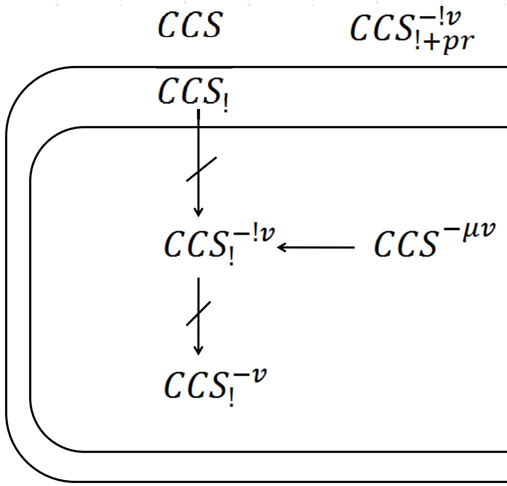
Discriminating the expressive power of process calculi through (un)decidability results



Gianluigi Zavattaro
FOCUS research team
University of Bologna/INRIA

General principle

- Consider:
  - two process calculi ***A*** and ***B***
  - a property ***c*** for ***A***, and a property ***d*** for ***B***
- If ***c*** is undecidable while ***d*** is decidable
  - there exists no computable encoding from ***A*** into ***B*** that maps ***c*** into ***d***
- When:
  - ***A*** and ***B*** are variants of the same process calculus
  - and ***c=d***

an expressiveness **gap** is proved between them

On the relationship between process calculi and WSTS

$$P, Q \ldots := 0 \mid \alpha.P \;\Big|\; P + Q \;\Big|\; P \mid Q \;\Big|\; (\nu a)P \;\Big|\; !P$$

$$\forall \quad \begin{tikzcd} s_1 \arrow[r, phantom, "\leq"] \arrow[d] & t_1 \arrow[d, dashed, "+"] \\ s_2 \arrow[r, phantom, "\leq"] & t_2 \end{tikzcd} \quad \exists$$

Gianluigi Zavattaro
FOCUS research team
University of Bologna/INRIA

Example

- The case of $CCS_!$ and CCS_{rec} [BGZ04]

$$P ::= \mathbf{0} \quad | \quad \alpha.P \quad | \quad P + P \quad | \quad P|P \quad | \quad (\nu x)P$$

$$\alpha ::= \tau \quad | \quad x \quad | \quad \overline{x}$$

$$P ::= \ldots \;|\; !P \qquad\qquad P ::= \ldots \;|\; recX.P \;|\; X$$

- In CCS_{rec} termination (of all computations) is undecidable while it is decidable for $CCS_!$

Some intuitions about the decidability proof

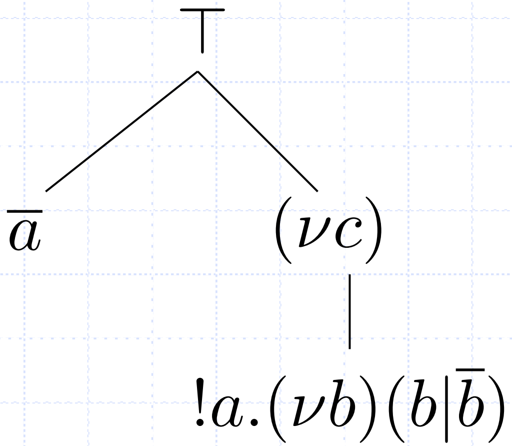
- Processes of $\text{CCS}_!$ can be seen as trees:

$$\overline{a}|(\nu c)!a.(\nu b)(b|\overline{b}) \rightarrow (\nu c)\big(!a.(\nu b)(b|\overline{b}) \mid (\nu b)(b|\overline{b})\big)$$

Some intuitions about the decidability proof

- Processes of $\text{CCS}_!$ can be seen as trees:

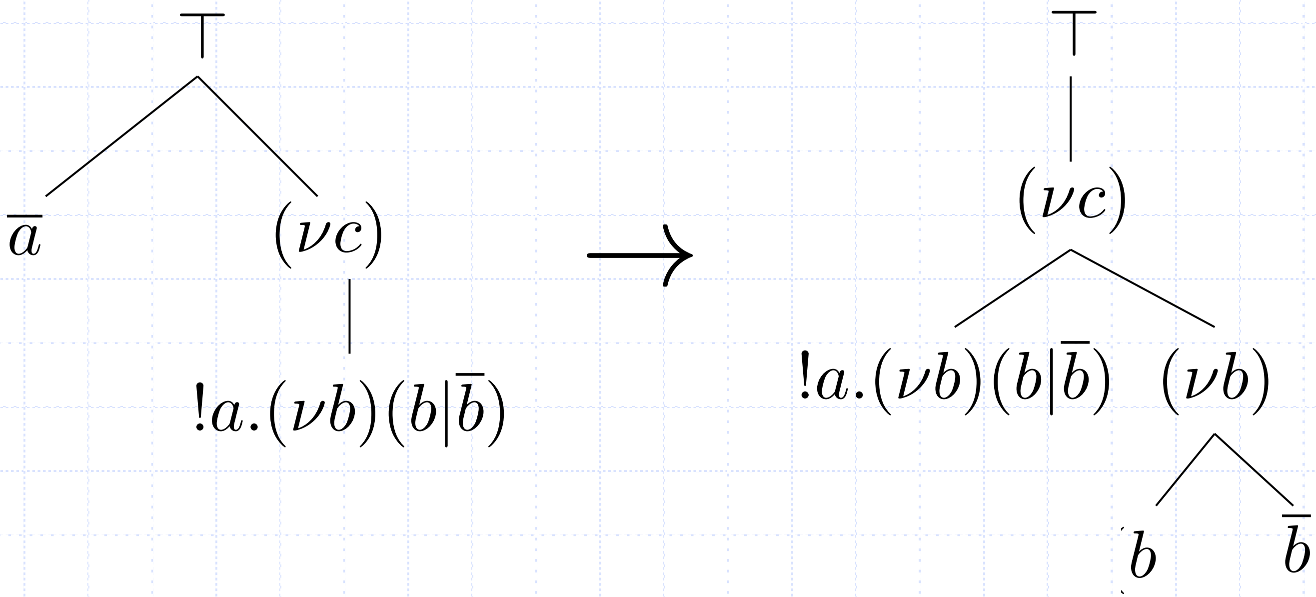
$$\overline{a}|(\nu c)!a.(\nu b)(b|\overline{b}) \rightarrow (\nu c)\big(!a.(\nu b)(b|\overline{b}) \mid (\nu b)(b|\overline{b})\big)$$



Some intuitions about the decidability proof

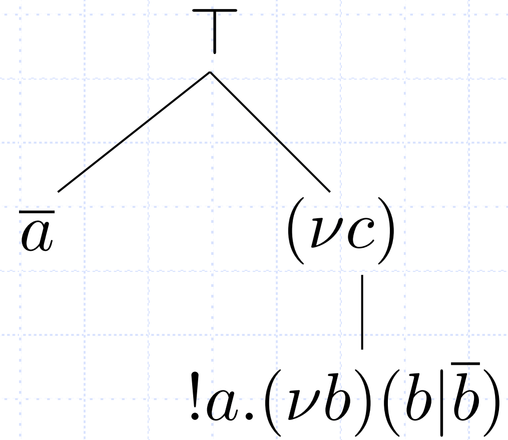
- Processes of CCS$_!$ can be seen as trees:

$$\overline{a}|(\nu c)!a.(\nu b)(b|\overline{b}) \rightarrow (\nu c)\big(!a.(\nu b)(b|\overline{b}) \mid (\nu b)(b|\overline{b})\big)$$



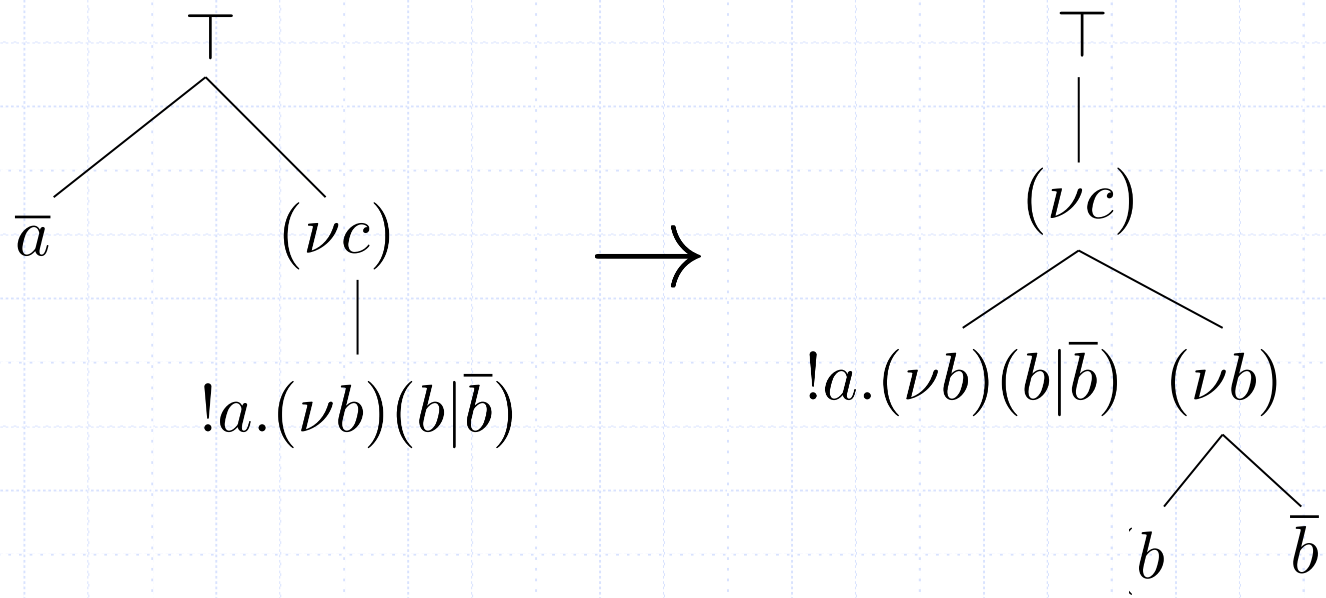
Some intuitions about the decidability proof

