Composing Snippets

Igor Benko
University of Waterloo

Jo Ebergen
Sun Microsystems Laboratories

Eilat, Israel, April 4, 2000

These slides were prepared to accompany "Composing Snippets," published in *Proceedings of the Sixth International Symposium on Advanced Research in Asynchronous Circuits and Systems*, Eilat, Israel, 4-6 Israel, 4-6 April 2000. pp. 23-33.

ECF processes

- a simple formalism
- specification of asynchronous circuits
- safety, progress
- network composition versus
  specification composition
- hierarchical design
- part-wise design

Related work

- trace theory

$$P = \langle \text{ ports, traces } \rangle$$

$$\text{WIRE} = \langle\ \{\, a, b\,\}, \{\, \varepsilon, a, ab, aba, \ldots\,\}\ \rangle$$

- split the trace set

$$P = \langle \text{ ports, success traces, failure traces } \rangle$$

$$\text{WIRE} = \langle\ \{\, a, b\,\}, \{\, \varepsilon, a, ab, \ldots\,\}, \{\, aa, \ldots\,\}\ \rangle$$

- <u>**E**</u>nhanced <u>**C**</u>haracteristic <u>**F**</u>unctions

$$P = \langle\ \mathbf{i}.P,\ \mathbf{o}.P,\ \mathbf{f}.P\ \rangle$$

$$\mathbf{f}.P\ :\ (\mathbf{i}.P \cup \mathbf{o}.P)^* \rightarrow \{\perp, \boldsymbol{\Delta}, \Box, \nabla, \top\}$$

The labels

- $\perp$ — failure
- Δ — demanding
- $\Box$ — indifferent
- ∇ — transient
- $\top$ — miracle

Refinement

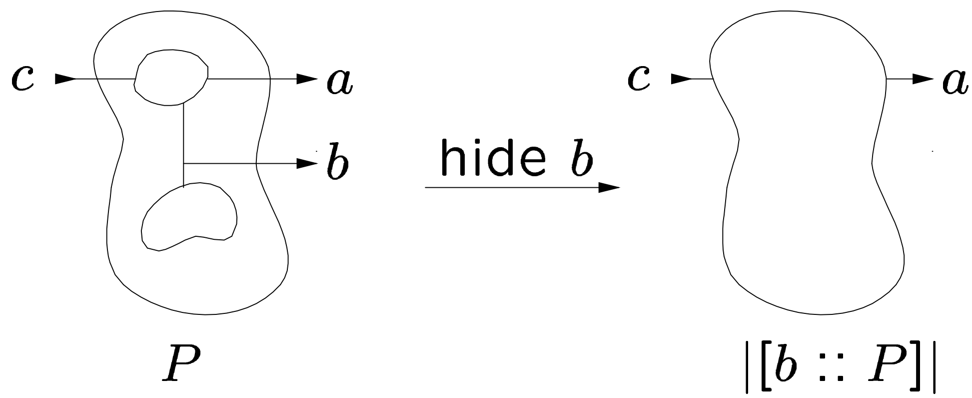
$$\bot \sqsubseteq \Delta \sqsubseteq \square \sqsubseteq \nabla \sqsubseteq \top$$

$$P \sqsubseteq Q \;\equiv\; Q \text{ ``implements'' } P$$

$$\equiv\; \forall t : \ \mathbf{f}.P.t \sqsubseteq \mathbf{f}.Q.t$$

$$\begin{tikzcd} \top & & \bot \\ \Delta \arrow[u, "b!"] \arrow[rr, bend left, "a?"] & & \nabla \arrow[u, "a?"] \arrow[ll, bend left, "b!"] \end{tikzcd} \quad \sqsubseteq \quad \begin{tikzcd} \top & & \bot \\ \square \arrow[u, "b!"] \arrow[rr, bend left, "a?"] & & \nabla \arrow[u, "a?"] \arrow[ll, bend left, "b!"] \end{tikzcd}$$

Hiding



Properties:

