Relating Process Languages for Security and Communication Correctness

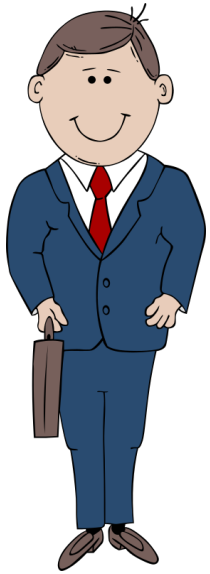
Daniele Nantes
Joint work with Jorge Perez

Universidade de Brasília/ Groningen University

May 2, 2018

1. Motivation
2. Contribution
   - Two-Buyer Contract Signing Protocol
3. Applied π-calculus with Global State
   - Syntax
   - Intruder Reasoning
   - Semantics
4. Multiparty Session Communication
   - Syntax
   - Semantics
5. Extension of A and S
   - Encoding
   - Typing
6. From Session Types to Formulas
7. Conclusions and Future Work

Two-Buyer Protocol

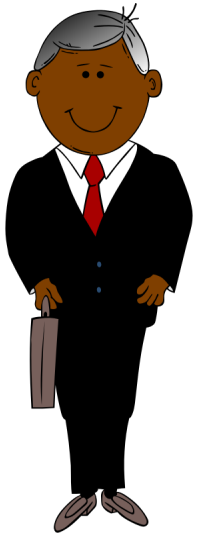


Buyer1

$\Longrightarrow$ $



$ \Longleftarrow$ $$



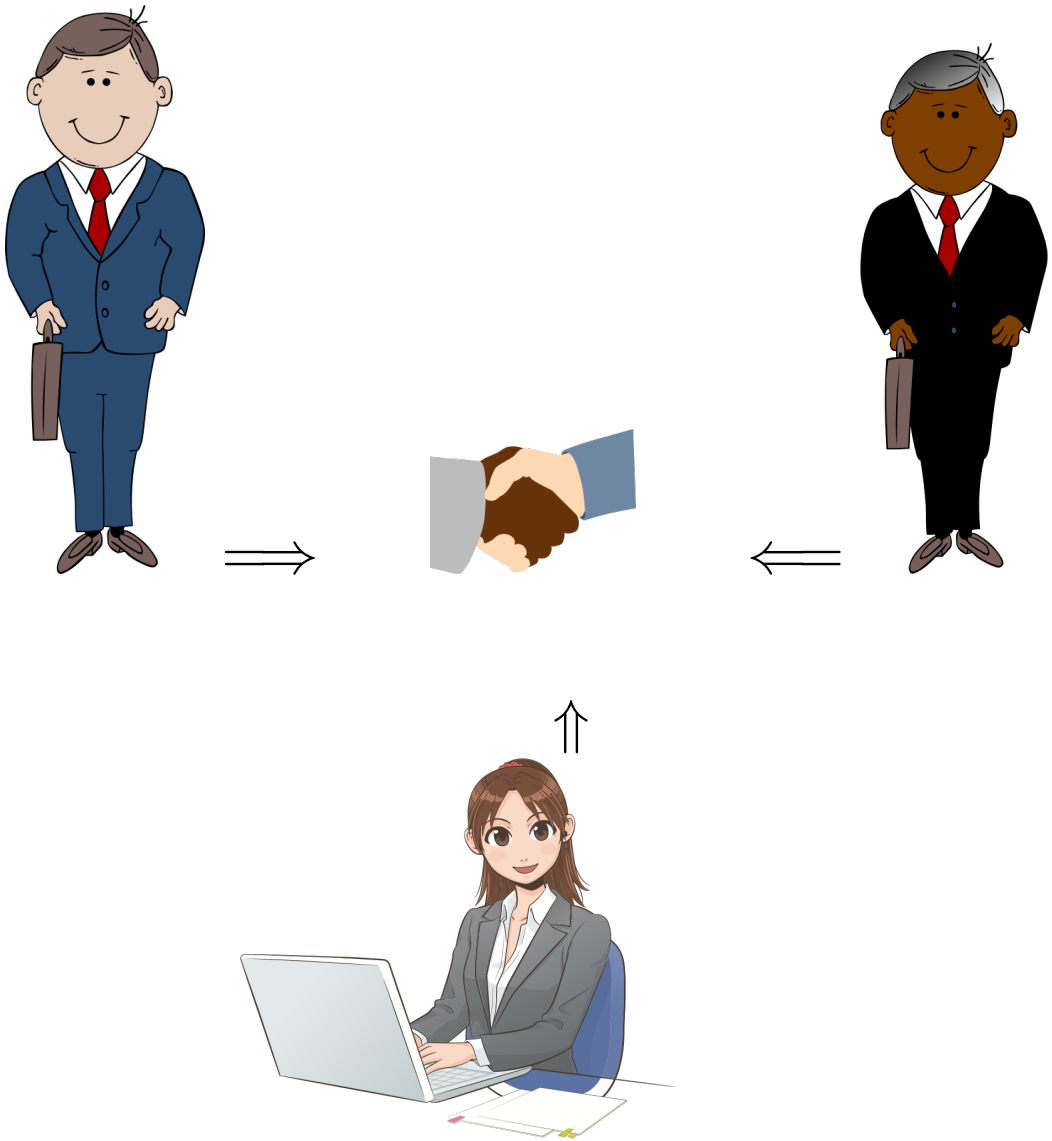
Buyer2



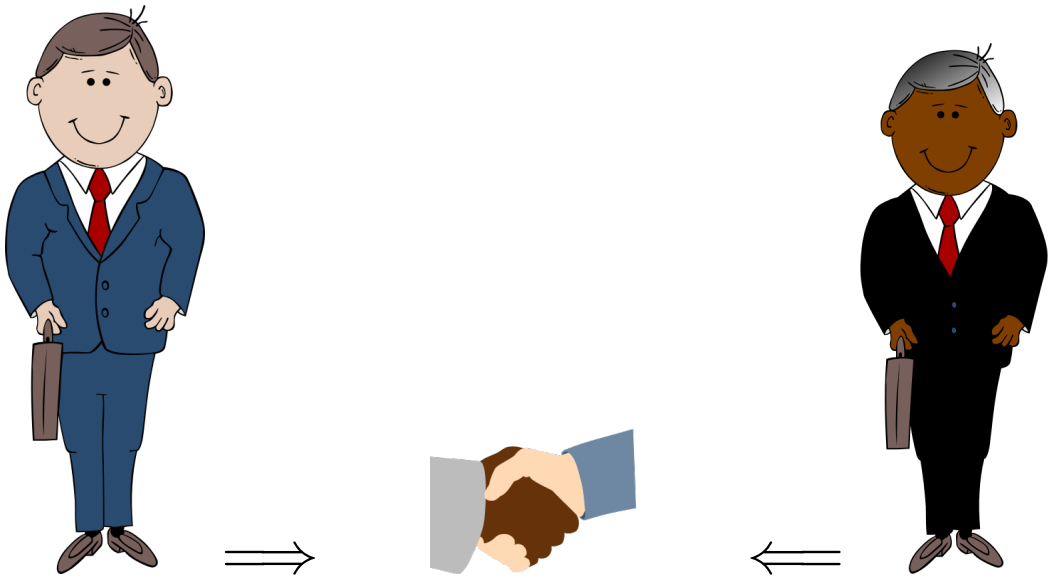
Let's do this!

