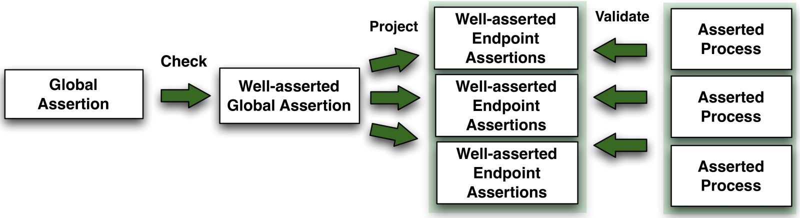
Amending Contracts for Choreographies

Laura Bocchi **Julien Lange** Emilio Tuosto

Department of Computer Science, University of Leicester

ICE'11
June 2011

Introduction



- Well-Assertedness
- Difficult to attain by hand
- Three algorithms:
  - Strengthening
  - Propagation
  - Lifting

cf. Honda et al. in POPL 2008
Bocchi et al. in CONCUR 2010

Background - Global Assertion

An example of global assertion:

$$
\begin{array}{ll}
\mathtt{A} \to \mathtt{B} : \{x \mid x > 0\}. & \\
\mathtt{B} \to \mathtt{C} : & \{x \geq 5\}\ \mathrm{gt} : \quad \mathtt{C} \to \mathtt{B} : \{y \mid \exists u.y = u \times 2\}, \\
 & \{x < 5\}\ \mathrm{le} : \quad \mathtt{C} \to \mathtt{B} : \{z \mid \exists u.z = u \times 2 + 1\}
\end{array}
$$

- Logic needs to be decidable
- Global assertions have to be closed
- ... and they must obey two conditions

Background - Global Assertion

An example of global assertion:

$$\mathtt{A} \to \mathtt{B} : \{x \mid x > 0\}$$

$$\mathtt{B} \to \mathtt{C}$$

$$\{x \geq 5\}\ \mathrm{gt} \qquad\qquad \{x < 5\}\ \mathrm{le}$$

$$\mathtt{C} \to \mathtt{B} : \{y \mid \exists u.y = u \times 2\} \quad \mathtt{C} \to \mathtt{B} : \{z \mid \exists u.z = u \times 2 + 1\}$$

- Logic needs to be decidable
- Global assertions have to be closed
- ... and they must obey two conditions

Background - Well-Assertedness

- **History Sensitivity (HS):** a participant having an obligation has enough information for choosing a set of values that guarantees it,

$$
\begin{aligned}
&\texttt{Alice} \rightarrow \texttt{Bob} : \{x \mid \mathsf{true}\}.\\
&\texttt{Bob} \rightarrow \texttt{Carol} : \{y \mid \mathsf{true}\}.\\
&\texttt{Carol} \rightarrow \texttt{Alice} : \{z \mid z > x\}
\end{aligned}
$$

Background - Well-Assertedness

- **History Sensitivity (HS):** a participant having an obligation has enough information for choosing a set of values that guarantees it,

$$\begin{aligned}&\texttt{Alice} \rightarrow \texttt{Bob} : \{x \mid \mathsf{true}\}.\\&\texttt{Bob} \rightarrow \texttt{Carol} : \{y \mid \mathsf{true}\}.\\&\texttt{Carol} \rightarrow \texttt{Alice} : \{z \mid z > x\}\end{aligned}$$

- **Temporal Satisfiability (TS):** the values sent in each interaction do not make predicates of future interactions unsatisfiable,

$$\begin{aligned}&\texttt{Alice} \rightarrow \texttt{Bob} : \{x \mid x > 10\}.\\&\texttt{Bob} \rightarrow \texttt{Alice} : \{y \mid y < x \wedge y > 10\}\end{aligned}$$

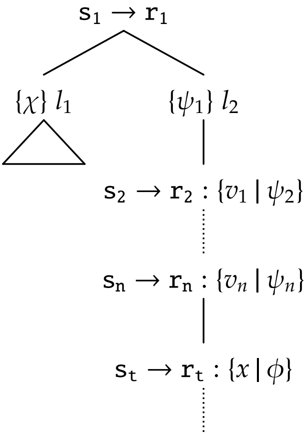
Outline

History Sensitivity

A participant having an obligation has enough information for choosing a set of values that guarantees it.

