Notation

- Assume $P$ contains port $\overline{b}$ and $Q$ contains port $a$
Notation

- Assume $P$ contains port $\bar{b}$ and $Q$ contains port $a$
- Define $P \bowtie Q$ linking $\bar{b}$ to $a$: $(P[c/b] \mid Q[c/a])\backslash\{c\}$
  where $c$ is a new port (not contained in $P$ or $Q$)
Notation

- Assume $P$ contains port $\overline{b}$ and $Q$ contains port $a$
- Define $P \leftarrow Q$ linking $\overline{b}$ to $a$: $(P[c/b] \mid Q[c/a]) \setminus \{c\}$ where $c$ is a new port (not contained in $P$ or $Q$)
- Example: buffers $B \overset{\text{def}}{=} i(x).o(x).B$

\[
\begin{align*}
B_1 & \equiv B[o_1/o] \\
B_{j+1} & \equiv B[o_j/i, o_{j+1}/o] \quad 1 \leq j < n-1 \\
B_n & \equiv B[o_{n-1}/i]
\end{align*}
\]
Notation

- Assume $P$ contains port $\overline{b}$ and $Q$ contains port $a$
- Define $P \bowtie Q$ linking $\overline{b}$ to $a$: $(P[c/b] \mid Q[c/a]) \backslash \{c\}$ where $c$ is a new port (not contained in $P$ or $Q$)
- Example: buffers $B \overset{\text{def}}{=} i(x).\overline{o}(x).B$

\[ B_1 \equiv B[\overline{o}_1/o] \]
\[ B_{j+1} \equiv B[\overline{o}_j/i, \overline{o}_{j+1}/o] \quad 1 \leq j < n - 1 \]
\[ B_n \equiv B[\overline{o}_{n-1}/i] \]

- Redo as $n$ $B$s with $\overline{o}$ linking $i$: $B \bowtie B \bowtie \ldots \bowtie B$
Sorting machine example

- Where a system of size \( n + 1 \) is defined in terms of a system of size \( n \). (From Milner’s book 136ff.)
Sorting machine example

- Where a system of size \( n + 1 \) is defined in terms of a system of size \( n \). (From Milner’s book 136ff.)

- Want a sorter \( \text{Sorter}_n, n \geq 0 \), capable of sorting \( n \)-length sequences of positive integers
Sorting machine example

- Where a system of size $n+1$ is defined in terms of a system of size $n$. (From Milner’s book 136ff.)
- Want a sorter $\text{Sorter}_n$, $n \geq 0$, capable of sorting $n$-length sequences of positive integers
- Assume $\text{Sorter}_n$ has ports in, out
Sorting machine example

- Where a system of size $n + 1$ is defined in terms of a system of size $n$. (From Milner’s book 136ff.)
- Want a sorter $Sorter_n$, $n \geq 0$, capable of sorting $n$-length sequences of positive integers
- Assume $Sorter_n$ has ports $\text{in}$, $\text{out}$
- It accepts exactly $n$ integers one by one at port $\text{in}$;
Sorting machine example

- Where a system of size $n + 1$ is defined in terms of a system of size $n$. (From Milner’s book 136ff.)
- Want a sorter $\text{Sorter}_n$, $n \geq 0$, capable of sorting $n$-length sequences of positive integers.
- Assume $\text{Sorter}_n$ has ports $\text{in}$, $\text{out}$.
- It accepts exactly $n$ integers one by one at port $\text{in}$;
- Then it delivers them one by one in descending order at $\text{out}$, terminated by a zero.
Sorting machine example

- Where a system of size \( n + 1 \) is defined in terms of a system of size \( n \). (From Milner's book 136ff.)
- Want a sorter \( \text{Sorter}_n, \ n \geq 0 \), capable of sorting \( n \)-length sequences of positive integers
- Assume \( \text{Sorter}_n \) has ports \( \text{in}, \text{out} \)
- It accepts exactly \( n \) integers one by one at port \( \text{in} \);
- Then it delivers them one by one in descending order at \( \text{out} \), terminated by a zero
- And returns to start state
A multiset is a set with possibly multiple elements

\[ \{1, 2, 1\} = \{2, 1, 1\} \neq \{1, 2\} \]

