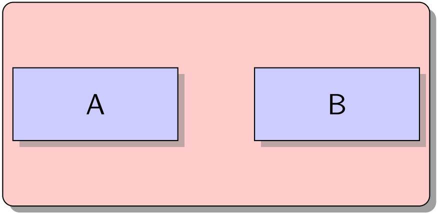
Separation-Based Reasoning for Deterministic Channel-Passing Concurrent Programs

Aimee Borda

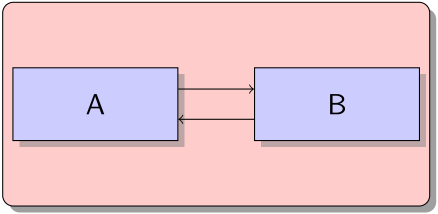
December 18, 2013

Table of Content

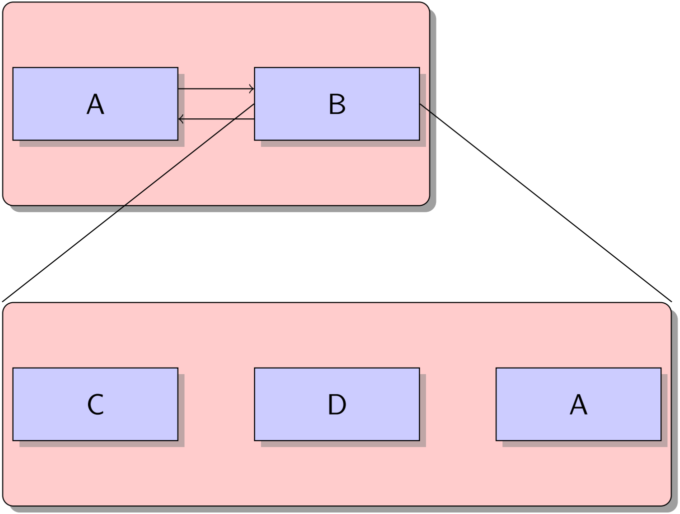
Compositional (localized) Proof Systems



Compositional (localized) Proof Systems



Compositional (localized) Proof Systems



Localized Reasoning

$$\textsc{sum}(l) = \boxed{\boxed{\textsc{sum}(l_1)} + \boxed{\textsc{sum}(l_2)}}$$

Localized Reasoning

$$\textsc{sum}(l) = \boxed{\boxed{\textsc{sum}(l_1)} + \boxed{\textsc{sum}(l_2)}}$$

$$l = l_1 \cdot l_2$$

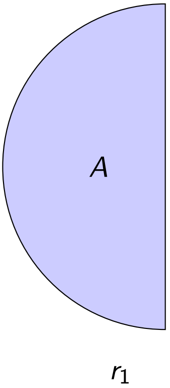
Localized Reasoning

$$\textsc{sum}(l) = \boxed{\underbrace{\boxed{\textsc{sum}(l_1)}}_{r_1} + \underbrace{\boxed{\textsc{sum}(l_2)}}_{r_2}}$$

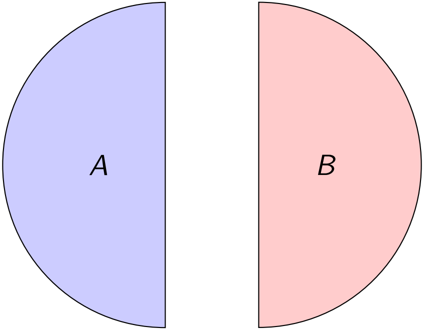
$$l = l_1 \cdot l_2$$

$$r_1 \cap r_2 = \emptyset$$

Separation Logic [Rey02]

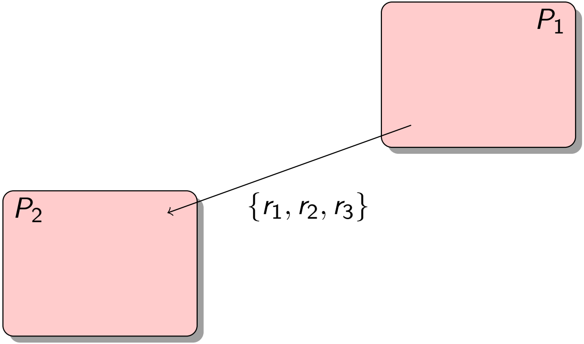


Separation Logic [Rey02]



