Sequential to Concurrent Transformation

Sequential to Concurrent Transformation

$x := y.\ x := x+1.\ z := y$

Sequential to Concurrent Transformation

$x := y.\ \ x := x+1.\ \ z := y$

$\text{start} \longrightarrow x := y \longrightarrow x := x+1 \longrightarrow z := y \longrightarrow \text{finish}$

Sequential to Concurrent Transformation

$$x := y.\ \ x := x+1.\ \ z := y$$

$$= \qquad x := y.\ \ (x := x+1 \ \| \ z := y)$$

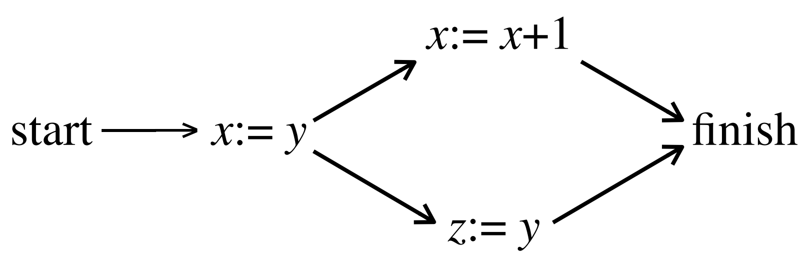
$$\text{start} \longrightarrow x := y \longrightarrow x := x+1 \longrightarrow z := y \longrightarrow \text{finish}$$

Sequential to Concurrent Transformation

$$x := y.\ x := x+1.\ z := y$$

$$=\quad x := y.\ (x := x+1 \parallel z := y)$$

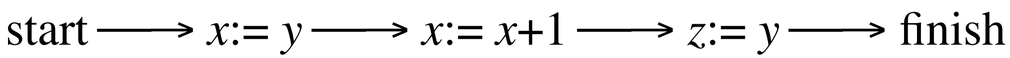
$$\text{start} \longrightarrow x := y \longrightarrow x := x+1 \longrightarrow z := y \longrightarrow \text{finish}$$

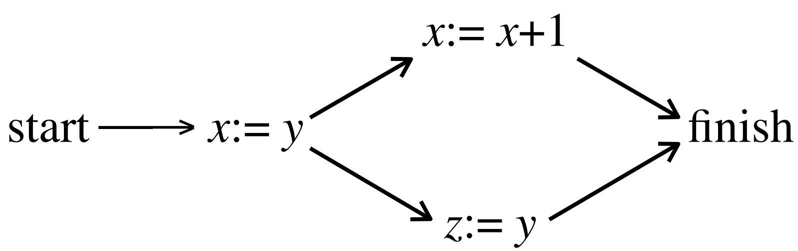


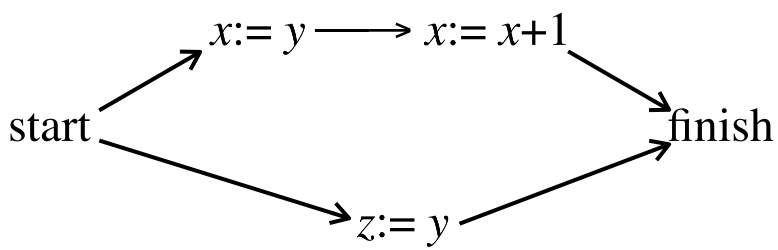
Sequential to Concurrent Transformation

$$x := y.\ \ x := x+1.\ \ z := y$$

$$= \quad x := y.\ \ (x := x+1 \ \| \ z := y)$$





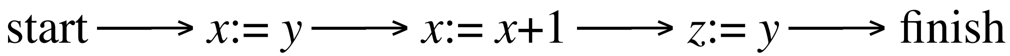


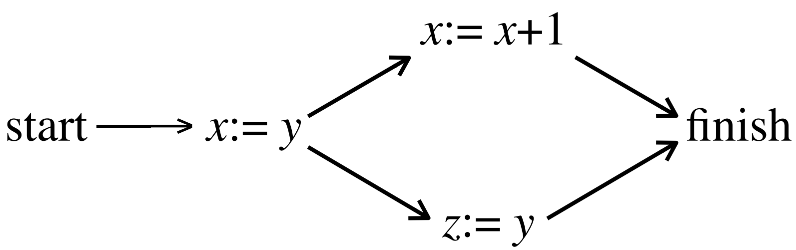
Sequential to Concurrent Transformation

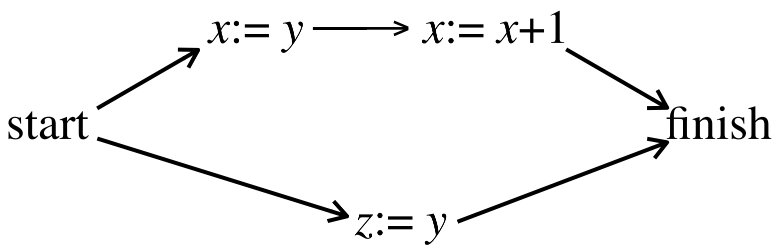
$$x{:=}\,y.\ \ x{:=}\,x{+}1.\ \ z{:=}\,y$$

$$=\quad x{:=}\,y.\ \ (x{:=}\,x{+}1\ \ \|\ \ z{:=}\,y)$$

$$=\quad (x{:=}\,y.\ \ x{:=}\,x{+}1)\ \ \|\ \ z{:=}\,y$$







Sequential to Concurrent Transformation

rules

Sequential to Concurrent Transformation

rules

Whenever two programs occur in sequence, and neither assigns to a variable appearing in the other, they can be placed in parallel.

example $x{:=}\, z.\;\; y{:=}\, z$ becomes $x{:=}\, z \;||\; y{:=}\, z$

Sequential to Concurrent Transformation

rules

Whenever two programs occur in sequence, and neither assigns to a variable appearing in the other, they can be placed in parallel.

example $x{:=}\,z.\ \ y{:=}\,z$ becomes $x{:=}\,z\ \|\ y{:=}\,z$

Whenever two programs occur in sequence, and neither assigns to a variable assigned in the other, and no variable assigned in the first appears in the second, they can be placed in parallel.

example $x{:=}\,y.\ \ y{:=}\,z$ becomes $x{:=}\,y\ \|\ y{:=}\,z$

Sequential to Concurrent Transformation

rules

Whenever two programs occur in sequence, and neither assigns to a variable appearing in the other, they can be placed in parallel.

example $x{:=}\, z.\;\; y{:=}\, z$ becomes $x{:=}\, z \;\|\; y{:=}\, z$

Whenever two programs occur in sequence, and neither assigns to a variable assigned in the other, and no variable assigned in the first appears in the second, they can be placed in parallel; a copy must be made of the initial value of any variable appearing in the first and assigned in the second.

example $x{:=}\, y.\;\; y{:=}\, z$ becomes $c{:=}\, y.\;\; (x{:=}\, c \;\|\; y{:=}\, z)$

Buffer

$$produce \;=\; \cdots\cdots b := e \cdots\cdots$$

$$consume \;=\; \cdots\cdots x := b \cdots\cdots$$

Buffer

$$produce = \cdots\cdots\cdots b := e \cdots\cdots\cdots$$

$$consume = \cdots\cdots\cdots x := b \cdots\cdots\cdots$$

$$control = produce.\ consume.\ control$$

Buffer

$$produce \;=\; \cdots\cdots b := e \cdots\cdots$$

$$consume \;=\; \cdots\cdots x := b \cdots\cdots$$

$$control \;=\; produce.\; consume.\; control$$

$$P \longrightarrow C \longrightarrow P \longrightarrow C \longrightarrow P \longrightarrow C \longrightarrow P \longrightarrow C \longrightarrow$$

