Observational and symbolic equivalence

Sergiu Bursuc
LSV, CNRS
INRIA project SECSI
École Normale Supérieure de Cachan
bursuc@lsv.ens-cachan.fr

Joint work with Hubert Comon-Lundh and Stéphanie Delaune

CoSyProofs Workshop, Atagawa, April 2009

Goal: decision procedure for observational equivalence

Current results:

- Automatic testing equivalence verification from spi-calculus specifications.
  [Durante, Sisto, Valenzano '03]
- Deciding framed bisimilarity.
  [Huttel '02]
- Automated verification of selected equivalences for the applied pi-calculus.
  [Blanchet, Abadi, Fournet '05]
- Symbolic bisimulation for the applied pi-calculus.
  [Delaune, Kremer, Ryan '07]
- A method for proving observational equivalence.
  [Cortier, Delaune '09]
- Deciding Security of Protocols against Off-line Guessing Attacks.
  [Baudet '05]

The restrictions

A simple class of processes:

$$
\begin{array}{l}
P_i ::= \\
0 \\
out(i,t).P_i \\
in(i,x).\text{if } \Phi(\overline{x}) \text{ then } P_i \\
(\nu\alpha)P_i
\end{array}
$$

A bounded number of sessions:

$$P_1 \mid \ldots \mid P_n$$

Observations, bisimulations, traces

Obsevational equivalence $P \sim_o P'$:
$\forall C.C[P]$ *behaves like* $C[P']$

Static equivalence $T \sim T'$:
$\forall \pi_1, \pi_2 . \pi_1[T]\downarrow = \pi_2[T]\downarrow \Leftrightarrow \pi_1[T']\downarrow = \pi_2[T']\downarrow$

Bisimulation $P \approx P'$:

$$\begin{cases} - \ \phi(P) \sim \phi(P') \\ - \ \forall a.(P \xrightarrow{a} Q \Leftrightarrow P' \xrightarrow{a} Q') \ \& \ Q \approx Q' \end{cases}$$

Trace equivalence $P \approx_t P'$:

$$\begin{cases} - \ \phi(P) \sim \phi(P') \\ - \ \forall w.(P \xrightarrow{w} Q \Leftrightarrow P' \xrightarrow{w} Q') \ \& \ \phi(P') \sim \phi(Q') \end{cases}$$

From concrete to symbolic traces

Concrete transitions: $E = (P_1 | \ldots | P_n, S, \sigma)$

- $P_i \xrightarrow{in(i,x)} \textit{if } \Phi(\overline{x}) \textit{ then } P_i'$

- $P_i \xrightarrow{out(i,t)} P_i'$

From concrete to symbolic traces

Concrete transitions: $E = (P_1 | \dots | P_n, S, \sigma)$

- $P_i \xrightarrow{in(i,x)} \textit{if } \Phi(\overline{x}) \textit{ then } P'_i$
  $\forall t \textit{ s.t. } S \vdash t$:
  - $E \xrightarrow{in(i,x)} (P_1 | \dots P'_i \dots | P_n, S \cup t, \sigma \uplus \{x \mapsto t\})$,
    if $\sigma \uplus \{x \mapsto t\} \models \Phi$
  - $E \xrightarrow{fail} (\bot, \bot, \bot)$, otherwise
- $P_i \xrightarrow{out(i,t)} P'_i$

From concrete to symbolic traces

Concrete transitions: $E = (P_1 | \ldots | P_n, S, \sigma)$

- $P_i \xrightarrow{in(i,x)} \mathit{if}\ \Phi(\overline{x})\ \mathit{then}\ P'_i$
  $\forall t\ s.t.\ S \vdash t$:
  - $E \xrightarrow{in(i,x)} (P_1 | \ldots P'_i \ldots | P_n, S \cup t, \sigma \uplus \{x \mapsto t\})$,
    if $\sigma \uplus \{x \mapsto t\} \models \Phi$
  - $E \xrightarrow{fail} (\bot, \bot, \bot)$, otherwise
- $P_i \xrightarrow{out(i,t)} P'_i$
  - $E \xrightarrow{out(i,t)} (P_1 | \ldots P'_i \ldots | P_n, S \cup t\sigma, \sigma)$