$r_1 \perp r_2$

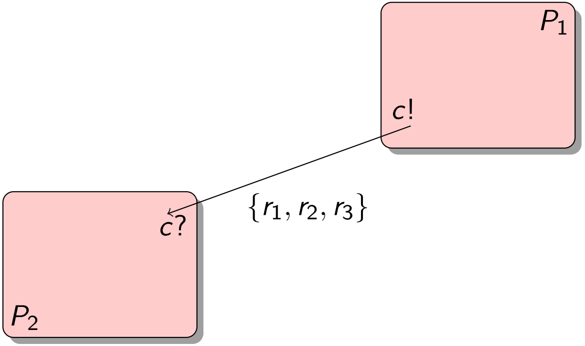
Resource Transfer [O'H07]



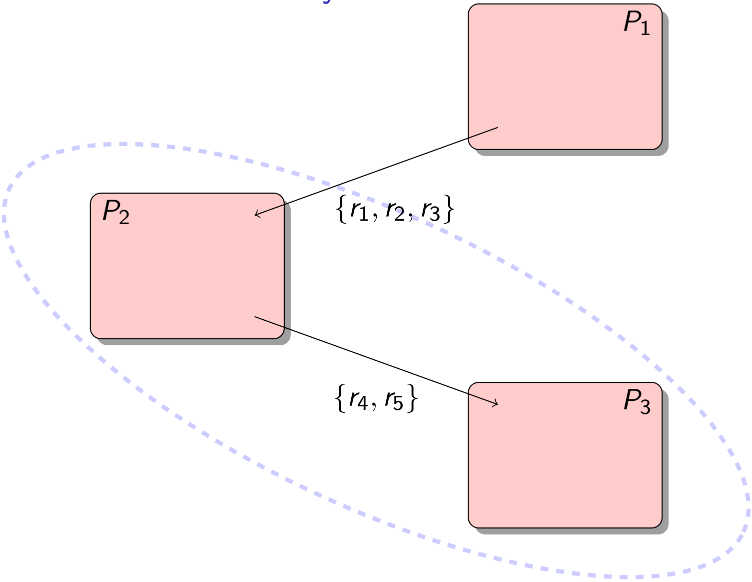
Separation-Based Reasoning for Message Passing Programs [FRS11]

$c \downarrow$	$c \uparrow$

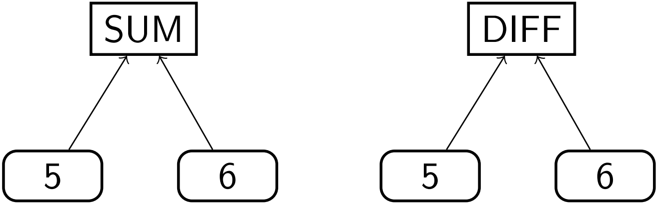
Communication Channels as Synchronization Mechanism



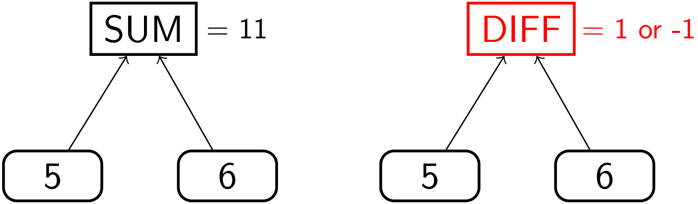
Channel Reuse - Dynamic Resource Transfer



Multiple-Sender and Single Receiver Pattern



Multiple-Sender and Single Receiver Pattern



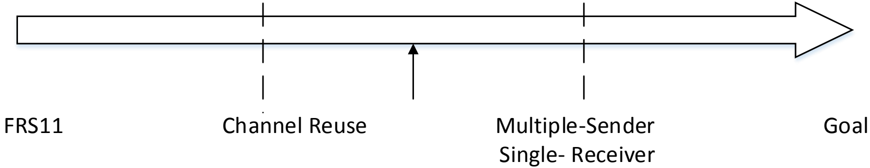
Semantic Satisfaction

$$\Gamma_{in}; \Gamma_{out}; b \vdash \{\varphi\} \;\; P \;\; \{\psi\} : \rho$$

implies

$$\Gamma_{in}; \Gamma_{out}; b \vDash \{\varphi\} \;\; P \;\; \{\psi\} : \rho$$

Technical Development Timeline



Proof of Soundness

$$\Gamma_{in}; \Gamma_{out}; b \vdash \{\varphi\} \ P \ \{\psi\} : \rho$$

(Data Analysis)