Buffer

$$produce = \cdots\cdots\cdots b := e \cdots\cdots\cdots$$

$$consume = \cdots\cdots\cdots x := b \cdots\cdots\cdots$$

$$control = produce \,.\, consume \,.\, control$$

Buffer

$$produce = \cdots\cdots\cdots b := e \cdots\cdots\cdots$$

$$consume = \cdots\cdots\cdots x := b \cdots\cdots\cdots$$

$$control = produce.\ newcontrol$$

$$newcontrol = consume.\ produce.\ newcontrol$$

Buffer

$$produce = \cdots\cdots b := e \cdots\cdots$$

$$consume = \cdots\cdots x := b \cdots\cdots$$

$$control = produce.\ newcontrol$$

$$newcontrol = (consume \parallel produce).\ newcontrol$$

Buffer

$$produce = \cdots\cdots b := e \cdots\cdots$$

$$consume = \cdots\cdots x := c \cdots\cdots$$

$$control = produce.\ newcontrol$$

$$newcontrol = c := b.\ (consume \parallel produce).\ newcontrol$$

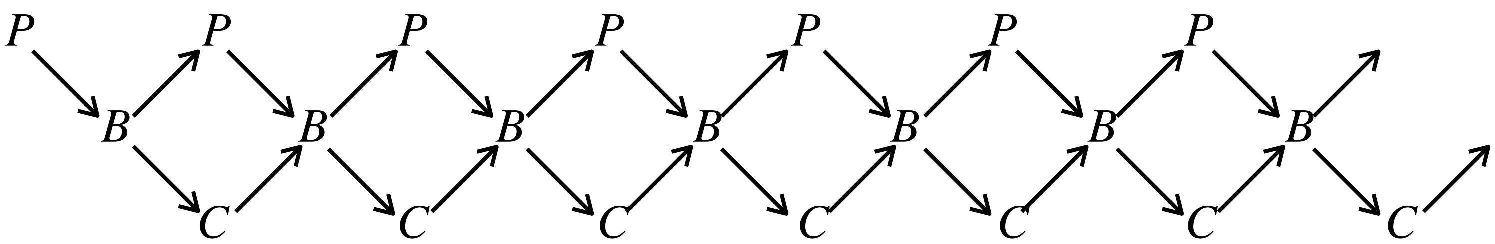
Buffer

$$\mathit{produce} = \cdots\cdots\cdots b := e \cdots\cdots\cdots$$

$$\mathit{consume} = \cdots\cdots\cdots x := c \cdots\cdots\cdots$$

$$\mathit{control} = \mathit{produce}.\ \mathit{newcontrol}$$

$$\mathit{newcontrol} = c := b.\ (\mathit{consume} \parallel \mathit{produce}).\ \mathit{newcontrol}$$