- Processes of $CCS_!$ can be seen as trees:
  - intermediary nodes are labeled with T or a restriction
  - leaves are labeled with sequential processes (top operator is neither parallel nor restriction)
  - a sequential process is son of its enclosing restriction (or T)



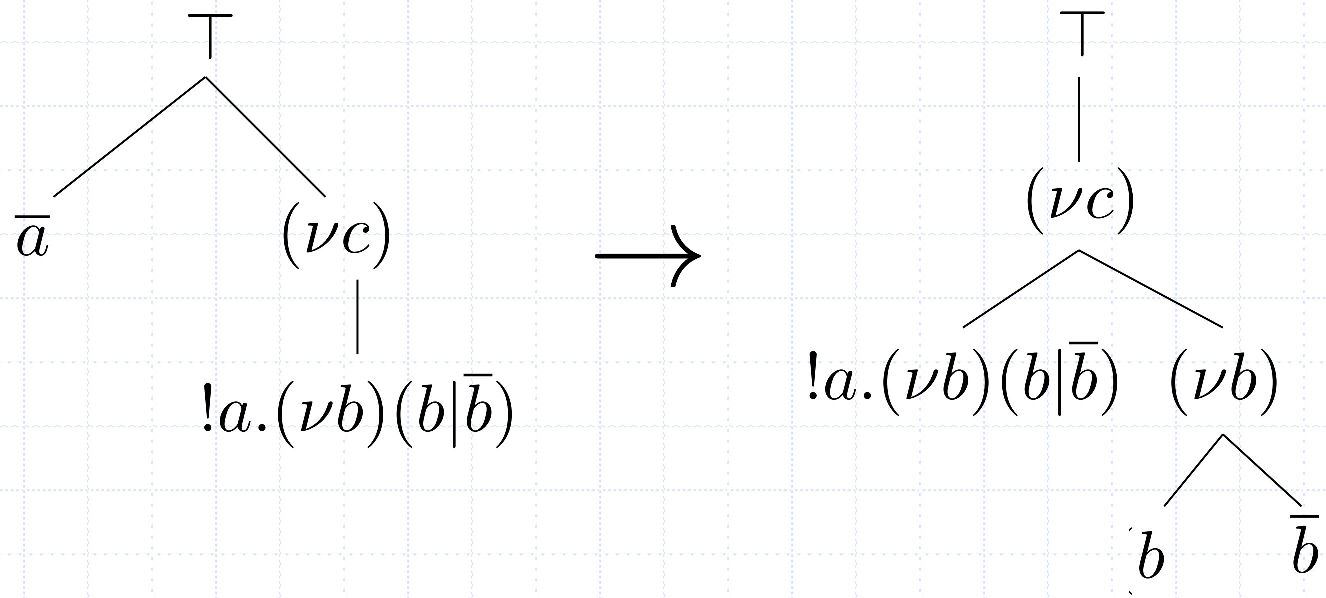
Some intuitions about the decidability proof

- Consider now a process *P*
  - its semantics can be seen as a transition system on trees