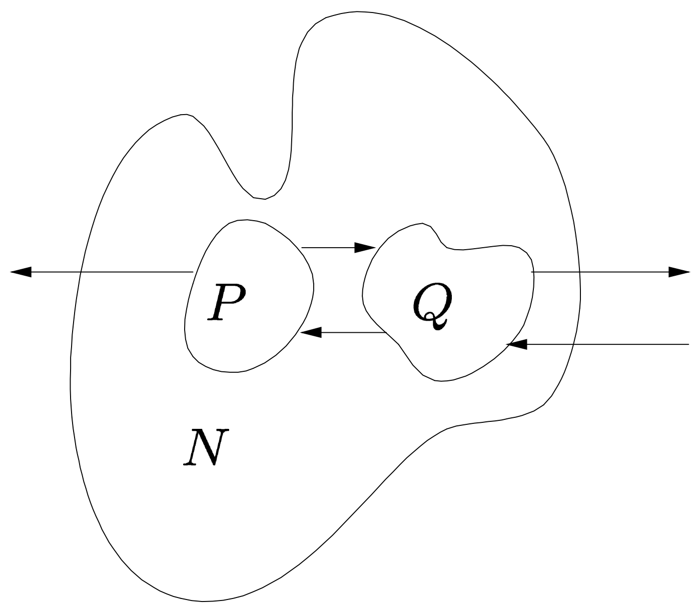
- hiding outputs only
- progress, safety taken into account
- ports can be hidden in an arbitrary order, monotonicity, distributivity over product

Process product

- an abstract operation
- no alphabet restrictions
- the basis for compositions

$$
\begin{aligned}
\mathbf{o}.P &= \mathbf{o}.P \cup \mathbf{o}.Q \\
\mathbf{i}.P &= (\mathbf{i}.P \cup \mathbf{i}.Q) - (\mathbf{o}.P \cup \mathbf{o}.Q) \\
\mathbf{f}.(P \times Q).t &= \mathbf{f}.P.(t \downarrow \mathbf{a}.P) \times \mathbf{f}.P.(t \downarrow \mathbf{a}.Q) \\
&\quad \text{unless reached } \top \text{ or } \bot
\end{aligned}
$$