Buffer

$$produce = \cdots\cdots\cdots b\,w := e.\ \ w := w+1 \cdots\cdots\cdots$$

$$consume = \cdots\cdots\cdots x := b\,r.\ \ r := r+1 \cdots\cdots\cdots$$

$$control = w := 0.\ \ r := 0.\ \ newcontrol$$

$$newcontrol = produce.\ \ consume.\ \ newcontrol$$

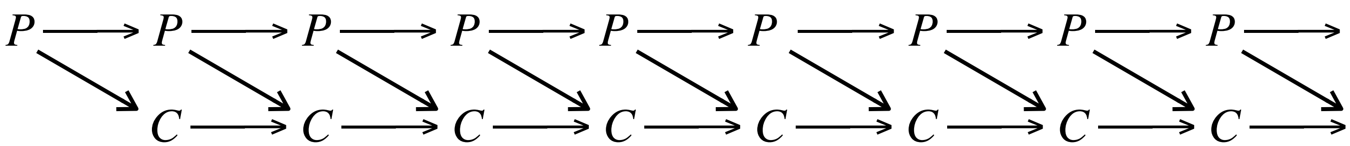
Buffer

$$produce \;=\; \cdots\cdots\cdots b\ w := e.\ \ w := w+1\cdots\cdots\cdots$$

$$consume \;=\; \cdots\cdots\cdots x := b\ r.\ \ r := r+1\cdots\cdots\cdots$$

$$control \;=\; w := 0.\ \ r := 0.\ \ newcontrol$$

$$newcontrol \;=\; produce.\ \ consume.\ \ newcontrol$$



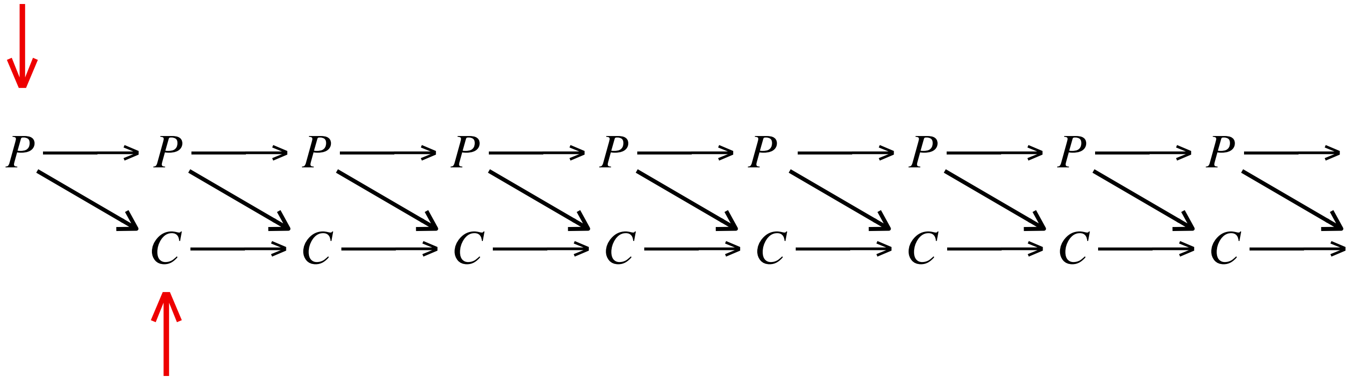
Buffer

$$produce \;=\; \cdots\cdots b\, w := e.\;\; w := w+1 \cdots\cdots$$

$$consume \;=\; \cdots\cdots x := b\, r.\;\; r := r+1 \cdots\cdots$$

$$control \;=\; w := 0.\;\; r := 0.\;\; newcontrol$$

$$newcontrol \;=\; produce.\;\; consume.\;\; newcontrol$$



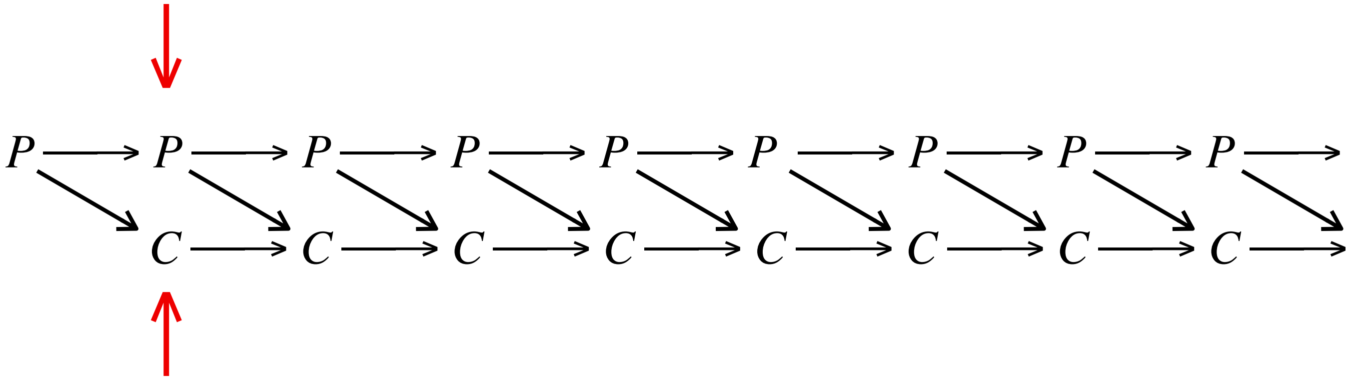
Buffer

$$produce \;=\; \cdots\cdots\, b\,w := e.\;\; w := w+1 \cdots\cdots$$

$$consume \;=\; \cdots\cdots\, x := b\,r.\;\; r := r+1 \cdots\cdots$$

$$control \;=\; w := 0.\;\; r := 0.\;\; newcontrol$$

$$newcontrol \;=\; produce.\;\; consume.\;\; newcontrol$$



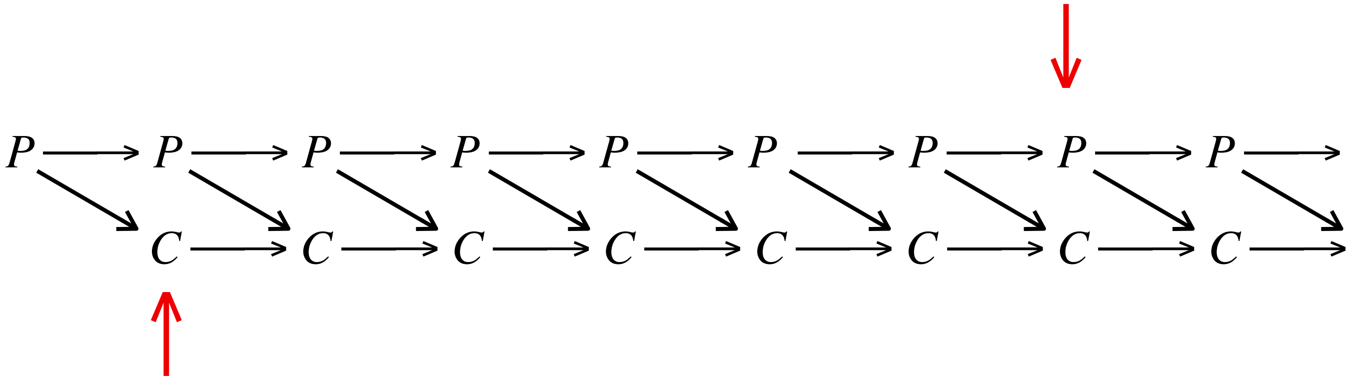
Buffer

$$produce \;=\; \cdots\cdots\, b\, w := e.\;\; w := w+1 \cdots\cdots$$

$$consume \;=\; \cdots\cdots\, x := b\, r.\;\; r := r+1 \cdots\cdots$$

$$control \;=\; w := 0.\;\; r := 0.\;\; newcontrol$$

$$newcontrol \;=\; produce.\;\; consume.\;\; newcontrol$$



Buffer

$produce \;=\; \cdots\cdots\cdots b\; w := e.\;\; w := mod\;(w+1)\; n\cdots\cdots\cdots$

$consume \;=\; \cdots\cdots\cdots x := b\; r.\;\; r := mod\;(r+1)\; n\cdots\cdots\cdots$

$control \;=\; w := 0.\;\; r := 0.\;\; newcontrol$

$newcontrol \;=\; produce.\;\; consume.\;\; newcontrol$

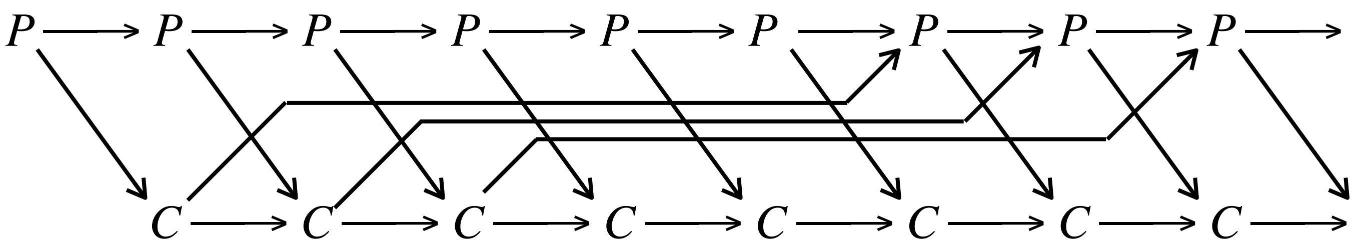
Buffer

$$produce = \cdots\cdots\cdots b\; w := e.\; w := mod\,(w+1)\; n\cdots\cdots\cdots$$

$$consume = \cdots\cdots\cdots x := b\; r.\; r := mod\,(r+1)\; n\cdots\cdots\cdots$$

$$control = w := 0.\; r := 0.\; newcontrol$$

$$newcontrol = produce.\; consume.\; newcontrol$$



Insertion Sort

define

$$sort \;=\; \langle n\cdot\ \forall i, j: 0,..n\cdot\ i{\le}j \Rightarrow L\, i \le L\, j\rangle$$

Insertion Sort

define

$$sort \;=\; \langle n\cdot\ \forall i, j: 0,..n\cdot\ i{\le}j \Rightarrow L\,i \le L\,j\rangle$$

$$sort'\,(\#L)$$

Insertion Sort

define

$$sort \;=\; \langle n\cdot\ \forall i, j{:}\ 0,..n\cdot\ i{\le}j \Rightarrow L\,i \le L\,j\rangle$$

$$sort'\ (\#L) \quad\Leftarrow\quad sort\ 0 \Rightarrow sort'\ (\#L)$$

Insertion Sort

define

$$sort \;=\; \langle n\cdot\ \forall i, j\colon 0,..n\cdot\ i{\le}j \Rightarrow L\,i \le L\,j\rangle$$

$$sort'\,(\#L) \quad\Leftarrow\quad sort\,0 \Rightarrow sort'\,(\#L)$$

$$sort\,0 \Rightarrow sort'\,(\#L) \quad\Leftarrow\quad \mathbf{for}\ n{:=}\ 0;..\#L\ \mathbf{do}\ sort\,n \Rightarrow sort'\,(n{+}1)\ \mathbf{od}$$

Insertion Sort

define

$$sort \;=\; \langle n\cdot\ \forall i, j: 0,..n\cdot\ i\le j \Rightarrow L\,i \le L\,j\rangle$$

$$sort'\,(\#L) \quad\Leftarrow\quad sort\,0 \Rightarrow sort'\,(\#L)$$

$$sort\,0 \Rightarrow sort'\,(\#L) \quad\Leftarrow\quad \mathbf{for}\ n{:=}\ 0;..\#L\ \mathbf{do}\ sort\,n \Rightarrow sort'\,(n{+}1)\ \mathbf{od}$$

$$sort\,n \Rightarrow sort'\,(n{+}1) \quad\Leftarrow$$

Insertion Sort

define

$$sort \;=\; \langle n\cdot\; \forall i, j{:}\; 0,..n\cdot\; i{\le}j \Rightarrow L\,i \le L\,j \rangle$$

$$sort'\,(\#L) \;\;\Leftarrow\;\; sort\,0 \Rightarrow sort'\,(\#L)$$

$$sort\,0 \Rightarrow sort'\,(\#L) \;\;\Leftarrow\;\; \textbf{for}\; n{:=}\,0;..\#L\; \textbf{do}\; sort\,n \Rightarrow sort'\,(n{+}1)\; \textbf{od}$$

$$sort\,n \Rightarrow sort'\,(n{+}1) \;\;\Leftarrow$$

$$\underset{0}{[}\; L\,0 \;\underset{1}{;}\; L\,1 \;\underset{2}{;}\; L\,2 \;\underset{3}{;}\; L\,3 \;\underset{4}{;}\; L\,4 \;\underset{5}{]}$$

Insertion Sort

define

$$sort \;=\; \langle n \cdot\ \forall i, j{:}\ 0,..n\cdot\ i{\le}j \Rightarrow L\,i \le L\,j\rangle$$

$$sort'\,(\#L) \quad\Leftarrow\quad sort\,0 \Rightarrow sort'\,(\#L)$$

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

$$sort\,n \Rightarrow sort'\,(n{+}1) \quad\Leftarrow$$

if $n{=}0$ **then**

$$\underset{0}{[}\ L\,0\ \underset{1}{;}\ L\,1\ \underset{2}{;}\ L\,2\ \underset{3}{;}\ L\,3\ \underset{4}{;}\ L\,4\ \underset{5}{]}$$