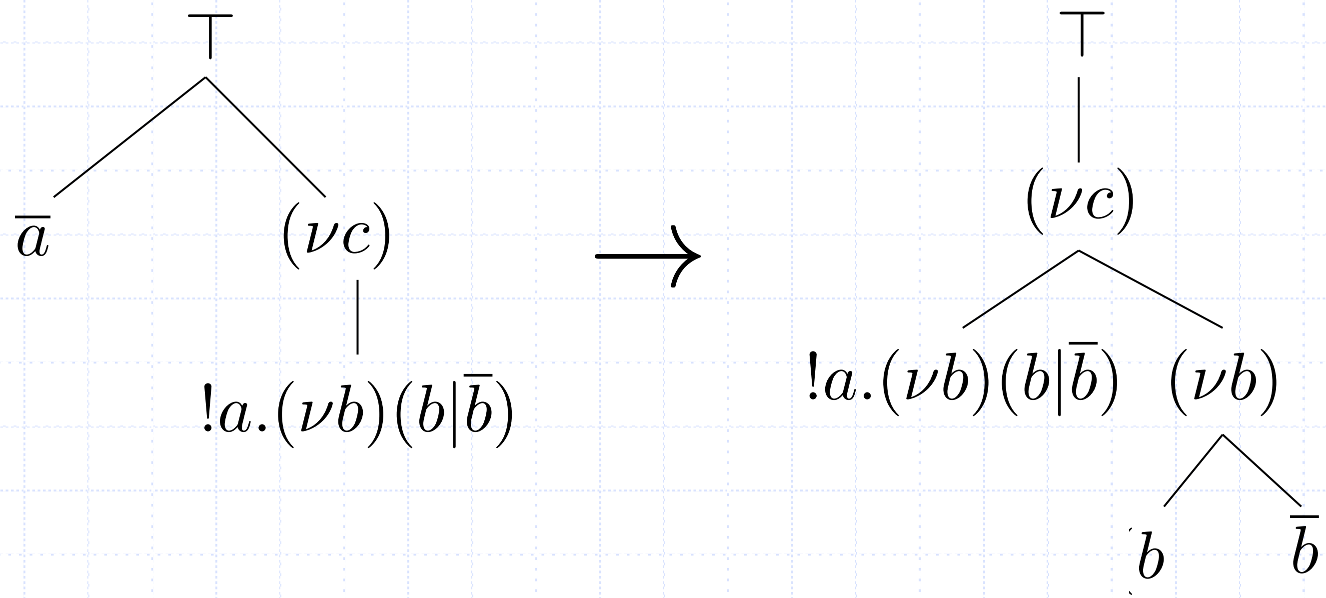
Some intuitions about the decidability proof

- Consider now a process *P*
  - the possible labels of trees are finite (finite names, finite sequential processes)



Some intuitions about the decidability proof

- Consider now a process *P*
  - all trees have a bounded depth (possibly unbounded width)



