The Problem of Observable Value in Interaction Nets

Southern and Midlands Logic Seminar

Marc Thatcher

University of Sussex

UCL, 24^{th} June 2026

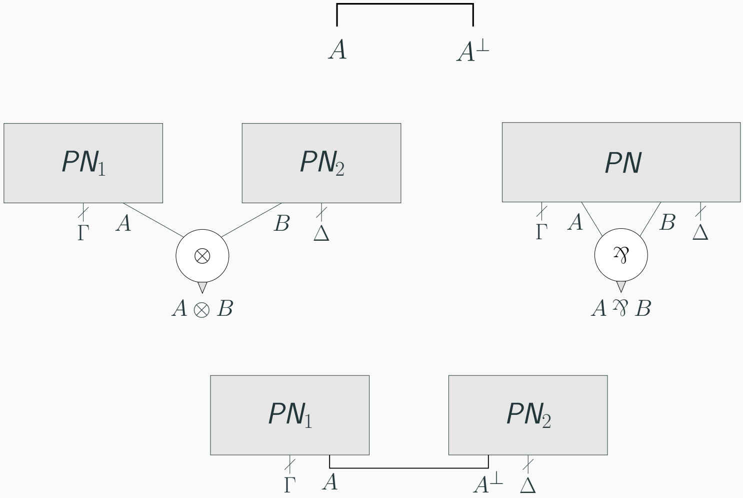
Multiplicative Linear Logic

$$\dfrac{}{\vdash A, A^\perp}\,\mathsf{ax}$$

$$\dfrac{\vdash \Gamma, A \qquad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\,\otimes \qquad\qquad \dfrac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, A \parr B, \Delta}\,\parr$$

$$\dfrac{\vdash \Gamma, A \qquad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta}\,\mathsf{cut}$$

Multiplicative Linear Logic Proof Nets

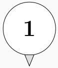


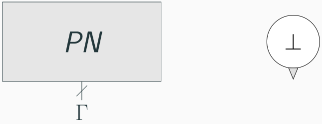
$+$ cut elimination: ax $-$ cut , $\otimes - \parr$

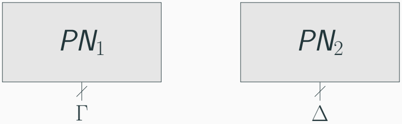
Units and Mix

$$\frac{}{\vdash 1}\ 1 \qquad\qquad \frac{\vdash \Gamma}{\vdash \bot, \Gamma}\ \bot$$

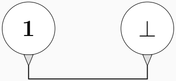
$$\frac{}{\vdash}\ \mathsf{mix}_0 \qquad\qquad \frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta}\ \mathsf{mix}_2$$


1

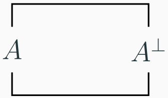

PN
Γ
⊥

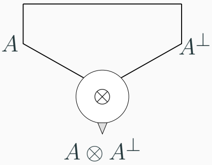

PN1
Γ
PN2
Δ

Units and Mix Proof Nets — Causes Problems



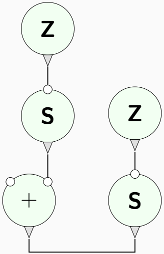
$\Rightarrow$?


A
$A^\perp$


A
$A^\perp$
$\otimes$
$A \otimes A^\perp$

Interaction nets

Interaction nets generalise MLL+units+mix proof structures with definable node labels and cut (*rewrite*) rules.



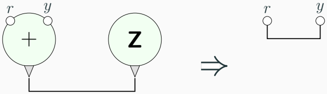
Interaction net = finite port graph ; nodes (*agent*) + edges (*wires*).

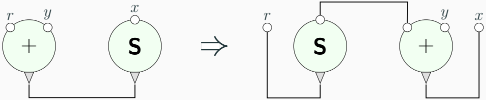
Agent label $\in \Sigma$; $ar : \Sigma \to \mathbb{N}$; $ar(\sigma)$ *auxiliary* ports .

May be cyclic, empty (mix_0), disjoint (mix_2).

Rewrite (cut) rules defined between *principal* ports.

Interaction rules replace cut



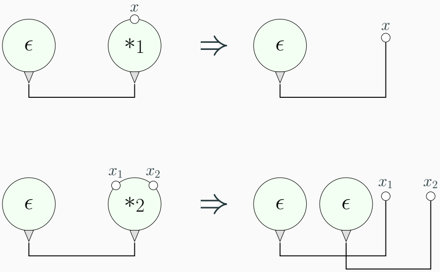


LHS (*active pair*) $=$ two agents connected only by principal port.

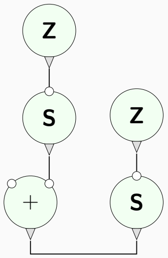
RHS $=$ net that preserves LHS's *interface* (set of free ports).