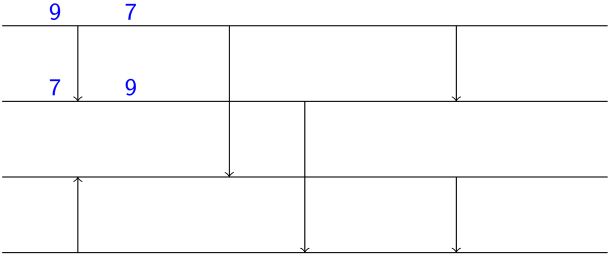
$$\lceil P \rceil_{\rho} \text{ is deterministic}$$

(Behavioral Analysis)

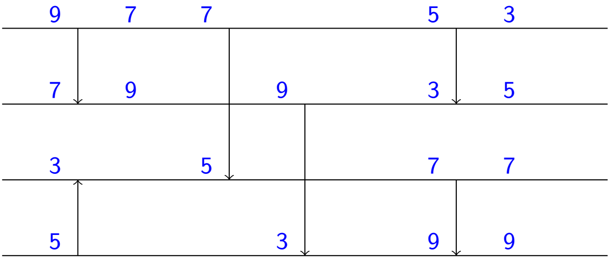
$$\Gamma_{in}; \Gamma_{out}; b \vDash \{\varphi\} \ P \ \{\psi\} : \rho$$

(P is deterministic)

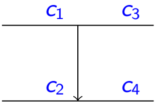
Sorting Networks [Knu98]



Sorting Networks [Knu98]

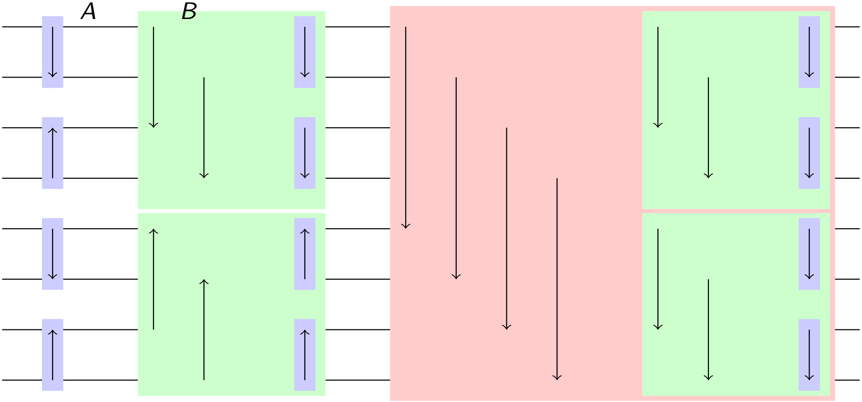


Our Implementation of SNs

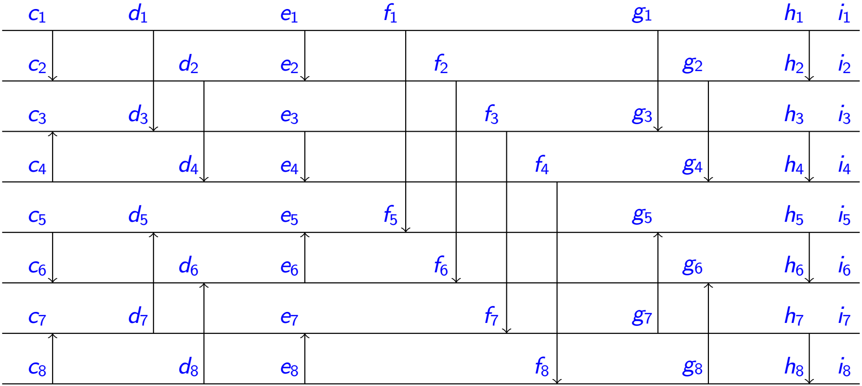


$$
\begin{aligned}
&c_1?(x_1).c_2?(x_2).\\
&\qquad \text{if } x_1 \le x_2 \text{ then } \big(c_3!\langle x_1\rangle \parallel c_4!\langle x_2\rangle\big)\\
&\qquad\qquad\qquad\qquad \text{else } \big(c_3!\langle x_2\rangle \parallel c_4!\langle x_1\rangle\big)
\end{aligned}
$$

