Hypha: static analysis of the linear π-calculus

Luca Padovani

(with contributions from Tzu-Chun Chen and Andrea Tosatto)

Dipartimento di Informatica – Università di Torino

What we do

1. linearity analysis
   $\Rightarrow$ partition channels into **linear** and **non-linear**

2. protocol analysis (optional)
   $\Rightarrow$ infer **communication structure**

3. deadlock freedom analysis
   $\Rightarrow$ no pending communications in **stable** states

4. lock freedom analysis
   $\Rightarrow$ pending communications in **all** states can be completed

How we do it

Theorem

If $\Gamma \vdash P$*, then the following properties hold:*

1. *P uses its channels according to Γ*
2. *P is deadlock free*
3. *P is lock free*

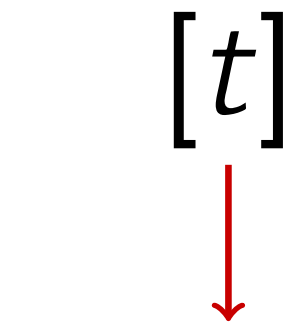
Definition (type reconstruction)

Given an untyped process P, find Γ such that $\Gamma \vdash P$, if there is one.

Information in channel types

$$[t]$$

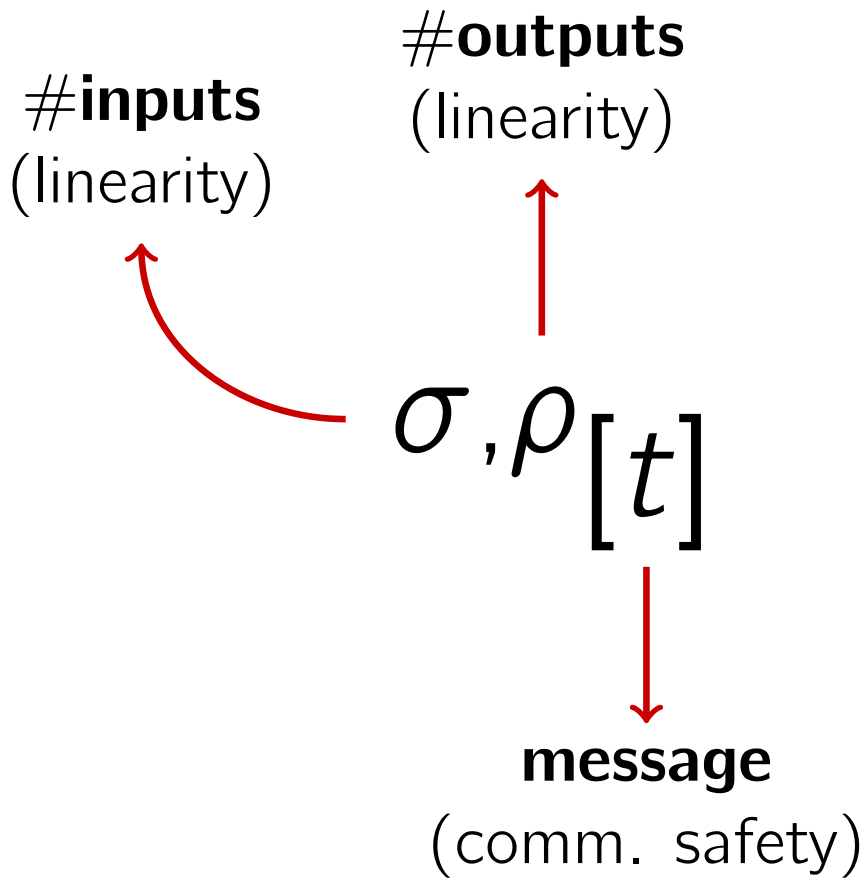
Information in channel types



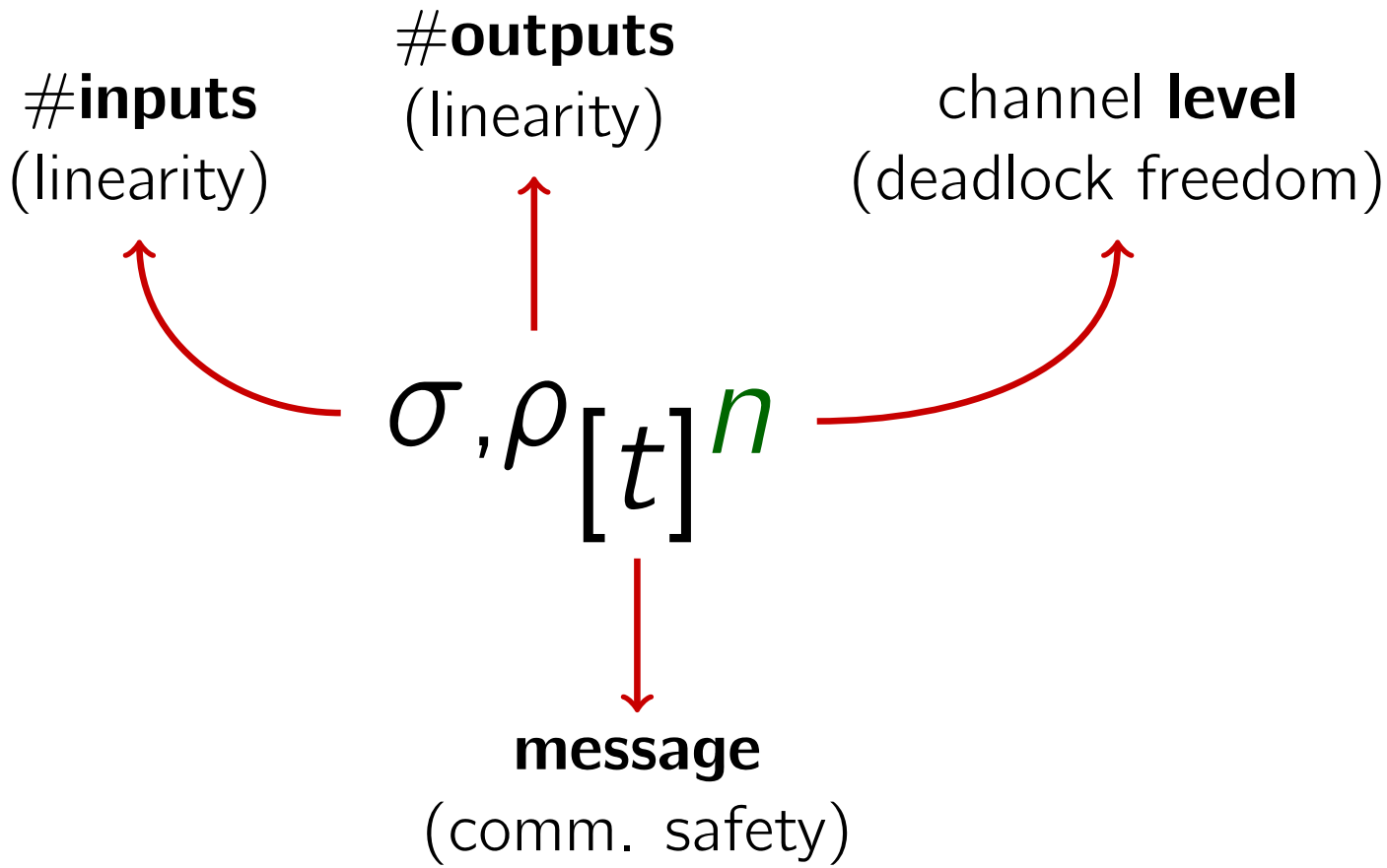
message

(comm. safety)