Insertion Sort

define

$$sort \;=\; \langle n\cdot\ \forall i, j\colon 0,..n\cdot\ i{\le}j \Rightarrow L\,i \le L\,j\rangle$$

$$sort'\,(\#L) \quad\Leftarrow\quad sort\,0 \Rightarrow sort'\,(\#L)$$

$$sort\,0 \Rightarrow sort'\,(\#L) \quad\Leftarrow\quad \mathbf{for}\ n{:=}\,0;..\#L\ \mathbf{do}\ sort\,n \Rightarrow sort'\,(n{+}1)\ \mathbf{od}$$

$$sort\,n \Rightarrow sort'\,(n{+}1) \quad\Leftarrow$$

if $n{=}0$ **then**

$$\underset{0}{[}\ L\,0\ \underset{1}{;}\ L\,1\ \underset{2}{;}\ L\,2\ \underset{3}{;}\ L\,3\ \underset{4}{;}\ L\,4\ \underset{5}{]}$$

Insertion Sort

define

$$sort \;=\; \langle n \cdot\ \forall i, j\!: 0,..n\cdot\ i{\le}j \Rightarrow L\,i \le L\,j \rangle$$

$$sort'\,(\#L) \quad \Leftarrow \quad sort\,0 \Rightarrow sort'\,(\#L)$$

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

$$sort\,n \Rightarrow sort'\,(n{+}1) \quad \Leftarrow$$

if $n{=}0$ **then**

$$[\ L\,0\ \ ;L\,1\ \ ;L\,2\ \ ;L\,3\ \ ;L\,4\ \ ]$$

0 1 2 3 4 5

Insertion Sort

define

$$sort \;=\; \langle n\cdot\; \forall i, j{:}\ 0,..n\cdot\ i{\le}j \Rightarrow L\,i \le L\,j\rangle$$

$$sort'\,(\#L) \quad\Leftarrow\quad sort\,0 \Rightarrow sort'\,(\#L)$$

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

$$sort\,n \Rightarrow sort'\,(n{+}1) \quad\Leftarrow$$

$$\textbf{if}\ n{=}0\ \textbf{then}\ ok$$

$$[\,L\,0\ \ ;L\,1\ \ ;L\,2\ \ ;L\,3\ \ ;L\,4\ \ ]$$

0 1 2 3 4 5

Insertion Sort

define

$$sort \;=\; \langle n\cdot\; \forall i, j{:}\, 0,..n\cdot\; i{\le}j \Rightarrow L\,i \le L\,j\rangle$$

$$sort'\,(\#L) \quad \Leftarrow \quad sort\,0 \Rightarrow sort'\,(\#L)$$

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

$$sort\,n \Rightarrow sort'\,(n{+}1) \quad \Leftarrow$$

if $n{=}0$ **then** *ok*

else

$$\underset{0}{[}\; L\,0 \;\; \underset{1}{;}\, L\,1 \;\; \underset{2}{;}\, L\,2 \;\; \underset{3}{;}\, L\,3 \;\; \underset{4}{;}\, L\,4 \;\; \underset{5}{]}$$

Insertion Sort

define

$$sort \;=\; \langle n\cdot\ \forall i, j: 0,..n\cdot\ i{\le}j \Rightarrow L\,i \le L\,j\rangle$$

$$sort'\,(\#L) \;\Leftarrow\; sort\,0 \Rightarrow sort'\,(\#L)$$

$$sort\,0 \Rightarrow sort'\,(\#L) \;\Leftarrow\; \textbf{for } n{:=}\,0;..\#L \textbf{ do } sort\,n \Rightarrow sort'\,(n{+}1) \textbf{ od}$$

$$sort\,n \Rightarrow sort'\,(n{+}1) \;\Leftarrow$$

if $n{=}0$ **then** *ok*

else if $L\,(n{-}1) \le L\,n$ **then** *ok*

$$[\,L\,0\ ;\,L\,1\ ;\,L\,2\ ;\,L\,3\ ;\,L\,4\ ]$$

0 1 2 3 4 5

Insertion Sort

define

$$sort \;=\; \langle n\cdot\ \forall i, j: 0,..n\cdot\ i{\le}j \Rightarrow L\, i \le L\, j\rangle$$

$$sort'\ (\#L) \quad \Leftarrow \quad sort\ 0 \Rightarrow sort'\ (\#L)$$

$$sort\ 0 \Rightarrow sort'\ (\#L) \quad \Leftarrow \quad \textbf{for}\ n{:=}\ 0;..\#L\ \textbf{do}\ sort\ n \Rightarrow sort'\ (n{+}1)\ \textbf{od}$$

$$sort\ n \Rightarrow sort'\ (n{+}1) \quad \Leftarrow$$

if $n{=}0$ **then** ok

else if $L\ (n{-}1) \le L\ n$ **then** ok

else

$[\ L\,0\ \ ;L\,1\ \ ;L\,2\ \ ;L\,3\ \ ;L\,4\ \ ]$

0 1 2 3 4 5

Insertion Sort

define

$$sort \;=\; \langle n\cdot\; \forall i, j{:}\; 0,..n\cdot\; i{\le}j \Rightarrow L\,i \le L\,j\rangle$$

$$sort'\,(\#L) \quad \Leftarrow \quad sort\,0 \Rightarrow sort'\,(\#L)$$

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

$sort\,n \Rightarrow sort'\,(n{+}1) \quad \Leftarrow$

if $n{=}0$ **then** ok

else if $L\,(n{-}1) \le L\,n$ **then** ok

else $swap\,(n{-}1)\,n$.

$$\underset{0}{[}\; L\,0 \;\underset{1}{;}\; L\,1 \;\underset{2}{;}\; L\,2 \;\underset{3}{;}\; L\,3 \;\underset{4}{;}\; L\,4 \;\underset{5}{]}$$

Insertion Sort

define

$$sort \;=\; \langle n\cdot\; \forall i, j\!: 0,..n\cdot\; i{\le}j \Rightarrow L\,i \le L\,j\rangle$$

$$swap \;=\; \langle i, j\!: 0,..\#L\cdot\;\; L\,i{:=}\,L\,j \,\|\, L\,j{:=}\,L\,i\rangle$$

$$sort'\,(\#L) \;\;\Leftarrow\;\; sort\,0 \Rightarrow sort'\,(\#L)$$

$$sort\,0 \Rightarrow sort'\,(\#L) \;\;\Leftarrow\;\; \mathbf{for}\; n{:=}\,0;..\#L \;\mathbf{do}\; sort\,n \Rightarrow sort'\,(n{+}1)\;\mathbf{od}$$

$$sort\,n \Rightarrow sort'\,(n{+}1) \;\;\Leftarrow$$

$$\mathbf{if}\; n{=}0 \;\mathbf{then}\; ok$$

$$\mathbf{else\ if}\; L\,(n{-}1) \le L\,n \;\mathbf{then}\; ok$$

$$\mathbf{else}\; swap\,(n{-}1)\,n.$$

$$\underset{0}{[}\, L\,0 \;\underset{1}{;}\, L\,1 \;\underset{2}{;}\, L\,2 \;\underset{3}{;}\, L\,3 \;\underset{4}{;}\, L\,4 \;\underset{5}{]}$$

Insertion Sort

define

$$sort \;=\; \langle n\cdot\;\; \forall i, j{:}\ 0,..n\cdot\ i{\le}j \Rightarrow L\,i \le L\,j\rangle$$

$$swap \;=\; \langle i, j{:}\ 0,..\#L\cdot\;\; L\,i{:=}\ L\,j \parallel L\,j{:=}\ L\,i\rangle$$

$$sort'\ (\#L) \quad \Leftarrow \quad sort\ 0 \Rightarrow sort'\ (\#L)$$

$sort\ 0 \Rightarrow sort'\ (\#L) \quad \Leftarrow \quad$ **for** $n{:=}\ 0;..\#L$ **do** $sort\ n \Rightarrow sort'\ (n+1)$ **od**

$sort\ n \Rightarrow sort'\ (n+1) \quad \Leftarrow$

if $n{=}0$ **then** ok

else if $L\ (n{-}1) \le L\ n$ **then** ok

else $swap\ (n{-}1)\ n.\;\; sort\ (n{-}1) \Rightarrow sort'\ n$ **fi fi**

$$\underset{0}{[}\ L\,0\ \underset{1}{;}\ L\,1\ \underset{2}{;}\ L\,2\ \underset{3}{;}\ L\,3\ \underset{4}{;}\ L\,4\ \underset{5}{]}$$