From concrete to symbolic traces

$$R_A(a,b) = \nu n.\ out(\mathrm{enc}(< a, n >, \mathrm{pub}(b))).\ in(\mathrm{enc}(< a, n >, \mathrm{pub}(a)).\ ok$$

$$R_B(a,b) = in(\mathrm{enc}(< a, x >, \mathrm{pub}(b))).\ out(\mathrm{enc}(< a, x >, \mathrm{pub}(a))).\ 0$$

From concrete to symbolic traces

$$R_A(a,b) \;=\; \nu n.\; out(\mathrm{enc}(< a, n >, \mathrm{pub}(b))).\; in(\mathrm{enc}(< a, n >, \mathrm{pub}(a))).\; ok$$

$$R_B(a,b) \;=\; in(\mathrm{enc}(< a, x >, \mathrm{pub}(b))).\; out(\mathrm{enc}(< a, x >, \mathrm{pub}(a))).\; 0$$

Sessions : $R_A(a,b) | R_B(a,b) | R_A(a,c) | R_B(a,c)$

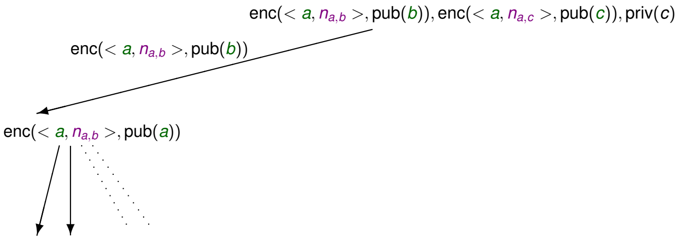
$$\mathrm{enc}(< a, n_{a,b} >, \mathrm{pub}(b)), \mathrm{enc}(< a, n_{a,c} >, \mathrm{pub}(c)), \mathrm{priv}(c)$$

From concrete to symbolic traces

$$R_A(a,b) \;=\; \nu n.\ out(\mathrm{enc}(<a,n>,\mathrm{pub}(b))).\ in(\mathrm{enc}(<a,n>,\mathrm{pub}(a))).\ ok$$

$$R_B(a,b) \;=\; in(\mathrm{enc}(<a,x>,\mathrm{pub}(b))).\ out(\mathrm{enc}(<a,x>,\mathrm{pub}(a))).\ 0$$

Sessions : $R_A(a,b) | R_B(a,b) | R_A(a,c) | R_B(a,c)$

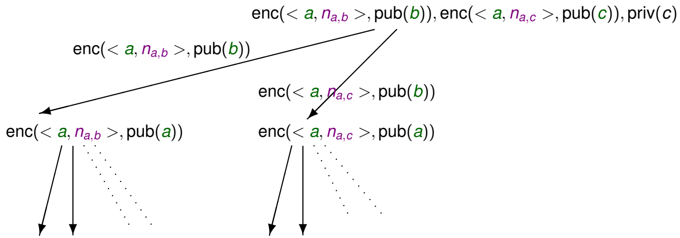


From concrete to symbolic traces

$$R_A(a,b) = \nu n.\ out(\mathrm{enc}(< a, n >, \mathrm{pub}(b))).\ in(\mathrm{enc}(< a, n >, \mathrm{pub}(a))).\ ok$$

$$R_B(a,b) = in(\mathrm{enc}(< a, x >, \mathrm{pub}(b))).\ out(\mathrm{enc}(< a, x >, \mathrm{pub}(a))).\ 0$$

Sessions : $R_A(a,b) | R_B(a,b) | R_A(a,c) | R_B(a,c)$

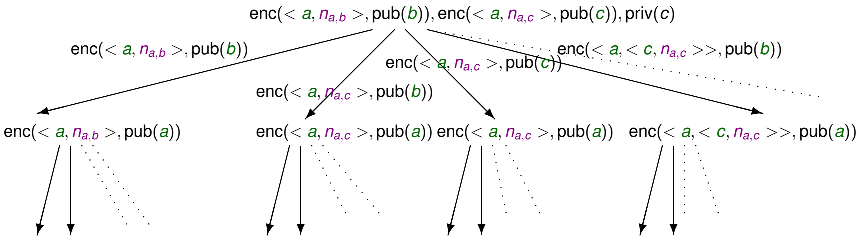


From concrete to symbolic traces

$$R_A(a,b) \;=\; \nu n.\ out(\text{enc}(<a,n>,\text{pub}(b))).\ in(\text{enc}(<a,n>,\text{pub}(a))).\ ok$$

$$R_B(a,b) \;=\; in(\text{enc}(<a,x>,\text{pub}(b))).\ out(\text{enc}(<a,x>,\text{pub}(a))).\ 0$$