- A participant *knows* a variable if s/he either sends or receives it,
- a sender *must know* all the variables which appear in a predicate s/he guarantees,
- otherwise, **history sensitivity** is violated!

Strengthening



- Assume no HS problems in ψs,
- $\mathtt{s}_\mathtt{t}$ does not know v (appearing in ϕ).
- Find a variable v_i:
  - v_i is known to $\mathtt{s}_\mathtt{t}$
  - $(\psi_1 \wedge \psi_2 \wedge \ldots \psi_n) \wedge \phi[v_i/v] \Rightarrow \phi$ holds?
- Replace ϕ by $\phi[v_i/v]$

Strengthening: Example

$$
\begin{array}{l}
\mathtt{A} \longrightarrow \mathtt{B} : \{v_0 \mid v_0 \leq 50\}. \\
\mathtt{B} \longrightarrow \mathtt{C} : \{v_1 \mid v_1 > v_0\}. \\
\mathtt{C} \longrightarrow \mathtt{D} : \{v_2\ v_3 \mid v_2 > v_1 \wedge v_3 \leq 70\}. \\
\mathtt{C} \longrightarrow \mathtt{D} : \quad \begin{array}{ll} \{v_1 \geq 0\}\ \mathsf{pos} : & \mathtt{D} \longrightarrow \mathtt{C} : \{v_4 \mid v_4 > v_0\}, \\ \{v_1 < 0\}\ \mathsf{neg} : & \mathtt{D} \longrightarrow \mathtt{C} : \{v_5 \mid v_5 < v_0\} \end{array}
\end{array}
$$

Strengthening: Example

$\mathtt{A} \longrightarrow \mathtt{B} : \{v_0 \mid v_0 \leq 50\}.$
$\mathtt{B} \longrightarrow \mathtt{C} : \{v_1 \mid v_1 > v_0\}.$
$\mathtt{C} \longrightarrow \mathtt{D} : \{v_2\ v_3 \mid v_2 > v_1 \wedge v_3 \leq 70\}.$
$\mathtt{C} \longrightarrow \mathtt{D} : \quad \{v_1 \geq 0\}\ \mathsf{pos} : \quad \mathtt{D} \longrightarrow \mathtt{C} : \{v_4 \mid v_4 > v_0\},$
$\qquad\qquad\quad \{v_1 < 0\}\ \mathsf{neg} : \quad \mathtt{D} \longrightarrow \mathtt{C} : \{v_5 \mid v_5 < v_0\}$

Strengthening: Example

$$
\begin{array}{l}
\mathtt{A} \rightarrow \mathtt{B} : \{v_0 \mid v_0 \leq 50\}. \\
\mathtt{B} \rightarrow \mathtt{C} : \{v_1 \mid v_1 > v_0\}. \\
\mathtt{C} \rightarrow \mathtt{D} : \{v_2\ v_3 \mid v_2 > v_1 \wedge v_3 \leq 70\}. \\
\mathtt{C} \rightarrow \mathtt{D} : \quad \{v_1 \geq 0\}\ \mathsf{pos} : \quad \mathtt{D} \rightarrow \mathtt{C} : \{v_4 \mid v_4 > v_0\}, \\
\qquad\qquad\quad \{v_1 < 0\}\ \mathsf{neg} : \quad \mathtt{D} \rightarrow \mathtt{C} : \{v_5 \mid v_5 < v_0\}
\end{array}
$$

Can we find v_i such that $\mathtt{D}$ knows v_i and

$$
\underbrace{(v_0 \leq 50 \wedge v_1 > v_0 \wedge v_2 > v_1 \wedge v_3 \leq 70)}_{\psi_1 \wedge \psi_2 \wedge \ldots \psi_n} \wedge \underbrace{(v_4 > v_i)}_{\phi[v_i/v]} \Rightarrow \underbrace{(v_4 > v_0)}_{\phi}
$$

holds?