Seller

Handshake Protocol



Handshake Protocol





correct? or secure?

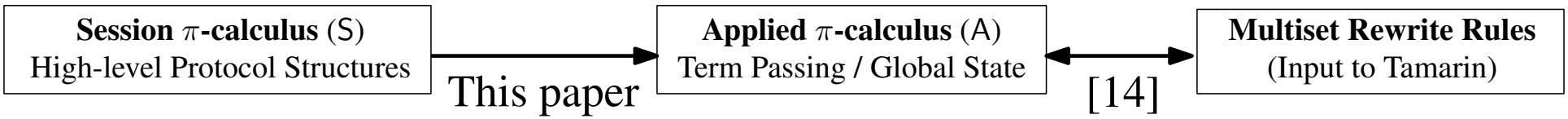
Issues: Synchrony, security, planning, distributed duties, implementing, bugs, choreography...



Work accepted in FORTE18

Two well developed approaches:

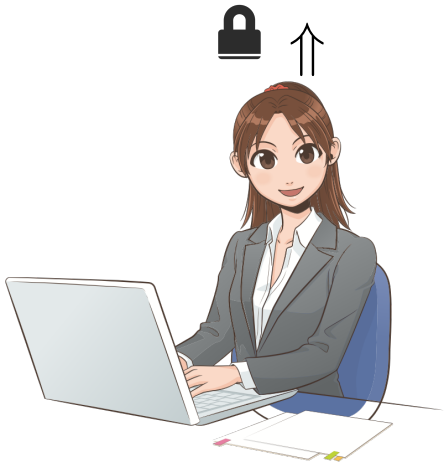
1. Abstract away the coreography and focus on the security issues
   - Pi-calculus (applied), term algebra, proof-assistants,...
2. Abstract away the security steps and focus on the correct message passing programs.
   - Session based communication, π-calculus variants equipped with type systems, some extensions with security properties such as access control and information flow ...