\( S \) ranges over multisets of integers and \( \max(S) \) \( \min(S) \) are maximum and minimum elements of \( S \)
A multiset is a set with possibly multiple elements

\[ \{1, 2, 1\} = \{2, 1, 1\} \neq \{1, 2\} \]

\( S \) ranges over multisets of integers and \( \max(S) \) \( \min(S) \) are maximum and minimum elements of \( S \)

Specification of sorter

\[ \text{Spec}_n \overset{\text{def}}{=} \text{in}(x_1) \ldots \text{in}(x_n) . \text{Hold}_n(\{x_1, \ldots, x_n\}) \]

\[ \text{Hold}_n(S) \overset{\text{def}}{=} \text{out}(\max(S)) . \text{Hold}_n(S - \{\max(S)\}) \]

\[ S \neq \emptyset \]

\[ \text{Hold}_n(\emptyset) \overset{\text{def}}{=} \text{out}(0) . \text{Spec}_n \]
Sorting machine specification

- A multiset is a set with possibly multiple elements

\[
\{1, 2, 1\} = \{2, 1, 1\} \neq \{1, 2\}
\]

- \(S\) ranges over multisets of integers and \(\max(S)\) \(\min(S)\) are maximum and minimum elements of \(S\)

- Specification of sorter

\[
\begin{align*}
\text{Spec}_n & \quad \overset{\text{def}}{=} \text{in}(x_1)\ldots\text{in}(x_n)\cdot\text{Hold}_n(\{x_1, \ldots, x_n\}) \\
\text{Hold}_n(S) & \quad \overset{\text{def}}{=} \overline{\text{out}}(\max(S))\cdot\text{Hold}_n(S - \{\max(S)\}) \\
\text{Hold}_n(\emptyset) & \quad \overset{\text{def}}{=} \overline{\text{out}}(0)\cdot\text{Spec}_n
\end{align*}
\]

- Alternatively assuming \(y_1 \geq \ldots \geq y_n\)

\[
\text{Hold}_n(\{y_1, \ldots, y_n\}) \overset{\text{def}}{=} \overline{\text{out}}(y_1)\ldots\overline{\text{out}}(y_n)\cdot\overline{\text{out}}(0)\cdot\text{Spec}_n
\]
Sorting machine implementation I

- Use $n$ simple cells $C$ and a barrier cell $B$
Sorting machine implementation I

- Use $n$ simple cells $C$ and a barrier cell $B$
- $C$ has ports in, down, up, out; $B$ just has in, out
Sorting machine implementation I

- Use $n$ simple cells $C$ and a barrier cell $B$
- $C$ has ports in, down, up, out; $B$ just has in, out
- Notation: $C \backslash C$ where down in first $C$ is linked to in of second $C$ and up of first $C$ is linked to out of second $C$ and then these ports are internalised (restricted upon)
Use $n$ simple cells $C$ and a barrier cell $B$

- $C$ has ports $\text{in, down, up, out}$; $B$ just has $\text{in, out}$

- **Notation**: $C \leftarrow C$ where $\text{down}$ in first $C$ is linked to $\text{in}$ of second $C$ and $\text{up}$ of first $C$ is linked to $\text{out}$ of second $C$ and then these ports are internalised (restricted upon)

- $\text{Sorter}_n \overset{\text{def}}{=} C \leftarrow \ldots \leftarrow C \leftarrow B$ ($n$ Cs)
Sorting machine implementation I

- Use $n$ simple cells $C$ and a barrier cell $B$
- $C$ has ports $\text{in, down, up, out}$; $B$ just has $\text{in, out}$
- Notation: $C \leadsto C$ where $\text{down}$ in first $C$ is linked to $\text{in}$ of second $C$ and $\text{up}$ of first $C$ is linked to $\text{out}$ of second $C$ and then these ports are internalised (restricted upon)
- $\text{Sorter}_n \overset{\text{def}}{=} C \leadsto \ldots \leadsto C \leadsto B \ (n \ Cs)$
- We need to define $B$ and $C$ so that: $\text{Sorter}_n \approx \text{Spec}_n$
Use $n$ simple cells $C$ and a barrier cell $B$

- $C$ has ports `in`, `down`, `up`, `out`; $B$ just has `in`, `out`

- Notation: $C \overset{c}{\leftarrow} C$ where `down` in first $C$ is linked to `in` of second $C$ and `up` of first $C$ is linked to `out` of second $C$ and then these ports are internalised (restricted upon)