Strengthening: Example

$$
\begin{array}{l}
\mathtt{A} \longrightarrow \mathtt{B} : \{v_0 \mid v_0 \leq 50\}. \\
\mathtt{B} \longrightarrow \mathtt{C} : \{v_1 \mid v_1 > v_0\}. \\
\mathtt{C} \longrightarrow \mathtt{D} : \{v_2\ v_3 \mid v_2 > v_1 \wedge v_3 \leq 70\}. \\
\mathtt{C} \longrightarrow \mathtt{D} : \quad \begin{array}{ll} \{v_1 \geq 0\}\ \mathsf{pos}: & \mathtt{D} \rightarrow \mathtt{C} : \{v_4 \mid v_4 > {\color{red} v_0}\}, \\ \{v_1 < 0\}\ \mathsf{neg}: & \mathtt{D} \rightarrow \mathtt{C} : \{v_5 \mid v_5 < {\color{red} v_0}\} \end{array}
\end{array}
$$

Can we find v_i such that D knows v_i and

$$
\underbrace{(v_0 \leq 50 \wedge v_1 > v_0 \wedge v_2 > v_1 \wedge v_3 \leq 70)}_{\psi_1 \wedge \psi_2 \wedge \ldots \psi_n} \wedge \underbrace{(v_4 > v_i)}_{\phi[v_i/v]} \Rightarrow \underbrace{(v_4 > v_0)}_{\phi}
$$

holds? Yes, D knows v_2 for which the equation above holds!

Strengthening: Example

$$\begin{array}{l}
\mathtt{A} \longrightarrow \mathtt{B} : \{v_0 \mid v_0 \leq 50\}. \\
\mathtt{B} \longrightarrow \mathtt{C} : \{v_1 \mid v_1 > v_0\}. \\
\mathtt{C} \longrightarrow \mathtt{D} : \{v_2\ v_3 \mid v_2 > v_1 \wedge v_3 \leq 70\}. \\
\mathtt{C} \longrightarrow \mathtt{D} : \quad \{v_1 \geq 0\}\ \mathsf{pos} : \quad \mathtt{D} \longrightarrow \mathtt{C} : \{v_4 \mid v_4 > {\color{blue}v_2}\}{\color{blue},} \\
\qquad\qquad\quad \{v_1 < 0\}\ \mathsf{neg} : \quad \mathtt{D} \longrightarrow \mathtt{C} : \{v_5 \mid v_5 < {\color{red}v_0}\}
\end{array}$$

Can we find v_i such that D knows v_i and

$$\underbrace{(v_0 \leq 50 \wedge v_1 > v_0 \wedge v_2 > v_1 \wedge v_3 \leq 70)}_{\psi_1 \wedge \psi_2 \wedge \ldots \psi_n} \wedge \underbrace{(v_4 > v_i)}_{\phi[v_i/v]} \Rightarrow \underbrace{(v_4 > v_0)}_{\phi}$$

holds? Yes, D knows v_2 for which the equation above holds!

Strengthening: Example

$$\begin{array}{l}
\mathtt{A} \longrightarrow \mathtt{B} : \{v_0 \mid v_0 \leq 50\}. \\
\mathtt{B} \longrightarrow \mathtt{C} : \{v_1 \mid v_1 > v_0\}. \\
\mathtt{C} \longrightarrow \mathtt{D} : \{v_2\ v_3 \mid v_2 > v_1 \wedge v_3 \leq 70\}. \\
\mathtt{C} \longrightarrow \mathtt{D} : \quad \{v_1 \geq 0\}\ \mathsf{pos} : \quad \mathtt{D} \longrightarrow \mathtt{C} : \{v_4 \mid v_4 > {\color{blue} v_2}\}{\color{blue},} \\
\qquad\qquad\quad \{v_1 < 0\}\ \mathsf{neg} : \quad \mathtt{D} \longrightarrow \mathtt{C} : \{v_5 \mid v_5 < {\color{red} v_0}\}
\end{array}$$

Can we find v_i such that $\mathtt{D}$ knows v_i and

$$\underbrace{(v_0 \leq 50 \wedge v_1 > v_0 \wedge v_2 > v_1 \wedge v_3 \leq 70)}_{\psi_1 \wedge \psi_2 \wedge \ldots \psi_n} \wedge \underbrace{(v_5 < v_i)}_{\phi[v_i/v]} \Rightarrow \underbrace{(v_5 < v_0)}_{\phi}$$

holds?

