \;\Leftarrow\;$ **for** $i{:=}\,0;..\#L$ **do** $L{:=}\, i{\rightarrow}L\,i + 1 \mid L$ **od**

For Loop

example: add 1 to each item in a list

$\#L'=\#L \ \wedge \ \forall n: \square L \cdot L'n = L\,n + 1$

define: $F\,i \ = \ \#L'=\#L$

$\wedge \ (\forall n: (0,..i)\cdot L'n = L\,n)$

$\wedge \ (\forall n: i,..\#L\cdot L'n = L\,n + 1)$

refine: $F\,0 \ \Leftarrow$ **for** $i{:=}\ 0;..\#L$ **do** $L{:=}\ i{\rightarrow}L\,i + 1 \mid L$ **od**

prove: $F\,i \ \Leftarrow \ i: 0,..\#L \wedge (L{:=}\ i{\rightarrow}L\,i + 1 \mid L.\ \ F(i{+}1))$

$F\,(\#L) \ \Leftarrow \ ok$

For Loop

special case: invariant

$$A\,m \Rightarrow A'n \quad \Leftarrow \quad \textbf{for}\ i{:=}\ m;..n\ \textbf{do}\ i{:}\ m,..n \wedge A\,i\ \Rightarrow\ A'(i{+}1)\ \textbf{od}$$

For Loop

special case: invariant

$$A\,m \Rightarrow A'n \;\Leftarrow\; \textbf{for } i{:=}\, m;..n \textbf{ do } i{:}\, m,..n \wedge A\,i \;\Rightarrow\; A'(i{+}1) \textbf{ od}$$

means

$$A\,i \Rightarrow A'n \;\Leftarrow\; i{:}\, m,..n \wedge (i{:}\, m,..n \;\wedge\; A\,i \Rightarrow A'(i{+}1).\;\; A(i{+}1) \Rightarrow A'n)$$ ✔

$$A\,n \Rightarrow A'n \;\Leftarrow\; ok$$ ✔

For Loop

special case: invariant

$$A\,m \Rightarrow A'n \quad \Leftarrow \quad \textbf{for } i{:=}\ m;..n \textbf{ do } i{:}\ m,..n \land A\,i \ \Rightarrow\ A'(i+1) \textbf{ od}$$

example: $x'=2^n$

For Loop

special case: invariant

$$A\,m \Rightarrow A'n \;\Leftarrow\; \textbf{for}\ i{:=}\ m;..n\ \textbf{do}\ i{:}\ m,..n \wedge A\,i \;\Rightarrow\; A'(i+1)\ \textbf{od}$$

example: $x'=2^n$

invariant: $A\,i \;=\; x=2^i$

For Loop

special case: invariant

$$A\,m \Rightarrow A'n \;\; \Leftarrow \;\; \textbf{for } i{:=}\, m;..n \textbf{ do } i{:}\; m,..n \land A\,i \;\Rightarrow\; A'(i{+}1) \textbf{ od}$$

example: $x'{=}2^n$

invariant: $A\,i \;\; = \;\; x{=}2^i$

refine: $x'{=}2^n \;\; \Leftarrow \;\; x{:=}\,1.\;\; A\,0 \Rightarrow A'n$

For Loop

special case: invariant

$$A\,m \Rightarrow A'n \quad \Leftarrow \quad \textbf{for } i := m;..n \textbf{ do } i{:}\ m,..n \wedge A\,i \ \Rightarrow\ A'(i+1) \textbf{ od}$$

example: $x'=2^n$

invariant: $A\,i \ = \ x=2^i$

refine: $x'=2^n \ \Leftarrow \ x:= 1.\ A\,0 \Rightarrow A'n$

$$A\,0 \Rightarrow A'n \quad \Leftarrow \quad \textbf{for } i := 0;..n \textbf{ do } i{:}\ 0,..n \wedge A\,i \ \Rightarrow\ A'(i+1) \textbf{ od}$$