Insertion Sort

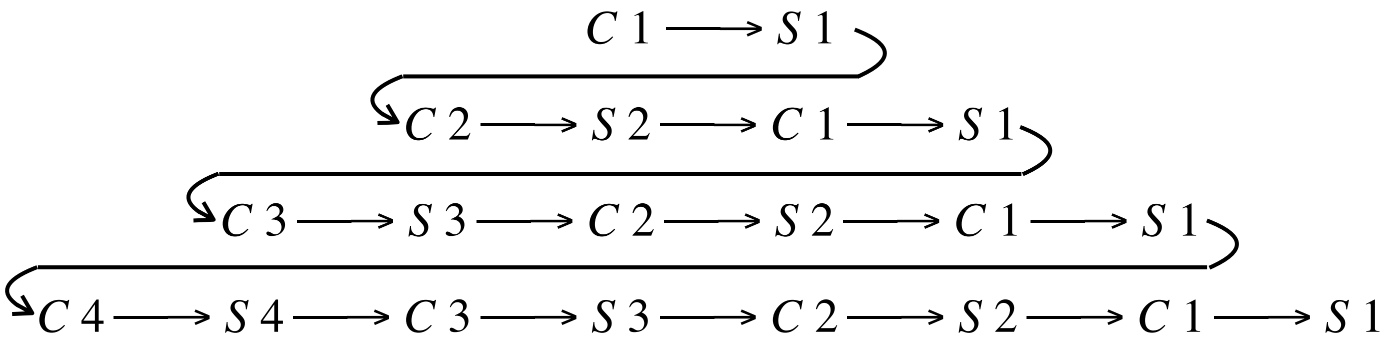
$$C\,1 \longrightarrow S\,1$$

$$C\,2 \longrightarrow S\,2 \longrightarrow C\,1 \longrightarrow S\,1$$

$$C\,3 \longrightarrow S\,3 \longrightarrow C\,2 \longrightarrow S\,2 \longrightarrow C\,1 \longrightarrow S\,1$$

$$C\,4 \longrightarrow S\,4 \longrightarrow C\,3 \longrightarrow S\,3 \longrightarrow C\,2 \longrightarrow S\,2 \longrightarrow C\,1 \longrightarrow S\,1$$

Insertion Sort



If *abs* $(i–j) > 1$ then $S\ i$ and $S\ j$ in parallel

Insertion Sort

$$C\,1 \longrightarrow S\,1$$

$$C\,2 \longrightarrow S\,2 \longrightarrow C\,1 \longrightarrow S\,1$$

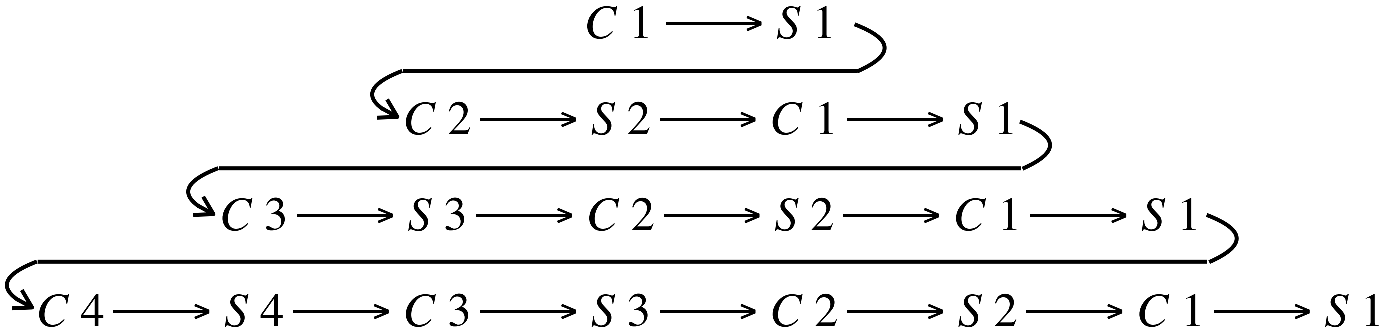
$$C\,3 \longrightarrow S\,3 \longrightarrow C\,2 \longrightarrow S\,2 \longrightarrow C\,1 \longrightarrow S\,1$$

$$C\,4 \longrightarrow S\,4 \longrightarrow C\,3 \longrightarrow S\,3 \longrightarrow C\,2 \longrightarrow S\,2 \longrightarrow C\,1 \longrightarrow S\,1$$