- $\text{Sorter}_n \overset{\text{def}}{=} C \overset{c}{\leftarrow} \ldots \overset{c}{\leftarrow} C \overset{c}{\leftarrow} B \; (n \; Cs)$

- We need to define $B$ and $C$ so that: $\text{Sorter}_n \approx \text{Spec}_n$

- Do it inductively
  1. Base Case: $B \approx \text{Spec}_0$
  2. General Step: $\text{Spec}_{n+1} \approx C \overset{c}{\leftarrow} \text{Spec}_n$
  3. Why? $\text{Sorter}_{n+1} \overset{\text{def}}{=} C \overset{c}{\leftarrow} \text{Sorter}_n$
Sorting machine implementation II

- $B$ is straightforward: $B \overset{\text{def}}{=} \overline{\text{out}}(0).B$
Sorting machine implementation II

- $B$ is straightforward: $B \overset{\text{def}}{=} \text{out}(0) . B$
- $C$ is more involved

\begin{align*}
C & \overset{\text{def}}{=} \text{in}(x) . C'(x) \\
C'(x) & \overset{\text{def}}{=} \text{down}(x) . C + \text{up}(y) . D(x, y) \\
D(x, y) & \overset{\text{def}}{=} \text{out}(\max\{x, y\}) . C''(\min\{x, y\}) \\
C''(x) & \overset{\text{def}}{=} \text{if } x = 0 \text{ then } \text{out}(0) . C \text{ else } C'(x)
\end{align*}
Sorting machine implementation II

- $B$ is straightforward: $B \overset{\text{def}}{=} \text{out}(0).B$
- $C$ is more involved

\[
\begin{align*}
C & \overset{\text{def}}{=} \text{in}(x).C'(x) \\
C'(x) & \overset{\text{def}}{=} \text{down}(x).C + \text{up}(y).D(x, y) \\
D(x, y) & \overset{\text{def}}{=} \text{out}(\max\{x, y\}).C''(\min\{x, y\}) \\
C''(x) & \overset{\text{def}}{=} \text{if } x = 0 \text{ then } \text{out}(0).C \text{ else } C'(x)
\end{align*}
\]

- Example: Sorter$_3$: $C \circ C \circ C \circ B$
Proof of correctness

- Base Case: $B \approx \text{Spec}_0$
Proof of correctness

- **Base Case:** \( B \approx \text{Spec}_0 \)
- **General Step:** \( \text{Spec}_{n+1} \approx C \sim \text{Spec}_n \approx \)
Proof of correctness

- **Base Case:** \( B \approx \text{Spec}_0 \)
- **General Step:** \( \text{Spec}_{n+1} \approx C \dashv \text{Spec}_n \approx \)
- \( (\text{in}(x_1).C'(x_1)) \dashv (\text{in}(z_1)\ldots\text{in}(z_n).\text{Hold}_n(\{z_1,\ldots,z_n\})) \approx \)
Proof of correctness

- **Base Case:** $B \approx \text{Spec}_0$
- **General Step:** $\text{Spec}_{n+1} \approx C \preceq \text{Spec}_n \approx$
- $(\text{in}(x_1).C'(x_1)) \prec (\text{in}(z_1)\ldots\text{in}(z_n).\text{Hold}_n(\{z_1,\ldots,z_n\})) \approx$
- $\text{in}(x_1).(\text{down}(x_1).C + \ldots) \prec$
  $(\text{in}(z_1)\ldots\text{in}(z_n).\text{Hold}_n(\{z_1,\ldots,z_n\})) \approx$
Proof of correctness

- **Base Case:** $B \approx \text{Spec}_0$
- **General Step:** $\text{Spec}_{n+1} \approx C \leftarrow \text{Spec}_n \approx$
  - $(\text{in}(x_1).C'(x_1)) \leftarrow (\text{in}(z_1)\ldots\text{in}(z_n).\text{Hold}_n\left({z_1, \ldots, z_n}\right)) \approx$
  - $\text{in}(x_1).(\overline{\text{down}}(x_1).C + \ldots) \leftarrow (\text{in}(z_1)\ldots\text{in}(z_n).\text{Hold}_n\left({z_1, \ldots, z_n}\right)) \approx$
  - $\text{in}(x_1).\tau.(C \leftarrow (\text{in}(z_2)\ldots\text{in}(z_n).\text{Hold}_n\left({x_1, z_2, \ldots, z_n}\right))) \approx \ldots$