Sessions : $R_A(a,b) | R_B(a,b) | R_A(a,c) | R_B(a,c)$

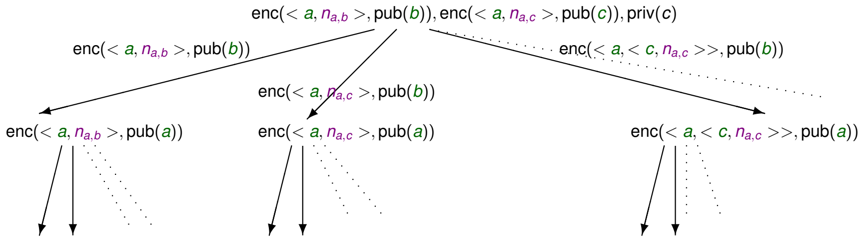


From concrete to symbolic traces

$R_A(a,b) \;=\; \nu n.\ out(\mathrm{enc}(<a,n>,\mathrm{pub}(b))).\ in(\mathrm{enc}(<a,n>,\mathrm{pub}(a))).\ ok$

$R_B(a,b) \;=\; in(\mathrm{enc}(<a,x>,\mathrm{pub}(b))).\ out(\mathrm{enc}(<a,x>,\mathrm{pub}(a))).\ 0$

Sessions : $R_A(a,b)|R_B(a,b)|R_A(a,c)|R_B(a,c)$



From concrete to symbolic traces

$$R_A(a,b) \;=\; \nu n.\ out(\mathrm{enc}(<a,n>,\mathrm{pub}(b))).\ in(\mathrm{enc}(<a,n>,\mathrm{pub}(a))).\ ok$$

$$R_B(a,b) \;=\; in(\mathrm{enc}(<a,x>,\mathrm{pub}(b))).\ out(\mathrm{enc}(<a,x>,\mathrm{pub}(a))).\ 0$$

Sessions : $R_A(a,b) | R_B(a,b) | R_A(a,c) | R_B(a,c)$

$\mathrm{enc}(<a,n_{a,b}>,\mathrm{pub}(b)), \mathrm{enc}(<a,n_{a,c}>,\mathrm{pub}(c)), \mathrm{priv}(c)$

$\mathrm{enc}(<a,x>,\mathrm{pub}(b))$

$\mathrm{enc}(<a,x>,\mathrm{pub}(a))$

$$\mathrm{enc}(<a,n_{a,b}>,\mathrm{pub}(b)), \mathrm{enc}(<a,n_{a,c}>,\mathrm{pub}(c)) \overset{?}{\vdash} \mathrm{enc}(<a,x>,\mathrm{pub}(b))$$

$$\mathrm{enc}(<a,n_{a,b}>,\mathrm{pub}(b)), \mathrm{enc}(<a,n_{a,c}>,\mathrm{pub}(c)), \mathrm{enc}(<a,x>,\mathrm{pub}(a)) \overset{?}{\vdash} \ldots$$

From concrete to symbolic traces

Symbolic transitions: $E = (P_1 | \ldots | P_n, S, \mathcal{C})$

- $P_i \xrightarrow{in(i,x)} \mathit{if}\ \Phi(\overline{x})\ \mathit{then}\ P_i'$

- $P_i \xrightarrow{out(i,t)} P_i'$

From concrete to symbolic traces

Symbolic transitions: $E = (P_1 | \ldots | P_n, S, \mathcal{C})$

- $P_i \xrightarrow{in(i,x)} \textit{if } \Phi(\overline{x}) \textit{ then } P_i'$
  - $E \xrightarrow{in(i,x)} (P_1 | \ldots P_i' \ldots | P_n, S \cup x, \mathcal{C} \wedge S \overset{?}{\vdash} x \wedge \Phi(\overline{x}))$,
    if $\mathcal{C} \wedge S \overset{?}{\vdash} x \wedge \Phi(\overline{x})$ is satisfiable
  - $E \xrightarrow{fail} (\bot, \bot, \bot)$, otherwise
- $P_i \xrightarrow{out(i,t)} P_i'$
  - $E \xrightarrow{out(i,t)} (P_1 | \ldots P_i' \ldots | P_n, S \cup t, \mathcal{C})$