For Loop

special case: invariant

$$A\,m \Rightarrow A'n \quad \Leftarrow \quad \textbf{for } i{:=}\, m;..n \textbf{ do } i{:}\ m,..n \wedge A\,i \ \Rightarrow\ A'(i+1) \textbf{ od}$$

example: $x'=2^n$

invariant: $A\,i \ = \ x=2^i$

refine: $x'=2^n \ \Leftarrow\ x{:=}\ 1.\ A\,0 \Rightarrow A'n$

$$A\,0 \Rightarrow A'n \quad \Leftarrow \quad \textbf{for } i{:=}\, 0;..n \textbf{ do } i{:}\ 0,..n \wedge A\,i \ \Rightarrow\ A'(i+1) \textbf{ od}$$

$$i{:}\ 0,..n \ \wedge\ A\,i \Rightarrow A'(i+1) \quad \Leftarrow \quad x{:=}\ 2{\times}x$$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.

$$[\ 4 ; -2 ; -8 ; 7 ; 3 ; 0 ; -1\ ]$$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.

$$[\ 4\ ;\ \underline{-2\ ;\ -8}\ ;\ 7\ ;\ 3\ ;\ 0\ ;\ -1\ ]$$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.

$$[\ 4\ ;\ -2\ ;\ -8\ ;\ 7\ ;\ 3\ ;\ 0\ ;\ -1\ ]$$

$s' \ = \ \Downarrow i\colon 0,..\#L\cdot \Downarrow j\colon 1,..\#L+1\cdot \Sigma\, L\, [i;..j]$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.

$$k$$

$$[\ 4\ ;\ -2\ ;\ -8\ ;\ 7\ ;\ 3\ ;\ 0\ ;\ -1\ ]$$

$$s' \;=\; \Downarrow i{:}\ 0,..\#L\cdot\ \Downarrow j{:}\ i{+}1,..\#L{+}1\cdot\ \Sigma\, L\,[i;..j]$$

$$A\,k \;=\; \quad s \;=\; (\Downarrow i{:}\ 0,..k\cdot\ \Downarrow j{:}\ i{+}1,..k{+}1\cdot\ \Sigma\, L\,[i;..j])$$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.

$$\overset{k}{\downarrow}$$

$$[\ 4\ ;\ -2\ ;\ -8\ ;\ 7\ ;\ 3\ ;\ 0\ ;\ -1\ ]$$

$$s' \ = \ \Downarrow i\colon 0,..\#L\cdot \Downarrow j\colon i{+}1,..\#L{+}1\cdot \Sigma L\,[i;..j] \quad \Leftarrow \quad s{:=}\,\infty.\ \ A\,0 \Rightarrow A'(\#L)$$

$$A\,k \quad = \qquad s \ = \ (\Downarrow i\colon 0,..k\cdot \Downarrow j\colon i{+}1,..k{+}1\cdot \Sigma L\,[i;..j])$$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.

k

$$[\ 4\ ;\ –2\ ;\ –8\ ;\ 7\ ;\ 3\ ;\ 0\ ;\ –1\ ]$$

$$s' \ = \ \Downarrow i\colon 0,..\#L\cdot \Downarrow j\colon i{+}1,..\#L{+}1\cdot \Sigma\, L\,[i;..j] \quad \Leftarrow \quad s{:=}\,\infty.\ \ A\,0 \Rightarrow A'(\#L)$$

$$A\,0 \Rightarrow A'(\#L) \quad \Leftarrow \quad \textbf{for}\ k{:=}\,0;..\#L\ \textbf{do}\ k\colon 0,..\#L \wedge A\,k \ \Rightarrow\ A'(k{+}1)\ \textbf{od}$$

$$A\,k \ = \quad s \ = \ (\Downarrow i\colon 0,..k\cdot \Downarrow j\colon i{+}1,..k{+}1\cdot \Sigma\, L\,[i;..j])$$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.