LHSs are unique.

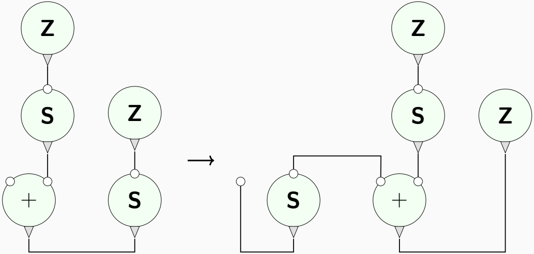
Interaction rules—erasure



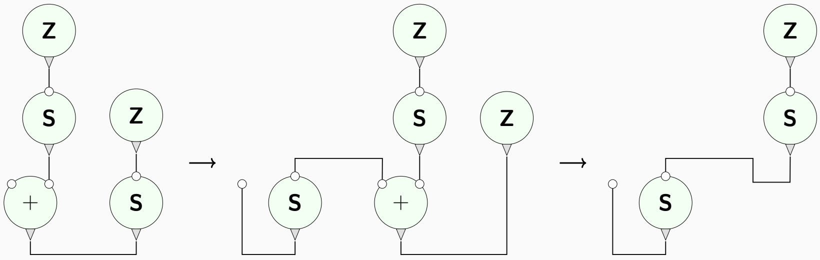
Interaction net evaluation



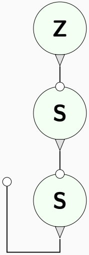
Interaction net evaluation



Interaction net evaluation



Result = normal form



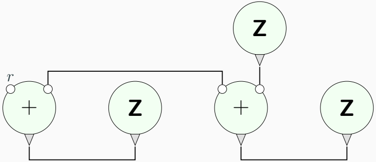
$$N \Downarrow M \iff N \to^* M \text{ and } \nexists M' \text{ s.t. } M \to M'$$

Interaction nets as programs

We wish to interpret interaction nets as programs.

Why?

- Turing complete.
- Explicit resource management.
- Parallel evaluation.



  - Hence `INPLA`, `Higher Order Co`, `tendrils`.

We will assume parallel rewrite steps.

Interaction nets as programs

Questions:

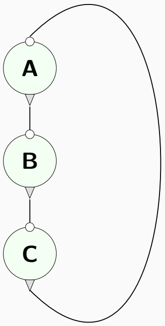
- What is input? Output?
- What are data?
- What are computations?
- What are results?

Recall:

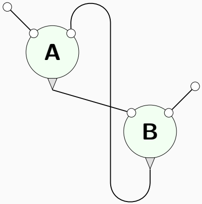
1. We can only reduce (compute) when there is an active pair.
2. Result of evaluation is normal form.

Problematic result #1

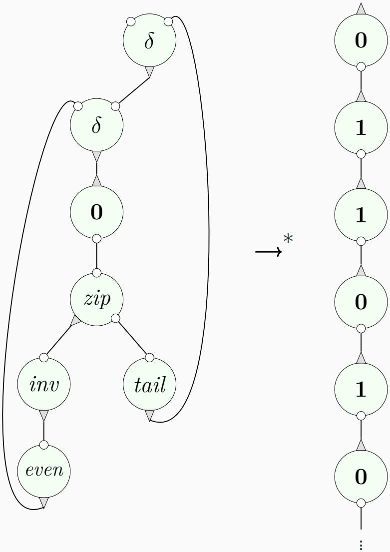
Problematic result #2



Problematic result #3



Problematic non-result



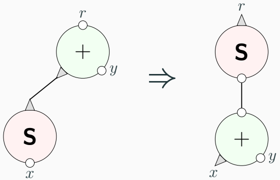
Function-constructor nets

$\Sigma = \Sigma_{\mathcal{F}} \cup \Sigma_{\mathcal{C}}$

$\Sigma_{\mathcal{F}} \cap \Sigma_{\mathcal{C}} = \varnothing$

$\Sigma_{\mathcal{F}} = \{\epsilon/0,\ \mathsf{id}/1,\ \delta/2,\ \ldots,\ F/n_F\}$ — principal port is *input*.

$\Sigma_{\mathcal{C}} = \{\mathsf{Z}/0,\ \mathsf{S}/1,\ :/2,\ \ldots,\ \ C/n_C\}$ — principal port is (only) *output*.

