LUDICS AND LOGICAL COMPLETENESS

Geometry of Interaction, Traced Monoidal Categories and Implicit Complexity Workshop, Kyoto, Japan.
28 August 2009

Completeness (Gödel 1929)

Duality proof — countermodels :

- *either* there exists a proof P such that $\vdash A$ is provable;
- *or* there exists a countermodel $\mathcal{M}$ such that $\mathcal{M} \models \neg A$.

One can *imagine* a debate on a general proposition A, where

- Player tries to justify A by giving a proof;
- Opponent tries to refute it by giving a countermodel.
- The completeness theorem states that exactly one of them wins.

Completeness (Gödel 1929)

Duality proof — countermodels :

- *either* there exists a proof P such that $\vdash A$ is provable;
- *or* there exists a countermodel $\mathcal{M}$ such that $\mathcal{M} \models \neg A$.

One can *imagine* a debate on a general proposition A, where

- Player tries to justify A by giving a proof;
- Opponent tries to refute it by giving a countermodel.
- The completeness theorem states that exactly one of them wins.

Completeness (Gödel 1929)

Duality proof — countermodels :

- *either* there exists a proof P such that $\vdash A$ is provable;
- *or* there exists a countermodel $\mathcal{M}$ such that $\mathcal{M} \models \neg A$.

One can *imagine* a debate on a general proposition A, where

- Player tries to justify A by giving a proof;
- Opponent tries to refute it by giving a countermodel.
- The completeness theorem states that exactly one of them wins.

Completeness (Gödel 1929)

Duality proof — countermodels :

- *either* there exists a proof P such that $\vdash A$ is provable;
- *or* there exists a countermodel $\mathcal{M}$ such that $\mathcal{M} \models \neg A$.

One can *imagine* a debate on a general proposition A, where

- Player tries to justify A by giving a proof;
- Opponent tries to refute it by giving a countermodel.
- The completeness theorem states that exactly one of them wins.

Completeness (Gödel 1929)

Duality proof — countermodels :

- *either* there exists a proof P such that $\vdash A$ is provable;
- *or* there exists a countermodel $\mathcal{M}$ such that $\mathcal{M} \models \neg A$.

One can *imagine* a debate on a general proposition A, where

- Player tries to justify A by giving a proof;
- Opponent tries to refute it by giving a countermodel.
- The completeness theorem states that exactly one of them wins.

Completeness (Gödel 1929)

Duality proof — countermodels :

- *either* there exists a proof P such that $\vdash A$ is provable;
- *or* there exists a countermodel $\mathcal{M}$ such that $\mathcal{M} \models \neg A$.

One can *imagine* a debate on a general proposition A, where

- Player tries to justify A by giving a proof;
- Opponent tries to refute it by giving a countermodel.
- The completeness theorem states that exactly one of them wins.

Proofs,Models,Completeness

Proofs:

- Finite.
- Provability defined by **induction on *proofs***.

Models:

- Infinite: arbitrary cardinality.
- Non standard models (Löwenheim — Skolem, Compactness Theorem).
- Satisfiability defined by **induction on *formulas***.

Completeness proof:

- Nondeterministic principles: König Lemma (Schütte), Zorn's Lemma (Henkin).

but . . . there is no (clear) interaction between proofs and models . . . .

Proofs,Models,Completeness

Proofs:

- Finite.
- Provability defined by **induction on *proofs***.

Models:

- Infinite: arbitrary cardinality.
- Non standard models (Löwenheim — Skolem, Compactness Theorem).
- Satisfiability defined by **induction on *formulas***.

Completeness proof:

- Nondeterministic principles: König Lemma (Schütte), Zorn's Lemma (Henkin).

but . . . there is no (clear) interaction between proofs and models . . . .

Proofs,Models,Completeness

Proofs:

- Finite.
- Provability defined by **induction on *proofs***.

Models:

- Infinite: arbitrary cardinality.
- Non standard models (Löwenheim — Skolem, Compactness Theorem).
- Satisfiability defined by **induction on *formulas***.

Completeness proof:

- Nondeterministic principles: König Lemma (Schütte), Zorn's Lemma (Henkin).

but . . . there is no (clear) interaction between proofs and models . . . .

Proofs,Models,Completeness

Proofs:

- Finite.
- Provability defined by **induction on *proofs***.

Models:

- Infinite: arbitrary cardinality.
- Non standard models (Löwenheim — Skolem, Compactness Theorem).
- Satisfiability defined by **induction on *formulas***.

Completeness proof:

- Nondeterministic principles: König Lemma (Schütte), Zorn's Lemma (Henkin).

but . . . there is no (clear) interaction between proofs and models . . . .

An interactive account of completeness

- We are interested in (models of) proofs rather than provability.
- QUESTION : What about the duality proofs — countermodels in Girard's ludics?
  ANSWER : *Proofs and models are objects of the same kind (designs) only distinguished by their structural properties.*

An interactive account of completeness

- We are interested in (models of) proofs rather than provability.
- QUESTION : What about the duality proofs — countermodels in Girard's ludics?

  ANSWER : *Proofs and models are objects of the same kind (designs) only distinguished by their structural properties.*

An interactive account of completeness

- We are interested in (models of) proofs rather than provability.
- QUESTION : What about the duality proofs — countermodels in Girard's ludics?
  ANSWER : *Proofs and models are objects of the same kind (designs) only distinguished by their structural properties.*

Completeness revisited (ludics, game semantics)

For any logical behaviour $\mathbf{A}$ (semantical type) and for any design P either:

- *either* P is a proof of $\vdash \mathbf{A}$, or
- there exists a model $M \models \mathbf{A}^{\perp}$ which *rejects* P.

M rejects P means that $M \not\perp P$ and hence, $P \notin \mathbf{A}$.

Proofs : Finite, deterministic, ✠-free *designs*

Models : Infinite, nondeterministic, linear *designs*

Completeness proof : a real interaction between proofs and models.

Completeness revisited (ludics, game semantics)

For any logical behaviour $\mathbf{A}$ (semantical type) and for any design P either:

- *either* P is a proof of $\vdash \mathbf{A}$, or
- there exists a model $M \models \mathbf{A}^{\perp}$ which *rejects* P.

M rejects P means that $M \not\perp P$ and hence, $P \notin \mathbf{A}$.

Proofs : Finite, deterministic, ✠-free *designs*

Models : Infinite, nondeterministic, linear *designs*

Completeness proof : a real interaction between proofs and models.

Completeness revisited (ludics, game semantics)

For any logical behaviour **A** (semantical type) and for any design P either:

- *either* P is a proof of $\vdash \mathbf{A}$, or
- there exists a model $M \models \mathbf{A}^{\perp}$ which *rejects* P.

M rejects P means that $M \not\perp P$ and hence, $P \notin \mathbf{A}$.

Proofs : Finite, deterministic, ✠-free *designs*

Models : Infinite, nondeterministic, linear *designs*

Completeness proof : a real interaction between proofs and models.

Completeness revisited (ludics, game semantics)

For any logical behaviour **A** (semantical type) and for any design P either:

- *either* P is a proof of $\vdash \mathbf{A}$, or
- there exists a model $M \models \mathbf{A}^{\perp}$ which *rejects* P.

M rejects P means that $M \not\perp P$ and hence, $P \notin \mathbf{A}$.