Trace equivalence

Concrete $P \approx_c P$:
$\forall w.$
$P \xrightarrow{w} (P', S, \sigma) \Leftrightarrow Q \xrightarrow{w} (Q', S', \sigma')$ &
$S \sim S'$

Symbolic $P \approx_s P$:
$\forall w.$
$P \xrightarrow{w} (P', S, \mathcal{C}) \Leftrightarrow Q \xrightarrow{w} (Q', S', \mathcal{C}')$ &
$\forall \sigma \in Sol(\mathcal{C}) \exists \sigma' \in Sol(\mathcal{C}').S\sigma \sim S'\sigma'$ &
$\forall \sigma' \in Sol(\mathcal{C}') \exists \sigma \in Sol(\mathcal{C}).S\sigma \sim S'\sigma'$

Trace equivalence

Concrete $P \approx_c P$:

$\forall w.$

$P \xrightarrow{w} (P', S, \sigma) \Leftrightarrow Q \xrightarrow{w} (Q', S', \sigma')$ &

$S \sim S'$

Symbolic $P \approx_s P$:

$\forall w.$

$P \xrightarrow{w} (P', S, \mathcal{C}) \Leftrightarrow Q \xrightarrow{w} (Q', S', \mathcal{C}')$ &

$\forall \sigma \in Sol(\mathcal{C}) \exists \sigma' \in Sol(\mathcal{C}').S\sigma \sim S'\sigma'$ &

$\forall \sigma' \in Sol(\mathcal{C}') \exists \sigma \in Sol(\mathcal{C}).S\sigma \sim S'\sigma'$

Equivalence of deducibility constraint systems

$$\mathcal{C} = \left\{ \begin{array}{ccc} S_1 & \overset{?}{\vdash} & x_1 \\ & \cdots & \\ S_n & \overset{?}{\vdash} & x_n \\ & \Phi(\overline{x}) & \\ S & & \end{array} \right. \qquad \mathcal{C}' = \left\{ \begin{array}{ccc} S'_1 & \overset{?}{\vdash} & x_1 \\ & \cdots & \\ S'_n & \overset{?}{\vdash} & x_n \\ & \Phi'(\overline{x}) & \\ S' & & \end{array} \right.$$

$\mathcal{C} \approx \mathcal{C}'$:

- $\forall \sigma \in Sol(\mathcal{C}) \exists \sigma' \in Sol(\mathcal{C}'). S\sigma \sim S'\sigma'$
- $\forall \sigma' \in Sol(\mathcal{C}') \exists \sigma \in Sol(\mathcal{C}). S\sigma \sim S'\sigma'$

Equivalence of deducibility constraint systems

$$\mathcal{C} = \begin{cases} S_1 & \stackrel{?}{\vdash} & u_1 \\ & \cdots & \\ S_n & \stackrel{?}{\vdash} & u_n \\ S & & \end{cases} \qquad \mathcal{C}' = \begin{cases} S'_1 & \stackrel{?}{\vdash} & u'_1 \\ & \cdots & \\ S'_n & \stackrel{?}{\vdash} & u'_n \\ S' & & \end{cases}$$

$\mathcal{C} \approx \mathcal{C}'$:

- $\forall \sigma \in Sol(\mathcal{C}) \exists \sigma' \in Sol(\mathcal{C}'). S\sigma \sim S'\sigma'$
- $\forall \sigma' \in Sol(\mathcal{C}') \exists \sigma \in Sol(\mathcal{C}). S\sigma \sim S'\sigma'$

Equivalence of deducibility constraint systems

$$\mathcal{C} = \begin{cases} S_1 & \overset{?}{\vdash} & u_1 \\ & \cdots & \\ S_n & \overset{?}{\vdash} & u_n \\ S & & \end{cases} \qquad \mathcal{C}' = \begin{cases} S'_1 & \overset{?}{\vdash} & u'_1 \\ & \cdots & \\ S'_n & \overset{?}{\vdash} & u'_n \\ S' & & \end{cases}$$