Strengthening: Example

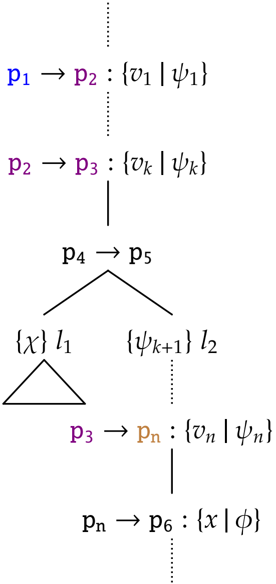
$$
\begin{array}{l}
\texttt{A} \rightarrow \texttt{B} : \{v_0 \mid v_0 \leq 50\}. \\
\texttt{B} \rightarrow \texttt{C} : \{v_1 \mid v_1 > v_0\}. \\
\texttt{C} \rightarrow \texttt{D} : \{v_2\ v_3 \mid v_2 > v_1 \land v_3 \leq 70\}. \\
\texttt{C} \rightarrow \texttt{D} : \quad \{v_1 \geq 0\}\ \mathsf{pos} : \quad \texttt{D} \rightarrow \texttt{C} : \{v_4 \mid v_4 > v_2\}, \\
\qquad\qquad\ \ \{v_1 < 0\}\ \mathsf{neg} : \quad \texttt{D} \rightarrow \texttt{C} : \{v_5 \mid v_5 < v_0\}
\end{array}
$$

Can we find v_i such that `D` knows v_i and

$$
\underbrace{(v_0 \leq 50 \land v_1 > v_0 \land v_2 > v_1 \land v_3 \leq 70)}_{\psi_1 \land \psi_2 \land \ldots \psi_n} \land \underbrace{(v_5 < v_i)}_{\phi[v_i/v]} \Rightarrow \underbrace{(v_5 < v_0)}_{\phi}
$$

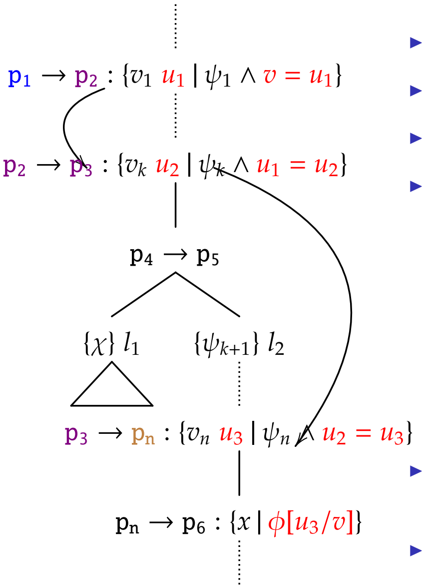
holds? No, `D` doesn't know any variable for which the equation above holds...

Propagation



- Assume no HS problems in ψs,
- p_n does not know v (appearing in ϕ),
- p_1 knows v.
- Find a chain of interactions:
  - the sender of the first interaction knows v,
  - the receiver of the last interaction is p_n,
  - the sender of a node is the receiver of the previous node in the chain.

Propagation



- Assume no HS problems in ψs,
- p_n does not know v (appearing in ϕ),
- p_1 knows v.
- Find a chain of interactions:
  - the sender of the first interaction knows v,
  - the receiver of the last interaction is p_n,
  - the sender of a node is the receiver of the previous node in the chain.
- "Propagate" v from the first node till the last node of the chain,
- ... and replace v in ϕ.

Propagation: Example

$$\texttt{A} \longrightarrow \texttt{B} : \{v_0 \mid v_0 \leq 50\}. \tag{1}$$

$$\texttt{B} \longrightarrow \texttt{C} : \{v_1 \mid v_1 > v_0\}. \tag{2}$$

$$\texttt{C} \longrightarrow \texttt{D} : \{v_2\ v_3 \mid v_2 > v_1 \wedge v_3 \leq 50\}. \tag{3}$$

$$\texttt{C} \longrightarrow \texttt{D} : \quad \{v_1 \geq 0\}\ \mathsf{pos} : \quad \texttt{D} \longrightarrow \texttt{C} : \{v_4 \mid v_4 > v_2\}, \tag{4}$$

$$\{v_1 < 0\}\ \mathsf{neg} : \quad \texttt{D} \longrightarrow \texttt{C} : \{v_5 \mid v_5 < {\color{red} v_0}\} \tag{5}$$