New Scenario

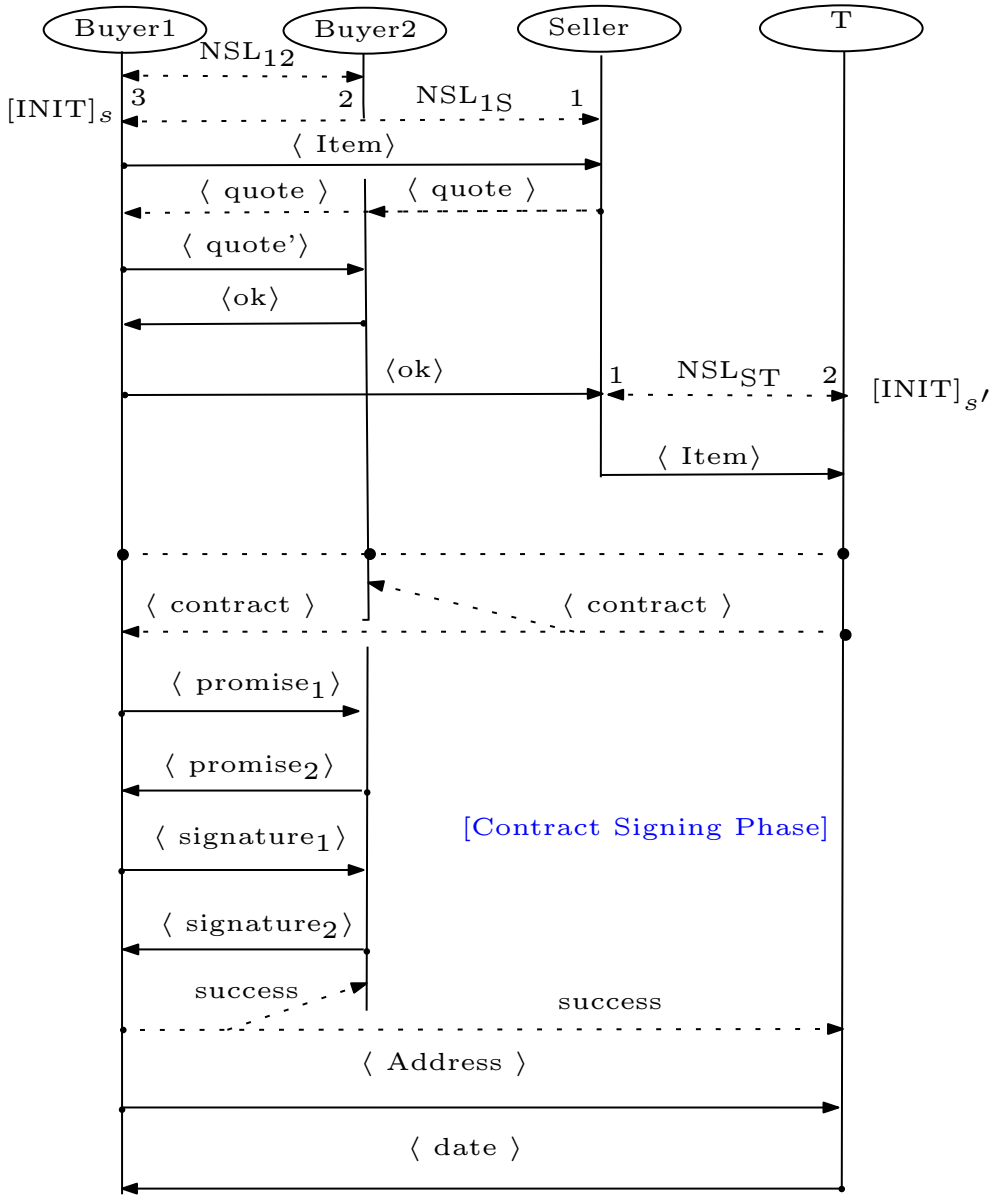


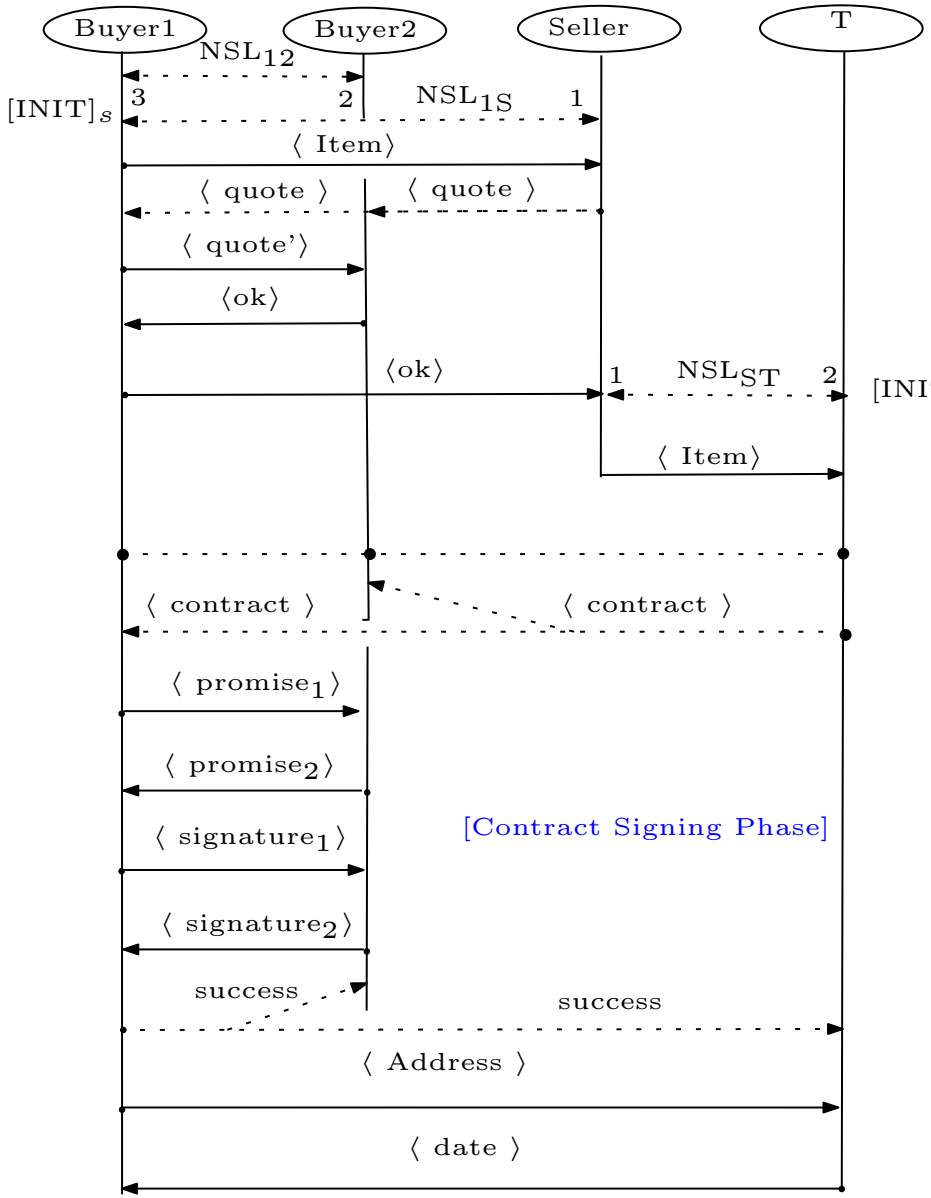
- Session Establishment
- Needham-Schroeder-Lowe-handshake
- Contract Signing



Trusted Party




Buyer1
Buyer2
Seller
T
NSL12
[INIT]s
3
2
NSL1S
1
⟨ Item⟩
⟨ quote ⟩
⟨ quote ⟩
⟨ quote'⟩
⟨ok⟩
⟨ok⟩
1
NSLST
2
[INIT]s'
⟨ Item⟩
⟨ contract ⟩
⟨ contract ⟩
⟨ promise1⟩
⟨ promise2⟩
[Contract Signing Phase]
⟨ signature1⟩
⟨ signature2⟩
success
success
⟨ Address ⟩
⟨ date ⟩



Phase 1

- NSL - session establishment
- purchase agreement
- delegation of the session to trusted party

Phase 2

- contract signing protocol
- exchange of signatures
- conflict solving

The security point of view formalism . . .

Work by Kremmer and Künnemann.

$$
\begin{array}{rl}
M, N ::= & x, y \mid p \mid n \mid f(M_1, \ldots, M_n) \quad (f \in \Sigma) \\
P, Q ::= & \mathbf{0} \mid \mathtt{out}(M, N); P \mid \mathtt{in}(M, N); P \mid \ P \mid Q \mid !P \mid \nu n; P \\
& \mid \mathtt{if}\ M = N\ \mathtt{then}\ P\ \mathtt{else}\ Q \\
& \mid \mathtt{insert}((M, N)); P \mid \mathtt{delete}\ M; P \mid \\
& \mid \mathtt{lookup}\ M\ \mathtt{as}\ x\ \mathtt{in}\ P\ \mathtt{else}\ Q \mid \\
& \mid \mathtt{lock}\ M; P \mid \mathtt{unlock}\ M; P \mid \mathtt{event}\ F; P
\end{array}
$$

Table: Syntax of A: Terms and Processes.

$$\frac{a \in (\mathrm{FN} \cup \mathrm{PN}) \setminus \tilde{n}}{\nu\tilde{n}.\sigma \vdash a}\ [\text{Name}] \qquad \frac{\nu\tilde{n}.\sigma \vdash t \quad t =_E t'}{\nu\tilde{n}.\sigma \vdash t'}\ [\text{Eq}]$$

$$\frac{x \in Dom(\sigma)}{\nu\tilde{n}.\sigma \vdash x\sigma}\ [\text{Frame}] \qquad \frac{\nu\tilde{n}.\sigma \vdash t_1 \ldots \nu\tilde{n}.\sigma \vdash t_n}{\nu\tilde{n}.\sigma \vdash f(t_1, \ldots, t_n)}\ [\text{App}]$$

Table: Deduction rules for A.

We use the well-known corrected version of the Needham-Schroeder-Lowe Public Key Protocol [15] to establish a secure session initiation.

$$(1)\ \overline{a}[\mathtt{n}] \longrightarrow a[\mathtt{i}] : a_n, a_i, aenc(\langle n_{a_n}, a_n\rangle, pk(a_i))$$
$$(2)\ \overline{a}[\mathtt{n}] \longleftarrow a[\mathtt{i}] : a_i, a_n, aenc(\langle n_{a_n}, n_{a_i}, a_i\rangle, pk(a_n))$$
$$(3)\ \overline{a}[\mathtt{n}] \longrightarrow a[\mathtt{i}] : a_n, a_i, aenc(\langle n_{a_i}, c_i\rangle, pk(a_i))$$

TAMARIN

Multiset Rewriting Rules

```
rule (modulo AC) isend:
   [ !KU( x ) ] --[ K( x ) ]-> [ In( x ) ]

rule (modulo AC) irecv:
   [ Out( x ) ] --> [ !KD( x ) ]

rule (modulo AC) 0:
   [ ] --[ Init( ) ]-> [ State_0( ) ]

rule (modulo AC) new_skB120:
   [ State_0( ), Fr( skB12 ) ] --> [ State_01( skB12 ) ]

rule (modulo AC) new_skB1S01:
   [ State_01( skB12 ), Fr( skB1S ) ]
  -->
   [ State_011( skB12, skB1S ) ]

rule (modulo AC) new_skB2011:
   [ State_011( skB12, skB1S ), Fr( skB2 ) ]
  -->
   [ State_0111( skB12, skB1S, skB2 ) ]

rule (modulo AC) new_skS0111:
   [ State_0111( skB12, skB1S, skB2 ), Fr( skS ) ]
  -->
```

SAPIC - TAMARIN

```
 */

begin

builtins: asymmetric-encryption

! ( new skA;
        event HonestA(pk(skA));
        out(pk(skA));
        !( in(pk(xB));
        new Na;
        event OUT_I_1(aenc(<Na,pk(skA)>,pk(xB))
        out(aenc( <Na,pk(skA) > ,pk(xB)));
        in(aenc(<Na,xNb,pk(xB)>,pk(skA)));
        event IN_I_2_nr(xNb,aenc(<Na,xNb,pk(xB)
        new k; out(aenc(<xNb,k>,pk(xB)));
        event SessionA(pk(skA),pk(xB),k)
        ) )
||
! ( new skB;
        event HonestB(pk(skB));
        out(pk(skB));
        !( in(aenc(<xNa,pk(xA)>,pk(skB)));
           event IN_R_1_ni(xNa,aenc(<xNa,pk(xA)
           new Nb;
           event OUT_R_1(aenc(<xNa,Nb,pk(skB)>,
           out(aenc(<xNa,Nb,pk(skB)>,pk(xA)));
           in(aenc(<Nb,xk>,pk(skB)));
           event SessionB(pk(xA),pk(skB),xk)
        ))
```

