Separation-Based Reasoning for Deterministic Channel-Passing Concurrent Programs

Aimee Borda

December 18, 2013
Table of Content

background
  Compositional Reasoning
  Separation Logic

Technical Development
  Resource Reuse
  Research Time-Line

Case-Study
  Overview
  Resource Reuse Patterns

Conclusion
  Future Work & Contributions
  Appendix
Compositional (localized) Proof Systems

A

B
Compositional (localized) Proof Systems
Compositional (localized) Proof Systems
Localized Reasoning

\[ \text{SUM}(l) = \text{SUM}(l_1) + \text{SUM}(l_2) \]
Localized Reasoning

\[ \text{SUM}(l) = \text{SUM}(l_1) + \text{SUM}(l_2) \]

\[ l = l_1 \cdot l_2 \]
Localized Reasoning

$$\text{SUM}(l) = \text{SUM}(l_1) + \text{SUM}(l_2)$$

$$l = l_1 \cdot l_2$$

$$r_1 \cap r_2 = \emptyset$$
Separation Logic [Rey02]
Separation Logic [Rey02]
Resource Transfer [O’H07]
Separation-Based Reasoning for Message Passing Programs [FRS11]
Communication Channels as Synchronization Mechanism
Channel Reuse - Dynamic Resource Transfer

\[
\{r_1, r_2, r_3\} \\
\{r_4, r_5\}
\]
Multiple-Sender and Single Receiver Pattern
Multiple-Sender and Single Receiver Pattern

\[
\text{SUM} = 11 \\
5 \quad 6
\]

\[
\text{DIFF} = 1 \text{ or } -1 \\
5 \quad 6
\]
Semantic Satisfaction

\[ \Gamma_{in}; \Gamma_{out}; b \vdash \{ \varphi \} \quad P \quad \{ \psi \} : \rho \]

implies

\[ \Gamma_{in}; \Gamma_{out}; b \models \{ \varphi \} \quad P \quad \{ \psi \} : \rho \]
Technical Development Timeline

FRS11 - Channel Reuse - Multiple-Sender Single-Receiver - Goal
Proof of Soundness

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

\[ [P]_\rho \text{ is deterministic} \]

(Data Analysis)

(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

\[
c_1 ? (x_1) . c_2 ? (x_2).
\]

if \( x_1 \leq x_2 \) then \((c_3 ! \langle x_1 \rangle \parallel c_4 ! \langle x_2 \rangle)\)
else \((c_3 ! \langle x_2 \rangle \parallel c_4 ! \langle x_1 \rangle)\)
Regular Pattern in SNs
Naïve Solution for SNs

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1$</td>
<td>$d_1$</td>
<td>$e_1$</td>
<td>$f_1$</td>
<td>$g_1$</td>
<td>$h_1$</td>
<td>$i_1$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_2$</td>
<td>$d_2$</td>
<td>$e_2$</td>
<td>$f_2$</td>
<td>$g_2$</td>
<td>$h_2$</td>
<td>$i_2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_3$</td>
<td>$d_3$</td>
<td>$e_3$</td>
<td>$f_3$</td>
<td>$g_3$</td>
<td>$h_3$</td>
<td>$i_3$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_4$</td>
<td>$d_4$</td>
<td>$e_4$</td>
<td>$f_4$</td>
<td>$g_4$</td>
<td>$h_4$</td>
<td>$i_4$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_5$</td>
<td>$d_5$</td>
<td>$e_5$</td>
<td>$f_5$</td>
<td>$g_5$</td>
<td>$h_5$</td>
<td>$i_5$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_6$</td>
<td>$d_6$</td>
<td>$e_6$</td>
<td>$f_6$</td>
<td>$g_6$</td>
<td>$h_6$</td>
<td>$i_6$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_7$</td>
<td>$d_7$</td>
<td>$e_7$</td>
<td>$f_7$</td>
<td>$g_7$</td>
<td>$h_7$</td>
<td>$i_7$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_8$</td>
<td>$d_8$</td>
<td>$e_8$</td>
<td>$f_8$</td>
<td>$g_8$</td>
<td>$h_8$</td>
<td>$i_8$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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