Proofs : Finite, deterministic, ✠-free *designs*

Models : Infinite, nondeterministic, linear *designs*

Completeness proof : a real interaction between proofs and models.

Completeness revisited (ludics, game semantics)

For any logical behaviour **A** (semantical type) and for any design P either:

- *either* P is a proof of $\vdash \mathbf{A}$, or
- there exists a model $M \models \mathbf{A}^{\perp}$ which *rejects* P.

M rejects P means that $M \not\perp P$ and hence, $P \notin \mathbf{A}$.

Proofs : Finite, deterministic, ✠-free *designs*

Models : Infinite, nondeterministic, linear *designs*

Completeness proof : a real interaction between proofs and models.

In this talk:

- We show a completeness result: ludics is a model for a variant of (propositional) polarized linear logic (with exponentials) = a constructive version of classical propositional logic.
- ...but before that: we explain what ludics is!

In this talk:

- We show a completeness result: ludics is a model for a variant of (propositional) polarized linear logic (with exponentials) = a constructive version of classical propositional logic.
- ...but before that: we explain what ludics is!

What is ludics? (I)

A purely interactive approach to logic.

Ludics arose as the study of the interaction between syntax and syntax, typically in cut-elimination. It was necessary to replace syntax with something more geometrical, and this is why ludics lies between syntax and semantics, as a 'semantics of syntax-as-syntax', a monist explanation of logic. The thesis of ludics, which was already present in the programmatic paper [Towards a geometry of interaction], is that logic reflects the hidden geometrical properties of something.

J.-Y. Girard, Locus Solum (2001).

What is ludics? (I)

A purely interactive approach to logic.

Ludics arose as the study of the interaction between syntax and syntax, typically in cut-elimination. It was necessary to replace syntax with something more geometrical, and this is why ludics lies between syntax and semantics, as a 'semantics of syntax-as-syntax', a monist explanation of logic. The thesis of ludics, which was already present in the programmatic paper [Towards a geometry of interaction], is that logic reflects the hidden geometrical properties of something.

J.-Y. Girard, Locus Solum (2001).

What is ludics? (II)

- Monism: An uniform framework in which syntax (proofs) and semantics (counterproofs, models) can be uniformly expressed.
- Designs: Untyped paraproofs
  - "untyped" : proofs from which the logical content has been almost erased.
  - "para" : proofs which might contain errors and might be incomplete.
- Interaction : Designs interact together via normalization which induces an orthogonality relation $\perp$ between designs in such a way that $P \perp M$ holds if the normalization of P applied to M terminates.
  - A proof P and "its model" $P^{\perp} := \{N : P \perp N\}$.
  - An automaton A and a datum D : A accepts D iff $A \perp D$.

What is ludics? (II)

- Monism: An uniform framework in which syntax (proofs) and semantics (counterproofs, models) can be uniformly expressed.
- Designs: Untyped paraproofs
  - "untyped" : proofs from which the logical content has been almost erased.
  - "para" : proofs which might contain errors and might be incomplete.
- Interaction : Designs interact together via normalization which induces an orthogonality relation $\perp$ between designs in such a way that $P \perp M$ holds if the normalization of P applied to M terminates.
  - A proof P and "its model" $P^{\perp} := \{N : P \perp N\}$.
  - An automaton A and a datum D : A accepts D iff $A \perp D$.

What is ludics? (II)

- Monism: An uniform framework in which syntax (proofs) and semantics (counterproofs, models) can be uniformly expressed.
- Designs: Untyped paraproofs
  - "untyped" : proofs from which the logical content has been almost erased.
  - "para" : proofs which might contain errors and might be incomplete.
- Interaction : Designs interact together via normalization which induces an orthogonality relation $\perp$ between designs in such a way that $P \perp M$ holds if the normalization of P applied to M terminates.
  - A proof P and "its model" $P^{\perp} := \{N : P \perp N\}$.
  - An automaton A and a datum D : A accepts D iff $A \perp D$.

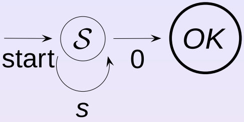
What is ludics? (II)

- Monism: An uniform framework in which syntax (proofs) and semantics (counterproofs, models) can be uniformly expressed.
- Designs: Untyped paraproofs
  - “untyped” : proofs from which the logical content has been almost erased.
  - “para” : proofs which might contain errors and might be incomplete.
- Interaction : Designs interact together via normalization which induces an orthogonality relation $\perp$ between designs in such a way that $P \perp M$ holds if the normalization of P applied to M terminates.
  - A proof P and “its model” $P^{\perp} := \{N : P \perp N\}$.
  - An automaton A and a datum D : A accepts D iff $A \perp D$.

What is ludics? (II)

- Monism: An uniform framework in which syntax (proofs) and semantics (counterproofs, models) can be uniformly expressed.
- Designs: Untyped paraproofs
  - "untyped" : proofs from which the logical content has been almost erased.
  - "para" : proofs which might contain errors and might be incomplete.
- Interaction : Designs interact together via normalization which induces an orthogonality relation $\perp$ between designs in such a way that $P \perp M$ holds if the normalization of P applied to M terminates.
  - A proof P and "its model" $P^{\perp} := \{N : P \perp N\}$.
  - An automaton A and a datum D : A accepts D iff $A \perp D$.

Example

$A =$ 

$$n = \underbrace{sssss \ldots s}_{n \text{ times}} 0$$

A dialogue between the automata and the datum.

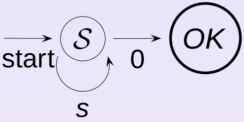
$$\begin{array}{rcl}
\mathrm{A} & := & x|\overline{\mathcal{S}}\langle \mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A}\rangle \\
0 & := & \mathcal{S}(x).x|\overline{\mathrm{zero}} \\
\mathrm{N}+1 & := & \mathcal{S}(x).x|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle
\end{array}$$

$$\begin{array}{rcl}
\mathrm{A}[0/x] & = & (\mathcal{S}(x).x|\overline{\mathrm{zero}})|\overline{\mathcal{S}}\langle \mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A}\rangle \\
& \longrightarrow & (\mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A})|\overline{\mathrm{zero}} \\
& \longrightarrow & OK\,.
\end{array}$$

$$\begin{array}{rcl}
\mathrm{A}[\mathrm{N}+1/x] & = & (\mathcal{S}(x).x|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle)|\overline{\mathcal{S}}\langle \mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A}\rangle \\
& \longrightarrow & (\mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A})|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle \\
& \longrightarrow & \mathrm{A}[\mathrm{N}/x].
\end{array}$$

Example

$A =$



$$n = \underbrace{sssss \ldots s}_{n \text{ times}} 0$$

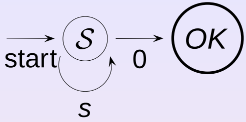
A dialogue between the automata and the datum.