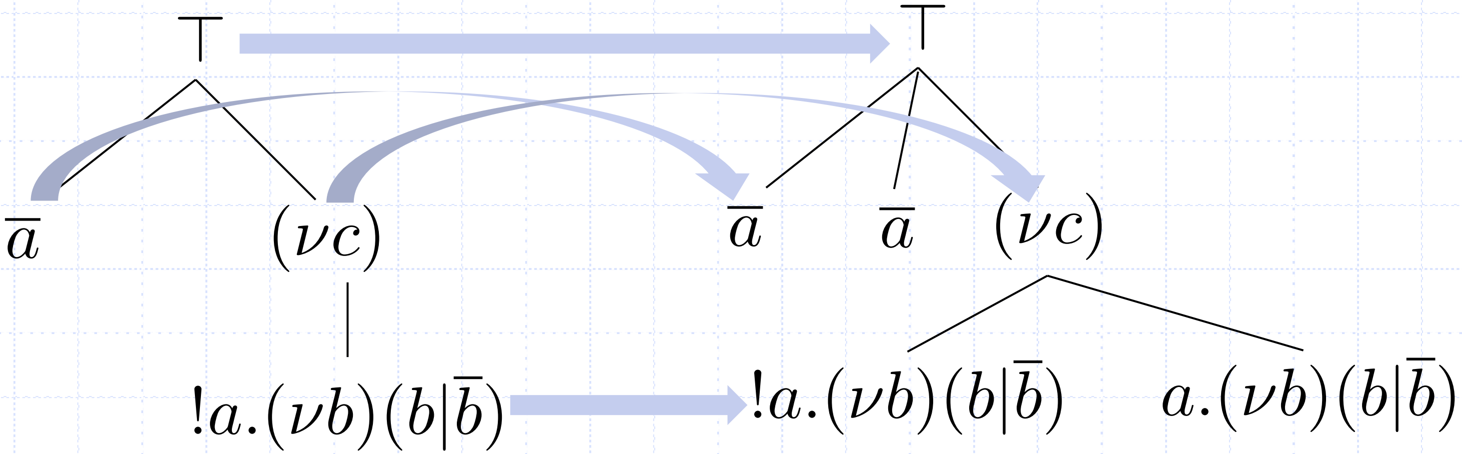
Well Quasi Ordering

- The set of trees
  - on a finite set of labels
  - with bounded depth

  has nice properties: it is a ***wqo*** for the ***rooted tree embedding*** ordering [Higman52]
- Well Quasi Ordering (wqo):
  - a reflexive and transitive relation $(S,\leq)$ is a wqo if given an infinite sequence $x_1,x_2,\ldots$ of elements in S, there exist $i<j$ s.t. $x_i \leq x_j$

Rooted tree embedding

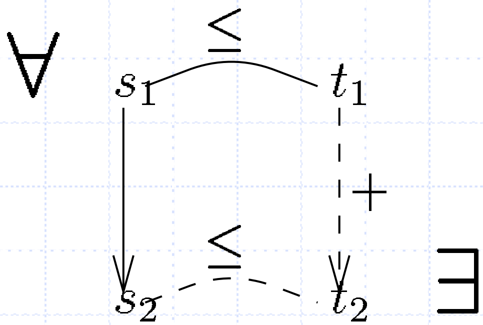
- Given two trees, the former can be embedded in the latter by keeping nodes at the same depth level



Well-Structured Transition Systems

- $(S, \rightarrow, \leq)$ is a WSTS if
  - $(S, \rightarrow)$ is a finitely branching transition system
  - $(S, \leq)$ is a wqo
  - Compatibility:

    for every $s_1 \rightarrow s_2$ and $s_1 \leq t_1$
    there exists $t_1 \rightarrow .. \rightarrow t_2$ s.t. $s_2 \leq t_2$



Infinite computations in WSTS

- s_0 has an infinite computation iff there exist $s_i \leq s_j$ s.t. $s_0 \rightarrow s_1 \rightarrow \ldots \rightarrow s_i \rightarrow \ldots \rightarrow s_j$
  - *only if* : follows from wqo
  - *if* : follows from compatibility
- WSTS are finitely branching:
  - so the existence of such s_i and s_j can be detected via a breadth-first search
- Conclusion: termination is decidable in WSTS (def: terminates iff no infinite computation)

Another example: HO^{f} [DPZ09]

- Variant of HOCORE, an asynchronous higher-order calculus (no restriction)
  - A process can be passed as it was received (without modifications)

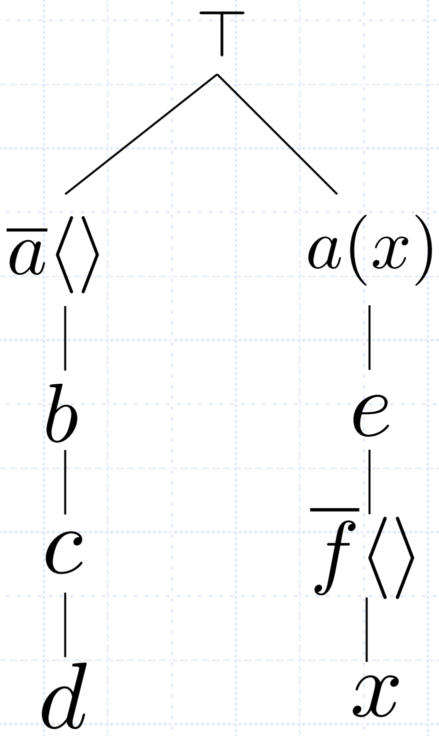
$$
\begin{array}{rcll}
P, Q & ::= & \overline{a}\langle x_1 \parallel \cdots \parallel x_k \parallel P \rangle & (\text{with } k \geq 0,\ \mathsf{fv}(P) = \emptyset) \\
& \mid & a(x).\,P & \\
& \mid & P \parallel Q & \\
& \mid & x & \\
& \mid & \mathbf{0} &
\end{array}
$$

Processes as trees

$$\overline{a}\langle b.c.d\rangle \parallel a(x).e.\overline{f}\langle x\rangle \;\rightarrow\; e.\overline{f}\langle b.c.d\rangle$$