Conclusion Remarks

Questions ?
The \textsc{LPar} Rule

\[
\begin{align*}
P_1 &\parallel P_2 \\
Q_1 &\rightarrow P_1 & R_1 \\
Q_2 &\rightarrow P_2 & R_2 \\
P_1 &\parallel P_2 \\
Q_1 &\rightarrow P_1 & R_1 \\
Q_2 &\rightarrow P_2 & R_2
\end{align*}
\]
The LPAR Rule

\[ \varphi_1 \quad P_1 \quad \psi_1 \quad \rho_1 \]

\[ \varphi_2 \quad P_2 \quad \psi_2 \quad \rho_2 \]

\[ \Gamma_i ; \Gamma_o ; b \vdash \{ \varphi_1 \} \quad P_1 \quad \{ \varphi_3 \ast \psi_1 \} : \rho_1 \quad \text{dom}(\Gamma) \subseteq \text{fn}(\varphi_3) \]

\[ \Gamma_i \backslash \Gamma ; \Gamma_o ; b \vdash \{ \varphi_2 \ast \varphi_3 \} \quad P_2 \quad \{ \psi_2 \} : \rho_2 \quad \varphi_2 \perp \varphi_3 \quad \psi_1 \perp \psi_2 \]

\[ \Gamma_i ; \Gamma_o ; b \vdash \{ \varphi_1 \ast \varphi_2 \} \quad P_1 \parallel P_2 \quad \{ \psi_1 \ast \psi_2 \} : \rho_1 \uplus \rho_2 \]
Vertical Reuse - SN
Multiple Sender and Single Receiver Checklist

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

- 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 LNIL Rule

\[
\text{LNIL} \quad \frac{\text{fn}(\varphi) \subseteq \text{dom}(\Gamma_i \cap \Gamma_o)}{\Gamma_i; \Gamma_o; b \vdash \{\varphi\} \; \text{nil} \; \{\varphi\} : \rho}
\]

\[
\{c\langle 5\rangle\} \; \text{nil} \parallel c?(x).c!\langle e\rangle \; \{c\langle 5\rangle\}
\]
Nested Permission Environment Update

\[
\Gamma_i; \Gamma_o \setminus \Gamma; b \vdash \{\varphi_1\} \quad P_1 \quad \{\varphi_3 \star \psi_1\} : \rho_1 \quad \text{dom}(\Gamma) \subseteq \text{fn}(\varphi_3)
\]

\[
\Gamma_i \setminus \Gamma; \Gamma_o; b \vdash \{\varphi_2 \star \varphi_3\} \quad P_2 \quad \{\psi_2\} : \rho_2 \quad \varphi_2 \perp \varphi_3 \quad \psi_1 \perp \psi_2
\]

\[
\Gamma_i; \Gamma_o; b \vdash \{\varphi_1 \star \varphi_2\} \quad P_1 \parallel P_2 \quad \{\psi_1 \star \psi_2\} : \rho_1 \uplus \rho_2
\]

\[
\{\langle 5 \rangle\} \quad c?(x).(c! \parallel \!d!) \parallel c?(x).d?(y).c!\langle 5 \rangle \quad \{\langle 5 \rangle\}
\]
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 than the process’s
Sequent Definition

\[ \vdash \{ \varphi \} \quad P \quad \{ \psi \} \]

\[ P, Q \triangleq \text{nil} \mid c?(x).P \mid c!(e) \mid P \parallel Q \mid \text{if } b \text{ then } P \text{ else } Q \mid f(\vec{x}) \]

\[ \varphi, \psi \triangleq \text{emp} \mid \text{blk}(c) \mid c(e) \mid \varphi \star \psi \]
Sequent Definition

\[ \Gamma \vdash \{ \varphi \} \quad \quad P \quad \{ \psi \} \quad : \quad \rho \]

E.g., \( \{ c \uparrow, d \downarrow \} \)
Sequent Definition

\[ b \vdash \{ \varphi \} \quad P \quad \{ \psi \} : \rho \]