Proof of correctness

- **Base Case:** \( B \approx \text{Spec}_0 \)
- **General Step:** \( \text{Spec}_{n+1} \approx C \leadsto \text{Spec}_n \approx \)
  
  \[(\text{in}(x_1).C'(x_1)) \leadsto (\text{in}(z_1)\ldots\text{in}(z_n).\text{Hold}_n(\{z_1,\ldots,z_n\})) \approx \]

  \[\text{in}(x_1).\text{(down}(x_1).C + \ldots) \leadsto \]
  
  \[(\text{in}(z_1)\ldots\text{in}(z_n).\text{Hold}_n(\{z_1,\ldots,z_n\})) \approx \]

  \[\text{in}(x_1).\tau.(C \leadsto (\text{in}(z_2)\ldots\text{in}(z_n).\text{Hold}_n(\{x_1,z_2,\ldots,z_n\}))))) \approx \ldots \]

  \[\text{in}(x_1)\ldots\text{in}(x_n).\text{in}(x_{n+1}).(C'(x_{n+1}) \leadsto \text{Hold}_n(\{x_1,\ldots,x_n\}))) \]
Proof of correctness

- **Base Case:** \( B \approx \text{Spec}_0 \)
- **General Step:** \( \text{Spec}_{n+1} \approx C \bowtie \text{Spec}_n \approx \)
  \[
  \text{(in}(x_1).C'(x_1)) \bowtie (\text{in}(z_1)\ldots\text{in}(z_n).\text{Hold}_n(\{z_1,\ldots,z_n\})) \approx
  \]
  \[
  \text{in}(x_1).\text{down}(x_1).C + \ldots \bowtie \\
  (\text{in}(z_1)\ldots\text{in}(z_n).\text{Hold}_n(\{z_1,\ldots,z_n\})) \approx
  \]
  \[
  \text{in}(x_1).\tau.(C \bowtie (\text{in}(z_2)\ldots\text{in}(z_n).\text{Hold}_n(\{x_1,z_2,\ldots,z_n\})))
  \approx \cdot
  \]
  \[
  \text{in}(x_1)\ldots\text{in}(x_n).\text{in}(x_{n+1}).(C'(x_{n+1}) \bowtie \text{Hold}_n(\{x_1,\ldots,x_n\}))
  \]
- Result follows using following lemma where if \( S \) is any multiset of size \( k \) and \( \{x\} \cup S = \{y_1,\ldots,y_{k+1}\} \) and \( y_1 \geq \ldots \geq y_{k+1} \) then
Proof of correctness

- **Base Case:** $B \approx \text{Spec}_0$
- **General Step:** $\text{Spec}_{n+1} \approx C \triangleleft \text{Spec}_n \approx (\text{in}(x_1).C'(x_1)) \triangleleft (\text{in}(z_1)...\text{in}(z_n).\text{Hold}_n(\{z_1,\ldots,z_n\})) \approx \text{in}(x_1).\text{down}(x_1).C + \ldots \triangleleft (\text{in}(z_1)...\text{in}(z_n).\text{Hold}_n(\{z_1,\ldots,z_n\})) \approx \text{in}(x_1).\tau.(C \triangleleft (\text{in}(z_2)...\text{in}(z_n).\text{Hold}_n(\{x_1,z_2,\ldots,z_n\}))) \approx \ldots \text{in}(x_1)...\text{in}(x_n).\text{in}(x_{n+1}).(C'(x_{n+1}) \triangleleft \text{Hold}_n(\{x_1,\ldots,x_n\}))$
- **Result follows using following lemma where if $S$ is any multiset of size $k$ and $\{x\} \cup S = \{y_1,\ldots,y_{k+1}\}$ and $y_1 \geq \ldots \geq y_{k+1}$ then**
  $$C'(x) \triangleleft \text{Hold}_n(S) \approx \tau.\text{out}(y_1)...\text{out}(y_{k+1}).\text{out}(0).(C \triangleleft \text{Spec}_n)$$