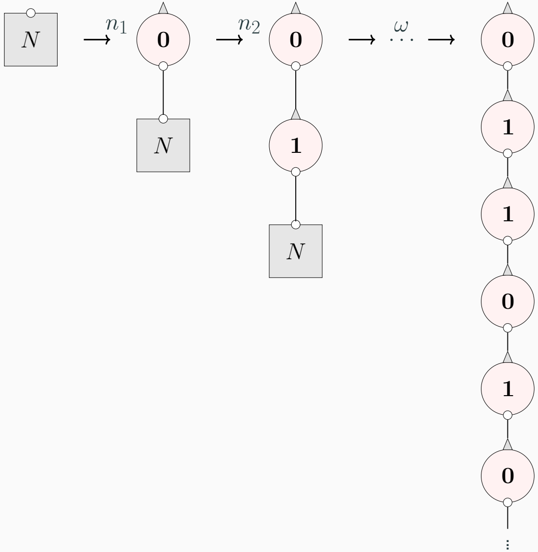


First-order system: constructors are function inputs and program outputs.

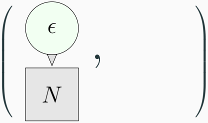
$\implies$ interface, $I = I_{in} \cup I_{out}$; $I_{in} \cap I_{out} = \varnothing$.

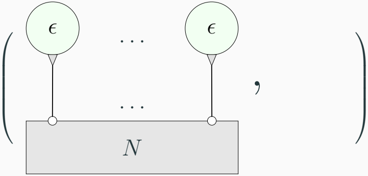
Consider nets with $I_{in} = \varnothing$.

Result = observational output

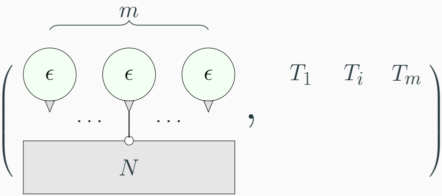


Function-constructor net semantics—set up

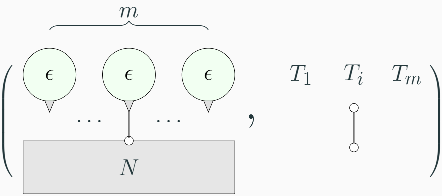




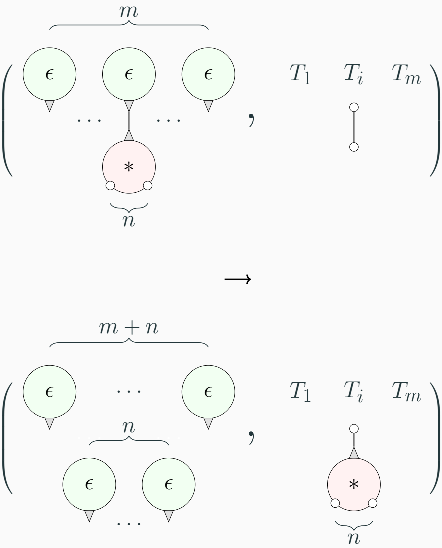
Function-constructor net semantics—dynamics #1



$\longrightarrow$



Function-constructor net semantics—dynamics #2



Observable normal form

Definition (Port observable normal form)
Let N be an interaction net with m output ports

and

N_ϵ be N with an ϵ agent attached to each output port.

Let $T_i^n(N)$ be the constructor tree accumulated at port i after n parallel reduction steps.

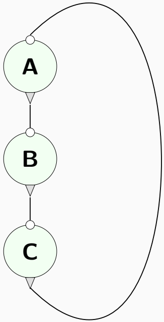
The *port observable normal form* of N at port i is

$$T_i(N) = \lim_{n \to \omega} T_i^n(N)$$

Definition (Observable normal form)
The *observable normal form* of net N is the forest of observable port normal forms.

Observable normal forms

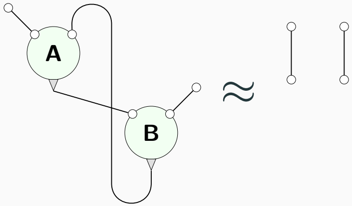
1. Inaccessible nets



$\approx$

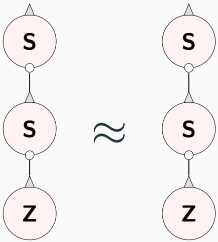
Observable normal forms

2. Vicious circles



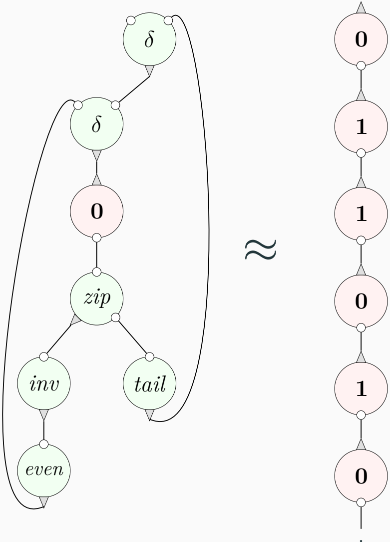
Observable normal forms

3. Finite (productive) nets



Observable normal forms

4. Infinite (productive) nets



How determine productivity?

How determine productivity?

Read thesis; chapter 5!