Standard Operations

$$\vdots$$

$$(\ldots, \mathcal{P} \cup^{\#} \{P \mid Q\}, \sigma, \mathcal{L}) \longrightarrow_{\mathsf{A}} (\ldots, \mathcal{P} \cup^{\#} \{P, Q\}, \sigma, \mathcal{L})$$

$$(\ldots, \mathcal{P} \cup^{\#} \{\mathtt{out}(M, N); P\}, \sigma, \mathcal{L}) \xrightarrow{K(M)}_{\mathsf{A}} (\ldots, \mathcal{P} \cup^{\#} \{P\}, \sigma \cup \{N/x\},$$

$$(\ldots, \mathcal{P} \cup^{\#} \{\mathtt{in}(M, N); P\}, \sigma, \mathcal{L}) \xrightarrow{K(\langle M, N\tau \rangle)}_{\mathsf{A}} (\ldots, \mathcal{P} \cup^{\#} \{P\tau\}, \sigma, \mathcal{L})$$

$$(\ldots, \mathcal{P} \cup^{\#} \{\mathtt{out}(M, N); P, \mathtt{in}(M', N'); Q\}, \sigma, \mathcal{L}) \longrightarrow_{\mathsf{A}} (\ldots, \mathcal{P} \cup^{\#} \{P, Q\tau\}, \sigma, \mathcal{L})$$

$$(\ldots, \mathcal{P} \cup^{\#} \{\mathtt{event}\ F; P\}, \sigma, \mathcal{L}) \xrightarrow{F}_{\mathsf{A}} (\ldots, \mathcal{P} \cup^{\#} \{P\}, \sigma, \mathcal{L})$$

Operations on Global State

$$(\mathcal{S}, \mathcal{P} \cup^{\#} \{\mathtt{insert}((M, N)); P\}, \sigma, \mathcal{L}) \longrightarrow_{\mathsf{A}} (\mathcal{S}[M \mapsto N], \mathcal{P} \cup^{\#} \{P\}, \sigma, \mathcal{L})$$

$$(\mathcal{S}, \mathcal{P} \cup^{\#} \{\mathtt{delete}\ M; P\}, \sigma, \mathcal{L}) \longrightarrow_{\mathsf{A}} (\mathcal{S}[M \mapsto \bot], \mathcal{P} \cup^{\#} \{P\}, \sigma, \mathcal{L})$$

$$(\mathcal{S}, \mathcal{P} \cup^{\#} \{\mathtt{lookup}\ M\ \mathtt{as}\ x\ \mathtt{in}\ P\ \mathtt{else}\ Q\}, \sigma, \mathcal{L}) \longrightarrow_{\mathsf{A}} (\mathcal{S}, \mathcal{P} \cup^{\#} \{P\{V/x\}\}, \sigma, \mathcal{L})$$

$$(\mathcal{S}, \mathcal{P} \cup^{\#} \{\mathtt{lookup}\ M\ \mathtt{as}\ x\ \mathtt{in}\ P\ \mathtt{else}\ Q\}, \sigma, \mathcal{L}) \longrightarrow_{\mathsf{A}} (\mathcal{S}, \mathcal{P} \cup^{\#} \{Q\}, \sigma, \mathcal{L})$$

$$(\mathcal{S}, \mathcal{P} \cup^{\#} \{\mathtt{lock}\ M; P\}, \sigma, \mathcal{L}) \longrightarrow_{\mathsf{A}} (\mathcal{S}, \mathcal{P} \cup^{\#} \{P\}, \sigma, \mathcal{L} \cup \{M\})$$

$$(\mathcal{S}, \mathcal{P} \cup^{\#} \{\mathtt{unlock}\ M; P\}, \sigma, \mathcal{L}) \longrightarrow_{\mathsf{A}} (\mathcal{S}, \mathcal{P} \cup^{\#} \{P\}, \sigma, \mathcal{L} \backslash M)$$

Table: Operational Semantics for A.

Definition (Trace atoms (2))

A *trace atom* has of one of the forms:

$$A ::= \bot \mid t_1 \approx t_2 \mid i \lessdot j \mid i \doteq k \mid F@i$$

The construction of *trace formula* φ respects the usual first-order convention:

$$\varphi, \psi ::= A \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \varphi \leftrightarrow \psi \mid (\exists x : \mathrm{s}).\varphi \mid (\forall x : \mathrm{s}).\varphi$$

Given a process P, in the definition below, tr denotes a trace in $\mathtt{traces}(P)$, $idx(\mathrm{tr})$ denotes the positions in tr, and tr_i denotes the i-th position in tr.

Definition (Satisfaction relation(2))

The satisfaction relation $(\mathrm{tr}, \theta) \vDash \varphi$ between a trace tr, a valuation θ, and a trace formula φ is defined as follows

$$
\begin{array}{llll}
(\mathrm{tr},\theta) \vDash \bot & \text{never} & (\mathrm{tr},\theta) \vDash t_1 \approx t_2 & \text{iff } t_1\theta =_E t_2\theta \\
(\mathrm{tr},\theta) \vDash i \lessdot j & \text{iff } \theta(i) < \theta(j) & (\mathrm{tr},\theta) \vDash \neg\varphi & \text{iff not } (\mathrm{tr},\theta) \vDash \varphi \\
(\mathrm{tr},\theta) \vDash i \doteq j & \text{iff } \theta(i) = \theta(j) & (\mathrm{tr},\theta) \vDash \varphi_1 \wedge \varphi_2 & \text{iff } (\mathrm{tr},\theta) \vDash \varphi_1 \text{ and } (\mathrm{tr},\theta) \vDash
\end{array}
$$

$$
(\mathrm{tr},\theta) \vDash F@i \quad \text{iff } \theta(i) \in idx(\mathrm{tr}) \text{ and } F\theta =_E \mathrm{tr}_{\theta(i)}
$$

$$
(\mathrm{tr},\theta) \vDash (\exists x : \mathtt{s}).\varphi \quad \text{iff there exists } u \in D(\mathtt{s}) \text{ such that } (\mathrm{tr}, \theta[x \mapsto u]) \vDash \varphi
$$

Satisfaction of $(\forall x : \mathtt{s})\varphi$, $\varphi \vee \psi$ and $\varphi \Rightarrow \psi$ can be obtained from the cases above.

Communication correctness formalism...