$$\begin{array}{rcl} \mathrm{A} & := & x|\overline{\mathcal{S}}\langle \mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A}\rangle \\ 0 & := & \mathcal{S}(x).x|\overline{\mathrm{zero}} \\ \mathrm{N}+1 & := & \mathcal{S}(x).x|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle \end{array}$$

$$\begin{array}{rcl} \mathrm{A}[0/x] & = & (\mathcal{S}(x).x|\overline{\mathrm{zero}})|\overline{\mathcal{S}}\langle \mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A}\rangle \\ & \longrightarrow & (\mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A})|\overline{\mathrm{zero}} \\ & \longrightarrow & OK\,. \end{array}$$

$$\begin{array}{rcl} \mathrm{A}[\mathrm{N}+1/x] & = & (\mathcal{S}(x).x|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle)|\overline{\mathcal{S}}\langle \mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A}\rangle \\ & \longrightarrow & (\mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A})|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle \\ & \longrightarrow & \mathrm{A}[\mathrm{N}/x]. \end{array}$$

Example

$A =$ 

$$n = \underbrace{sssss\ldots s}_{n \text{ times}} 0$$

A dialogue between the automata and the datum.

$$\begin{array}{rcl}
\mathrm{A} & := & x|\overline{\mathcal{S}}\langle \mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A}\rangle \\
0 & := & \mathcal{S}(x).x|\overline{\mathrm{zero}} \\
\mathrm{N}+1 & := & \mathcal{S}(x).x|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle
\end{array}$$

$$\begin{array}{rcl}
\mathrm{A}[0/x] & = & \big(\mathcal{S}(x).x|\overline{\mathrm{zero}}\big)|\overline{\mathcal{S}}\langle \mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A}\rangle \\
& \longrightarrow & (\mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A})|\overline{\mathrm{zero}} \\
& \longrightarrow & OK\,.
\end{array}$$

$$\begin{array}{rcl}
\mathrm{A}[\mathrm{N}+1/x] & = & \big(\mathcal{S}(x).x|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle\big)|\overline{\mathcal{S}}\langle \mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A}\rangle \\
& \longrightarrow & \big(\mathrm{zero}.OK + \mathrm{succ}(x).\mathrm{A}\big)|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle \\
& \longrightarrow & \mathrm{A}[\mathrm{N}/x].
\end{array}$$

What is ludics? (III)

The core of ludics : focalization

Positive	Negative
$\otimes$	$⅋$
$\oplus$	$\&$
$\mathbf{0}$	$\top$
$\mathbf{1}$	$\bot$
$?$	$!$

- Negative = reversible, deterministic: $\dfrac{\vdash \Sigma, A, B}{\vdash \Sigma, A ⅋ A} \Updownarrow$
- Positive = irreversible, nondeterministic: $\dfrac{\vdash \Sigma_1, A \qquad \vdash \Sigma_2, B}{\vdash \Sigma, A \otimes B} \Downarrow$

What is ludics? (III)

The core of ludics : focalization

Positive	Negative
$\otimes$	$\parr$
$\oplus$	$\&$
$\mathbf{0}$	$\top$
$\mathbf{1}$	$\bot$
$?$	$!$

- Negative = reversible, deterministic: $\dfrac{\vdash \Sigma, A, B}{\vdash \Sigma, A \parr A} \Updownarrow$
- Positive = irreversible, nondeterministic: $\dfrac{\vdash \Sigma_1, A \qquad \vdash \Sigma_2, B}{\vdash \Sigma, A \otimes B} \Downarrow$

What is ludics? (III)

The core of ludics : focalization

Positive	Negative
$\otimes$	$\parr$
$\oplus$	$\&$
$\mathbf{0}$	$\top$
$\mathbf{1}$	$\bot$
$?$	$!$

- Negative = reversible, deterministic: $\dfrac{\vdash \Sigma, A, B}{\vdash \Sigma, A \parr A} \Updownarrow$
- Positive = irreversible, nondeterministic: $\dfrac{\vdash \Sigma_1, A \qquad \vdash \Sigma_2, B}{\vdash \Sigma, A \otimes B} \Downarrow$

What is ludics? (III)

The core of ludics : focalization

Positive	Negative
$\otimes$	$⅋$
$\oplus$	$\&$
$\mathbf{0}$	$\top$
$\mathbf{1}$	$\bot$
$?$	$!$

- Negative = reversible, deterministic: $\dfrac{\vdash \Sigma, A, B}{\vdash \Sigma, A ⅋ A} \Updownarrow$
- Positive = irreversible, nondeterministic: $\dfrac{\vdash \Sigma_1, A \qquad \vdash \Sigma_2, B}{\vdash \Sigma, A \otimes B} \Downarrow$

What is ludics? (IV)

- $\vdash N_1, \ldots, N_m, P_1, \ldots, P_n$ choose a negative formula (if any) and keep decomposing until one get to atoms or positive subformulas;
- $\vdash P_1, \ldots, P_n$ choose a positive formula and keep decomposing it up to atoms or negative subformulas.

(Andreoli 92) The focalization discipline is a complete proof-search strategy.

What is ludics? (IV)

- $\vdash N_1, \ldots, N_m, P_1, \ldots, P_n$ choose a negative formula (if any) and keep decomposing until one get to atoms or positive subformulas;
- $\vdash P_1, \ldots, P_n$ choose a positive formula and keep decomposing it up to atoms or negative subformulas.

(Andreoli 92) The focalization discipline is a complete proof-search strategy.

What is ludics? (V)

Synthetic connectives

- Focalization allows synthetic connectives: clusters of connectives of the same polarity.
- $N \otimes (M_1 \oplus M_2)$ can be written as $\overline{a}\langle N, M_1, M_2\rangle$. Think $\overline{a}$ as a “generalized” ternary connective $\_ \otimes (\_ \oplus \_)$.

$$\dfrac{\Sigma_1, N \qquad \dfrac{\vdash \Sigma_2, M_1}{\vdash \Sigma_2, M_1 \oplus M_2}\oplus_1}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes \qquad \dfrac{\Sigma_1, N \qquad \dfrac{\vdash \Sigma_2, M_2}{\vdash \Sigma_2, M_1 \oplus M_2}\oplus_2}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes$$

$$\dfrac{\Sigma_1, N \qquad \vdash \Sigma_2, M_1}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes\oplus_1 \qquad \dfrac{\Sigma_1, N \qquad \vdash \Sigma_2, M_2}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes\oplus_2$$

- Alternation of positive and negative layers.

What is ludics? (V)

Synthetic connectives

- Focalization allows synthetic connectives: clusters of connectives of the same polarity.
- $N \otimes (M_1 \oplus M_2)$ can be written as $\overline{a}\langle N, M_1, M_2\rangle$. Think $\overline{a}$ as a "generalized" ternary connective $\_ \otimes (\_ \oplus \_)$.

$$\dfrac{\Sigma_1, N \qquad \dfrac{\vdash \Sigma_2, M_1}{\vdash \Sigma_2, M_1 \oplus M_2}\oplus_1}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes \qquad \dfrac{\Sigma_1, N \qquad \dfrac{\vdash \Sigma_2, M_2}{\vdash \Sigma_2, M_1 \oplus M_2}\oplus_2}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes$$

$$\dfrac{\Sigma_1, N \qquad \vdash \Sigma_2, M_1}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes\oplus_1 \qquad \dfrac{\Sigma_1, N \qquad \vdash \Sigma_2, M_2}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes\oplus_2$$

- Alternation of positive and negative layers.