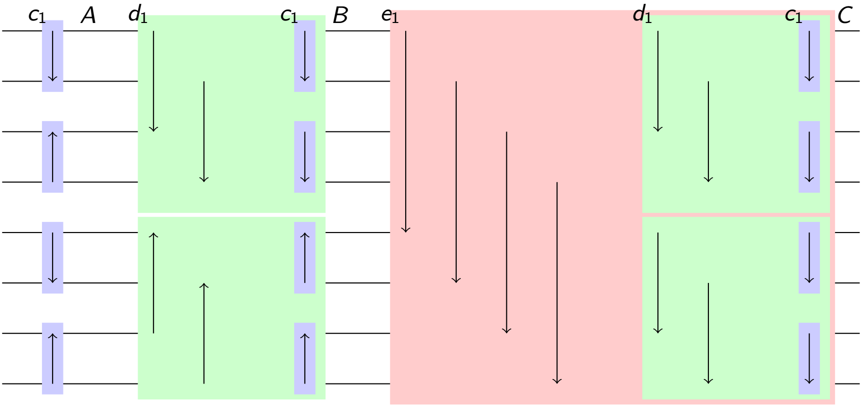
Regular Pattern in SNs



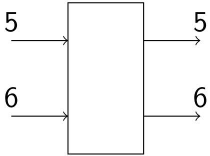
Naïve Solution for SNs

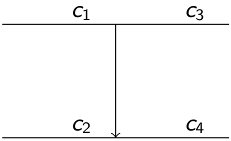


Horizontal Reuse in SNs

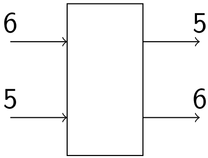


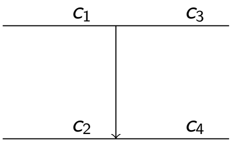
Vertical Reuse in SNs



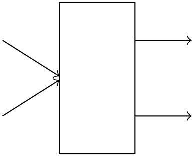


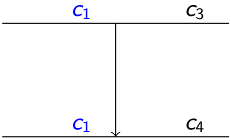
Vertical Reuse in SNs





Vertical Reuse in SNs





Contributions

- Separation-Based Logic for Stable Process for the pre- and postconditions
- *Separation-based* Proof System for *Message-Passing*, *Deterministic* and *Terminating* Programs
- Proof of Soundness of Proof System
- Message-passing Implementation of Sorting Network resorting to resource reuse
- Proof of Correctness for the Implementation
- Preliminary Design of Second Proof System where channels can be shared
- An innovative Proof Technique for proving Soundness

Future Work

- More Resource Reuse Pattern
- Enhanced Languages
  - Name-Passing Channels
  - Scoping Construct
- Logical Framework Improvement

Bibliography

Adrian Francalanza, Julian Rathke, and Vladimiro Sassone.
Permission-based separation logic for message-passing concurrency.
Logical Methods in Computer Science, 7(3), 2011.

Donald E. Knuth.
The art of computer programming, volume 3: (2nd ed.) sorting and searching.
Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.

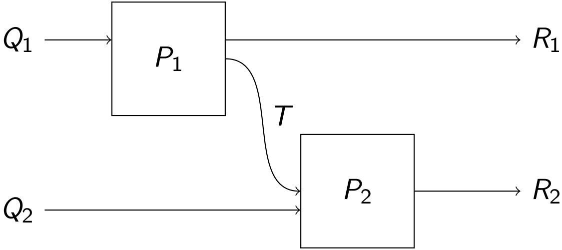
Peter W. O'Hearn.
Resources, concurrency, and local reasoning.
Theor. Comput. Sci., 375(1-3):271–307, 2007.

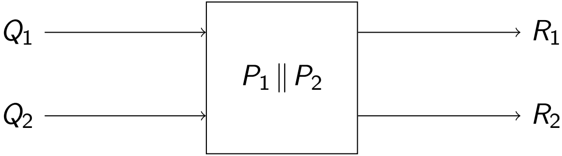
John C. Reynolds.
Separation logic: A logic for shared mutable data structures.
In *LICS*, pages 55–74, 2002.