$$\overset{k}{\downarrow}$$

$$[\ 4\ ;\ -2\ ;\ -8\ ;\ 7\ ;\ 3\ ;\ 0\ ;\ -1\ ]$$

$s' \ = \ \Downarrow i\colon 0,..\#L\cdot \Downarrow j\colon i{+}1,..\#L{+}1\cdot \Sigma\, L\,[i;..j] \quad \Longleftarrow \quad s{:=}\,\infty.\ \ A\,0 \Rightarrow A'(\#L)$

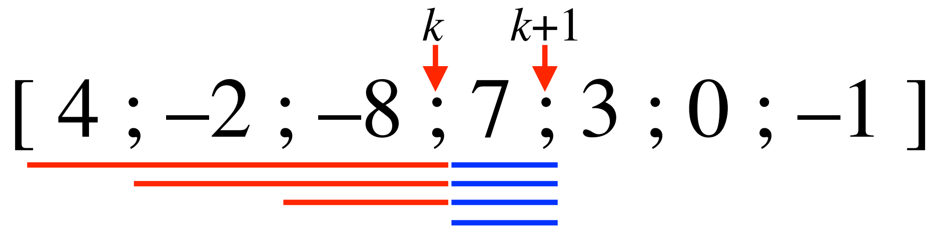
$A\,0 \Rightarrow A'(\#L) \quad \Longleftarrow \quad$ **for** $k{:=}\,0;..\#L$ **do** $k\colon 0,..\#L \wedge A\,k \ \Rightarrow\ A'(k{+}1)$ **od**

$k\colon 0,..\#L \wedge A\,k \ \Rightarrow\ A'(k{+}1) \ \Longleftarrow$

$A\,k \ \ = \qquad s \ = \ (\Downarrow i\colon 0,..k\cdot \Downarrow j\colon i{+}1,..k{+}1\cdot \Sigma\, L\,[i;..j])$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.



$s' \;=\; \Downarrow i{:}\ 0,..\#L\cdot\ \Downarrow j{:}\ i{+}1,..\#L{+}1\cdot\ \Sigma\, L\, [i;..j] \quad\Leftarrow\quad s{:=}\ \infty.\ \ A\,0 \Rightarrow A'(\#L)$

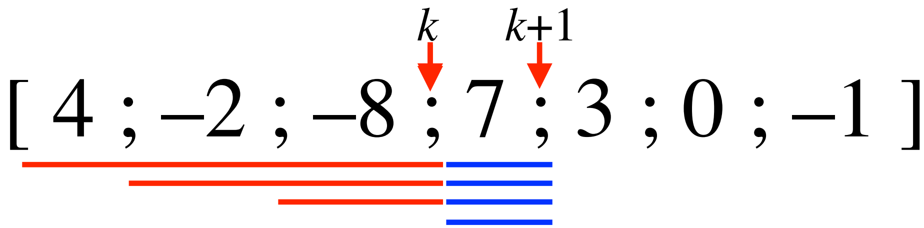
$A\,0 \Rightarrow A'(\#L) \quad\Leftarrow\quad$ **for** $k{:=}\ 0;..\#L$ **do** $k{:}\ 0,..\#L \wedge A\,k \;\Rightarrow\; A'(k{+}1)$ **od**

$k{:}\ 0,..\#L \wedge A\,k \;\Rightarrow\; A'(k{+}1) \quad\Leftarrow$

$A\,k \;=\; \quad s \;=\; (\Downarrow i{:}\ 0,..k\cdot\ \Downarrow j{:}\ i{+}1,..k{+}1\cdot\ \Sigma\, L\, [i;..j])$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.