What is ludics? (V)

Synthetic connectives

- Focalization allows synthetic connectives: clusters of connectives of the same polarity.
- $N \otimes (M_1 \oplus M_2)$ can be written as $\overline{a}\langle N, M_1, M_2\rangle$. Think $\overline{a}$ as a "generalized" ternary connective $\_ \otimes (\_ \oplus \_)$.

$$\dfrac{\Sigma_1, N \qquad \dfrac{\vdash \Sigma_2, M_1}{\vdash \Sigma_2, M_1 \oplus M_2}\oplus_1}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes \qquad \dfrac{\Sigma_1, N \qquad \dfrac{\vdash \Sigma_2, M_2}{\vdash \Sigma_2, M_1 \oplus M_2}\oplus_2}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes$$

$$\dfrac{\Sigma_1, N \qquad \vdash \Sigma_2, M_1}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes\oplus_1 \qquad \dfrac{\Sigma_1, N \qquad \vdash \Sigma_2, M_2}{\vdash \Sigma, N \otimes (M_1 \oplus M_2)}\otimes\oplus_2$$

- Alternation of positive and negative layers.

Computational ludics (I)

Designs (Terui 08) $\approx$ infinitary lambda terms (Böhm trees) + named applications + named and superimposed abstractions.

cf.

- the "concrete syntax" (Curien 05) $\approx$ abstract Böhm trees,
- the correspondence with linear π-calculus (Faggian-Piccolo 07).

Signature: $\mathcal{A} = (A, \mathrm{ar})$

A is a set of names,
$\mathrm{ar} : A \longrightarrow \mathbb{N}$ *gives an arity to each name.*

Computational ludics (II)

The set of designs is coinductively defined by:

$$
\begin{array}{rcll}
P & ::= & \maltese & \text{Daimon} \\
 & | & \Omega & \text{Divergence} \\
 & | & N_0|\overline{a}\langle N_1, \ldots, N_n\rangle & \text{Application} \\
N & ::= & x & \text{Variable} \\
 & | & \sum a(\vec{x}).P_a & \text{Abstraction}
\end{array}
$$

- where $ar(a) = n, \vec{x} = x_1, \ldots, x_n$
- $\sum a(\vec{x}).P_a$ is built from $\{a(\vec{x}).P_a\}_{a \in A}$.

Compare it with:

$$
\begin{array}{rcl}
P & ::= & (N_0)N_1 \ldots N_n \\
N & ::= & x \mid \lambda x_1 \cdots x_n.P
\end{array}
$$

Computational ludics (II)

The set of designs is coinductively defined by:

$$\begin{array}{rcll} P & ::= & \maltese & \text{Daimon} \\ & | & \Omega & \text{Divergence} \\ & | & N_0|\overline{a}\langle N_1,\ldots,N_n\rangle & \text{Application} \\ N & ::= & x & \text{Variable} \\ & | & \sum a(\vec{x}).P_a & \text{Abstraction} \end{array}$$

- where $ar(a) = n$, $\vec{x} = x_1,\ldots,x_n$
- $\sum a(\vec{x}).P_a$ is built from $\{a(\vec{x}).P_a\}_{a\in A}$.

Compare it with:

$$\begin{array}{rcl} P & ::= & (N_0)N_1\ldots N_n \\ N & ::= & x \mid \lambda x_1\cdots x_n.P \end{array}$$

Reduction

- Ω allows partial branching:

$$a(\vec{x}).P + b(\vec{y}).Q \;:=\; a(\vec{x}).P + b(\vec{y}).Q + c(\vec{z}).\Omega + d(\vec{z}).\Omega + \cdots$$

- Reduction rule:

$$(\sum a(x_1, \ldots, x_n).P_a) \,|\, \overline{a}\langle N_1, \ldots, N_n\rangle \;\longrightarrow\; P_a[N_1/x_1, \ldots, N_n/x_n].$$

- Compare it with

$$(\lambda x_1 \cdots x_n.P)N_1 \cdots N_n \;\longrightarrow\; P[N_1/x_1, \ldots, N_n/x_n]$$

Reduction

- Ω allows partial branching:

$$a(\vec{x}).P + b(\vec{y}).Q \;:=\; a(\vec{x}).P + b(\vec{y}).Q + c(\vec{z}).\Omega + d(\vec{z}).\Omega + \cdots$$

- Reduction rule:

$$(\sum a(x_1, \ldots, x_n).P_a) \,|\, \overline{a}\langle N_1, \ldots, N_n \rangle \;\longrightarrow\; P_a[N_1/x_1, \ldots, N_n/x_n].$$

- Compare it with

$$(\lambda x_1 \cdots x_n.P)N_1 \cdots N_n \;\longrightarrow\; P[N_1/x_1, \ldots, N_n/x_n]$$

Reduction

- Ω allows partial branching:

$$a(\vec{x}).P+b(\vec{y}).Q \;:=\; a(\vec{x}).P+b(\vec{y}).Q+c(\vec{z}).\Omega + d(\vec{z}).\Omega + \cdots$$

- Reduction rule:

$$(\sum a(x_1,\ldots,x_n).P_a)\,|\overline{a}\langle N_1,\ldots,N_n\rangle \;\longrightarrow\; P_a[N_1/x_1,\ldots,N_n/x_n].$$

- Compare it with

$$(\lambda x_1\cdots x_n.P)N_1\cdots N_n \;\longrightarrow\; P[N_1/x_1,\ldots,N_n/x_n]$$

Orthogonality

A positive design P is one of the following forms:

$x\|\overline{a}\langle N_1,\ldots,N_n\rangle$	Head normal form
$(\sum a(\vec{x}).P_a)\,\|\overline{a}\langle N_1,\ldots,N_n\rangle$	Cut
$\maltese$	Daimon
Ω	Divergence

- Dichotomy: For any closed positive design P,

$$P \longrightarrow^* \maltese \text{ or diverges.}$$

- Orthogonality: Suppose $fv(P) \subseteq \{x_0\}$ and $fv(M) = \emptyset$.

$$P \perp M \iff P[M/x_0] \longrightarrow^* \maltese.$$

Compare it with:

$$\pi \perp \pi' \iff \pi\pi' \text{ is nilpotent.}$$

Orthogonality

A positive design P is one of the following forms:

$x \,\vert\, \overline{a}\langle N_1, \ldots, N_n \rangle$	Head normal form
$(\sum a(\vec{x}).P_a) \,\vert\, \overline{a}\langle N_1, \ldots, N_n \rangle$	Cut
$\maltese$	Daimon
Ω	Divergence

- Dichotomy: For any closed positive design P,

$$P \longrightarrow^* \maltese \text{ or diverges.}$$

- Orthogonality: Suppose $fv(P) \subseteq \{x_0\}$ and $fv(M) = \emptyset$.

$$P \perp M \iff P[M/x_0] \longrightarrow^* \maltese.$$

Compare it with:

$$\pi \perp \pi' \iff \pi\pi' \text{ is nilpotent.}$$

Orthogonality

A positive design P is one of the following forms:

$x\|\overline{a}\langle N_1, \ldots, N_n\rangle$	Head normal form
$(\sum a(\vec{x}).P_a)\ \|\overline{a}\langle N_1, \ldots, N_n\rangle$	Cut
$\maltese$	Daimon
Ω	Divergence