Conclusion Remarks

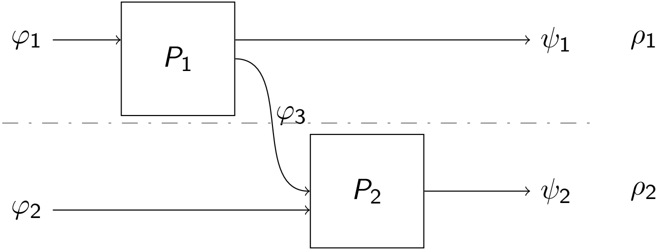
Questions ?

The LPAR Rule



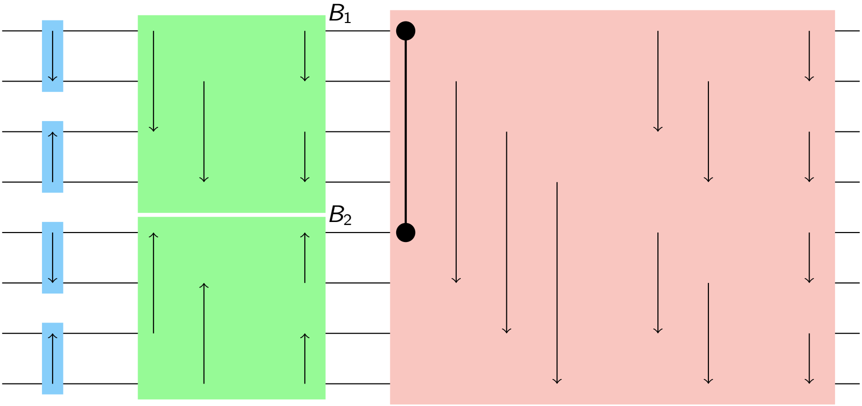


The lPar Rule



$$\frac{\begin{array}{ll}\Gamma_i\,;\ \Gamma_o \backslash \Gamma; b \vdash \{\varphi_1\}\ P_1\ \{\varphi_3 \star \psi_1\} : \rho_1 & \textbf{dom}(\Gamma) \subseteq \mathrm{fn}(\varphi_3) \\ \Gamma_i \backslash \Gamma\ ;\ \Gamma_o;\ b \vdash \{\varphi_2 \star \varphi_3\}\ P_2\ \{\psi_2\} : \rho_2 & \varphi_2 \perp \varphi_3 \quad \psi_1 \perp \psi_2\end{array}}{\Gamma_i; \Gamma_o; b \vdash \{\varphi_1 \star \varphi_2\}\ P_1 \parallel P_2\ \{\psi_1 \star \psi_2\} : \rho_1 \uplus \rho_2}$$

Vertical Reuse - SN



Multiple Sender and Single Receiver Checklist

$$c!\langle 5\rangle \parallel c!\langle 6\rangle \parallel c?(x).(c?(y).d!\langle x+y\rangle)$$

- Permissions Analysis
  - Frozen Permissions
  - Permission Bags
- Data Analysis
  - Number of I/O operations
  - Operation performed on the Data
  - Frozen Data

Multiple Sender and Single Receiver Checklist

$$c!\langle 7\rangle \parallel c!\langle 5\rangle \parallel c!\langle 6\rangle \parallel c?(x).(c?(y).d!\langle x+y\rangle)$$

- Permissions Analysis
  - Frozen Permissions
  - Permission Bags
- Data Analysis
  - Number of I/O operations
  - Operation performed on the Data
  - Frozen Data

Multiple Sender and Single Receiver Checklist

$$c!\langle 5\rangle \parallel c!\langle 6\rangle \parallel c?(x).(c?(y).d!\langle x-y\rangle)$$

- Permissions Analysis
  - Frozen Permissions
  - Permission Bags
- Data Analysis
  - Number of I/O operations
  - Operation performed on the Data
  - Frozen Data

Multiple Sender and Single Receiver Checklist

$$c!\langle 5\rangle \parallel c!\langle 6\rangle \parallel c?(x).(d!\langle x\rangle \parallel c?(y).d!\langle x+y+y\rangle)$$

- Permissions Analysis
  - Frozen Permissions
  - Permission Bags
- Data Analysis
  - Number of I/O operations
  - Operation performed on the Data
  - Frozen Data