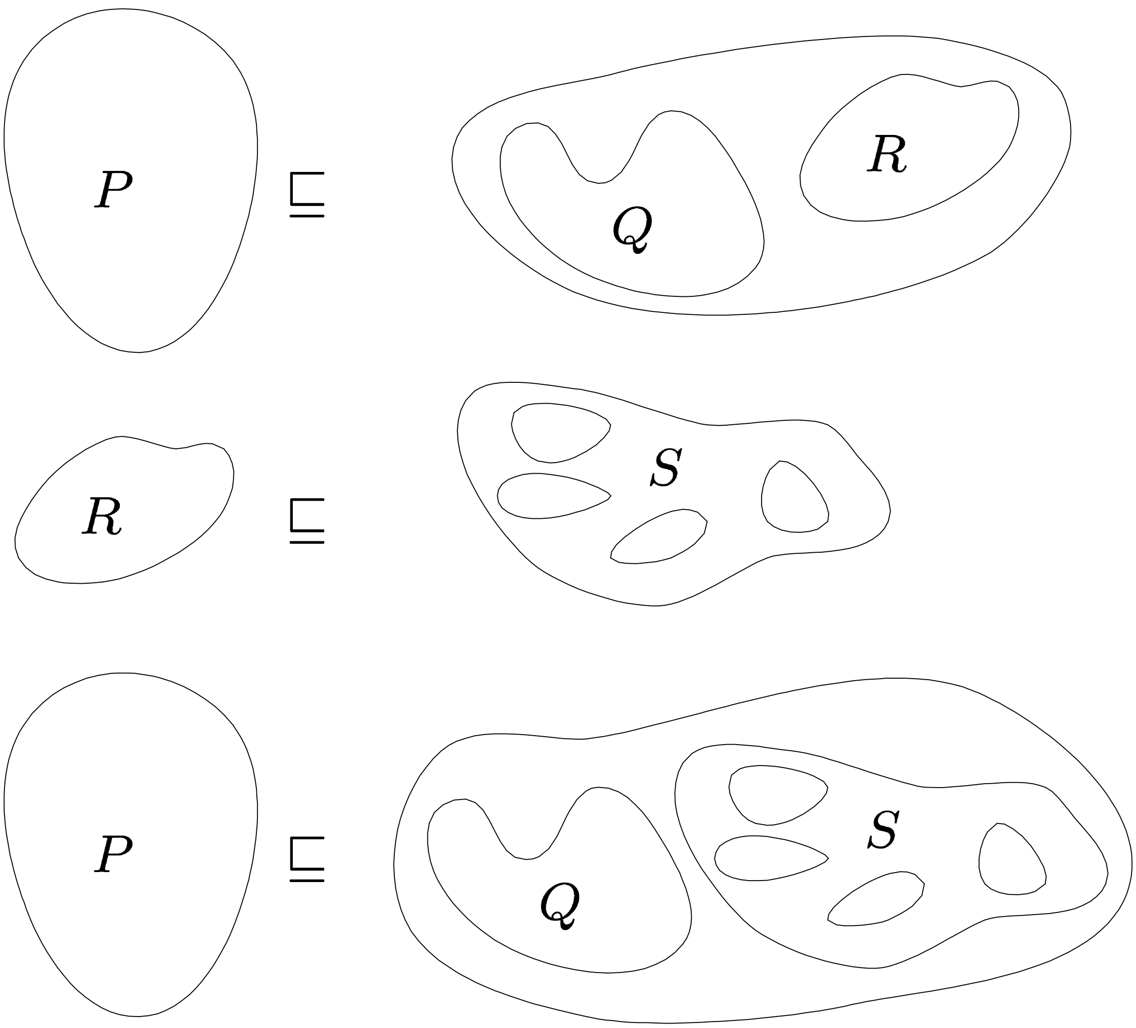
Network composition



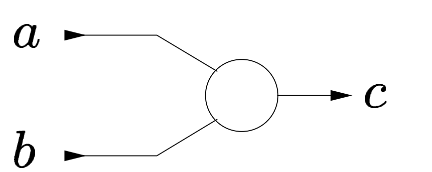
$$N = P \parallel Q$$

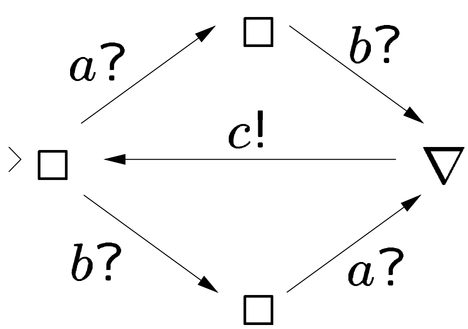
- only defined when $\mathbf{o}.P \cap \mathbf{o}.Q = \emptyset$
- $P \parallel Q \;=\; P \times Q$

Hierarchical design

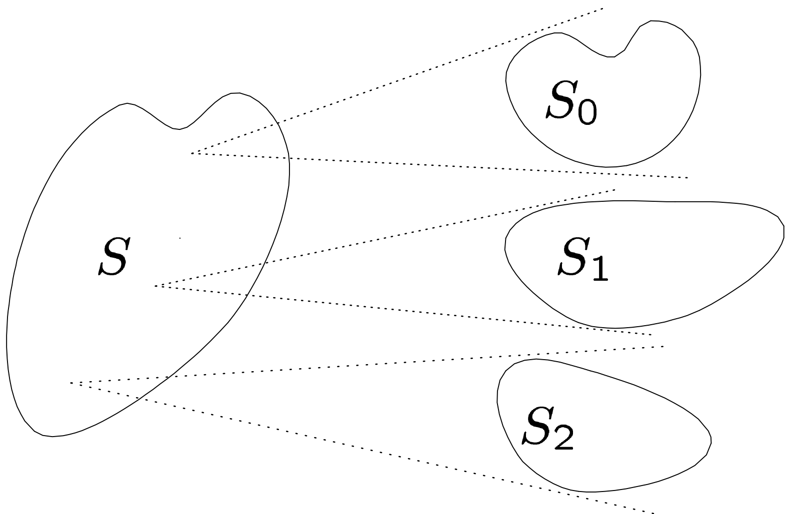


Specification composition: JOIN





Specification composition



$$S = S_0 \,\&\, S_1 \,\&\, S_2$$

S_0, S_1, S_2: snippets

Specification composition: restrictions

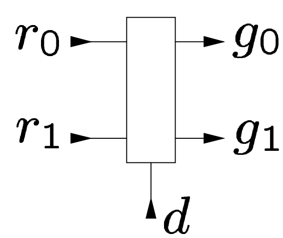
process	...	snippet
product	...	rounded product $\otimes$
refinement	...	output-persistent refinement $\sqsubseteq_o$

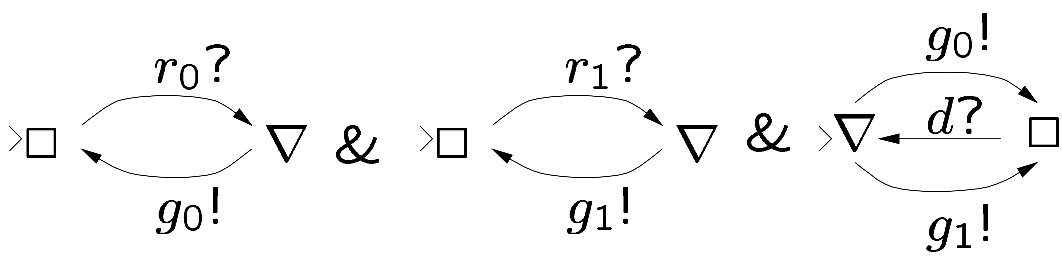
Snippets and specification composition

Specification composition:

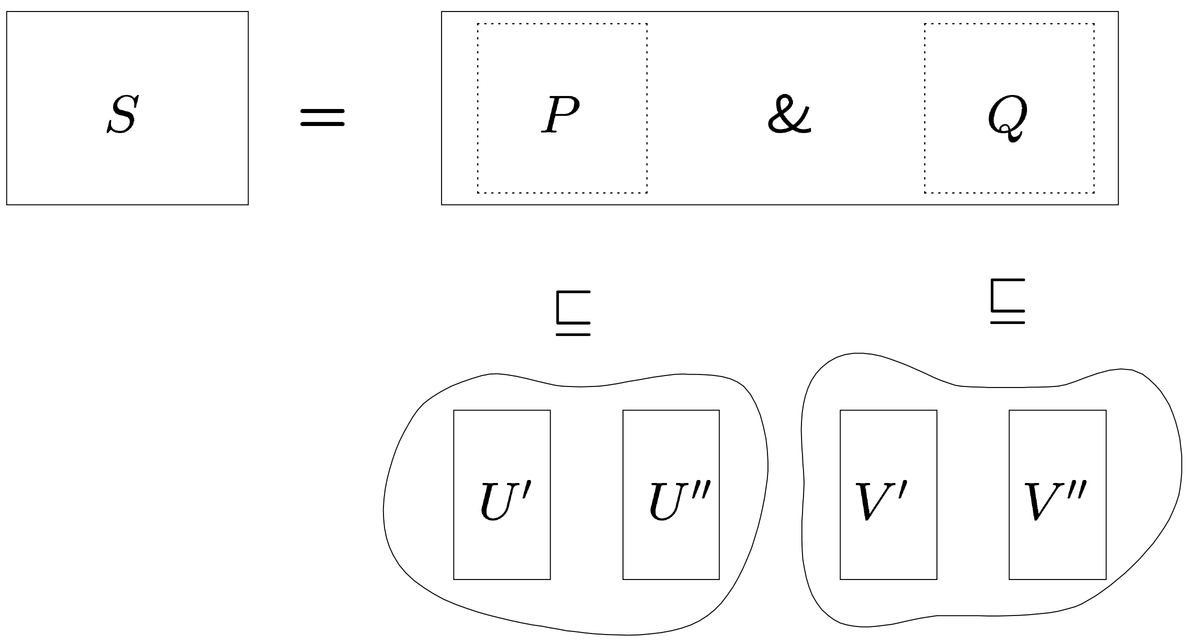
- only when $\mathbf{o}.P \cap \mathbf{i}.Q = \emptyset$ and $\mathbf{i}.P \cap \mathbf{o}.Q = \emptyset$
- defined as $P \,\&\, Q \;=\; P \otimes Q$
- snippets closed under $\&$
- $\&$ is commutative, associative,
  and monotonic with respect to $\sqsubseteq_o$

Specification composition: SEQUENCER

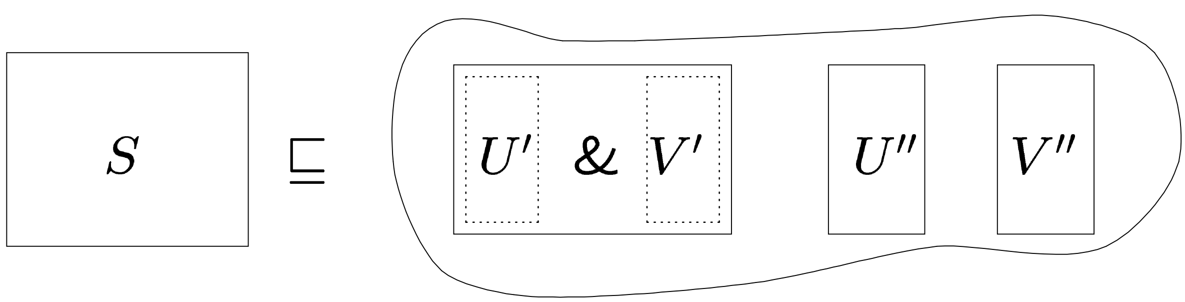




Part-wise design



U' and V' have common outputs



Conclusions

- ECF model addresses safety, progress
- hiding - safety, progress taken into account
- formal definition of snippets
- specification composition versus
  network composition
- design: hierarchical, part-wise