- Dichotomy: For any closed positive design P,

$$P \longrightarrow^* \maltese \text{ or diverges.}$$

- Orthogonality: Suppose $fv(P) \subseteq \{x_0\}$ and $fv(M) = \emptyset$.

$$P \perp M \iff P[M/x_0] \longrightarrow^* \maltese.$$

Compare it with:

$$\pi \perp \pi' \iff \pi\pi' \text{ is nilpotent.}$$

Orthogonality

A positive design P is one of the following forms:

$x\|\overline{a}\langle N_1,\ldots,N_n\rangle$	Head normal form
$(\sum a(\vec{x}).P_a)\,\|\overline{a}\langle N_1,\ldots,N_n\rangle$	Cut
$\maltese$	Daimon
Ω	Divergence

- Dichotomy: For any closed positive design P,

$$P \longrightarrow^* \maltese \text{ or diverges.}$$

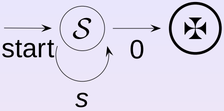
- Orthogonality: Suppose $fv(P) \subseteq \{x_0\}$ and $fv(M) = \emptyset$.

$$P \perp M \iff P[M/x_0] \longrightarrow^* \maltese.$$

Compare it with:

$$\pi \perp \pi' \iff \pi\pi' \text{ is nilpotent.}$$

Example: termination

$A =$ 

$$n = \underbrace{sssss \ldots s}_{n \text{ times}} 0$$

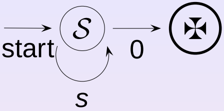
$$\begin{array}{rcl}
\mathrm{A} & := & x|\overline{\mathcal{S}}\langle \mathrm{zero}.\maltese + \mathrm{succ}(x).\mathrm{A}\rangle \\
0 & := & \mathcal{S}(x).x|\overline{\mathrm{zero}} \\
\mathrm{N}+1 & := & \mathcal{S}(x).x|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle
\end{array}$$

$$\begin{array}{rcl}
\mathrm{A}[0/x] & = & (\mathcal{S}(x).x|\overline{\mathrm{zero}})|\overline{\mathcal{S}}\langle \mathrm{zero}.\maltese + \mathrm{succ}(x).\mathrm{A}\rangle \\
& \longrightarrow & (\mathrm{zero}.\maltese + \mathrm{succ}(x).\mathrm{A})|\overline{\mathrm{zero}} \\
& \longrightarrow & \maltese.
\end{array}$$

$$\begin{array}{rcl}
\mathrm{A}[\mathrm{N}+1/x] & = & (\mathcal{S}(x).x|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle)|\overline{\mathcal{S}}\langle \mathrm{zero}.\maltese + \mathrm{succ}(x).\mathrm{A}\rangle \\
& \longrightarrow & (\mathrm{zero}.\maltese + \mathrm{succ}(x).\mathrm{A})|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle \\
& \longrightarrow & \mathrm{A}[\mathrm{N}/x].
\end{array}$$

Example: termination

$A =$



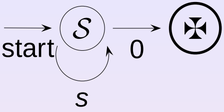
$$n = \underbrace{sssss \ldots s}_{n \text{ times}} 0$$

$$\begin{array}{rcl}
\mathrm{A} & := & x|\overline{\mathcal{S}}\langle \mathrm{zero}.\maltese + \mathrm{succ}(x).\mathrm{A}\rangle \\
0 & := & \mathcal{S}(x).x|\overline{\mathrm{zero}} \\
\mathrm{N}+1 & := & \mathcal{S}(x).x|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle
\end{array}$$

$$\begin{array}{rcl}
\mathrm{A}[0/x] & = & (\mathcal{S}(x).x|\overline{\mathrm{zero}})|\overline{\mathcal{S}}\langle \mathrm{zero}.\maltese + \mathrm{succ}(x).\mathrm{A}\rangle \\
& \longrightarrow & (\mathrm{zero}.\maltese + \mathrm{succ}(x).\mathrm{A})|\overline{\mathrm{zero}} \\
& \longrightarrow & \maltese.
\end{array}$$

$$\begin{array}{rcl}
\mathrm{A}[\mathrm{N}+1/x] & = & (\mathcal{S}(x).x|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle)|\overline{\mathcal{S}}\langle \mathrm{zero}.\maltese + \mathrm{succ}(x).\mathrm{A}\rangle \\
& \longrightarrow & (\mathrm{zero}.\maltese + \mathrm{succ}(x).\mathrm{A})|\overline{\mathrm{succ}}\langle \mathrm{N}\rangle \\
& \longrightarrow & \mathrm{A}[\mathrm{N}/x].
\end{array}$$

Example: termination

$A =$ 

$$n = \underbrace{sssss \ldots s}_{n \text{ times}} 0$$

$$\begin{array}{rcl}
\mathsf{A} & := & x|\overline{\mathcal{S}}\langle \mathsf{zero}.\maltese + \mathsf{succ}(x).\mathsf{A}\rangle \\
0 & := & \mathcal{S}(x).x|\overline{\mathsf{zero}} \\
\mathsf{N}+1 & := & \mathcal{S}(x).x|\overline{\mathsf{succ}}\langle \mathsf{N}\rangle
\end{array}$$

$$\begin{array}{rcl}
\mathsf{A}[0/x] & = & \big(\mathcal{S}(x).x|\overline{\mathsf{zero}}\big)|\overline{\mathcal{S}}\langle \mathsf{zero}.\maltese + \mathsf{succ}(x).\mathsf{A}\rangle \\
 & \longrightarrow & (\mathsf{zero}.\maltese + \mathsf{succ}(x).\mathsf{A})|\overline{\mathsf{zero}} \\
 & \longrightarrow & \maltese.
\end{array}$$

$$\begin{array}{rcl}
\mathsf{A}[\mathsf{N}+1/x] & = & \big(\mathcal{S}(x).x|\overline{\mathsf{succ}}\langle \mathsf{N}\rangle\big)|\overline{\mathcal{S}}\langle \mathsf{zero}.\maltese + \mathsf{succ}(x).\mathsf{A}\rangle \\
 & \longrightarrow & \big(\mathsf{zero}.\maltese + \mathsf{succ}(x).\mathsf{A}\big)|\overline{\mathsf{succ}}\langle \mathsf{N}\rangle \\
 & \longrightarrow & \mathsf{A}[\mathsf{N}/x].
\end{array}$$

Example: nontermination

$$
\begin{aligned}
P &:= x|\overline{a}\langle N\rangle \\
N &:= a(x).P \\
M &:= b(y).P
\end{aligned}
$$

$$
\begin{aligned}
P[N/x] &= (a(x).P)|\overline{a}\langle N\rangle \\
&\longrightarrow P[N/x].
\end{aligned}
$$

$$
\begin{aligned}
P[M/x] &= (b(x).P)|\overline{a}\langle N\rangle \\
&\longrightarrow \Omega.
\end{aligned}
$$

Example: nontermination

$$
\begin{array}{rcl}
P & := & x|\overline{a}\langle N\rangle \\
N & := & a(x).P \\
M & := & b(y).P
\end{array}
$$

$$
\begin{array}{rcl}
P[N/x] & = & \big(a(x).P\big)|\overline{a}\langle N\rangle \\
& \longrightarrow & P[N/x].
\end{array}
$$

$$
\begin{array}{rcl}
P[M/x] & = & \big(b(x).P\big)|\overline{a}\langle N\rangle \\
& \longrightarrow & \Omega.
\end{array}
$$