Propagation: Example

$$
\begin{aligned}
&\texttt{A} \longrightarrow \texttt{B} : \{v_0 \mid v_0 \le 50\}. && (1)\\
&\texttt{B} \longrightarrow \texttt{C} : \{v_1 \mid v_1 > v_0\}. && (2)\\
&\texttt{C} \longrightarrow \texttt{D} : \{v_2\ v_3 \mid v_2 > v_1 \wedge v_3 \le 50\}. && (3)\\
&\texttt{C} \longrightarrow \texttt{D} : \quad \{v_1 \ge 0\}\ \mathsf{pos} : \quad \texttt{D} \longrightarrow \texttt{C} : \{v_4 \mid v_4 > v_2\}, && (4)\\
&\qquad\qquad\quad \{v_1 < 0\}\ \mathsf{neg} : \quad \texttt{D} \longrightarrow \texttt{C} : \{v_5 \mid v_5 < {\color{red}v_0}\} && (5)
\end{aligned}
$$

Find a chain of *interactions* such that:

- the sender of the first interaction knows v_0, e.g. (1)
- the receiver of the last interaction is `D`, e.g. (3), and
- the sender for each interaction is the receiver of the previous interaction in the chain, e.g. (1) $\rightarrow$ (2) $\rightarrow$ (3)

Propagation: Example

$$
\begin{aligned}
&\mathtt{A} \longrightarrow \mathtt{B} : \{v_0 \mid v_0 \leq 50\}. && (1)\\
&\mathtt{B} \longrightarrow \mathtt{C} : \{v_1 \mid v_1 > v_0\}. && (2)\\
&\mathtt{C} \longrightarrow \mathtt{D} : \{v_2\ v_3 \mid v_2 > v_1 \wedge v_3 \leq 50\}. && (3)\\
&\mathtt{C} \longrightarrow \mathtt{D} : \quad \{v_1 \geq 0\}\ \textsf{pos} : \quad \mathtt{D} \longrightarrow \mathtt{C} : \{v_4 \mid v_4 > v_2\}, && (4)\\
&\qquad\qquad\quad \{v_1 < 0\}\ \textsf{neg} : \quad \mathtt{D} \longrightarrow \mathtt{C} : \{v_5 \mid v_5 < \color{red}{v_0}\} && (5)
\end{aligned}
$$

In the chain (1) $\rightarrow$ (2) $\rightarrow$ (3):

Propagation: Example

$$
\begin{aligned}
&\mathtt{A} \rightarrow \mathtt{B} : \{v_0\ u_1 \mid v_0 \leq 50 \wedge u_1 = v_0\}. && (1)\\
&\mathtt{B} \rightarrow \mathtt{C} : \{v_1 \mid v_1 > v_0\}. && (2)\\
&\mathtt{C} \rightarrow \mathtt{D} : \{v_2\ v_3 \mid v_2 > v_1 \wedge v_3 \leq 50\}. && (3)\\
&\mathtt{C} \rightarrow \mathtt{D} : \quad \{v_1 \geq 0\}\ \mathsf{pos} : \quad \mathtt{D} \rightarrow \mathtt{C} : \{v_4 \mid v_4 > v_2\}, && (4)\\
&\qquad\qquad\ \{v_1 < 0\}\ \mathsf{neg} : \quad \mathtt{D} \rightarrow \mathtt{C} : \{v_5 \mid v_5 < v_0\} && (5)
\end{aligned}
$$

In the chain (1) $\rightarrow$ (2) $\rightarrow$ (3):

- $u_1 = v_0$ is added to the predicate of the first interaction,

Propagation: Example

$$
\begin{aligned}
&\mathtt{A} \to \mathtt{B} : \{v_0\ u_1 \mid v_0 \leq 50 \wedge u_1 = v_0\}. && (1)\\
&\mathtt{B} \to \mathtt{C} : \{v_1\ u_2 \mid v_1 > v_0 \wedge u_2 = u_1\}. && (2)\\
&\mathtt{C} \to \mathtt{D} : \{v_2\ v_3\ u_3 \mid v_2 > v_1 \wedge v_3 \leq 50 \wedge u_3 = u_2\}. && (3)\\
&\mathtt{C} \to \mathtt{D} : \quad \{v_1 \geq 0\}\ \mathsf{pos} : \quad \mathtt{D} \to \mathtt{C} : \{v_4 \mid v_4 > v_2\}, && (4)\\
&\qquad\qquad\ \{v_1 < 0\}\ \mathsf{neg} : \quad \mathtt{D} \to \mathtt{C} : \{v_5 \mid v_5 < v_0\} && (5)
\end{aligned}
$$

In the chain (1) $\to$ (2) $\to$ (3):