If $abs\,(i-j) > 1$ then $S\,i$ and $S\,j$ in parallel

If $abs\,(i-j) > 1$ then $S\,i$ and $C\,j$ in parallel

Insertion Sort

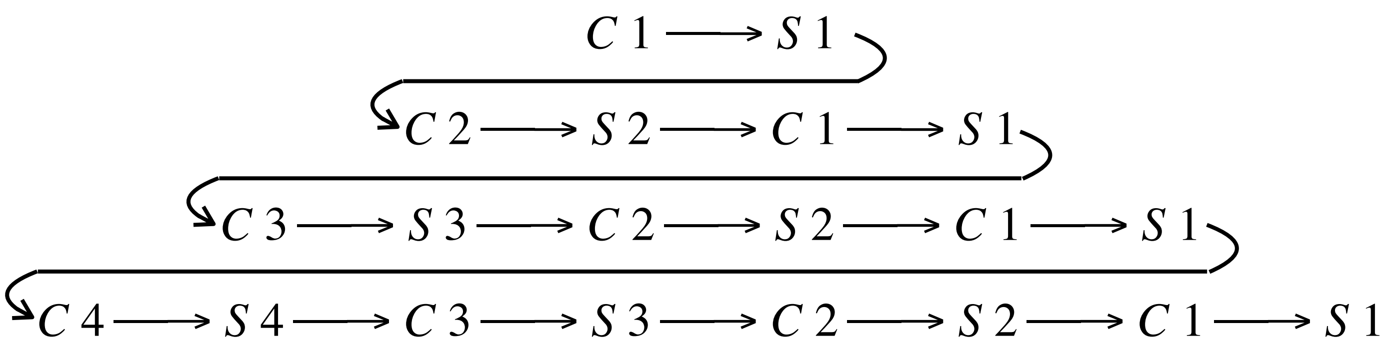


If *abs* $(i–j) > 1$ then $S\,i$ and $S\,j$ in parallel

If *abs* $(i–j) > 1$ then $S\,i$ and $C\,j$ in parallel

$C\,i$ and $C\,j$ in parallel

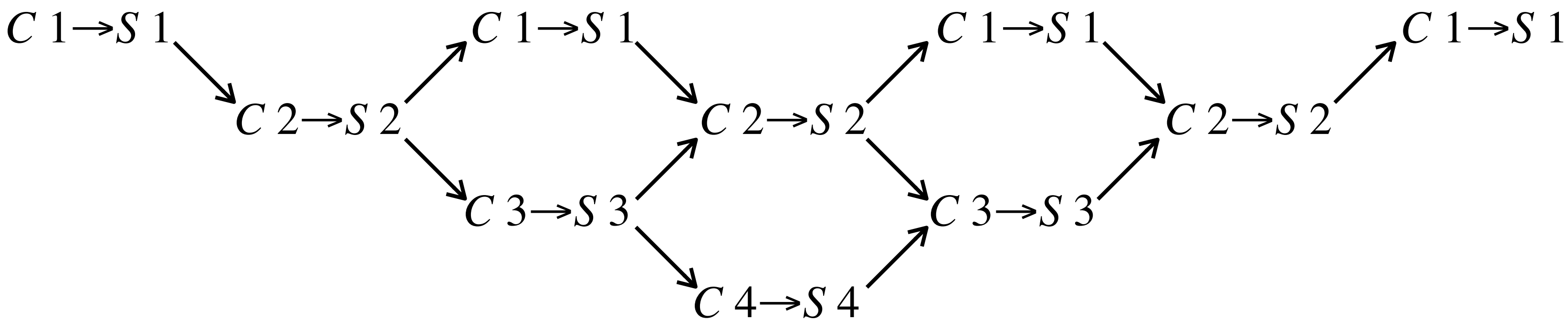
Insertion Sort



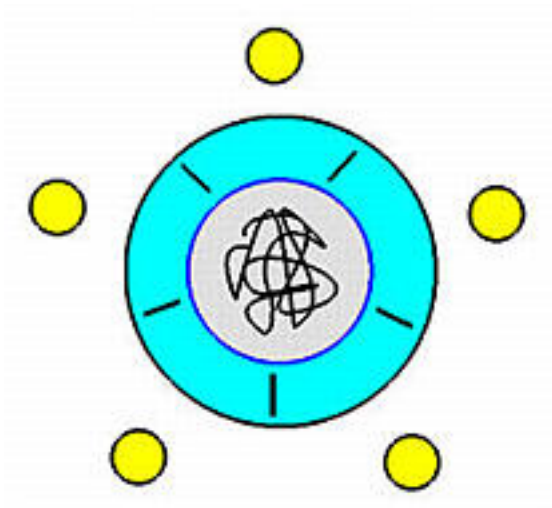
If *abs* (i–j) > 1 then *S i* and *S j* in parallel

If *abs* (i–j) > 1 then *S i* and *C j* in parallel

C i and *C j* in parallel



Dining Philosophers



Dining Philosophers

$$life \;=\; P\,0 \parallel P\,1 \parallel P\,2 \parallel P\,3 \parallel P\,4$$

$$P\,i \;=\; think\,i.\;(up\,i \parallel up(i \oplus 1)).\;eat\,i.\;(down\,i \parallel down(i \oplus 1)).\;P\,i$$

$$up\,i \;=\; chopstick\,i := \top$$

$$down\,i \;=\; chopstick\,i := \bot$$

$$eat\,i \;=\; (\text{uses } chopstick\,i \text{ and } chopstick(i \oplus 1))$$

$$think\,i \;=\; (\text{does not use any } chopstick)$$

Dining Philosophers

$$\mathit{life} \;=\; P\,0 \parallel P\,1 \parallel P\,2 \parallel P\,3 \parallel P\,4 \quad \longleftarrow$$

$$P\,i \;=\; \mathit{think}\,i.\;(\mathit{up}\,i \parallel \mathit{up}(i\oplus 1)).\;\mathit{eat}\,i.\;(\mathit{down}\,i \parallel \mathit{down}(i\oplus 1)).\;P\,i$$

$$\mathit{up}\,i \;=\; \mathit{chopstick}\,i := \top$$

$$\mathit{down}\,i \;=\; \mathit{chopstick}\,i := \bot$$

$$\mathit{eat}\,i \;=\; (\text{uses } \mathit{chopstick}\,i \text{ and } \mathit{chopstick}(i\oplus 1))$$

$$\mathit{think}\,i \;=\; (\text{does not use any } \mathit{chopstick})$$

Dining Philosophers

$$life \;=\; P\,0 \parallel P\,1 \parallel P\,2 \parallel P\,3 \parallel P\,4$$

$$P\,i \;=\; think\,i.\;(up\,i \parallel up(i\oplus 1)).\;eat\,i.\;(down\,i \parallel down(i\oplus 1)).\;P\,i \qquad \longleftarrow$$

$$up\,i \;=\; chopstick\,i := \top$$

$$down\,i \;=\; chopstick\,i := \bot$$

$eat\,i$ = (uses *chopstick* i and *chopstick*$(i\oplus 1)$)

$think\,i$ = (does not use any *chopstick*)

Dining Philosophers

$$life \;=\; P\,0 \parallel P\,1 \parallel P\,2 \parallel P\,3 \parallel P\,4$$

$$P\,i \;=\; think\;i.\;(up\;i \parallel up(i\oplus 1)).\;eat\;i.\;(down\;i \parallel down(i\oplus 1)).\;P\,i$$

$$up\;i \;=\; chopstick\;i := \top \quad \longleftarrow$$

$$down\;i \;=\; chopstick\;i := \bot \quad \longleftarrow$$

$eat\;i$ = (uses *chopstick i* and *chopstick*$(i\oplus 1)$)

$think\;i$ = (does not use any *chopstick*)

Dining Philosophers

$$life = P\,0 \parallel P\,1 \parallel P\,2 \parallel P\,3 \parallel P\,4$$

$$P\,i = think\;i.\;(up\;i \parallel up(i \oplus 1)).\;eat\;i.\;(down\;i \parallel down(i \oplus 1)).\;P\,i$$

$$up\;i = chopstick\;i := \top$$

$$down\;i = chopstick\;i := \bot$$

$eat\;i$ = (uses *chopstick i* and *chopstick*($i \oplus 1$)) ⟵

$think\;i$ = (does not use any *chopstick*) ⟵

Dining Philosophers

$$\mathit{life} \;=\; P\,0 \,\|\, P\,1 \,\|\, P\,2 \,\|\, P\,3 \,\|\, P\,4 \qquad ✗$$

$$P\,i \;=\; \mathit{think}\ i.\ (\mathit{up}\ i \,\|\, \mathit{up}(i{\oplus}1)).\ \mathit{eat}\ i.\ (\mathit{down}\ i \,\|\, \mathit{down}(i{\oplus}1)).\ P\,i$$

$$\mathit{up}\ i \;=\; \mathit{chopstick}\ i := \top$$

$$\mathit{down}\ i \;=\; \mathit{chopstick}\ i := \bot$$

eat i = (uses *chopstick i* and *chopstick*($i{\oplus}1$))

think i = (does not use any *chopstick*)

Dining Philosophers

$$life \;=\; (P\,0 \vee P\,1 \vee P\,2 \vee P\,3 \vee P\,4).\ life$$

$$P\,i \;=\; think\,i.\ up\,i.\ up(i \oplus 1).\ eat\,i.\ down\,i.\ down(i \oplus 1)$$

$$up\,i \;=\; chopstick\,i := \top$$

$$down\,i \;=\; chopstick\,i := \bot$$

$$eat\,i \;=\; (\text{uses } chopstick\,i \text{ and } chopstick(i \oplus 1)\,)$$

$$think\,i \;=\; (\text{does not use any } chopstick\,)$$

Dining Philosophers

$$life \;=\; (P\,0 \vee P\,1 \vee P\,2 \vee P\,3 \vee P\,4).\ life \quad \longleftarrow$$

$$P\,i \;=\; think\ i.\ up\ i.\ up(i \oplus 1).\ eat\ i.\ down\ i.\ down(i \oplus 1)$$

$$up\ i \;=\; chopstick\ i := \top$$

$$down\ i \;=\; chopstick\ i := \bot$$

$eat\ i$ = (uses *chopstick* i and $chopstick(i \oplus 1)$)

$think\ i$ = (does not use any *chopstick*)

Dining Philosophers

life	=	$(P\,0 \vee P\,1 \vee P\,2 \vee P\,3 \vee P\,4).\ \mathit{life}$
P i	=	$\mathit{think}\ i.\ \mathit{up}\ i.\ \mathit{up}(i{\oplus}1).\ \mathit{eat}\ i.\ \mathit{down}\ i.\ \mathit{down}(i{\oplus}1)$ ⟵
up i	=	$\mathit{chopstick}\ i := \top$
down i	=	$\mathit{chopstick}\ i := \bot$
eat i	=	(uses *chopstick i* and $\mathit{chopstick}(i{\oplus}1)$)
think i	=	(does not use any *chopstick*)