Ludics and Game Semantics

$$\begin{array}{ccc} \textbf{Ludics} & \qquad & \textbf{Game Semantics} \\ \text{Untyped strategies } (\textit{designs}) & & \text{Typed } \textit{strategies} \\ \Big\downarrow{\scriptstyle \perp\perp} & & \Big\uparrow \\ \text{Types } (\textit{Behaviours}) & & \text{Types } (\textit{Arenas, Games}) \end{array}$$

- Game Semantics: All strategies are typed. Types GUARANTEE that strategies compose well.
- Ludics : Strategies are untyped (all given on a universal arena) Strategies can ALWAYS interact with each other, and interaction may terminate well ($\perp$) or not (deadlock, Ω)

Ludics and Game Semantics

Ludics	Game Semantics
Untyped strategies (*designs*)	Typed *strategies*
$\downarrow \perp\perp$	$\uparrow$
Types (*Behaviours*)	Types (*Arenas, Games*)

- Game Semantics: All strategies are typed. Types GUARANTEE that strategies compose well.
- Ludics : Strategies are untyped (all given on a universal arena) Strategies can ALWAYS interact with each other, and interaction may terminate well ($\perp$) or not (deadlock, Ω)

Ludics and Game Semantics

Ludics | **Game Semantics**

$$\begin{tikzcd} \text{Untyped strategies (\textit{designs})} \arrow[d, "\perp\perp"'] \\ \text{Types (\textit{Behaviours})} \end{tikzcd}$$

$$\begin{tikzcd} \text{Typed \textit{strategies}} \\ \text{Types (\textit{Arenas, Games})} \arrow[u] \end{tikzcd}$$

- Game Semantics: All strategies are typed. Types GUARANTEE that strategies compose well.
- Ludics : Strategies are untyped (all given on a universal arena) Strategies can ALWAYS interact with each other, and interaction may terminate well ($\perp$) or not (deadlock, Ω)

Nondeterminism: why

- An interactive account and of contraction — duplication rule:

$$\frac{P(x,y) \vdash x : \mathbf{P},\ y : \mathbf{P}}{P(z,z) \vdash z : \mathbf{P}}$$

  where:
  - $\mathbf{P}$ is a positive logical type;
  - $P(x,y)$ is a positive design with free variables in $\{x,y\}$;
  - $P(z,z)$ is a positive design with free variable z.
- Two different readings of the rule:

  Top Down *Contraction*: an *identification* of free variables.

  Bottom Up *Duplication*: an arbitrary *bi-partition* of occurrences of z.

Nondeterminism: why

- An interactive account and of contraction — duplication rule:

$$\frac{P(x,y) \vdash x : \mathbf{P},\ y : \mathbf{P}}{P(z,z) \vdash z : \mathbf{P}}$$

  where:
  - $\mathbf{P}$ is a positive logical type;
  - $P(x,y)$ is a positive design with free variables in $\{x,y\}$;
  - $P(z,z)$ is a positive design with free variable z.
- Two different readings of the rule:

  Top Down *Contraction*: an *identification* of free variables.

  Bottom Up *Duplication*: an arbitrary *bi-partition* of occurrences of z.

Nondeterminism: why

- An interactive account and of contraction — duplication rule:

$$\frac{P(x,y) \vdash x : \mathbf{P},\ y : \mathbf{P}}{P(z,z) \vdash z : \mathbf{P}}$$

  where:
  - $\mathbf{P}$ is a positive logical type;
  - $P(x,y)$ is a positive design with free variables in $\{x,y\}$;
  - $P(z,z)$ is a positive design with free variable z.
- Two different readings of the rule:

  Top Down *Contraction*: an *identification* of free variables.

  Bottom Up *Duplication*: an arbitrary *bi-partition* of occurrences of z.

Failure of completeness

Write $P \models \Gamma$ for the interpretation of the sequent $P \vdash \Gamma$.
Semantically, we have to show that:

$$\star \quad P(x,y) \models x : \mathbf{P},\ y : \mathbf{P} \iff P(z,z) \models z : \mathbf{P}$$

In general, $\star$ does not hold in a *uniform* setting....
We need to *enlarge* the universe of designs.
We introduce (universal) nondeterminism.

Failure of completeness

Write $P \models \Gamma$ for the interpretation of the sequent $P \vdash \Gamma$.
Semantically, we have to show that:

$$\bigstar \quad P(x,y) \models x : \mathbf{P},\ y : \mathbf{P} \iff P(z,z) \models z : \mathbf{P}$$

In general, $\bigstar$ does not hold in a *uniform* setting....
We need to *enlarge* the universe of designs.
We introduce (universal) nondeterminism.

Failure of completeness

Write $P \models \Gamma$ for the interpretation of the sequent $P \vdash \Gamma$.
Semantically, we have to show that:

$$\bigstar \quad P(x,y) \models x : \mathbf{P},\ y : \mathbf{P} \iff P(z,z) \models z : \mathbf{P}$$

In general, $\bigstar$ does not hold in a *uniform* setting....
We need to *enlarge* the universe of designs.
We introduce (universal) nondeterminism.

Failure of completeness

Write $P \models \Gamma$ for the interpretation of the sequent $P \vdash \Gamma$.
Semantically, we have to show that:

$$\bigstar \quad P(x, y) \models x : \mathbf{P},\ y : \mathbf{P} \iff P(z, z) \models z : \mathbf{P}$$

In general, $\bigstar$ does not hold in a *uniform* setting....
We need to *enlarge* the universe of designs.
We introduce (universal) nondeterminism.

Failure of completeness

Write $P \models \Gamma$ for the interpretation of the sequent $P \vdash \Gamma$.
Semantically, we have to show that:

$$\bigstar \quad P(x,y) \models x : \mathbf{P},\ y : \mathbf{P} \iff P(z,z) \models z : \mathbf{P}$$

In general, $\bigstar$ does not hold in a *uniform* setting....
We need to *enlarge* the universe of designs.
We introduce (universal) nondeterminism.

Designs

Coinductively defined terms given by the following grammar:

$$
\begin{array}{rcll}
P & ::= & \Omega \;\Big|\; \bigwedge_I Q_i & \textit{positive designs} \\
Q_i & ::= & N_0 | \overline{a}\langle N_1, \ldots, N_n \rangle & \textit{predesigns} \\
N & ::= & x \;\Big|\; \sum a(\vec{x}).P_a & \textit{negative designs}
\end{array}
$$

- $\maltese$ is now defined as the empty conjunction $\bigwedge_\emptyset$. $\bigwedge_{\{i\}} Q_i$ is simply written as Q_i.
- A designs is *deterministic* if in any occurrence of subdesign $\bigwedge_I Q_i$, I is either empty (and hence $\bigwedge_I Q_i = \maltese$) or a singleton.