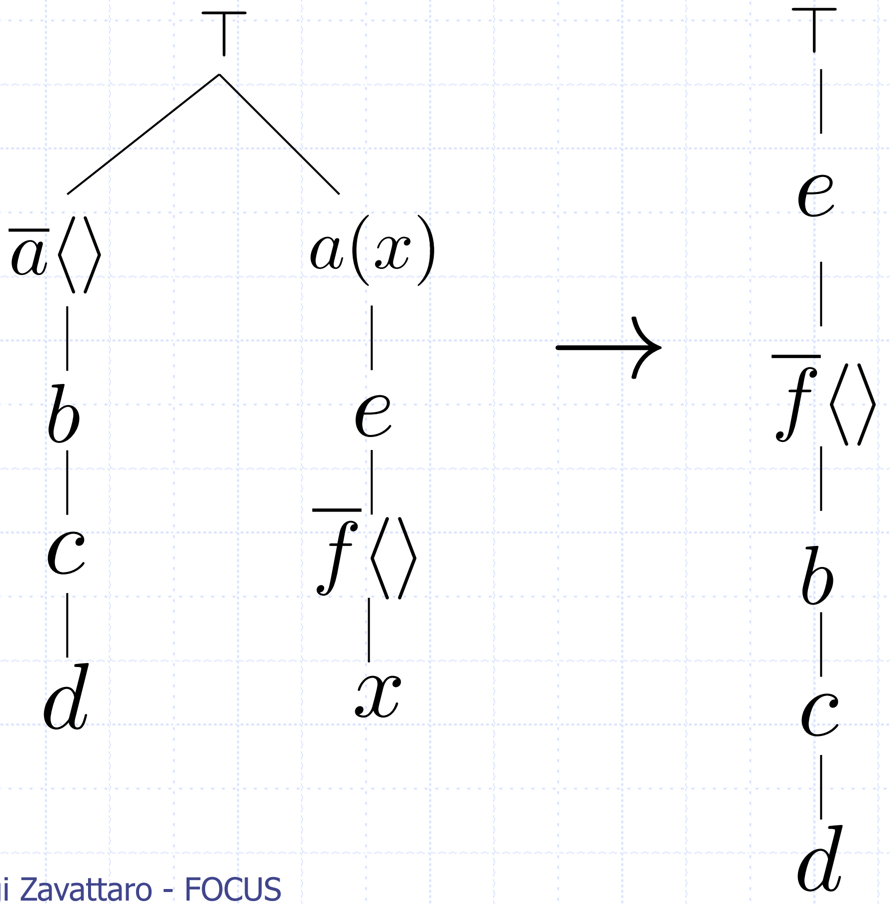
Processes as trees

$$\overline{a}\langle b.c.d\rangle \parallel a(x).e.\overline{f}\langle x\rangle \;\rightarrow\; e.\overline{f}\langle b.c.d\rangle$$