Dining Philosophers

$$life \;=\; (P\,0 \vee P\,1 \vee P\,2 \vee P\,3 \vee P\,4).\; life$$

$$P\,i \;=\; think\;i.\; up\;i.\; up(i\oplus 1).\; eat\;i.\; down\;i.\; down(i\oplus 1)$$

$$up\;i \;=\; chopstick\;i := \top$$

$$down\;i \;=\; chopstick\;i := \bot$$

$$eat\;i \;=\; (\text{uses}\;\; chopstick\;i\;\;\text{and}\;\; chopstick(i\oplus 1)\;)$$

$$think\;i \;=\; (\text{does not use any}\;\; chopstick\;)$$

$(think\;i.\,think\;j)$ becomes $(think\;i \,\|\, think\;j)$.
$(think\;i.\,up\;j)$ and $(up\;j.\,think\;i)$ become $(think\;i \,\|\, up\;j)$.
$(think\;i.\,down\;j)$ and $(down\;j.\,think\;i)$ become $(think\;i \,\|\, down\;j)$.
$(think\;i.\,eat\;j)$ and $(eat\;j.\,think\;i)$ become $(think\;i \,\|\, eat\;j)$.
If $i \neq j$, $(up\;i.\,up\;j)$ becomes $(up\;i \,\|\, up\;j)$.
If $i \neq j$, $(up\;i.\,down\;j)$ and $(down\;j.\,up\;i)$ become $(up\;i \,\|\, down\;j)$.
If $i \neq j$, $(down\;i.\,down\;j)$ becomes $(down\;i \,\|\, down\;j)$.
If $i \neq j \wedge i\oplus 1 \neq j$, $(eat\;i.\,up\;j)$ and $(up\;j.\,eat\;i)$ become $(eat\;i \,\|\, up\;j)$.
If $i \neq j \wedge i\oplus 1 \neq j$, $(eat\;i.\,down\;j)$ and $(down\;j.\,eat\;i)$ become $(eat\;i \,\|\, down\;j)$.
If $i \neq j \wedge i\oplus 1 \neq j \wedge i \neq j\oplus 1$, $(eat\;i.\,eat\;j)$ becomes $(eat\;i \,\|\, eat\;j)$.

Dining Philosophers

$$life = (P\,0 \lor P\,1 \lor P\,2 \lor P\,3 \lor P\,4).\ life$$

$$P\,i = think\ i.\ up\ i.\ up(i \oplus 1).\ eat\ i.\ down\ i.\ down(i \oplus 1)$$

$$up\ i = chopstick\ i := \top$$

$$down\ i = chopstick\ i := \bot$$

$eat\ i$ = (uses $chopstick\ i$ and $chopstick(i \oplus 1)$)

$think\ i$ = (does not use any $chopstick$)

$(think\ i.\ think\ j)$ becomes $(think\ i \,\|\, think\ j)$. ←
$(think\ i.\ up\ j)$ and $(up\ j.\ think\ i)$ become $(think\ i \,\|\, up\ j)$.
$(think\ i.\ down\ j)$ and $(down\ j.\ think\ i)$ become $(think\ i \,\|\, down\ j)$.
$(think\ i.\ eat\ j)$ and $(eat\ j.\ think\ i)$ become $(think\ i \,\|\, eat\ j)$.
If $i \neq j$, $(up\ i.\ up\ j)$ becomes $(up\ i \,\|\, up\ j)$.
If $i \neq j$, $(up\ i.\ down\ j)$ and $(down\ j.\ up\ i)$ become $(up\ i \,\|\, down\ j)$.
If $i \neq j$, $(down\ i.\ down\ j)$ becomes $(down\ i \,\|\, down\ j)$.
If $i \neq j \land i \oplus 1 \neq j$, $(eat\ i.\ up\ j)$ and $(up\ j.\ eat\ i)$ become $(eat\ i \,\|\, up\ j)$.
If $i \neq j \land i \oplus 1 \neq j$, $(eat\ i.\ down\ j)$ and $(down\ j.\ eat\ i)$ become $(eat\ i \,\|\, down\ j)$.
If $i \neq j \land i \oplus 1 \neq j \land i \neq j \oplus 1$, $(eat\ i.\ eat\ j)$ becomes $(eat\ i \,\|\, eat\ j)$.

Dining Philosophers

$$life = (P\,0 \lor P\,1 \lor P\,2 \lor P\,3 \lor P\,4).\ life$$

$$P\,i = think\ i.\ up\ i.\ up(i\oplus 1).\ eat\ i.\ down\ i.\ down(i\oplus 1)$$

$$up\ i = chopstick\ i := \top$$

$$down\ i = chopstick\ i := \bot$$

$eat\ i$ = (uses $chopstick\ i$ and $chopstick(i\oplus 1)$)

$think\ i$ = (does not use any $chopstick$)

$(think\ i.\ think\ j)$ becomes $(think\ i \,\|\, think\ j)$.
$(think\ i.\ up\ j)$ and $(up\ j.\ think\ i)$ become $(think\ i \,\|\, up\ j)$. ←
$(think\ i.\ down\ j)$ and $(down\ j.\ think\ i)$ become $(think\ i \,\|\, down\ j)$. ←
$(think\ i.\ eat\ j)$ and $(eat\ j.\ think\ i)$ become $(think\ i \,\|\, eat\ j)$. ←
If $i \ne j$, $(up\ i.\ up\ j)$ becomes $(up\ i \,\|\, up\ j)$.
If $i \ne j$, $(up\ i.\ down\ j)$ and $(down\ j.\ up\ i)$ become $(up\ i \,\|\, down\ j)$.
If $i \ne j$, $(down\ i.\ down\ j)$ becomes $(down\ i \,\|\, down\ j)$.
If $i \ne j \land i\oplus 1 \ne j$, $(eat\ i.\ up\ j)$ and $(up\ j.\ eat\ i)$ become $(eat\ i \,\|\, up\ j)$.
If $i \ne j \land i\oplus 1 \ne j$, $(eat\ i.\ down\ j)$ and $(down\ j.\ eat\ i)$ become $(eat\ i \,\|\, down\ j)$.
If $i \ne j \land i\oplus 1 \ne j \land i \ne j\oplus 1$, $(eat\ i.\ eat\ j)$ becomes $(eat\ i \,\|\, eat\ j)$.

Dining Philosophers

$$life \;=\; (P\,0 \lor P\,1 \lor P\,2 \lor P\,3 \lor P\,4).\; life$$

$$P\,i \;=\; think\,i.\; up\,i.\; up(i \oplus 1).\; eat\,i.\; down\,i.\; down(i \oplus 1)$$

$$up\,i \;=\; chopstick\,i := \top$$

$$down\,i \;=\; chopstick\,i := \bot$$

$eat\,i \;=\;$ (uses $chopstick\,i$ and $chopstick(i \oplus 1)$)

$think\,i \;=\;$ (does not use any $chopstick$)

$(think\,i.\, think\,j)$ becomes $(think\,i \parallel think\,j)$.
$(think\,i.\, up\,j)$ and $(up\,j.\, think\,i)$ become $(think\,i \parallel up\,j)$.
$(think\,i.\, down\,j)$ and $(down\,j.\, think\,i)$ become $(think\,i \parallel down\,j)$.
$(think\,i.\, eat\,j)$ and $(eat\,j.\, think\,i)$ become $(think\,i \parallel eat\,j)$.
If $i \ne j$, $(up\,i.\, up\,j)$ becomes $(up\,i \parallel up\,j)$. ←
If $i \ne j$, $(up\,i.\, down\,j)$ and $(down\,j.\, up\,i)$ become $(up\,i \parallel down\,j)$.
If $i \ne j$, $(down\,i.\, down\,j)$ becomes $(down\,i \parallel down\,j)$.
If $i \ne j \land i \oplus 1 \ne j$, $(eat\,i.\, up\,j)$ and $(up\,j.\, eat\,i)$ become $(eat\,i \parallel up\,j)$.
If $i \ne j \land i \oplus 1 \ne j$, $(eat\,i.\, down\,j)$ and $(down\,j.\, eat\,i)$ become $(eat\,i \parallel down\,j)$.
If $i \ne j \land i \oplus 1 \ne j \land i \ne j \oplus 1$, $(eat\,i.\, eat\,j)$ becomes $(eat\,i \parallel eat\,j)$.