- $u_1 = v_0$ is added to the predicate of the first interaction,
- $u_i = u_{i-1}$ is added the ith interaction's predicate $(1 < i \leq 3)$,

Propagation: Example

$$
\begin{aligned}
&\mathtt{A} \rightarrow \mathtt{B} : \{v_0\ u_1 \mid v_0 \leq 50 \wedge u_1 = v_0\}. && (1)\\
&\mathtt{B} \rightarrow \mathtt{C} : \{v_1\ u_2 \mid v_1 > v_0 \wedge u_2 = u_1\}. && (2)\\
&\mathtt{C} \rightarrow \mathtt{D} : \{v_2\ v_3\ u_3 \mid v_2 > v_1 \wedge v_3 \leq 50 \wedge u_3 = u_2\}. && (3)\\
&\mathtt{C} \rightarrow \mathtt{D} : \quad \{v_1 \geq 0\}\ \mathsf{pos} : \quad \mathtt{D} \rightarrow \mathtt{C} : \{v_4 \mid v_4 > v_2\}, && (4)\\
&\qquad\qquad\quad \{v_1 < 0\}\ \mathsf{neg} : \quad \mathtt{D} \rightarrow \mathtt{C} : \{v_5 \mid v_5 < u_3\} && (5)
\end{aligned}
$$

In the chain (1) → (2) → (3):

- $u_1 = v_0$ is added to the predicate of the first interaction,
- $u_i = u_{i-1}$ is added the ith interaction's predicate ($1 < i \leq 3$),
- u_3 replaces v_0 in the "problematic" interaction's predicate

Outline

Temporal Satisfiability

The values sent in each interaction do not make predicates of future interactions unsatisfiable.

- All the values satisfying the predicates before an interaction must allow to instantiate its interaction variables.
- At least one branch can be chosen, and each branch must satisfy temporal satisfiability.

Lifting: Example

$$\mathtt{A} \longrightarrow \mathtt{B} : \{x \mid x < 10\}. \tag{1}$$
$$\mathtt{A} \longrightarrow \mathtt{B} : \{y \mid y > 8\}. \tag{2}$$
$$\mathtt{B} \longrightarrow \mathtt{A} : \{z \mid x > z \wedge \; z > 6 \; \wedge \; y \neq z\} \tag{3}$$

Lifting: Example

$$\mathtt{A} \longrightarrow \mathtt{B} : \{x \mid x < 10\}. \tag{1}$$

$$\mathtt{A} \longrightarrow \mathtt{B} : \{y \mid y > 8\}. \tag{2}$$

$$\mathtt{B} \longrightarrow \mathtt{A} : \{z \mid x > z \wedge\ z > 6\ \wedge\ y \neq z\} \tag{3}$$

Lifting: Example

$$
\begin{aligned}
&\mathtt{A} \longrightarrow \mathtt{B} : \{x \mid x < 10\}. && (1)\\
&\mathtt{A} \longrightarrow \mathtt{B} : \{y \mid y > 8\}. && (2)\\
&\mathtt{B} \longrightarrow \mathtt{A} : \{z \mid x > z \wedge \ z > 6 \ \wedge \ y \neq z\} && (3)
\end{aligned}
$$

- Identify which part of the predicate is *in conflict* with the previous predicates . . .

Lifting: Conflict

- Assume that z is introduced at the node where TS is violated, and ϕ is its predicate.
- Search for β such that
  - $\phi \iff \gamma \wedge \beta$, and
  - β *is in conflict* on z with γ in $\psi_1 \wedge \ldots \wedge \psi_k$.

Conflict

The predicate β *is in conflict* on z with γ *in* $\psi_1 \wedge \ldots \wedge \psi_k$ iff

$$\psi_1 \wedge \ldots \wedge \psi_k \Rightarrow \exists z.\gamma \quad \text{and} \quad \psi_1 \wedge \ldots \wedge \psi_k \not\Rightarrow \exists z.(\gamma \wedge \beta)$$

Lifting: Example

$$\texttt{A} \longrightarrow \texttt{B} : \{x \mid x < 10\}. \tag{1}$$
$$\texttt{A} \longrightarrow \texttt{B} : \{y \mid y > 8\}. \tag{2}$$
$$\texttt{B} \longrightarrow \texttt{A} : \{z \mid x > z \wedge\ z > 6\ \wedge\ y \neq z\} \tag{3}$$

- In our case we have that
  - $\underbrace{(x < 10 \wedge y > 8)}_{\psi_1 \wedge \ldots \wedge \psi_k} \Rightarrow \exists z.\, \underbrace{y \neq z}_{\gamma}$ and
  - $\underbrace{(x < 10 \wedge y > 8)}_{\psi_1 \wedge \ldots \wedge \psi_k} \not\Rightarrow \exists z.\, \underbrace{x > z \wedge\ z > 6}_{\gamma \wedge \beta}$
- Thus, $x > z \wedge\ z > 6$ is *in conflict*, and we have to “lift” it . . .