Designs

Coinductively defined terms given by the following grammar:

$$
\begin{array}{rcll}
P & ::= & \Omega \;\Big|\; \bigwedge_I Q_i & \textit{positive designs} \\
Q_i & ::= & N_0 | \overline{a}\langle N_1, \ldots, N_n \rangle & \textit{predesigns} \\
N & ::= & x \;\Big|\; \sum a(\vec{x}).P_a & \textit{negative designs}
\end{array}
$$

- ✠ is now defined as the empty conjunction $\bigwedge_\emptyset$. $\bigwedge_{\{i\}} Q_i$ is simply written as Q_i.
- A designs is *deterministic* if in any occurrence of subdesign $\bigwedge_I Q_i$, I is either empty (and hence $\bigwedge_I Q_i =$ ✠) or a singleton.

Designs

Coinductively defined terms given by the following grammar:

$$\begin{array}{rclr} P & ::= & \Omega \mid \bigwedge_I Q_i & \textit{positive designs} \\ Q_i & ::= & N_0|\overline{a}\langle N_1, \ldots, N_n\rangle & \textit{predesigns} \\ N & ::= & x \mid \sum a(\vec{x}).P_a & \textit{negative designs} \end{array}$$

- ✠ is now defined as the empty conjunction $\bigwedge_{\emptyset}$. $\bigwedge_{\{i\}} Q_i$ is simply written as Q_i.
- A designs is *deterministic* if in any occurrence of subdesign $\bigwedge_I Q_i$, I is either empty (and hence $\bigwedge_I Q_i = ✠$) or a singleton.

Normalization: Reduction

The **reduction relation** $\longrightarrow$ is defined over the set of positive designs as follows:

$$\begin{array}{rcl} \Omega & \longrightarrow & \Omega; \\ Q \wedge \bigwedge \left(\sum a(\vec{x}).P_a \mid \overline{a}\langle \vec{N} \rangle \right) & \longrightarrow & Q \wedge \bigwedge \left(P_a[\vec{N}/\vec{x}] \right). \end{array}$$

Given two positive designs Q, R, we define:

Convergence : $Q \Downarrow R$, if $Q \longrightarrow^* R$ and R is a conjunction of head normal forms (no cuts);

Divergence : $Q \Uparrow$, otherwise. $Q \longrightarrow^* \Omega$, $Q \longrightarrow \ldots \longrightarrow \ldots$

Normalization: Reduction

The **reduction relation** $\longrightarrow$ is defined over the set of positive designs as follows:

$$\begin{array}{rcl} \Omega & \longrightarrow & \Omega; \\ Q \wedge \bigwedge \left(\sum a(\vec{x}).P_a \mid \overline{a}\langle \vec{N} \rangle \right) & \longrightarrow & Q \wedge \bigwedge \left(P_a[\vec{N}/\vec{x}] \right). \end{array}$$

Given two positive designs Q, R, we define:

Convergence : $Q \Downarrow R$, if $Q \longrightarrow^* R$ and R is a conjunction of head normal forms (no cuts);

Divergence : $Q \Uparrow$, otherwise. $Q \longrightarrow^* \Omega$, $Q \longrightarrow \ldots \longrightarrow \ldots$

Normalization: Normal Form

The normal form function $[\![\ ]\!] : \mathcal{D} \longrightarrow \mathcal{D}$ is defined by corecursion as follows:

$$
\begin{array}{rcll}
[\![x]\!] & = & x; & \\
[\![P]\!] & = & \Omega, & \text{if } P \Uparrow; \\
 & = & \bigwedge_I x_i|\overline{a}_i\langle[\![\vec{N}_i]\!]\rangle & \text{if } P \Downarrow \bigwedge_I x_i|\overline{a}_i\langle\vec{N}_i\rangle; \\
[\![\sum a(\vec{x}).P_a]\!] & = & \sum a(\vec{x}).[\![P_a]\!]. &
\end{array}
$$

- $(a(\vec{x}).\maltese)|\overline{a}\langle\vec{N}\rangle = (a(\vec{x}).\bigwedge\emptyset)|\overline{a}\langle\vec{N}\rangle = \bigwedge\emptyset = \maltese$
- The dichotomy between $\maltese$ and Ω in the closed case is maintained: $[\![\bigwedge_I Q_i]\!] = \maltese$ iff any reduction sequence from any Q_i is finite.
- $\bigwedge$ is *universal*: $[\![Q_1 \bigwedge Q_2]\!] = \maltese$ iff $[\![Q_1]\!] = \maltese$ and $[\![Q_2]\!] = \maltese$.

Normalization: Normal Form

The normal form function $[\![\ ]\!] : \mathcal{D} \longrightarrow \mathcal{D}$ is defined by corecursion as follows:

$$
\begin{array}{rcll}
[\![x]\!] & = & x; & \\
[\![P]\!] & = & \Omega, & \text{if } P \Uparrow; \\
 & = & \bigwedge_I x_i|\overline{a}_i\langle[\![\vec{N}_i]\!]\rangle & \text{if } P \Downarrow \bigwedge_I x_i|\overline{a}_i\langle\vec{N}_i\rangle; \\
[\![\sum a(\vec{x}).P_a]\!] & = & \sum a(\vec{x}).[\![P_a]\!]. &
\end{array}
$$

- $(a(\vec{x}).\maltese)|\overline{a}\langle\vec{N}\rangle = (a(\vec{x}).\bigwedge\emptyset)|\overline{a}\langle\vec{N}\rangle = \bigwedge\emptyset = \maltese$
- The dichotomy between $\maltese$ and Ω in the closed case is maintained: $[\![\bigwedge_I Q_i]\!] = \maltese$ iff any reduction sequence from any Q_i is finite.
- $\bigwedge$ is *universal*: $[\![Q_1 \bigwedge Q_2]\!] = \maltese$ iff $[\![Q_1]\!] = \maltese$ and $[\![Q_2]\!] = \maltese$.

Normalization: Normal Form

The normal form function $[\![\ ]\!] : \mathcal{D} \longrightarrow \mathcal{D}$ is defined by corecursion as follows:

$$
\begin{array}{rcll}
[\![x]\!] & = & x; & \\
[\![P]\!] & = & \Omega, & \text{if } P \Uparrow; \\
 & = & \bigwedge_I x_i|\overline{a}_i\langle[\![\vec{N}_i]\!]\rangle & \text{if } P \Downarrow \bigwedge_I x_i|\overline{a}_i\langle\vec{N}_i\rangle; \\
[\![\sum a(\vec{x}).P_a]\!] & = & \sum a(\vec{x}).[\![P_a]\!]. &
\end{array}
$$

- $(a(\vec{x}).\maltese)|\overline{a}\langle\vec{N}\rangle = (a(\vec{x}).\bigwedge\emptyset)|\overline{a}\langle\vec{N}\rangle = \bigwedge\emptyset = \maltese$
- The dichotomy between $\maltese$ and Ω in the closed case is maintained: $[\![\bigwedge_I Q_i]\!] = \maltese$ iff any reduction sequence from any Q_i is finite.
- $\bigwedge$ is *universal*: $[\![Q_1 \bigwedge Q_2]\!] = \maltese$ iff $[\![Q_1]\!] = \maltese$ and $[\![Q_2]\!] = \maltese$.