$s' \ = \ \Downarrow i\colon 0,..\#L\cdot \Downarrow j\colon i{+}1,..\#L{+}1\cdot \Sigma\, L\, [i;..j] \ \ \Longleftarrow \ \ s{:=}\,\infty.\ \ A\,0 \Rightarrow A'(\#L)$

$A\,0 \Rightarrow A'(\#L) \ \ \Longleftarrow \ \ \textbf{for}\ k{:=}\,0;..\#L\ \textbf{do}\ k\colon 0,..\#L \wedge A\,k \ \Rightarrow \ A'(k{+}1)\ \textbf{od}$

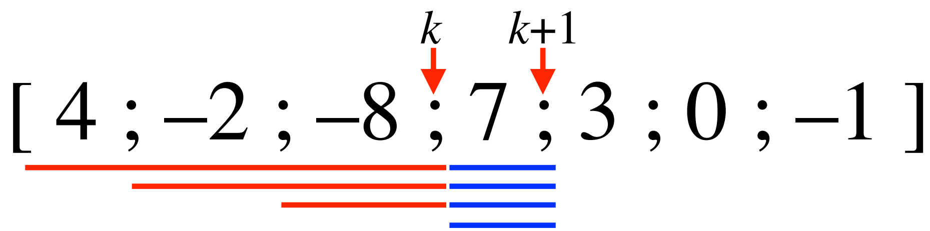
$k\colon 0,..\#L \wedge A\,k \ \Rightarrow \ A'(k{+}1) \ \Longleftarrow$

$$A\,k \ = \quad s \ = \ (\Downarrow i\colon 0,..k\cdot \Downarrow j\colon i{+}1,..k{+}1\cdot \Sigma\, L\, [i;..j])$$

$$\wedge \quad c \ = \ (\Downarrow i\colon 0,..k\cdot \Sigma\, L\, [i;..k])$$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.



$s' \;=\; \Downarrow i\!: 0,..\#L\cdot \Downarrow j\!: i{+}1,..\#L{+}1\cdot \Sigma\, L\, [i;..j] \quad \Leftarrow \quad s{:=}\,\infty.\;\; c{:=}\,\infty.\;\; A\,0 \Rightarrow A'(\#L)$

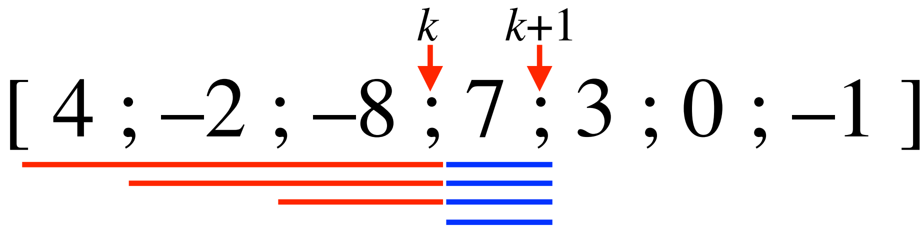
$A\,0 \Rightarrow A'(\#L) \quad \Leftarrow \quad$ **for** $k{:=}\,0;..\#L$ **do** $k\!: 0,..\#L \wedge A\,k \;\Rightarrow\; A'(k{+}1)$ **od**

$k\!: 0,..\#L \wedge A\,k \;\Rightarrow\; A'(k{+}1) \;\; \Leftarrow$

$$\begin{aligned} A\,k \;=\;\; & s \;=\; (\Downarrow i\!: 0,..k\cdot \Downarrow j\!: i{+}1,..k{+}1\cdot \Sigma\, L\,[i;..j]) \\ \wedge\;\; & c \;=\; (\Downarrow i\!: 0,..k\cdot \Sigma\, L\,[i;..k]) \end{aligned}$$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.



$$s' \;=\; \Downarrow i{:}\ 0,..\#L\cdot\ \Downarrow j{:}\ i{+}1,..\#L{+}1\cdot\ \Sigma\, L\,[i;..j] \quad\Leftarrow\quad s{:=}\,\infty.\ \ c{:=}\,\infty.\ \ A\,0 \Rightarrow A'(\#L)$$

$$A\,0 \Rightarrow A'(\#L) \quad\Leftarrow\quad \textbf{for}\ k{:=}\,0;..\#L\ \textbf{do}\ k{:}\ 0,..\#L \wedge A\,k \;\Rightarrow\; A'(k{+}1)\ \textbf{od}$$

