An Equivalence Result Between Linear Logic and Process Calculi

Alessandro Bruni
(Joint work with Eike Ritter and Carsten Schürmann)

Center for Information Security and Trust

Øresund Security Day 2019

IT UNIVERSITY OF COPENHAGEN

Problem: precisely analysing security protocols

Example

```
free c: channel.
free s: channel[private].
query attacker(new secret_).
process
  (new secret_:bitstring; out(s, secret_) |
    in(s, x:bitstring); in(s, y:bitstring); out(c, x))
```

Shows a false attack in ProVerif (and other tools)

1. Can we use linear logic to reason precisely about concurrent communicating processes, security protocols in particular?
2. Is there a semantic gap between linear logic formulas with their turnstyle relation and process algebras with their reductions?

Short answer: Yes, and yes!

Long answer

Let's start simple:

- CCS: $P, Q ::= 0 \mid \overline{a} \mid a.P \mid (P \mid Q)$
- LL: $A, B ::= 1 \mid a \mid A \multimap B \mid A \otimes B$

Example:

$$\overline{a} \mid a.\overline{b} \mid b.\overline{c} \to \overline{b} \mid b.\overline{c} \to \overline{c}$$

We can prove in linear logic:

1. $a \otimes (a \multimap b) \otimes (b \multimap c) \vdash b \otimes (b \multimap c)$
2. $a \otimes (a \multimap b) \otimes (b \multimap c) \vdash c$

But also:

3. $a \otimes (a \multimap b) \otimes (b \multimap c) \vdash a \otimes (a \multimap c)$

Semantics

Stuctural equivalence:

$$P \mid 0 \equiv P \qquad P \mid Q \equiv Q \mid P \qquad P \mid (Q \mid R) \equiv (P \mid Q) \mid R$$

Reaction semantics for CCS:

$$a.P \mid \overline{a} \rightarrow P \qquad \frac{P \rightarrow P'}{P \mid Q \rightarrow P' \mid Q} \qquad \frac{P \equiv \circ \rightarrow \circ \equiv Q}{P \rightarrow Q}$$

Reduction in n steps:

$$P \rightarrow^0 Q \text{ iff } P \equiv Q \qquad P \rightarrow^{i+1} Q \text{ iff } P \rightarrow P' \text{ and } P' \rightarrow^i Q$$

Translation into Linear Logic

$$[\![a.P]\!] = a \multimap [\![P]\!] \qquad [\![0]\!] = 1 \qquad [\![\overline{a}]\!] = a \qquad [\![P \mid Q]\!] = [\![P]\!] \otimes [\![Q]\!]$$

The weird one out

$$\begin{array}{c} \overline{a} \mid a.\overline{b} \mid b.\overline{c} \not\rightarrow \quad \overline{a} \mid a.\overline{c} \\ \downarrow \\ [\![\cdot]\!] \\ \downarrow \\ a \otimes (a \multimap b) \otimes (b \multimap c) \vdash \quad a \otimes (a \multimap c) \end{array}$$

Let's look at the proof:

$$\dfrac{\dfrac{a \vdash a \qquad \dfrac{\dfrac{a \vdash a \qquad \dfrac{b \vdash b \quad c \vdash c}{b, b \multimap c \vdash c}\multimap L}{a, a \multimap b, b \multimap c \vdash c}\multimap L}{a \multimap b, b \multimap c \vdash a \multimap c}\multimap R}{a, a \multimap b, b \multimap c \vdash a \otimes (a \multimap c)}\otimes R}{a \otimes (a \multimap b) \otimes (b \multimap c) \vdash a \otimes (a \multimap c)}\otimes L^2$$

Annotated Linear Logic

$$\frac{}{A \vdash^0 A}\ ax \qquad \frac{\Delta \vdash^i C}{\Delta, 1 \vdash^i C}\ 1L \qquad \frac{}{\cdot \vdash^0 1}\ 1R$$

$$\frac{\Delta_1 \vdash^i A \quad \Delta_2, B \vdash^j C}{\Delta_1, \Delta_2, A \multimap B \vdash^{i+j+1} C}\ {\multimap}L \qquad \frac{B \vdash^i C}{a \multimap B \vdash^i a \multimap C}\ {\multimap}S$$

$$\frac{\Delta, A, B \vdash^i C}{\Delta, A \otimes B \vdash^i C}\ {\otimes}L \qquad \frac{\Delta_1 \vdash^i A \quad \Delta_2 \vdash^j B}{\Delta_1, \Delta_2 \vdash^{i+j} A \otimes B}\ {\otimes}R$$

(The index i on $\vdash^i$ counts the ${\multimap}L$ applications in the current branch)

Is this a logic?