Lifting: Example

$$\mathtt{A} \longrightarrow \mathtt{B} : \{x \mid x < 10\}. \tag{1}$$
$$\mathtt{A} \longrightarrow \mathtt{B} : \{y \mid y > 8\}. \tag{2}$$
$$\mathtt{B} \longrightarrow \mathtt{A} : \{z \mid x > z \wedge \ z > 6 \ \wedge \ y \neq z\} \tag{3}$$

- “Lift” the predicate $z > 6 \wedge x > z$:

Lifting: Example

$$\mathtt{A} \rightarrow \mathtt{B} : \{x \mid x < 10\}. \tag{1}$$
$$\mathtt{A} \rightarrow \mathtt{B} : \{y \mid y > 8\}. \tag{2}$$
$$\mathtt{B} \rightarrow \mathtt{A} : \{z \mid x > z \land \ z > 6 \ \land \ y \neq z\} \tag{3}$$

- "Lift" the predicate $z > 6 \land x > z$:
- For each interaction above (3) which introduces a variable appearing in $z > 6 \land x > z$,
- add $z > 6 \land x > z$ in the predicate and quantify the variables accordingly:
  - $\forall$: the variables that A doesn't know and are introduced *before* ($\rightarrow$ none)
  - $\exists$: the variables that are introduced *later* ($\rightarrow z$)

Lifting: Example

$$\mathtt{A} \longrightarrow \mathtt{B} : \{x \mid x < 10 \wedge \exists z. x > z > 6\}. \qquad (1)$$
$$\mathtt{A} \longrightarrow \mathtt{B} : \{y \mid y > 8\}. \qquad (2)$$
$$\mathtt{B} \longrightarrow \mathtt{A} : \{z \mid x > z \wedge \ z > 6 \ \wedge \ y \neq z\} \qquad (3)$$

- "Lift" the predicate $z > 6 \wedge x > z$:
- For each interaction above (3) which introduces a variable appearing in $z > 6 \wedge x > z$,
- add $z > 6 \wedge x > z$ in the predicate and quantify the variables accordingly:
  - $\forall$: the variables that `A` doesn't know and are introduced *before* ($\rightarrow$ none)
  - $\exists$: the variables that are introduced *later* ($\rightarrow z$)

Lifting: Example

$$\mathtt{A} \longrightarrow \mathtt{B} : \{x \mid x < 10 \wedge \exists z.x > z > 6\}. \quad (1)$$
$$\mathtt{A} \longrightarrow \mathtt{B} : \{y \mid y > 8\}. \quad (2)$$
$$\mathtt{B} \longrightarrow \mathtt{A} : \{z \mid x > z \wedge \ z > 6 \ \wedge \ y \neq z\} \quad (3)$$

- "Lift" the predicate $z > 6 \wedge x > z$:
- For each interaction above (3) which introduces a variable appearing in $z > 6 \wedge x > z$,
- add $z > 6 \wedge x > z$ in the predicate and quantify the variables accordingly:
  - $\forall$: the variables that A doesn't know and are introduced *before* ($\rightarrow$ none)
  - $\exists$: the variables that are introduced *later* ($\rightarrow z$)
- ... and check that the updated predicate is *satisfiable* (otherwise the algorithm is not applicable)
- ... repeat the algorithm.

Outline

A Methodology

1. An architect designs a choreography
2. The architect is notified if there are any HS or TS problems
3. Using one of the 3 algorithms, solutions or hints of solution may be offered
4. The architect chooses which problem should be solved first
5. Steps 2 to 4 are repeated until all the problems have been solved

Outline

Conclusions and Future Work

- Properties
  - Branching and recursion
  - Correctness
  - TS/HS preservation
  - Underlying type preservation
- Relation between algorithms
  - Strengthening vs Propagation
  - Strengthening vs Lifting
- Implementation

Thanks!