The $\textsc{lNil}$ Rule

$$\textsc{lNil}\quad \frac{\mathrm{fn}(\varphi) \subseteq \mathbf{dom}(\Gamma_i \cap \Gamma_o)}{\Gamma_i; \Gamma_o; b \vdash \{\varphi\}\ \mathsf{nil}\ \{\varphi\} : \rho}$$

$$\{c\langle 5\rangle\}\ \mathsf{nil} \parallel c?(x).c!\langle e\rangle\ \{c\langle 5\rangle\}$$

Nested Permission Environment Update

$$\frac{\begin{array}{cc} \Gamma_i \,;\, \Gamma_o \backslash \Gamma; b \vdash \{\varphi_1\}\ P_1\ \{\varphi_3 \star \psi_1\} : \rho_1 & \mathbf{dom}(\Gamma) \subseteq \mathrm{fn}(\varphi_3) \\ \Gamma_i \backslash \Gamma \,;\, \Gamma_o;\ b \vdash \{\varphi_2 \star \varphi_3\}\ P_2\ \{\psi_2\} : \rho_2 & \varphi_2 \perp \varphi_3 \quad \psi_1 \perp \psi_2 \end{array}}{\Gamma_i; \Gamma_o; b \vdash \{\varphi_1 \star \varphi_2\}\ P_1 \parallel P_2\ \{\psi_1 \star \psi_2\} : \rho_1 \uplus \rho_2}$$

$$\{c\langle 5\rangle\}\ c?(x).(c! \parallel d!) \parallel c?(x).d?(y).c!\langle 5\rangle\ \{c\langle 5\rangle\}$$

Changes from [FRS11]

- Logical Formula Satisfaction
- Proof of Soundness - from 2 tier to 1 tier
- Removed the Confined Processes Semantics – permission describe the sequent's footprint rather then the process's

Sequent Definition

$$\vdash \quad \{\varphi\} \quad P \quad \{\psi\}$$

$$P, Q \triangleq \mathsf{nil} \mid c?(x).P \mid c!\langle e\rangle \mid P \parallel Q \mid \mathsf{if}\ b\ \mathsf{then}\ P\ \mathsf{else}\ Q \mid f(\vec{x})$$

$$\varphi, \psi \triangleq \mathbf{emp} \mid \mathbf{blk}(c) \mid c\langle e\rangle \mid \varphi \star \psi$$

Sequent Definition

$$\vdash \quad \{\varphi\} \qquad P \qquad\qquad \{\psi\} \qquad\qquad : \rho$$

E.g., $\quad \{c\uparrow, d\downarrow\}$

Sequent Definition

$$b \quad \vdash \quad \{\varphi\} \quad P \quad \{\psi\} \quad : \rho$$

E.g., $\quad x = y + 1 \vdash \{c\langle x\rangle\} \;\; P \;\; \{c\langle y\rangle\}$

Sequent Definition

$$\Gamma_i \quad ; \quad \Gamma_o \quad ; \quad b \quad \vdash \quad \{\varphi\} \quad P \quad \{\psi\} \quad : \rho$$

$$E.g., c : \{c \uparrow, d \downarrow\}$$

$$E.g., \begin{array}{rcl} \Gamma_i & = & c : \{c \uparrow, d \downarrow\} \\ \Gamma_o & = & c : \{c \uparrow, e \uparrow\} \end{array}$$

Logical Formula Satisfaction

$$
\begin{array}{lll}
\Gamma\,,\ P\,,\ \mu \vDash \textbf{emp} & \text{iff} & P \equiv \mathsf{nil} \\[1em]
\Gamma\,,\ P\,,\ \mu \vDash c\langle e\rangle & \text{iff} & P \equiv c!\langle e'\rangle \text{ and } e \Downarrow v, e' \Downarrow v \text{ and } \Gamma(c) \subseteq \mu \\[1em]
\Gamma\,,\ P\,,\ \mu \vDash \varphi_1 \star \varphi_2 & \text{iff} & P \equiv P_1 \,\|\, P_2 \text{ and } \Gamma\,,\ P_1\,,\ \mu_1 \vDash \varphi_1 \text{ and} \\
 & & \Gamma\,,\ P_2\,,\ \mu_2 \vDash \varphi_2 \text{ and } \mu = \mu_1 \uplus \mu_2 \\[1em]
\Gamma\,,\ P\,,\ \mu \vDash \textbf{blk}(c) & \text{iff} & P \equiv c?(x).P' \text{ and } c \in \textbf{dom}(\Gamma) \text{ and } c \downarrow \mu
\end{array}
$$