$\mathcal{C} \approx \mathcal{C}'$:

- $\forall \sigma \in Sol(\mathcal{C}) \exists \sigma' \in Sol(\mathcal{C}'). S\sigma \sim S'\sigma'$
- $\forall \sigma' \in Sol(\mathcal{C}') \exists \sigma \in Sol(\mathcal{C}). S\sigma \sim S'\sigma'$

$$\mathcal{C} = \begin{cases} a & \overset{?}{\vdash} & x \\ a, x, \{a\}_{\{a\}_k}, \{x\}_k & & \end{cases} \qquad \mathcal{C}' = \begin{cases} a & \overset{?}{\vdash} & x \\ a, x, \{a\}_{\{<a,a>\}_k}, \{x\}_k & & \end{cases}$$

Equivalence of deducibility constraint systems

$$\mathcal{C} = \begin{cases} S_1 & \overset{?}{\vdash} & u_1 \\ & \cdots & \\ S_n & \overset{?}{\vdash} & u_n \\ S & & \end{cases} \qquad \mathcal{C}' = \begin{cases} S'_1 & \overset{?}{\vdash} & u'_1 \\ & \cdots & \\ S'_n & \overset{?}{\vdash} & u'_n \\ S' & & \end{cases}$$

$\mathcal{C} \approx \mathcal{C}'$:

- $\forall \sigma \in Sol(\mathcal{C}) \exists \sigma' \in Sol(\mathcal{C}'). S\sigma \sim S'\sigma'$
- $\forall \sigma' \in Sol(\mathcal{C}') \exists \sigma \in Sol(\mathcal{C}). S\sigma \sim S'\sigma'$

Procedure:

- $\mathcal{C} \rightsquigarrow (\mathcal{C}_1, S_1), \ldots, (\mathcal{C}_n, S_n)$ and $\mathcal{C}' \rightsquigarrow (\mathcal{C}'_1, S'_1), \ldots, (\mathcal{C}'_m, S'_m)$
-

Equivalence of deducibility constraint systems

$$\mathcal{C} = \begin{cases} S_1 & \overset{?}{\vdash} & u_1 \\ & \cdots & \\ S_n & \overset{?}{\vdash} & u_n \\ S & & \end{cases} \qquad \mathcal{C}' = \begin{cases} S'_1 & \overset{?}{\vdash} & u'_1 \\ & \cdots & \\ S'_n & \overset{?}{\vdash} & u'_n \\ S' & & \end{cases}$$

$\mathcal{C} \approx \mathcal{C}'$:

- $\forall \sigma \in Sol(\mathcal{C}) \exists \sigma' \in Sol(\mathcal{C}'). S\sigma \sim S'\sigma'$
- $\forall \sigma' \in Sol(\mathcal{C}') \exists \sigma \in Sol(\mathcal{C}). S\sigma \sim S'\sigma'$

Procedure:

- $\mathcal{C} \rightsquigarrow (\mathcal{C}_1, S_1), \ldots, (\mathcal{C}_n, S_n)$ and $\mathcal{C}' \rightsquigarrow (\mathcal{C}'_1, S'_1), \ldots, (\mathcal{C}'_m, S'_m)$
- $\mathcal{C} \approx \mathcal{C}' \Leftrightarrow$
  - $\forall i \exists j. S_i \sim S'_j$
  - $\forall j \exists i. S_i \sim S'_j$

Elements of a solution

$$\mathcal{C} = \left\{ \begin{array}{rcl} a & \overset{?}{\vdash} & x \\ a, x, \{a\}_{\{a\}_k}, \{x\}_k & & \end{array} \right. \qquad \mathcal{C}' = \left\{ \begin{array}{rcl} a & \overset{?}{\vdash} & x \\ a, x, \{a\}_{\{<a,a>\}_k}, \{x\}_k & & \end{array} \right.$$

Elements of a solution

$$\mathcal{C} = \begin{cases} & a \overset{?}{\vdash} x \\ a, x, \{a\}_{\{a\}_k}, \{x\}_k & \end{cases} \qquad \mathcal{C}' = \begin{cases} & a \overset{?}{\vdash} x \\ a, x, \{a\}_{\{<a,a>\}_k}, \{x\}_k & \end{cases}$$