Asynchronous Multiparty Communication [1]

$$u ::= x \mid a \quad \text{(Identifiers)} \qquad n ::= s \mid a \quad \text{(Names)}$$
$$e ::= v \mid x \mid e = e' \mid \ldots \quad \text{(Expressions)}$$
$$c ::= s[\mathtt{p}] \mid x \quad \text{(Channels)}$$
$$v ::= a \mid \mathtt{true} \mid \mathtt{false} \mid s[\mathtt{p}] \quad \text{(Values)}$$
$$m ::= (\mathtt{q} \triangleright \mathtt{p} : v) \mid (\mathtt{q} \triangleright \mathtt{p} : c) \mid (\mathtt{q} \triangleright \mathtt{p} : l) \quad \text{(Messages)}$$
$$h ::= h \cdot m \mid \emptyset \quad \text{(Queue)}$$

$$\begin{array}{rcll}
P & ::= & \overline{u}[\mathtt{p}](y).P & \text{(Req)} \\
& \mid & u[\mathtt{p}](y).P & \text{(Acc)} \\
& \mid & c!\langle \mathtt{p}, e \rangle.P & \text{(Send)} \\
& \mid & c?(\mathtt{p}, x).P & \text{(Recv)} \\
& \mid & c!\langle\langle \mathtt{p}, c \rangle\rangle.P & \text{(Deleg)} \\
& \mid & c?((\mathtt{q}, y)).P & \text{(Recep)} \\
& \mid & c \oplus \langle \mathtt{p}, l \rangle.P & \text{(Select)} \\
& \mid & c \,\&\, (\mathtt{p}, \{l_i : P_i\}_{i \in I}) & \text{(Branch)} \\
& \mid & \text{if } e \text{ then } P \text{ else } Q & \text{(Condit.)} \\
& \mid & P \mid Q & \text{(Parallel)} \\
& \mid & \mathbf{0} & \text{(Inaction)} \\
& \mid & (\nu n) P & \text{(N.Hiding)} \\
& \mid & s[\widetilde{\mathtt{p}}] : h & \text{(M. Queue)}
\end{array}$$

Table: Process syntax and naming conventions for S.

Process specifications for B_i and S are as follows:

$$B_1 = \overline{a}[3](y).y[3]!\langle 1, \text{“Title”}\rangle.y[3]?(1, x_1).y[3]!\langle 2, x_1 \ \mathit{div}\ 2\rangle.y[3]\&(2, \{\mathtt{ok} : B_1^{sct}, \neg\mathtt{ok} : \mathbf{0}\})$$

$$B_2 = a[2](y).y[2]?(1, x_2).y[2]?(3, x_3).\mathtt{if}\ x_2 - x_3 \leq 99\ \mathtt{then}\ y[2] \oplus \langle\{1, 3\}, \mathtt{ok}\rangle.B_2^{sct}$$
$$\mathtt{else}\ y[2] \oplus \langle\{1, 3\}, \neg\mathtt{ok}\rangle.\mathbf{0}$$

$$S = a[1](y).y[1]?(3, x_1).y[1]!\langle\{2, 3\}, \text{quote}\rangle.y[1]\&(2, \{\mathtt{ok} : \overline{b}[2](z).y[2]!\langle\langle 1, y\rangle\rangle.$$
$$z[2]?(1, x_4).y[1]!\langle 2, \text{date}\rangle.\mathbf{0}, \neg\mathtt{ok} : \mathbf{0}\}))$$

Operational Semantics

$$
\begin{array}{rcll}
a[\mathtt{p}_1](y)P_1 \mid \ldots \mid a[\mathtt{p}_{n-1}](y)P_{n-1} \mid \overline{a}[\mathtt{p}_n](y).P_n & \longrightarrow_{\mathrm{S}} & & [\mathrm{Init}] \\
(\nu s)(P_1\{s[\mathtt{p}_1]/y\} \mid \ldots \mid P_{n-1}\{s[\mathtt{p}_{n-1}]/y\} \mid P_n\{s[\mathtt{p}_n]/y\} \mid s[\widetilde{\mathtt{p}}] : \emptyset) & & & \\
s[\mathtt{p}]!\langle \mathtt{q}, e\rangle.P \mid s : h & \longrightarrow_{\mathrm{S}} & P \mid s : h \cdot (\mathtt{p} \triangleright \mathtt{q} : v) \quad (e \downarrow v) & [\mathrm{Send}] \\
s[\mathtt{p}]!\langle\langle \mathtt{q}, s'[\mathtt{p}']\rangle\rangle.P \mid s : h & \longrightarrow_{\mathrm{S}} & P \mid s : h \cdot (\mathtt{p} \triangleright \mathtt{q} : s'[\mathtt{p}']) & [\mathrm{Deleg}] \\
s[\mathtt{p}] \oplus \langle \mathtt{q}, l\rangle.P \mid s : h & \longrightarrow_{\mathrm{S}} & P \mid s : h \cdot (\mathtt{p} \triangleright \mathtt{q} : l) & [\mathrm{Sel}] \\
s[\mathtt{p}]?(\mathtt{q}, x).P \mid s : (\mathtt{q} \triangleright \mathtt{p} : v) \cdot h & \longrightarrow_{\mathrm{S}} & P\{v/x\} \mid s[\widetilde{\mathtt{p}}] : h & [\mathrm{Recv}] \\
s[\mathtt{p}]?((\mathtt{q}, y)).P \mid s : (\mathtt{q} \triangleright \mathtt{p} : s'[\mathtt{p}']) \cdot h & \longrightarrow_{\mathrm{S}} & P\{s'[\mathtt{p}']/y\} \mid s[\widetilde{\mathtt{p}}] : h & [\mathrm{SRecv}] \\
s[\mathtt{p}] \,\&(\mathtt{q}, \{l_i : P_i\}_{i \in I}) \mid s : (\mathtt{q} \triangleright \mathtt{p} : l_j) \cdot h & \longrightarrow_{\mathrm{S}} & P_j \mid s : h \; (j \in I) & [\mathrm{Branch}] \\
\text{if } e \text{ then } P \text{ else } Q & \longrightarrow_{\mathrm{S}} & P \quad (e \downarrow \mathtt{true}) & [\mathrm{If\text{-}T}] \\
P \equiv P' \text{ and } P' \longrightarrow_{\mathrm{S}} Q' \text{ and } Q \equiv Q' \Rightarrow P & \longrightarrow_{\mathrm{S}} & Q & [\mathrm{Str}] \\
P \longrightarrow_{\mathrm{S}} P' \Rightarrow E[P] & \longrightarrow_{\mathrm{S}} & E[P'] & [\mathrm{Ctx}]
\end{array}
$$

Table: Reduction rules for S (Rule [If-F] omitted).

Implementing NSL- Session Establishment

$$
\begin{aligned}
[\![\overline{a}[3](\widetilde{y_3}).P]\!] = \nu s; P_{31}; P_{32}; \texttt{insert}((\widetilde{s_{ij}}, \emptyset)); \texttt{event}\ \mathsf{init}(\widetilde{s_{ij}});\\
\texttt{event}\ \mathsf{sndchann}(pk(ska_{31}), pk(y_1), s); \texttt{out}(u_1, s);\\
\texttt{event}\ \mathsf{sndchann}(pk(ska_{32}), pk(y_2), s); \texttt{out}(u_2, s); [\![P]\!]\\
P_{3i} = \nu ska_{3i}; \texttt{out}(c, pk(ska_{3i})); \texttt{event}\ \mathsf{honest}(pk(ska_{3i})); \texttt{in}(c, pk(y_i));\\
\nu n_{31}; \texttt{event}\ \mathsf{sndnonce}(pk(ska_{3i}), pk(y_i), aenc(\langle n_{3i}, pk(ska_{3i})\rangle, pk(y_i)))\\
\texttt{out}(c, aenc(\langle n_{3i}, pk(ska_{3i})\rangle, pk(y_i)))); \texttt{in}(c, aenc(\langle n_{3i}, u_i, pk(y_i)\rangle, pk(ska_{3i}\\
\texttt{event}\ \mathsf{rcvnonce}(pk(y_i), pk(ska_{3i}), aenc(\langle n_{3i}, u_i, pk(y_i)\rangle, pk(ska_{3i})))\\
[\![a[i](\widetilde{y_i}).P]\!] = \nu\ ska_i; \texttt{in}(c, pk(x_i)); \texttt{event}\ \mathsf{honest}(pk(ska_i)); \texttt{in}(c, aenc(\langle y, pk(x_i)\rangle, pk(\\
\texttt{event}\ \mathsf{rcvnonce}(pk(x_i), pk(ska_i), aenc(\langle y, pk(x_i)\rangle, pk(ska_i)))\\
\nu n_i; \texttt{event}\ \mathsf{sndnonce}(pk(ska_i), pk(x_i), aenc(\langle y, n_i, pk(ska_i)\rangle, pk(x_i)))\\
\texttt{out}(c, aenc(\langle y, n_i, pk(ska_i)\rangle, pk(x_i))); \texttt{in}(n_i, z);\\
\texttt{event}\ \mathsf{rcvchann}(pk(x_i), pk(ska_i), z); [\![P]\!]
\end{aligned}
$$