Semantic Definition

$$\Gamma_{in}; \Gamma_{out}; b \vdash \{\varphi\} \;\; P \;\; \{\psi\} : \rho$$

implies

$$\Gamma_{in}; \Gamma_{out}; b \models \{\varphi\} \;\; P \;\; \{\psi\} : \rho$$

Semantic Definition

$$\Gamma_{in}; \Gamma_{out}; b \vdash \{\varphi\}\ \ P\ \ \{\psi\} : \rho$$

implies

$$\forall \sigma, Q, \mu. \quad \Gamma_{in}, Q\sigma, \mu \vDash \varphi\sigma \text{ and } \rho \perp \mu \text{ and } b\sigma \Downarrow \mathsf{tt}$$

$$\text{implies} \quad (P \parallel Q)\sigma \Downarrow R\sigma \text{ and } \Gamma_{out}, R\sigma, \mu \uplus \rho \vDash \psi\sigma$$

Semantic Definition

$$\Gamma_{in}; \Gamma_{out}; b \vdash \{\varphi\} \;\; P \;\; \{\psi\} : \rho$$

implies

$$\forall \sigma, Q, \mu. \quad \Gamma_{in}, Q\sigma, \mu \vDash \varphi\sigma \text{ and } \rho \perp \mu \text{ and } b\sigma \Downarrow \mathsf{tt}$$

implies $\quad (P \parallel Q)\sigma \Downarrow R\sigma \text{ and } \Gamma_{out}, R\sigma, \mu \uplus \rho \vDash \psi\sigma$

Semantic Definition

$$\Gamma_{in}; \Gamma_{out}; b \vdash \{\varphi\}\ \ P\ \ \{\psi\} : \rho$$

implies

$$\forall \sigma, Q, \mu. \quad \Gamma_{in}, Q\sigma, \mu \vDash \varphi\sigma \text{ and } \rho \perp \mu \text{ and } b\sigma \Downarrow \mathsf{tt}$$

implies $\quad (P \parallel Q)\sigma \Downarrow R\sigma$ and $\Gamma_{out}, R\sigma, \mu \uplus \rho \vDash \psi\sigma$

Semantic Definition

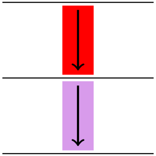
$$\Gamma_{in}; \Gamma_{out}; b \vdash \{\varphi\}\;\; P \;\;\{\psi\} : \rho$$

implies

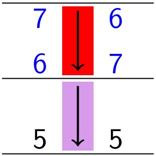
$$\forall \sigma, Q, \mu. \quad \Gamma_{in}, Q\sigma, \mu \vDash \varphi\sigma \text{ and } \rho \perp \mu \text{ and } b\sigma \Downarrow \mathsf{tt}$$

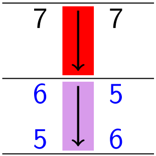
implies $\quad (P \,\|\, Q)\sigma \Downarrow R\sigma$ and $\Gamma_{out}, R\sigma, \mu \uplus \rho \vDash \psi\sigma$

Race Conditions in SNs

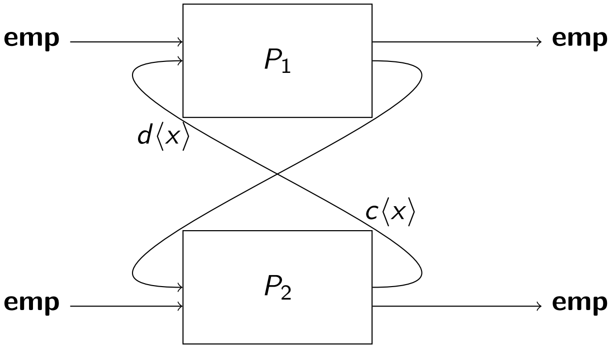


Race Conditions in SNs





Deadlocks



$$c?(x).d!\langle x\rangle \parallel d?(y).c!\langle y\rangle$$