Notation

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

Example: buffers $B \overset{\text{def}}{=} i(x).o(x).B$

\[
\begin{align*}
B_1 & \equiv B[o_1/o] \\
B_{j+1} & \equiv B[o_j/i, o_{j+1}/o] \quad 1 \leq j < n - 1 \\
B_n & \equiv B[o_{n-1}/i]
\end{align*}
\]
Assume $P$ contains port $b$ and $Q$ contains port $a$
Define $P \rightsquigarrow Q$ linking $b$ 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 $B \overset{\text{def}}{=} i(x).\sigma(x).B$

Want a sorter $\text{Sorter}_n$, $n \geq 0$, capable of sorting $n$-length sequences of positive integers

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

Assume $\text{Sorter}_n$ has ports $\text{in}$, $\text{out}$
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 \( \text{Sorter}_n \), \( n \geq 0 \), 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

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 \( S \)
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 \( S \).

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

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

\( \text{Hold}_n(\emptyset) \) def \(\overset{\text{def}}{=}\) \(\text{out}(0).\text{Spec}_n\)

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
Use $n$ simple cells $C$ and a barrier cell $B$

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

**Notation:** $C \triangleright C$ 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)

$\text{Sorter}_n \overset{\text{def}}{=} C \triangleright \ldots \triangleright C \triangleright B \ (n \ Cs)$

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:** $\text{Spec}_{n+1} \approx C \triangleright \text{Spec}_n$
3. **Why?** $\text{Sorter}_{n+1} \overset{\text{def}}{=} C \triangleright \text{Sorter}_n$
Sorting machine implementation II

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

Proof of correctness

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

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

- **Base Case:** $B \approx \text{Spec}_0$
- **General Step:** $\text{Spec}_{n+1} \approx C \bowtie \text{Spec}_n \approx$

- $(\text{in}(x_1).C'(x_1)) \bowtie (\text{in}(z_1) \ldots \text{in}(z_n).\text{Hold}_n({z_1, \ldots, z_n})) \approx$

- $\text{in}(x_1).\tau.((\text{in}(z_2) \ldots \text{in}(z_n).\text{Hold}_n({x_1, z_2, \ldots, z_n}))) \approx$
Proof of correctness

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

Proof of correctness

- **Base Case:** $B \approx \text{Spec}_0$
- **General Step:** $\text{Spec}_{n+1} \approx C \dashv \text{Spec}_n \approx$
- $(\text{in}(x_1).C'(x_1)) \dashv (\text{in}(z_1) \ldots \text{in}(z_n).\text{Hold}_n(\{z_1, \ldots, z_n\})) \approx$
- $\text{in}(x_1).((\text{down}(x_1).C + \ldots) \dashv$
  $(\text{in}(z_1) \ldots \text{in}(z_n).\text{Hold}_n(\{z_1, \ldots, z_n\})) \approx$
- $\text{in}(x_1).\tau.(C \dashv (\text{in}(z_2) \ldots \text{in}(z_n).\text{Hold}_n(\{x_1, z_2, \ldots, z_n\})))$
  $\approx \vdash$
- $\text{in}(x_1) \ldots \text{in}(x_n).\text{in}(x_{n+1}).(C'(x_{n+1}) \dashv \text{Hold}_n(\{x_1, \ldots, x_n\}))$
- **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**
  
  $C'(x) \dashv \text{Hold}_n(S) \approx$
  
  $\tau.\text{out}(y_1) \ldots \text{out}(y_{k+1}).\text{out}(0).(C \dashv \text{Spec}_n)$