E.g., \( x = y + 1 \vdash \{ c\langle x \rangle \} \quad P \quad \{ c\langle y \rangle \} \)
\[ \Gamma_i ; \Gamma_o ; b \vdash \{ \varphi \} \quad P \quad \{ \psi \} : \rho \]

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

E.g., \( \Gamma_i = c : \{ c \uparrow, d \downarrow \} \)
\( \Gamma_o = c : \{ c \uparrow, e \uparrow \} \)
Logical Formula Satisfaction

\[ \Gamma, P, \mu \models \text{emp} \quad \text{iff} \quad P \equiv \text{nil} \]

\[ \Gamma, P, \mu \models c\langle e \rangle \quad \text{iff} \quad P \equiv c!(e') \text{ and } e \downarrow v, e' \downarrow v \text{ and } \Gamma(c) \subseteq \mu \]

\[ \Gamma, P, \mu \models \varphi_1 \star \varphi_2 \quad \text{iff} \quad P \equiv P_1 \parallel P_2 \text{ and } \Gamma, P_1, \mu_1 \models \varphi_1 \text{ and } \Gamma, P_2, \mu_2 \models \varphi_2 \text{ and } \mu = \mu_1 \cup \mu_2 \]

\[ \Gamma, P, \mu \models \text{blk}(c) \quad \text{iff} \quad P \equiv c?(x).P' \text{ and } c \in \text{dom}(\Gamma) \text{ and } c \downarrow \mu \]
Semantic Definition

\[ \Gamma_{in}; \Gamma_{out}; b \vdash \{ \varphi \} \quad P \quad \{ \psi \} : \rho \]

implies

\[ \Gamma_{in}; \Gamma_{out}; b \models \{ \varphi \} \quad P \quad \{ \psi \} : \rho \]
Semantic Definition

\[ \Gamma_{in}; \Gamma_{out}; b \vdash \{ \varphi \} \quad P \quad \{ \psi \} : \rho \]

implies

\[ \forall \sigma, Q, \mu. \quad \Gamma_{in}, Q\sigma, \mu \models \varphi\sigma \text{ and } \rho \perp \mu \text{ and } b\sigma \downarrow \text{tt} \]

implies

\[ (P \parallel Q)\sigma \downarrow R\sigma \text{ and } \Gamma_{out}, R\sigma, \mu \uplus \rho \models \psi\sigma \]
**Semantic Definition**

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

implies

\[ \forall \sigma, Q, \mu. \ \Gamma_{in}, Q\sigma, \mu \vdash \varphi\sigma \ \text{and} \ \rho \perp \mu \ \text{and} \ b\sigma \Downarrow \text{tt} \]

implies

\[ (P \parallel Q)\sigma \Downarrow R\sigma \ \text{and} \ \Gamma_{out}, R\sigma, \mu \uplus \rho \vdash \psi\sigma \]
Semantic Definition

\[
\Gamma_{\text{in}}; \Gamma_{\text{out}}; b \vdash \{ \varphi \} \quad P \quad \{ \psi \} : \rho
\]

implies

\[
\forall \sigma, Q, \mu. \quad \Gamma_{\text{in}}, Q\sigma, \mu \vdash \varphi\sigma \quad \text{and} \quad \rho \perp \mu \quad \text{and} \quad b\sigma \Downarrow \text{tt}
\]

implies

\[
(P \parallel Q)\sigma \Downarrow R\sigma \quad \text{and} \quad \Gamma_{\text{out}}, R\sigma, \mu \uplus \rho \vdash \psi\sigma
\]
Semantic Definition

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

\text{implies}

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

\text{implies} \quad (P \parallel Q)\sigma \Downarrow R\sigma \text{ and } \Gamma_{out}, R\sigma, \mu \uplus \rho \models \psi\sigma
Race Conditions in SNs
Race Conditions in SNs

\[ \begin{array}{c|c|c}
7 & 6 & 5 \\
6 & 7 & 5 \\
5 & 5 & \\
\end{array} \] 

\[ \begin{array}{c|c|c}
7 & 7 & \\
6 & 5 & \\
5 & 6 & \\
\end{array} \]
Deadlocks

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