- Guess equalities among subterms: $\theta_1, \ldots, \theta_n$ and $\theta'_1, \ldots, \theta'_m$
- $\mathcal{C} \rightsquigarrow \mathcal{C}\theta_1, \ldots, \mathcal{C}\theta_n$ and $\mathcal{C}' \rightsquigarrow \mathcal{C}'\theta'_1, \ldots, \mathcal{C}'\theta'_m$

Elements of a solution

$$\mathcal{C} = \left\{ \begin{array}{rcl} a & \stackrel{?}{\vdash} & x \\ a, x, \{a\}_{\{a\}_k}, \{x\}_k & & \end{array} \right. \quad \mathcal{C}' = \left\{ \begin{array}{rcl} a & \stackrel{?}{\vdash} & x \\ a, x, \{a\}_{\{<a,a>\}_k}, \{x\}_k & & \end{array} \right.$$

- Guess equalities among subterms: $\theta_1, \ldots, \theta_n$ and $\theta'_1, \ldots, \theta'_m$
- $\mathcal{C} \rightsquigarrow \mathcal{C}\theta_1, \ldots, \mathcal{C}\theta_n$ and $\mathcal{C}' \rightsquigarrow \mathcal{C}'\theta'_1, \ldots, \mathcal{C}'\theta'_m$
- Does not work (see example 2 next)

Elements of a solution

$$\mathcal{C} = \left\{ \begin{array}{rcl} a & \stackrel{?}{\vdash} & x \\ a, x, \{a\}_{\{a\}_k}, \{x\}_k & & \end{array} \right. \quad \mathcal{C}' = \left\{ \begin{array}{rcl} a & \stackrel{?}{\vdash} & x \\ a, x, \{a\}_{\{<a,a>\}_k}, \{x\}_k & & \end{array} \right.$$

$$\mathcal{C} = \left\{ \begin{array}{rcl} a & \stackrel{?}{\vdash} & x \\ a, x, \{a\}_{\{a\}_k} & & \end{array} \right. \quad \mathcal{C}' = \left\{ \begin{array}{rcl} a & \stackrel{?}{\vdash} & x \\ a, x, \{a\}_{\{<a,a>\}_k} & & \end{array} \right.$$

Elements of a solution

$$\mathcal{C} = \begin{cases} a \overset{?}{\vdash} x \\ a, x, \{a\}_{\{a\}_k}, \{x\}_k \end{cases} \quad \mathcal{C}' = \begin{cases} a \overset{?}{\vdash} x \\ a, x, \{a\}_{\{<a,a>\}_k}, \{x\}_k \end{cases}$$

$$\mathcal{C} = \begin{cases} a \overset{?}{\vdash} x \\ a, x, \{a\}_{\{a\}_k} \end{cases} \quad \mathcal{C}' = \begin{cases} a \overset{?}{\vdash} x \\ a, x, \{a\}_{\{<a,a>\}_k} \end{cases}$$

$$\mathcal{C} = \begin{cases} a \overset{?}{\vdash} x \\ a, x, \{a\}_{\{a\}_k}, \{x\}_k \overset{?}{\vdash} a \\ a, x, \{a\}_{\{a\}_k}, \{x\}_k \end{cases} \quad \mathcal{C}' = \begin{cases} a \overset{?}{\vdash} x \\ a, x, \{a\}_{\{<a,a>\}_k}, \{x\}_k \overset{?}{\vdash} a \\ a, x, \{a\}_{\{<a,a>\}_k}, \{x\}_k \end{cases}$$

Equalities must not be guessed in an arbitrary way, but linked to a proof

Constraint solving rules that...