Implementing Intra-Session Communication

$$
\begin{aligned}
[\![c_{pq}!\langle e : \mathtt{msg}\rangle.P]\!] = {} & \mathtt{lock}\ c_{pq}; \\
& \mathtt{lookup}\ c_{pq}\ \mathtt{as}\ x\ \mathtt{in} \\
& \mathtt{insert}((c_{pq}, x \cdot \langle \mathtt{msg}, v\rangle)); \mathtt{event}\ \mathsf{out}(c_{pq}, v); \\
& \mathtt{unlock}\ c_{pq}; [\![P]\!] \qquad e \downarrow v \\
[\![c_{pq}?(x).P]\!] = {} & \mathtt{lock}\ c_{qp}; \\
& \mathtt{lookup}\ c_{qp}\ \mathtt{as}\ z_v\ \mathtt{in} \\
& \mathtt{if}\ \mathit{fst}(z_v) = \langle \mathtt{msg}, z\rangle\ \mathtt{then} \\
& \quad \mathtt{insert}((c_{qp}, \mathit{snd}(z_v))); \mathtt{event}\ \mathsf{inp}(c_{pq}, \mathit{fst}(z_v)); \\
& \mathtt{unlock}\ c_{qp}; [\![P\{z/x\}]\!])
\end{aligned}
$$

$$
[\![c_{pq} \oplus \langle l : \mathtt{lbl}\rangle.P]\!] = \mathtt{lock}\ c_{pq}; \mathtt{lookup}\ c_{pq}\ \mathtt{as}\ x\ \mathtt{in}\ (\mathtt{insert}((c_{pq}, x \cdot \langle \mathtt{lbl}, l\rangle))); \mathtt{event}\ \mathsf{sel}(c_{pq}, l); \mathtt{unlock}\ c_{pq}; [\![P]\!]
$$

$$
[\![\mathbf{0}]\!] = \mathtt{event}\ \mathsf{close} \qquad [\![s[\widetilde{\mathrm{p}}] : h]\!] = \mathbf{0}
$$

$$
[\![(\nu s)P]\!] = \nu s; [\![P]\!] \quad [\![P \mid Q]\!] = [\![P]\!] \mid [\![Q]\!]
$$

Theorem (Completeness)

Let $P \in \mathsf{S}^\star$. If $P \longrightarrow_{\mathsf{S}^\star} P'$ then for all process configuration C, there exists a process configuration C' such that $C[\![P]\!] \longrightarrow_{\mathsf{A}}^* C'[\![P']\!]$.

Theorem (Soundness)

Let $P \in \mathsf{S}^\star$ and C be such that $C[\![P]\!] \longrightarrow_{\mathsf{A}_P} R$. Then there exist $P' \in \mathsf{S}^\star$, a C', and λ such that $R \longrightarrow_{\mathsf{A}}^* C'[\![P']\!]$ and $P \xrightarrow{\lambda} P'$.

$S ::= \texttt{bool} \mid \texttt{nonce} \mid \texttt{msg} \mid \texttt{temp} \mid \ldots \mid G$ Sorts

$U ::= S \mid T$ Exchange Types

Global Types

$$G ::= \texttt{p} \to \texttt{q} : \langle U \rangle.G \mid \texttt{p} \to \texttt{q} : \{l_i : G_i\}_{i \in I} \mid \texttt{end}$$

Local Types

$$T ::= !\langle \texttt{p}, U \rangle.T \mid ?(\texttt{p}, U).T \mid \oplus \langle \texttt{p}, \{l_i : T_i\} \rangle \mid \&(\texttt{p}, \{l_i : T_i\}) \mid \texttt{end}$$

Table: Global and Local Types

$$
\begin{array}{lllll}
G_{\text{init}}: & (I.1) & 3 \to 1: & \langle \text{Title} \rangle & \\
& (I.2) & 1 \to \{2,3\}: & \langle \text{quote} \rangle & \\
& (I.3) & 3 \to 2: & \langle \text{quote'} \rangle & \\
& (I.4) & 2 \to \{1,3\}: & \begin{cases} \texttt{ok} & : G_{\text{contract}} \\ \neg\texttt{ok} & : \texttt{end} \end{cases} &
\end{array}
$$

$$
\begin{array}{l}
G_b: \\
(1') \; 1 \to 2 : \langle T \rangle \\
\\
T = (G_{\text{contract}})|_1
\end{array}
$$

$$
\begin{array}{lllll}
G_{\text{contract}}: & (c.1) & 1 \to \{2,3\}: & \langle \text{contract} \rangle & \\
& (c.2) & 3 \to 2: & \langle \text{promise} \rangle & \\
& (c.3) & 2 \to 3: & \begin{cases} \texttt{ok}: & 2 \to 3 : \langle \text{promise} \rangle \\ & 3 \to 2 : \begin{cases} \texttt{ok} : G_{\text{sign}} \\ \neg\texttt{ok} : 3 \to 1 : \texttt{abort} \end{cases} \\ \neg\texttt{ok}: & \texttt{end} \end{cases} &
\end{array}
$$

$$
\begin{array}{lllll}
G_{\text{sign}}: & (s.1) & 3 \to 2: & \langle \text{signature}_1 \rangle & \\
& (s.1) & 2 \to 3: & \begin{cases} \texttt{ok}: & 2 \to 3 : \langle \text{signature}_2 \rangle \\ & 3 \to 1 : \begin{cases} \texttt{success}: & 3 \to 1 : \langle \text{address} \rangle \\ & 1 \to 3 : \langle \text{date} \rangle \\ \neg\texttt{success}: & 1 \to 3 : G_{\text{resolve}_1} \end{cases} \\ \neg\texttt{ok}: & 2 \to 1 : G_{\text{resolve}_2} \end{cases} &
\end{array}
$$

Figure: Global Types for the Trusted Buyer-Seller Protocol.