Any Questions?

Assertion Tree

We often consider Global Assertions as trees, e.g. the global assertion

$$\begin{array}{ll} \mathtt{A} \to \mathtt{B} : \{x \mid x > 0\}. & \\ \mathtt{B} \to \mathtt{C} : & \{x \geq 5\}\ \mathrm{gt} : \mathtt{C} \to \mathtt{B} : \{y \mid \psi\} \\ & \{x < 5\}\ \mathrm{le} : \mathtt{C} \to \mathtt{B} : \{z \mid \phi\} \end{array}$$

can be represented by its parsing tree (called *Assertion Tree*)

- $\mathtt{A} \to \mathtt{B} : \{x \mid x > 0\}$
  - $\mathtt{B} \to \mathtt{C}$
    - $\{x \geq 5\}\ \mathrm{gt}$
      - $\mathtt{C} \to \mathtt{B} : \{y \mid \psi\}$
    - $\{x < 5\}\ \mathrm{le}$
      - $\mathtt{C} \to \mathtt{B} : \{z \mid \phi\}$

In the following, $\mathcal{G}$ ranges over global assertions, T over assertion trees, n over nodes, and $n \in T$ denotes that n is a node of T.

Some Definitions

We will use the following functions:

- $snd_T(n)$ returns the sender/selector for $n \in T$
- $rcv_T(n)$ returns the receiver for $n \in T$
- $var_T(n)$ returns the set of variables introduced at node $n \in T$
- $cst_T(n)$ returns the predicate of $n \in T$
- $\texttt{PRED}_T(n)$ returns the conjunction of all the predicates on the path from the root of T until the parent of n.

Strengthening: In General

- Let $n \in T$, where HS is violated for the variable v and let $\psi = cst_T(n)$.
- *Strengthening* consists in trying to find a variable v' such that
  - v' is known to $snd_T(n)$, and
  - $\texttt{PRED}_T(n) \wedge \psi[v'/v] \Rightarrow \psi$ holds.
- If such v' can be found, then *Strengthening* returns a new tree T', obtained from T by replacing ψ by $\psi[v'/v]$ in n.

Propagation: In General

- Let $n \in T$ a node where HS is violated for the variable v.
- Propagation consists in finding a *chain* of (ordered) *interaction* nodes $n_1 \ldots n_t$ such that:
  - $snd_T(n_1)$ knows v,
  - $rcv_T(n_t) = rcv_T(n)$, and
  - for $1 \leq i < t$, $rcv_T(n_i) = snd_T(n_{i+1})$

Propagation: In General

- Let $n \in T$ a node where HS is violated for the variable v.
- Propagation consists in finding a *chain* of (ordered) *interaction* nodes $n_1 \dots n_t$ such that:
  - $snd_T(n_1)$ knows v,
  - $rcv_T(n_t) = rcv_T(n)$, and
  - for $1 \leq i < t$, $rcv_T(n_i) = snd_T(n_{i+1})$
- and returning a new tree T', updated from T such that
  - $cst_{T'}(n_1) = cst_T(n_1) \wedge (u_1 = v)$
  - for $2 \leq i < t$, $cst_{T'}(n_i) = cst_T(n_i) \wedge (u_i = u_{i-1})$
  - $cst_{T'}(n_t) = cst_T(n_t)[u_{t-1}/u]$
  - all the other nodes of T remain unchanged.

Propagation: In General

- Let $n \in T$ a node where HS is violated for the variable v.
- Propagation consists in finding a *chain* of (ordered) *interaction* nodes $n_1 \dots n_t$ such that:
  - $snd_T(n_1)$ knows v,
  - $rcv_T(n_t) = rcv_T(n)$, and
  - for $1 \leq i < t$, $rcv_T(n_i) = snd_T(n_{i+1})$
- and returning a new tree T', updated from T such that
  - $cst_{T'}(n_1) = cst_T(n_1) \wedge (u_1 = v)$
  - for $2 \leq i < t$, $cst_{T'}(n_i) = cst_T(n_i) \wedge (u_i = u_{i-1})$
  - $cst_{T'}(n_t) = cst_T(n_t)[u_{t-1}/u]$
  - all the other nodes of T remain unchanged.

where each u_i variable is fresh and added in the corresponding node.