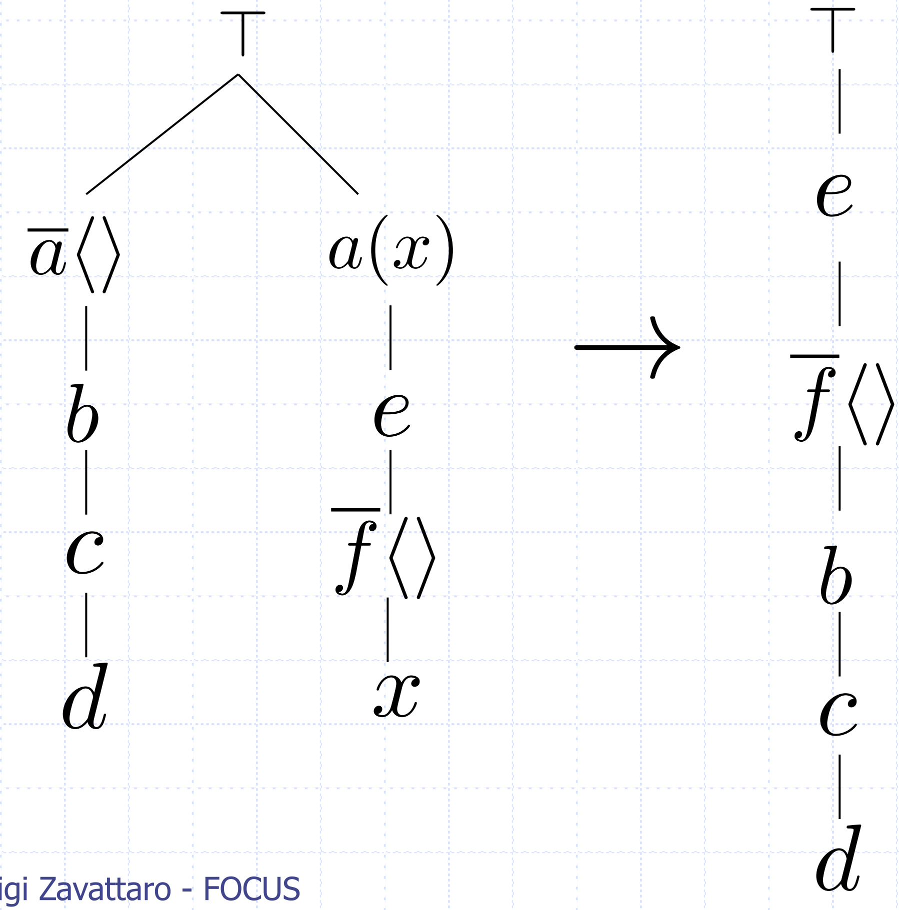
Processes as trees

$$\overline{a}\langle b.c.d\rangle \parallel a(x).e.\overline{f}\langle x\rangle \;\rightarrow\; e.\overline{f}\langle b.c.d\rangle$$



Processes as trees

$$\overline{a}\langle b.c.d\rangle \parallel a(x).e.\overline{f}\langle x\rangle \;\rightarrow\; e.\overline{f}\langle b.c.d\rangle$$



- As processes can only be forwarded, tree depth cannot grow indefinitely

What about HOCORE?

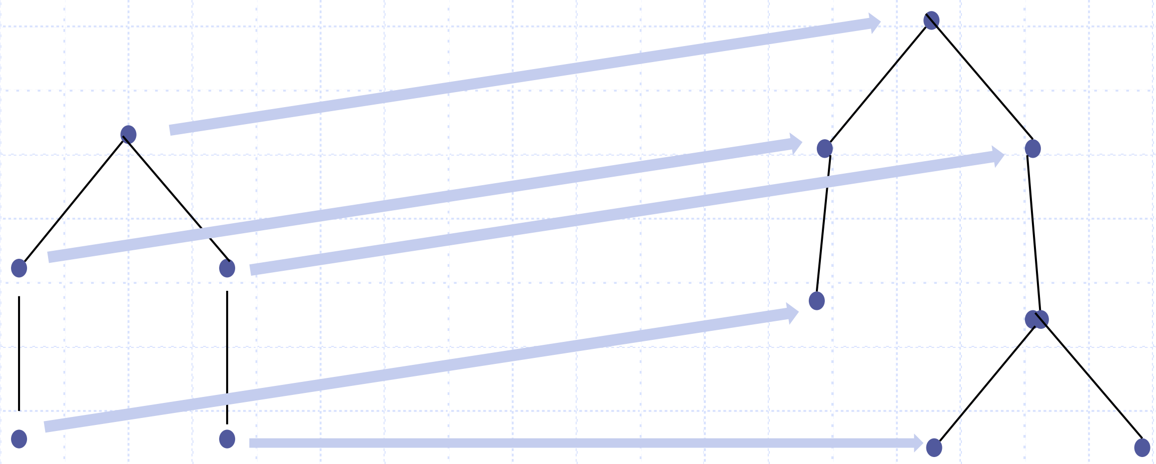
- In HOCORE processes of unbunded length can be reached

$$\overline{a}\langle\rangle \parallel\ !a(x).\overline{a}\langle b.x\rangle \ \rightarrow^{n}\ \overline{a}\langle \underbrace{b.\cdots.b}_{n}\rangle \parallel\ !a(x).\overline{a}\langle b.x\rangle$$

  - Hence, it is not always guaranteed that the corresponding trees have bounded depth

Trees with unbounded depth

- Treee embedding is a wqo for trees with unbounded depth [Kruskal60]



- Problem: this ordering breaks compatibility!
  - one process could be embedded in a 'different' one

Another example: CCS^{Δ} and CCS^{tc} [BZ09]

- CCS without restriction extended with operators for process interruption:
  - From CSP: Δ

$$\frac{P \xrightarrow{\alpha} P'}{P \triangle Q \xrightarrow{\alpha} P' \triangle Q} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P \triangle Q \xrightarrow{\alpha} Q'}$$

  - From usual programming lang.: *try-catch*

$$\frac{P \xrightarrow{\alpha} P' \quad \alpha \neq \mathtt{throw}}{\mathtt{try}\, P\, \mathtt{catch}\, Q \xrightarrow{\alpha} \mathtt{try}\, P'\, \mathtt{catch}\, Q} \qquad \frac{P \xrightarrow{\mathtt{throw}} P'}{\mathtt{try}\, P\, \mathtt{catch}\, Q \xrightarrow{\tau} Q}$$