Yes! It has Cut-elimination:

Theorem (Cut)

If $\Delta_1 \vdash^i A$ *and* $\Delta_2, A \vdash^j C$*, then* $\Delta_1, \Delta_2 \vdash^{i+j} C$*.*

Proof.

By induction on i and then structural induction on the derivations. $\square$

Soundness and Completeness

Theorem (Completeness)

Let $\mathcal{P}$ be a list of processes, Q a process, $i \in \mathbb{N}$. If $[\![\mathcal{P}]\!] \vdash^i [\![Q]\!]$ then $\prod_{P \in \mathcal{P}} P \rightarrow^i Q$.

Theorem (Soundness)

Let $\mathcal{P}$ be a list of processes, Q a process, $i \in \mathbb{N}$. If $\prod_{P \in \mathcal{P}} P \rightarrow^i Q$ then $[\![\mathcal{P}]\!] \vdash^i [\![Q]\!]$.

Moving to the π-calculus

Processes:

$$
\begin{aligned}
P, Q ::= {} & 0 \\
& \mid \mathsf{out}(M, N) \\
& \mid \mathsf{in}(M, x); P \\
& \mid\ !\, P \\
& \mid P \mid Q \\
& \mid \mathsf{new}\ u; P \\
& \mid \mathsf{let}\ x = g(M)\ \mathsf{in}\ P \\
& \mid \mathsf{if}\ M = N\ \mathsf{then}\ P \\
& \mid \mathsf{reduc}\ \forall x_1, \ldots, x_n; g(M_1, \ldots, M_n) = N
\end{aligned}
$$

A Translation for the Applied Pi-calculus

$$
\begin{aligned}
[\![\mathrm{in}(M, x); P]\!] &= \forall\, x.\, \mathrm{msg}(M, x) \multimap [\![P]\!] \\
[\![\mathrm{out}(M, N)]\!] &= \mathrm{msg}(M, N) \\
[\![\mathrm{new}\; u; P]\!] &= \exists\, u.\, [\![P]\!] \\
[\![P \mid Q]\!] &= [\![P]\!] \otimes [\![Q]\!] \\
[\![\mathrm{let}\; x = g(\vec{M})\; \mathrm{in}\; P]\!] &= \left(\exists\, c.\, \mathrm{red}(c, g(\vec{M})) \otimes \forall\, x.\, \mathrm{res}(c, x) \multimap [\![P]\!]\right) \\
[\![\mathrm{if}\; M{=}N\; \mathrm{then}\; P]\!] &= (\exists\, c.\, \mathrm{eq}(c, M) \otimes (\mathrm{eq}(c, N) \multimap [\![P]\!])) \\
[\![!P]\!] &= ![\![P]\!] \\
[\![0]\!] &= 1 \\
[\![\mathrm{reduc}\; \forall \vec{x};\; g(\vec{M}) \rightarrow N]\!] &= !\forall\, c, \vec{x}.\, \mathrm{red}(c, g(\vec{M})) \multimap \mathrm{res}(c, N)
\end{aligned}
$$

Proofs (WIP)

Operational semantics and proof system with explicit substitutions:

$$\Gamma; \rho; \mathcal{P} \to \Gamma'; \rho'; \mathcal{P}'$$

$$\Gamma; \Delta[\rho] \vdash A[\rho']$$

Lemma (Soundness)

Let $\Gamma; \rho; \mathcal{P}$ *and* $\Gamma'; \rho'; \mathcal{P}'$ *be two configurations, let* $K = [\![\mathcal{P}]\!]$ *and* $K' = [\![\mathcal{P}']\!]$*. If* $\Gamma; \rho; \mathcal{P} \to \Gamma'; \rho'; \mathcal{P}'$ *then* $\cdot; \exists\Gamma.\ K[\rho] \vdash \exists\Gamma'.\ K'[\rho']$*.*

Completeness

(WIP)

It's not Curry-Howard, but close

- Curry-Howard isomorphisms relate programs and logic formulas, e.g.:
- natural deduction $\leftrightarrow$ λ-calculus, linear logic $\leftrightarrow$ π-calculus
- Here we rather use linear logic as a logical framework for reasoning about concurrent communicating systems
- The approach extends to analyzing for example cryptographic protocols, as shown

Outlook

- The power of $a \otimes (a \multimap b) \otimes (b \multimap c) \vdash a \otimes (a \multimap c)$ (Resolution)
- Skolemizing intuitionistic linear logic