Dining Philosophers

$life$	$=$	$(P\,0 \lor P\,1 \lor P\,2 \lor P\,3 \lor P\,4).\ life$
$P\ i$	$=$	$think\ i.\ up\ i.\ up(i\oplus 1).\ eat\ i.\ down\ i.\ down(i\oplus 1)$
$up\ i$	$=$	$chopstick\ i := \top$
$down\ i$	$=$	$chopstick\ i := \bot$
$eat\ i$	$=$	(uses $chopstick\ i$ and $chopstick(i\oplus 1)$)
$think\ i$	$=$	(does not use any $chopstick$)

$(think\ i.\ think\ j)$ becomes $(think\ i \,\|\, think\ j)$.
$(think\ i.\ up\ j)$ and $(up\ j.\ think\ i)$ become $(think\ i \,\|\, up\ j)$.
$(think\ i.\ down\ j)$ and $(down\ j.\ think\ i)$ become $(think\ i \,\|\, down\ j)$.
$(think\ i.\ eat\ j)$ and $(eat\ j.\ think\ i)$ become $(think\ i \,\|\, eat\ j)$.
If $i \neq j$, $(up\ i.\ up\ j)$ becomes $(up\ i \,\|\, up\ j)$.
If $i \neq j$, $(up\ i.\ down\ j)$ and $(down\ j.\ up\ i)$ become $(up\ i \,\|\, down\ j)$. ←
If $i \neq j$, $(down\ i.\ down\ j)$ becomes $(down\ i \,\|\, down\ j)$.
If $i \neq j \land i\oplus 1 \neq j$, $(eat\ i.\ up\ j)$ and $(up\ j.\ eat\ i)$ become $(eat\ i \,\|\, up\ j)$.
If $i \neq j \land i\oplus 1 \neq j$, $(eat\ i.\ down\ j)$ and $(down\ j.\ eat\ i)$ become $(eat\ i \,\|\, down\ j)$.
If $i \neq j \land i\oplus 1 \neq j \land i \neq j\oplus 1$, $(eat\ i.\ eat\ j)$ becomes $(eat\ i \,\|\, eat\ j)$.

Dining Philosophers

$$life \;=\; (P\,0 \vee P\,1 \vee P\,2 \vee P\,3 \vee P\,4).\; life$$

$$P\,i \;=\; think\;i.\; up\;i.\; up(i \oplus 1).\; eat\;i.\; down\;i.\; down(i \oplus 1)$$

$$up\;i \;=\; chopstick\;i := \top$$

$$down\;i \;=\; chopstick\;i := \bot$$

$$eat\;i \;=\; (\text{uses } chopstick\;i \text{ and } chopstick(i \oplus 1))$$

$$think\;i \;=\; (\text{does not use any } chopstick)$$

$(think\;i.\,think\;j)$ becomes $(think\;i \,\|\, think\;j)$.
$(think\;i.\,up\;j)$ and $(up\;j.\,think\;i)$ become $(think\;i \,\|\, up\;j)$.
$(think\;i.\,down\;j)$ and $(down\;j.\,think\;i)$ become $(think\;i \,\|\, down\;j)$.
$(think\;i.\,eat\;j)$ and $(eat\;j.\,think\;i)$ become $(think\;i \,\|\, eat\;j)$.
If $i \neq j$, $(up\;i.\,up\;j)$ becomes $(up\;i \,\|\, up\;j)$.
If $i \neq j$, $(up\;i.\,down\;j)$ and $(down\;j.\,up\;i)$ become $(up\;i \,\|\, down\;j)$.
If $i \neq j$, $(down\;i.\,down\;j)$ becomes $(down\;i \,\|\, down\;j)$. ←
If $i \neq j \wedge i \oplus 1 \neq j$, $(eat\;i.\,up\;j)$ and $(up\;j.\,eat\;i)$ become $(eat\;i \,\|\, up\;j)$.
If $i \neq j \wedge i \oplus 1 \neq j$, $(eat\;i.\,down\;j)$ and $(down\;j.\,eat\;i)$ become $(eat\;i \,\|\, down\;j)$.
If $i \neq j \wedge i \oplus 1 \neq j \wedge i \neq j \oplus 1$, $(eat\;i.\,eat\;j)$ becomes $(eat\;i \,\|\, eat\;j)$.

Dining Philosophers

$$life \;=\; (P\,0 \vee P\,1 \vee P\,2 \vee P\,3 \vee P\,4).\; life$$

$$P\,i \;=\; think\,i.\; up\,i.\; up(i\oplus 1).\; eat\,i.\; down\,i.\; down(i\oplus 1)$$

$$up\,i \;=\; chopstick\,i := \top$$

$$down\,i \;=\; chopstick\,i := \bot$$

$$eat\,i \;=\; (\text{uses } chopstick\,i \text{ and } chopstick(i\oplus 1))$$

$$think\,i \;=\; (\text{does not use any } chopstick)$$

$(think\,i.\,think\,j)$ becomes $(think\,i \,\|\, think\,j)$.
$(think\,i.\,up\,j)$ and $(up\,j.\,think\,i)$ become $(think\,i \,\|\, up\,j)$.
$(think\,i.\,down\,j)$ and $(down\,j.\,think\,i)$ become $(think\,i \,\|\, down\,j)$.
$(think\,i.\,eat\,j)$ and $(eat\,j.\,think\,i)$ become $(think\,i \,\|\, eat\,j)$.
If $i \neq j$, $(up\,i.\,up\,j)$ becomes $(up\,i \,\|\, up\,j)$.
If $i \neq j$, $(up\,i.\,down\,j)$ and $(down\,j.\,up\,i)$ become $(up\,i \,\|\, down\,j)$.
If $i \neq j$, $(down\,i.\,down\,j)$ becomes $(down\,i \,\|\, down\,j)$.
If $i \neq j \wedge i\oplus 1 \neq j$, $(eat\,i.\,up\,j)$ and $(up\,j.\,eat\,i)$ become $(eat\,i \,\|\, up\,j)$. ←
If $i \neq j \wedge i\oplus 1 \neq j$, $(eat\,i.\,down\,j)$ and $(down\,j.\,eat\,i)$ become $(eat\,i \,\|\, down\,j)$. ←
If $i \neq j \wedge i\oplus 1 \neq j \wedge i \neq j\oplus 1$, $(eat\,i.\,eat\,j)$ becomes $(eat\,i \,\|\, eat\,j)$.

Dining Philosophers

$$life = (P\,0 \vee P\,1 \vee P\,2 \vee P\,3 \vee P\,4).\ life$$

$$P\,i = think\ i.\ up\ i.\ up(i\oplus 1).\ eat\ i.\ down\ i.\ down(i\oplus 1)$$

$$up\ i = chopstick\ i := \top$$

$$down\ i = chopstick\ i := \bot$$

$$eat\ i = (\text{uses } chopstick\ i \text{ and } chopstick(i\oplus 1))$$

$$think\ i = (\text{does not use any } chopstick)$$

$(think\ i.\ think\ j)$ becomes $(think\ i \| think\ j)$.
$(think\ i.\ up\ j)$ and $(up\ j.\ think\ i)$ become $(think\ i \| up\ j)$.
$(think\ i.\ down\ j)$ and $(down\ j.\ think\ i)$ become $(think\ i \| down\ j)$.
$(think\ i.\ eat\ j)$ and $(eat\ j.\ think\ i)$ become $(think\ i \| eat\ j)$.
If $i \neq j$, $(up\ i.\ up\ j)$ becomes $(up\ i \| up\ j)$.
If $i \neq j$, $(up\ i.\ down\ j)$ and $(down\ j.\ up\ i)$ become $(up\ i \| down\ j)$.
If $i \neq j$, $(down\ i.\ down\ j)$ becomes $(down\ i \| down\ j)$.
If $i \neq j \wedge i\oplus 1 \neq j$, $(eat\ i.\ up\ j)$ and $(up\ j.\ eat\ i)$ become $(eat\ i \| up\ j)$.
If $i \neq j \wedge i\oplus 1 \neq j$, $(eat\ i.\ down\ j)$ and $(down\ j.\ eat\ i)$ become $(eat\ i \| down\ j)$.
If $i \neq j \wedge i\oplus 1 \neq j \wedge i \neq j\oplus 1$, $(eat\ i.\ eat\ j)$ becomes $(eat\ i \| eat\ j)$. ←