Example

The projection of G_{init} onto participant 3 (Buyer1), under session s is: $s[3] : !\langle 1, \mathit{string}\rangle.?(1, \mathit{int}).!\langle 2, \mathit{int}\rangle.\&(2, \{\texttt{ok} : (G_{\text{contract}}|_3), \neg\texttt{ok} : \texttt{end}\})$.

$$\Gamma, u : S \vdash u : S \text{ [Name]} \quad \Gamma \vdash \texttt{true}, \texttt{false} : \texttt{bool} \text{ [Bool]} \quad \frac{\Gamma \vdash e_i : \texttt{bool}}{\Gamma \vdash e_1 \text{ and } e_2 : \texttt{bool}} \text{ [And]}$$

$$\frac{\Gamma \vdash \mathtt{u} : G \quad \Gamma \vdash P \triangleright \Delta, y : G|_{\mathtt{p}}}{\Gamma \vdash \overline{u}[p](y).P \triangleright \Delta} \text{ [MCast]} \qquad \frac{\Gamma \vdash \mathtt{u} : G \quad \Gamma \vdash P \triangleright \Delta, y : G|_{\mathtt{p}}}{\Gamma \vdash u[p](y).P \triangleright \Delta} \text{ [MAcc]}$$

$$\frac{\Gamma \vdash e : S \quad \Gamma \vdash P \triangleright \Delta, c : T}{\Gamma \vdash c!\langle \mathtt{q}, e\rangle.P \triangleright \Delta, c : !\langle \mathtt{q}, S\rangle.T} \text{ [Send]} \quad \frac{\Gamma, x : S \vdash P \triangleright \Delta, c : T}{\Gamma \vdash c?(\mathtt{q}, x).P \triangleright \Delta, c : ?(\mathtt{q}, S).T} \text{ [Rcv]} \quad \frac{\Delta \text{ end}}{\Gamma \vdash \mathbf{0} \triangleright \Delta} \text{ [Inact]}$$

$$\frac{\Gamma \vdash P \triangleright \Delta, c : T}{\Gamma \vdash c!\langle \mathtt{q}, \langle c'\rangle\rangle.P \triangleright \Delta, c : !\langle \mathtt{q}, \mathtt{T}\rangle.T, c' : \mathtt{T}} \text{ [Deleg]} \quad \frac{\Gamma \vdash P \triangleright \Delta, c : \mathtt{T}, c' : \mathtt{T}'}{\Gamma \vdash c?(\mathtt{q}, (y)).P \triangleright \Delta, c : ?(\mathtt{q}, \mathtt{T}').\mathtt{T}} \text{ [Srcv]}$$

$$\frac{\Gamma \vdash P \triangleright \Delta, c : T_j \quad j \in I}{\Gamma \vdash c \oplus \langle \mathtt{q}, l_j\rangle.P \triangleright \Delta, c : \oplus\langle \mathtt{q}, \{l_i : T_i\}\rangle} \text{ [Sel]} \quad \frac{\Gamma \vdash P_i \triangleright \Delta, c : T_i \quad \forall i \in I}{\Gamma \vdash c \,\&(\mathtt{q}, \{l_i : P_i\}_{i \in I}) \triangleright \Delta, c : \oplus\langle \mathtt{q}, \{l_i : T_i\}\rangle} \text{ [Bran]}$$

$$\frac{\Gamma \vdash P \triangleright \Delta \quad \Gamma \vdash Q \triangleright \Delta'}{\Gamma \vdash P \mid Q \triangleright \Delta, \Delta'} \text{ [Par]} \quad \frac{\Gamma \vdash e : \texttt{bool} \quad \Gamma \vdash P \triangleright \Delta \quad \Gamma \vdash Q \triangleright \Delta}{\Gamma \vdash \mathsf{if}\ e\ \mathsf{then}\ P\ \mathsf{else}\ Q \triangleright \Delta} \text{ [If]} \quad \frac{\Gamma, a : G \vdash P \triangleright \Delta}{\Gamma \vdash (\nu a)P \triangleright \Delta} \text{ [NRes]}$$

Table 22. Typing rules (from [11]). In [Sel] and [Bran], $i \in I$. In rules [MCast] and [MAcc], $\mathtt{p} = mp(G)$ and $\mathtt{p} < mp(G)$, resp.

Definition (Local Formula)

Given a local type T and an endpoint $s[\mathtt{p}]$, its *local formula* $\Phi_{s[\mathtt{p}]}(T)$ is defined inductively as follows:

$$
\begin{array}{lcl}
\Phi_{s[\mathtt{p}]}(!\langle \mathtt{q}, S\rangle.T) & = & \exists i, z.(\mathsf{out}(s_{\mathtt{pq}}, z)@i \wedge \psi(\Phi_{s[\mathtt{p}]}(T)) \\
\Phi_{s[\mathtt{p}]}(?(\mathtt{q}, U).T) & = & \exists i, z.(\mathsf{inp}(s_{\mathtt{pq}}, z)@i \wedge \psi(\Phi_{s[\mathtt{p}]}(T)) \\
\Phi_{s[\mathtt{p}]}(\oplus\langle \mathtt{q}, \{l_i : T_i\}_{i\in I}\rangle) & = & \exists i. \bigvee_{j\in I}(\mathsf{sel}(s_{\mathtt{pq}}, l_j)@i \wedge \psi(\Phi_{s[\mathtt{p}]}(T_j)) \\
\Phi_{s[\mathtt{p}]}(\&(\mathtt{q}, \{l_i : T_i\}_{i\in I})) & = & \exists i. \bigvee_{j\in I}(\mathsf{bra}(s_{\mathtt{pq}}, l_j)@i \wedge \psi(\Phi_{s[\mathtt{p}]}(T_j)) \\
\Phi_{s[\mathtt{p}]}(\mathtt{end}) & = & \exists i.\mathsf{close}@i.
\end{array}
$$

where $\psi(\Phi_{s[\mathtt{p}]}(T)) := \forall k.(\Phi_{s[\mathtt{p}]}(T)(k) \Rightarrow i \lessdot k))$ the quantified variables have sorts $i, j, k : \mathtt{temp}$ and $z : S$, and variables i and z are fresh. The extension of $\Phi(_)$ to *session environments*, denoted $\widehat{\Phi}(_)$, is as expected: $\widehat{\Phi}(\Delta, s[\mathtt{p}] : T) = \widehat{\Phi}(\Delta) \wedge \Phi_{s[\mathtt{p}]}(T)$.

Theorem

Let $\Gamma \vdash P \triangleright \Delta$ *be a well-typed* S *process. Also, let* $\mathrm{tr} \in \texttt{traces}([\![([\![P]\!])]\!])$. *Then there exists a* θ *such that* $(\mathrm{tr}, \theta) \vDash \widehat{\Phi}(\Delta)$.

Theorem

Let tr *and* φ *be a trace and a local formula, respectively. Suppose* θ *is an instantiation such that* $(\mathrm{tr}, \theta) \vDash \varphi$. *Then there is a* $P \in \mathsf{S}$ *such that*

$$\Gamma_\varphi \vdash P \triangleright \Phi^{-1}(\varphi) \qquad \textit{where } \Gamma_\varphi = \{\theta(x) : \mathit{sort}(x) \mid x \in \mathit{dom}(\theta)\}$$

Example

The projection of G_{init} onto participant 3 (Buyer1), under session s is:
$s[3] : !\langle 1, \mathit{string}\rangle.?(1, \mathit{int}).!\langle 2, \mathit{int}\rangle.\&(2, \{\texttt{ok} : (G_{\text{contract}}|_3), \neg\texttt{ok} : \texttt{end}\})$.