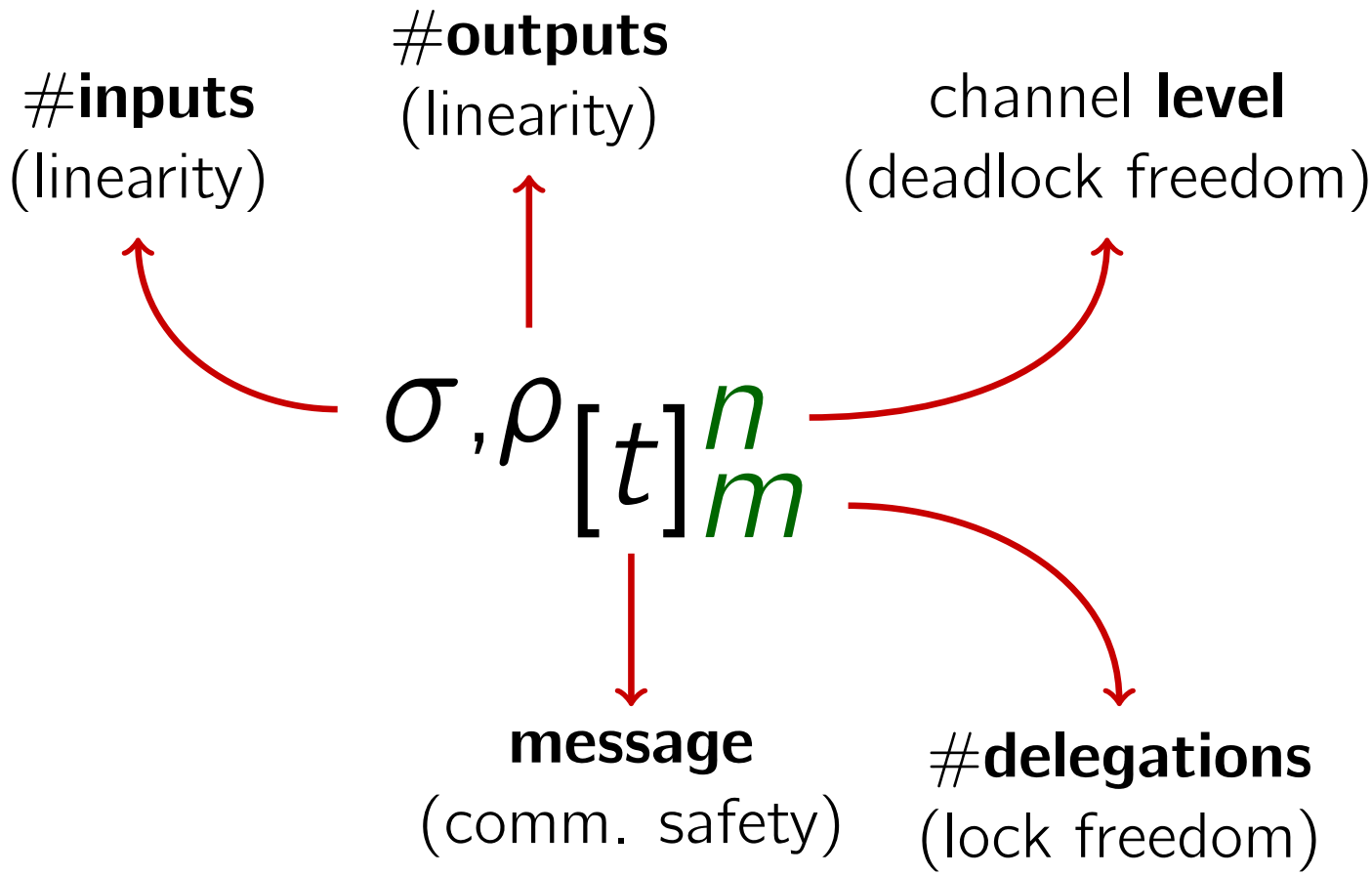
Information in channel types



Information in channel types



Information in channel types



Example: the server for mathematical operations

with sessions

```
*server?c.
 case c? of
 { Quit ⇒ {}
 ; Plus ⇒ c?x.
          c?y.
          c!(x + y).
          server!c
 ; Neg  ⇒ c?x.
          c!(0 - x).
          server!c }
```

Example: the server for mathematical operations

with sessions

```
*server?c.
 case c? of
 { Quit ⇒ {}
 ; Plus ⇒ c?x.
          c?y.
          c!(x + y).
          server!c
 ; Neg  ⇒ c?x.
          c!(0 - x).
          server!c }
```

with linear channels

```
*server?c.
 case c? of
 { Quit    ⇒ {}
 ; Plus c1 ⇒ c1?(x, c2).
             c2?(y, c3).
             new c4 in c3!(x + y, c4).
             server!c4
 ; Neg c1  ⇒ c1?(x, c2).
             new c3 in c2!(0 - x, c3).
             server!c3 }
```

Demo roadmap

1. server
   - linearity analysis
   - a glimpse at the constraints
   - protocol analysis
2. server + Alice
   - linearity/protocol analysis
   - deadlock analysis
   - lock analysis
3. server + (lazy) Alice + (lazy) Bob
   - deadlock analysis
   - lock analysis
4. Fibonacci (time permitting)
   - recursive (deadlock free but not lock free)
   - Kahn process network (lock free)

A few words (and numbers) on performances

Analysis of N-dimensional hypercubes with full-duplex channels

N	**Proc.**	**Chan.**	**Lin.**	**Gen.+Sat.**	**Levels**	**Tickets**	**Overall**
1	5	8	0.021	0.006	0.002	0.003	0.032
2	25	80	0.128	0.051	0.009	0.012	0.200
3	125	600	1.439	0.844	0.069	0.124	2.477
4	625	4000	33.803	26.422	1.116	3.913	65.254

- type reconstruction is exponential (cf. Mairson)
- integer programming is NP-hard
- >10GB memory use for $N = 4$

A few words (and numbers) on performances

Analysis of N-dimensional hypercubes with full-duplex channels

N	**Proc.**	**Chan.**	**Lin.**	**Gen.+Sat.**	**Levels**	**Tickets**	**Overall**
1	5	8	0.021	0.006	0.002	0.003	0.032
2	25	80	0.128	0.051	0.009	0.012	0.200
3	125	600	1.439	0.844	0.069	0.124	2.477
4	625	4000	33.803	26.422	1.116	3.913	65.254

- type reconstruction is exponential (cf. Mairson)
- integer programming is NP-hard
- $>$10GB memory use for $N = 4$

Paperware and software

- Padovani, **Deadlock and Lock Freedom in the Linear π-Calculus** (LICS'14)
- Padovani, **Type Reconstruction for the Linear π-Calculus with Composite and Equi-Recursive Types** (FoSSaCS'14)
- Padovani, Chen, Tosatto, **Type Reconstruction Algorithms for Deadlock-Free and Lock-Free Linear π-Calculi** (COORDINATION'15)
- Padovani and Tosatto, **Hypha** (`http://di.unito.it/hypha`)

`Slides, papers, links on my home page`