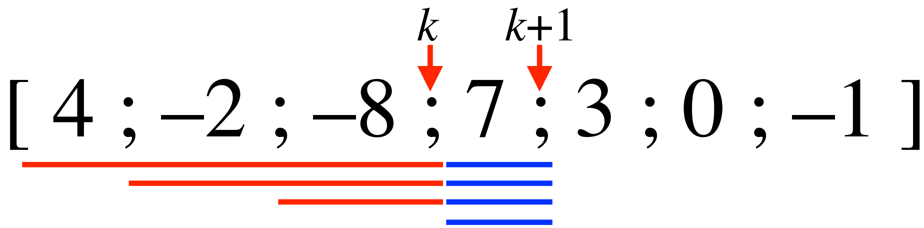
$$k{:}\ 0,..\#L \wedge A\,k \;\Rightarrow\; A'(k{+}1) \quad\Leftarrow\quad c{:=}\,(c + L\,k){\downarrow}(L\,k)$$

$$A\,k \;=\; \quad s \;=\; (\Downarrow i{:}\ 0,..k\cdot\ \Downarrow j{:}\ i{+}1,..k{+}1\cdot\ \Sigma\, L\,[i;..j])$$

$$\wedge \quad c \;=\; (\Downarrow i{:}\ 0,..k\cdot\ \Sigma\, L\,[i;..k])$$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.



$$s' \;=\; \Downarrow i\colon 0,..\#L\cdot \Downarrow j\colon i{+}1,..\#L{+}1\cdot \Sigma L\,[i;..j] \quad \Leftarrow \quad s{:=}\,\infty.\;\; c{:=}\,\infty.\;\; A\,0 \Rightarrow A'(\#L)$$

$$A\,0 \Rightarrow A'(\#L) \quad \Leftarrow \quad \textbf{for } k{:=}\,0;..\#L \textbf{ do } k\colon 0,..\#L \wedge A\,k \;\Rightarrow\; A'(k{+}1) \textbf{ od}$$

$$k\colon 0,..\#L \wedge A\,k \;\Rightarrow\; A'(k{+}1) \quad \Leftarrow \quad c{:=}\,c{\downarrow}0 + L\,k$$

$$A\,k \;=\; \quad s \;=\; (\Downarrow i\colon 0,..k\cdot \Downarrow j\colon i{+}1,..k{+}1\cdot \Sigma L\,[i;..j])$$

$$\wedge \quad c \;=\; (\Downarrow i\colon 0,..k\cdot \Sigma L\,[i;..k])$$

Minimum Sum Segment

Given a list L of integers, possibly including negatives, write a program to find the minimum sum of any nonempty segment.

$$[\,4\,;\,-2\,;\,-8\,\overset{k}{;}\,7\,\overset{k+1}{;}\,3\,;\,0\,;\,-1\,]$$

$$s' \;=\; \Downarrow i{:}\ 0,..\#L\cdot\ \Downarrow j{:}\ i+1,..\#L+1\cdot\ \Sigma\, L\,[i;..j] \quad\Leftarrow\quad s{:=}\ \infty.\ \ c{:=}\ \infty.\ \ A\,0 \Rightarrow A'(\#L)$$

$$A\,0 \Rightarrow A'(\#L) \quad\Leftarrow\quad \textbf{for}\ k{:=}\ 0;..\#L\ \textbf{do}\ k{:}\ 0,..\#L \wedge A\,k \;\Rightarrow\; A'(k+1)\ \textbf{od}$$

$$k{:}\ 0,..\#L \wedge A\,k \;\Rightarrow\; A'(k+1) \quad\Leftarrow\quad c{:=}\ c\downarrow 0 + L\,k.\ \ s{:=}\ s\downarrow c$$

$$A\,k \;=\; \qquad s \;=\; (\Downarrow i{:}\ 0,..k\cdot\ \Downarrow j{:}\ i+1,..k+1\cdot\ \Sigma\, L\,[i;..j])$$

$$\wedge \qquad c \;=\; (\Downarrow i{:}\ 0,..k\cdot\ \Sigma\, L\,[i;..k])$$