| (par                |
|---------------------|
| ((i_1 ())           |
| (seq                |
| (choose             |
| k                   |
| unit                |
| (backtrack i 2 k 2) |
| (err "Failure!"))   |
| (let x =            |
| (choose             |
| k 1                 |
| 1                   |
| (seq                |
| (backtrack i 2 k 3) |
| 4)                  |
| (backtrack i 1 k))  |
| in                  |
| (let y =            |
| (recv i_2)          |
| in                  |
| (let z =            |
| (recv i 3)          |
| in                  |
| (if (< x y)         |
| (if (< y z)         |
| (seg                |
| (collect k)         |
| (collect k 1)       |
| (send i 2 1)        |
| (send i 3 1)        |
| (err                |
| "Success!"))        |
| (backtrack          |
| i 1                 |
| L 1\\               |

```
(par
((i_1 ())
 (seq
  (choose
  k
  unit
  (backtrack i_2 k_2)
  (err "Failure!"))
  (let x =
   (choose
    k 1
    1
    (seq
     (backtrack i_2 k_3)
    (backtrack i_1 k))
   in
   (let y =
     (recvi 2)
     in
     (let z =
      (recv i_3)
      in
      (if (< x y)
       (if (< y z)
         (seq
          (collect k)
          (collect k 1)
          (send i 2 1)
          (send i_3 1)
          (err
          "Success!"))
         (backtrack
          i 1
          _
|<sub>2</sub> 1\\
```

```
(par
((i_1 ())
 (seq
  (choos
  k
  unit
  (backt
  (err "F
  (let x =
   (choc
    k 1
    1
    (seq
    (bac
    4)
    (bacl
   in
   (let y
    (rec
    in
    (let:
      (re
      in
      (if
       (it
```

| ie  |
|---|
| track i_2 k_2) -ailure!")) = ose  |
| :ktrack i_2 k_3)  |
| ktrack i_1 k))  |
| =<br>v i_2)   |
| z =<br>:cv i_3)   |
| <pre>(&lt; x y) f (&lt; y z) (seq   (collect k)   (collect k_1)   (send i_2 1)   (send i_3 1)   (err    "Success!")) (backtrack   i_1</pre> |

```
(par
((i 1
 ((k
   (seq
   (choose
    k
    (backtrack i_2 k_2)
    (err "Failure!")
    unit)
   (let x =
     (choose
      k_1
      1
      (seq
      (backtrack i_2 k_3)
      (backtrack i_1 k))
     in
     (let y =
      (recvi 2)
      in
      (let z =
        (recv i_3)
        in
        (if (< x y)
         (if (< y z)
           (seq
           (collect k)
           (collect k_1)
           (send i_2 1)
           (send i_3 1)
           (err
            "Success!"))
           (backtrack
```

```
(par
((i<sub>1</sub>
  ((k
   (seq
    (choose
     k
     (backtrack i_2 k_2)
     (err "Failure!")
     unit)
    (let x =
     (choose
      k_1
      1
      (seq
       (backtrack i_2 k_
      (backtrack i_1 k))
     in
     (let y =
       (recv i_2)
       in
       (let z =
        (recv i_3)
        in
        (if (< x y)
          (if (< y z)
           (seq
            (collect k)
            (collect k_1)
            (send i_2 1)
            (send i_3 1)
            (err
             "Success!"))
```

(backtrack

|         | (par                 | (par                 |
|---------|----------------------|----------------------|
|         | ((i_1                | ((i_1                |
|         | ( <mark>(</mark> k   | ( <mark>(</mark> k   |
|         | (seq                 | (seq                 |
|         | (choose              | (choose              |
|         | k                    | k                    |
| k_2)    | (backtrack i_2 k_2)  | (backtrack i_2 k_2)  |
| )       | (err "Failure!")     | (err "Failure!")     |
|         | unit)                | unit)                |
|         | (let x =             | (let x =             |
|         | (choose              | (choose              |
|         | k_1                  | k_1                  |
|         | 1                    |                      |
|         | (seq                 | (seq                 |
| _2 k_3) | (backtrack i_2 k_3)  | (backtrack i_2 k_3)  |
|         | 4)                   | 4)                   |
| 1 k))   | (backtrack i_1 k))   | (backtrack i_1 k))   |
|         | in                   | in                   |
|         | (let y =             | (let y =             |
|         | (recv i_2)           | (recv i_2)           |
|         | in<br>(lat =         | in<br>(lab =         |
|         | (let z = (recvi : 2) | (let z = (recvi : 2) |
|         | (recv i_3)           | (recv i_3)           |
|         | in<br>(if ( < x y)   | in<br>(if ( < x y)   |
|         | (if (< x y))         | (if (< x y))         |
|         | (if (< y z)<br>(seq  | (if (< y z)<br>(seq  |
| k)      | (collect k)          | (collect k)          |
| k_1)    | (collect k 1)        | (collect k 1)        |
| 2 1)    | (send i 2 1)         | (send i_2 1)         |
| 3 1)    | (send i_3 1)         | (send i 3 1)         |
| J ± /   | (err                 | (err                 |
| ss!"))  | "Success!"))         | "Success!"))         |
| ck      | (backtrack           | (backtrack           |
|         | 1 1                  | ; 1                  |

(pa ((i\_ ((

| (par                |
|---------------------|
| ((i_1               |
| ((k                 |
| (seq                |
| (choose             |
| k                   |
| (backtrack i 2 k 2) |
| (err "Failure!")    |
| unit)               |
| (let x =            |
| (choose             |
| k 1                 |
| 1                   |
| (seq                |
| (backtrack i 2 k 3) |
| 4)                  |
| (backtrack i 1 k))  |
| in                  |
| (let y =            |
| (recv i_2)          |
| in                  |
| (let z =            |
| (recv i 3)          |
| in                  |
| (if (< x y)         |
| (if (< yz)          |
| (seq                |
| (collect k)         |
| (collect k 1)       |
| (send i 2 1)        |
| (send i <u>3</u> 1) |
| (err                |
| "Success!"))        |
| (backtrack          |
| i 1                 |
|                     |

```
(par
((i 1
 ((k
   (seq
    (choose
    k
    (backtrack i_2 k_2)
    (err "Failure!")
    unit)
    (let x =
     (choose
      k_1
      1
      (seq
      (backtrack i_2 k_3)
      4)
      (backtrack i_1 k))
     in
     (let y =
      (recvi 2)
      in
      (let z =
        (recv i_3)
        in
        (if (< x y)
         (if (< y z)
          (seq
           (collect k)
           (collect k_1)
           (send i_2 1)
           (send i_3 1)
           (err
            "Success!"))
           (backtrack
           : 1
```

```
(par
((i 1
  ((k
   (seq
    (choose
    k
     (backtra
    (err "Fa
    unit)
    (let x =
     (choose
      k_1
      1
      (seq
      (backtı
     in
     (let y =
      (recv
      in
```

(backt

(let z :

in

(recv

(if (<

(if (

(Se

(0

(0

(5

(5

(e

(b

4)