Example

The projection of G_{init} onto participant 3 (Buyer1), under session s is:
$s[3] : !\langle 1, \mathit{string}\rangle.?(1, \mathit{int}).!\langle 2, \mathit{int}\rangle.\&(2, \{\mathtt{ok} : (G_{\text{contract}}|_3), \neg\mathtt{ok} : \mathtt{end}\})$.
The local formula associated is:

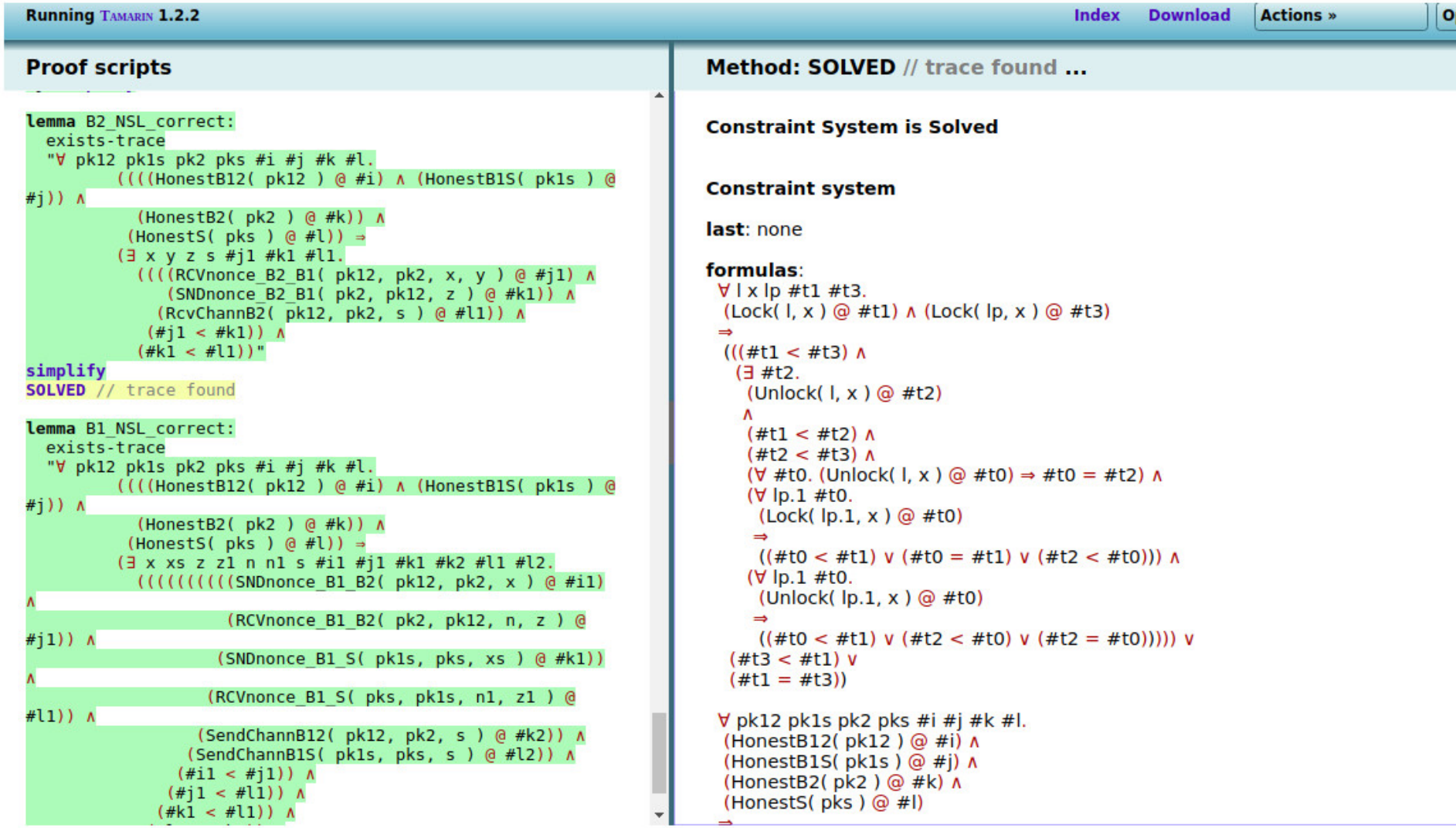
$$\begin{aligned}\Phi_{s[3]}(T) = \exists i_1, z_1.\mathrm{out}(s_{31}, z_1)@i_1 \wedge (\exists i_2 z_2.\mathrm{inp}(s_{31}, z_2)@i_2 \wedge (\exists i_3 z_3.\mathrm{out}(s_{32}, z_3))@i_3 \\ \wedge (\exists i_4 i_5 z_4.((\mathrm{bra}(s_{32}, \mathtt{ok})@i_4 \wedge \Phi_{s[3]}(T')) \vee \mathrm{bra}(s_{32}, \neg\mathtt{ok})@i_4 \wedge \mathrm{close}@i_5))) \\ \wedge ((i_1 < i_2 < i_3 < i_4 \wedge \psi(\Phi_{s[3]}(T'))) \vee (i_1 < i_2 < i_4 < i_5 \wedge \psi(\Phi_{s[3]}(T'))))\end{aligned}$$

where T' is the projection of G_{contract} onto participant 3.

We now consider properties associated to fidelity/safety of processes with respect to their local types. The lemma below ensures protocol fidelity of B_1 and B_2 with respect to the corresponding projections of the global type $G_{\texttt{init}}$, presented in Figure 1. The corresponding local formula can be obtained following Definition 6:

```
lemma B1_B2_protocol_fidelity:
exists-trace
(Ex x y z s #j #j1 #k #k1 #l.out(s31, x)@j & inp(s31, z)@k & out(s32, s)@l & j <
k & k < l
  & ((bra(s32, ok))@j1 & l < j1) | (bra(s32, ¬ok)@k1 & l < k1 & Φ(Gcontract|3))) &
(Ex x y z s #j #j1 #k #k1 #l. inp(s21, z)@k & inp(s23, s)@l & j < k & k < l
  &((sel(s23, ok)@j1 & l < j1 & Φ(Gcontract|2)) | (sel(s23, ¬ok)@k1 & l < k1))
```

Tamarin Proof Script



What we did:

1. We have connected two distinct process models: the calculus S for multiparty session- based communication [1] and the calculus A for the analysis of security protocols [2].
2. we have used SAPIC/Tamarin to implement our encodings and the verification technique for communication correctness, based on local formulas

What we did:

1. We have connected two distinct process models: the calculus S for multiparty session- based communication [1] and the calculus A for the analysis of security protocols [2].
2. we have used SAPIC/Tamarin to implement our encodings and the verification technique for communication correctness, based on local formulas

What we want to do:

1. To explore our approach to process specification and verifi- cation in the setting of ProVerif, whose input language is a typed applied π-calculus.
2. We also plan to connect our approach with existing type systems for secure information flow and access control in multiparty sessions.

References

1. K. Honda, N. Yoshida, M. Carbone. *Multiparty Asynchronous Session Types.* J. ACM 63(1): 9:1-9:67, 2016.
2. S.Kremmer, R. Künnemann. *Automated analysis of security protocols with global state.* Journal of Computer Security 24(5): 583-616, 2016.
3. D. Nantes and J. Perez. *Relating Process Languages for Security and Communication Correctness(Extended Abstract)*. In Proc. of International Conference on Formal Techniques for Distributed Objects, Components, and Systems, 2018.