- $\mathcal{C} \rightsquigarrow^* \mathcal{C}_1, \ldots, \mathcal{C}_n$
- Do not miss any solution σ of $\mathcal{C}$
- Do not miss any proof π in $\mathcal{C}\sigma$
- Do compute all and only equalities that are needed for π

Local theories

A deduction system is *local* if, whenever v is deducible from H, then there is a proof of $H \vdash v$ whose all intermediate steps are either subterms of H or subterms of v

Local theories

A deduction system is *local* if, whenever v is deducible from H, then there is a proof of $H \vdash v$ whose all intermediate steps are either subterms of H or subterms of v

Example:

$$\dfrac{\mathrm{enc}(a, < b, c >) \quad \dfrac{b \quad c}{< b, c >}}{a}$$

$$\dfrac{\mathrm{enc}(a, b) \quad \dfrac{\dfrac{b \quad c}{< b, c >}}{b}}{a}$$

Local theories

A deduction system is *local* if, whenever v is deducible from H, then there is a proof of $H \vdash v$ whose all intermediate steps are either subterms of H or subterms of v

$$\dfrac{\dfrac{\vdots}{v_1} \quad \ldots \quad \dfrac{\vdots}{v_n}}{v} \qquad \text{(from } H\text{)}$$

Local theories

A deduction system is *local* if, whenever v is deducible from H, then there is a proof of $H \vdash v$ whose all intermediate steps are either subterms of H or subterms of v

$$\dfrac{\dfrac{\begin{array}{c}H\\ \vdots\end{array}}{v_1} \quad \dots \quad \dfrac{\vdots}{v_n}}{v}$$

$v_1, \dots, v_n \in \mathrm{St}(H, v)$

Local theories

A deduction system is *local* if, whenever v is deducible from H, then there is a proof of $H \vdash v$ whose all intermediate steps are either subterms of H or subterms of v

$$\cfrac{\cfrac{\overset{\displaystyle H}{\vdots}}{v_1} \quad \ldots \quad \cfrac{\vdots}{v_n}}{v}$$

$v_1, \ldots, v_n \in \mathrm{St}(H)$ (for decompositions)

Locality, decompositions and constraint solving rules

Locality:

equalities are trigered by decompositions $\Longrightarrow$
solved forms are systems with only compositional proofs

Locality, decompositions and constraint solving rules

Locality:
equalities are trigered by decompositions $\Longrightarrow$
solved forms are systems with only compositional proofs

Minimal unsolved constraint: $S_1 \overset{?}{\vdash}_c w_1, \ldots, S_i \overset{?}{\vdash}_c w_i, S \overset{?}{\vdash} v, \ldots$

- Last inference rule $\dfrac{u_1 \;\ldots\; u_n}{u}$

Locality, decompositions and constraint solving rules

Locality:
equalities are trigered by decompositions $\Longrightarrow$
solved forms are systems with only compositional proofs

Minimal unsolved constraint: $S_1 \overset{?}{\vdash}_c w_1, \ldots, S_i \overset{?}{\vdash}_c w_i, S \overset{?}{\vdash} v, \ldots$

- Last inference rule $\dfrac{u_1 \ \ldots \ u_n}{u}$
- Constraint solving rule (decomposition)

$$S \overset{?}{\vdash} v \rightsquigarrow \left\{ \begin{array}{l} S \overset{?}{\vdash} v_1, \ldots, S \overset{?}{\vdash} v_n \\ v = u, v_1 = u_1, \ldots, v_n = u_n, \end{array} \right. \\ \textit{where } v_1, \ldots, v_n \in \mathrm{St}(H)$$

A way to represent all intruder actions

- $\mathcal{C} \rightsquigarrow \mathcal{C}_1, \ldots, \mathcal{C}_n$ and $\mathcal{C}' \rightsquigarrow \mathcal{C}'_1, \ldots, \mathcal{C}'_n$

A way to represent all intruder actions

- $\mathcal{C} \rightsquigarrow \mathcal{C}_1, \ldots, \mathcal{C}_n$ and $\mathcal{C}' \rightsquigarrow \mathcal{C}'_1, \ldots, \mathcal{C}'_n$
- $\mathcal{C} \approx \mathcal{C}'$ iff
  - $\forall i \exists j. S(\mathcal{C}_i) \sim S(\mathcal{C}'_j)$
  - $\forall j \exists i. S(\mathcal{C}_i) \sim S(\mathcal{C}'_j)$

Questions

Design: Are simple processes enough for what we need? If not, can we extend symbolic methods to a larger class?

Analysis:
- Static equivalence: more theories? finer relations?
- Locality: what theories are local? how to handle non-local theories?

Proof theory: are the restrictions that come from the application necessary for decidability?