Expressiveness gap

- In CCS^{tc} termination is undecidable
- It is decidable in CCS^{Δ} because tree embedding preserves compatibility:

$(P2|(P4\Delta Q2))\Delta Q1 \leq$

$P1 \mid (P2|P3|(P6|(P4\Delta Q2)\ \Delta Q3)|(P5\Delta Q3))\Delta Q1$

$$\begin{tikzcd}
\varepsilon \arrow[rr] \arrow[d] & & P1 \arrow[d] & \\
P2,\Delta Q1 \arrow[rr] \arrow[d] & & P2,P3,\Delta Q1 \arrow[dl] \arrow[dr] & \\
P4,\Delta Q2 \arrow[dr] & P6,\Delta Q3 \arrow[d] & & P5,\Delta Q3 \\
& P4,\Delta Q2 & &
\end{tikzcd}$$

Conclusion

- WSTS revealed as an interesting meta-model for capturing interesting (topological) properties of process calculi
  - e.g. in more recent works on wireless process calculi we considered orderings on graphs (induced subgraph ordering [Ding92])
- In many cases, WSTS allowed us to prove decidability of termination in calculi where an existential version of termination (at least one computation terminates) is undecidable
  - this holds for both $CCS_!$, Ho^f, and CCS^{Δ}

Example: nondeterministic RAM encoding in HO^f

$$[\![(1,0,0)]\!]_{\mathsf{M}} ::= \overline{p_1} \parallel \prod_{i=1}^{n} [\![(i:I_i)]\!]_{\mathsf{M}} \parallel loop.\,\textsc{Div} \parallel \overline{set_0}\langle \mathbf{0} \rangle \parallel \overline{set_1}\langle \mathbf{0} \rangle$$

$\textsc{Instructions}\ (i:I_i)$

$$\begin{aligned}
[\![(i:\mathtt{INC}(r_j))]\!]_{\mathsf{M}} &= !p_i.\,(\overline{u_j} \parallel set_j(x).\,\overline{set_j}\langle x \parallel \textsc{Inc}_j \rangle \parallel \overline{p_{i+1}}) \\
[\![(i:\mathtt{DECJ}(r_j,s))]\!]_{\mathsf{M}} &= \quad !p_i.\,\overline{m_i} \\
&\quad \parallel\ !m_i.\,(\overline{loop} \parallel u_j.\,loop.\,set_j(x).\,\overline{set_j}\langle x \parallel \textsc{Dec}_j \rangle \parallel \overline{p_{i+1}}) \\
&\quad \parallel\ !m_i.\,set_j(x).\,(x \parallel \overline{set_j}\langle \mathbf{0} \rangle \parallel \overline{p_s}))
\end{aligned}$$

where

$$\textsc{Inc}_j = \overline{loop} \parallel check_j.\,loop \qquad\qquad \textsc{Dec}_j = \overline{check_j}$$

Example: nondeterministic RAM encoding in CCS$^\Delta$

$$
\begin{aligned}
[\![(i, c_1, \ldots, c_n)]\!] \quad &= \\
&\overline{p_i} \mid [\![(1 : I_1)]\!] \mid \ldots \mid [\![(m : I_m)]\!] \mid \prod_{\sum_{j=1}^{n} c_j} \overline{loop} \mid LOOP \mid \\
&[\![r_1 = c_1]\!] \quad \mid \ldots \mid [\![r_n = c_n]\!] \quad \mid !nr_1.[\![r_1 = 0]\!] \quad \mid \ldots \mid !nr_n.[\![r_n = 0]\!]
\end{aligned}
$$

$$
[\![(i : I_i)]\!] \quad : \quad !p_i.(\overline{inc_j}.\overline{loop} \mid \overline{p_{i+1}}) \qquad \text{if } I_i = Succ(r_j)
$$

$$
\begin{aligned}
[\![(i : I_i)]\!] \quad : \quad & !p_i.(\ \tau.(\overline{loop} \mid \overline{dec_j}.loop.loop.\overline{p_{i+1}}) + \\
& \qquad \tau.\overline{zero_j}.ack.\overline{p_s}\ ) \qquad \text{if } I_i = DecJump(r_j, s)
\end{aligned}
$$

$$
[\![r_j = c_j]\!] \quad : \quad (!inc_j.dec_j \mid \prod_{c_j} dec_j\ ) \triangle (zero_j.\overline{nr_j}.\overline{ack})
$$

$$
LOOP \quad : \quad loop.(\bar{l} \mid !l.\bar{l})
$$