Normalization: Normal Form

The normal form function $[\![\ ]\!] : \mathcal{D} \longrightarrow \mathcal{D}$ is defined by corecursion as follows:

$$\begin{array}{rcll} [\![x]\!] & = & x; & \\ [\![P]\!] & = & \Omega, & \text{if } P \Uparrow; \\ & = & \bigwedge_I x_i|\overline{a}_i\langle[\![\vec{N}_i]\!]\rangle & \text{if } P \Downarrow \bigwedge_I x_i|\overline{a}_i\langle\vec{N}_i\rangle; \\ [\![\sum a(\vec{x}).P_a]\!] & = & \sum a(\vec{x}).[\![P_a]\!]. & \end{array}$$

- $(a(\vec{x}).\maltese)|\overline{a}\langle\vec{N}\rangle = (a(\vec{x}).\bigwedge\emptyset)|\overline{a}\langle\vec{N}\rangle = \bigwedge\emptyset = \maltese$
- The dichotomy between $\maltese$ and Ω in the closed case is maintained: $[\![\bigwedge_I Q_i]\!] = \maltese$ iff any reduction sequence from any Q_i is finite.
- $\bigwedge$ is *universal*: $[\![Q_1 \bigwedge Q_2]\!] = \maltese$ iff $[\![Q_1]\!] = \maltese$ and $[\![Q_2]\!] = \maltese$.

Example

$$x|\overline{a}\langle y\rangle \;\wedge\; a(x).x|\overline{b}\langle y\rangle \mid \overline{a}\langle z\rangle \;\wedge\; b(x).(c(y).\maltese \mid \overline{c}\langle t\rangle) \mid \overline{b}\langle u\rangle \longrightarrow$$

$$x|\overline{a}\langle y\rangle \;\wedge\; z|\overline{b}\langle y\rangle \;\wedge\; c(y).\maltese \mid \overline{c}\langle t\rangle \longrightarrow x|\overline{a}\langle y\rangle \;\wedge\; z|\overline{b}\langle y\rangle.$$

Some definitions

- P is total if $P \neq \Omega$.
- T is linear if for any subterm $N_0|a\langle N_1, \ldots, N_n\rangle$, $fv(N_0), \ldots, fv(N_n)$ are pairwise disjoint.
- x is an identity if it occurs as $N_0|\overline{a}\langle N_1, \ldots, x, \ldots, N_n\rangle$.

Orthogonality

We consider only total, cut-free and identity free designs.

- P is **closed** if $\mathrm{fv}(P) = \emptyset$, **atomic** if $\mathrm{fv}(P) \subseteq \{x_0\}$ for a certain fixed variable x_0.
- N is **atomic** if $\mathrm{fv}(N) = \emptyset$.
- P, N are **orthogonal** $P \perp N$ when $P[N/x_0] = \maltese$.
- For $\mathbf{X}$ a set of atomic designs (same polarity):

$$\mathbf{X}^{\perp} := \{E : \forall D \in \mathbf{X},\ D \perp E\}.$$

- A **behaviour** (**interactive type**) $\mathbf{G}$ is a set of designs of the same polarity such that

$$\mathbf{G}^{\perp\perp} = \mathbf{G}.$$

Orthogonality

We consider only total, cut-free and identity free designs.

- P is **closed** if $\mathrm{fv}(P) = \emptyset$, **atomic** if $\mathrm{fv}(P) \subseteq \{x_0\}$ for a certain fixed variable x_0.
- N is **atomic** if $\mathrm{fv}(N) = \emptyset$.
- P, N are **orthogonal** $P \perp N$ when $P[N/x_0] = \maltese$.
- For $\mathbf{X}$ a set of atomic designs (same polarity):

$$\mathbf{X}^{\perp} := \{E : \forall D \in \mathbf{X},\ D \perp E\}.$$

- A **behaviour** (**interactive type**) $\mathbf{G}$ is a set of designs of the same polarity such that

$$\mathbf{G}^{\perp\perp} = \mathbf{G}.$$

Orthogonality

We consider only total, cut-free and identity free designs.

- P is **closed** if $\mathrm{fv}(P) = \emptyset$, **atomic** if $\mathrm{fv}(P) \subseteq \{x_0\}$ for a certain fixed variable x_0.
- N is **atomic** if $\mathrm{fv}(N) = \emptyset$.
- P, N are **orthogonal** $P \perp N$ when $P[N/x_0] = \maltese$.
- For $\mathbf{X}$ a set of atomic designs (same polarity):

$$\mathbf{X}^{\perp} := \{E : \forall D \in \mathbf{X},\ D \perp E\}.$$

- A **behaviour** (**interactive type**) $\mathbf{G}$ is a set of designs of the same polarity such that

$$\mathbf{G}^{\perp\perp} = \mathbf{G}.$$

Orthogonality

We consider only total, cut-free and identity free designs.

- P is **closed** if $\mathrm{fv}(P) = \emptyset$, **atomic** if $\mathrm{fv}(P) \subseteq \{x_0\}$ for a certain fixed variable x_0.
- N is **atomic** if $\mathrm{fv}(N) = \emptyset$.
- P, N are **orthogonal** $P \perp N$ when $P[N/x_0] = \maltese$.
- For **X** a set of atomic designs (same polarity):

$$\mathbf{X}^{\perp} := \{E : \forall D \in \mathbf{X},\ D \perp E\}.$$

- A **behaviour** (**interactive type**) **G** is a set of designs of the same polarity such that

$$\mathbf{G}^{\perp\perp} = \mathbf{G}.$$

Logical Connectives

Fix a linear order on variables: $x_0, x_1, x_2 \ldots$.

- An *n-ary logical connective* α is a finite set of negative actions $\alpha = \{a_1(\vec{x}_1), \ldots, a_n(\vec{x}_n)\}$, where $\vec{x}_1, \ldots, \vec{x}_n$ are taken over $\{x_1, \ldots, x_n\}$.
- Given an n-ary logical connective α and behaviours $\mathbf{N}_1, \ldots, \mathbf{N}_n, \mathbf{P}_1, \ldots, \mathbf{P}_n$ we define:

$$\overline{a}\langle \mathbf{N}_1, \ldots, \mathbf{N}_m \rangle := \{x_0 | \overline{a}\langle N_1, \ldots, N_m \rangle : N_i \in \mathbf{N}_i, 1 \leq i \leq m\}$$

PC: $\overline{\alpha}\langle \mathbf{N}_1, \ldots, \mathbf{N}_n \rangle := \left(\bigcup_{a \in \alpha} \overline{a}\langle \mathbf{N}_{i_1}, \ldots, \mathbf{N}_{i_m} \rangle\right)^{\perp\perp}$
where $i_1, \ldots, i_m \in \{1, \ldots, n\}$

NC: $\alpha(\mathbf{P}_1, \ldots, \mathbf{P}_n) := \overline{\alpha}\langle \mathbf{P}_1{}^{\perp}, \ldots, \mathbf{P}_n{}^{\perp} \rangle^{\perp}$

- $\left(\overline{\alpha}\langle \mathbf{N}_1, \ldots, \mathbf{N}_n \rangle\right)^{\perp} = \alpha\langle \mathbf{N}_1{}^{\perp}, \ldots, \mathbf{N}_n{}^{\perp} \rangle$.

Logical Connectives

Fix a linear order on variables: $x_0, x_1, x_2 \ldots$.

- An *n-ary logical connective* α is a finite set of negative actions $\alpha = \{a_1(\vec{x}_1), \ldots, a_n(\vec{x}_n)\}$, where $\vec{x}_1, \ldots, \vec{x}_n$ are taken over $\{x_1, \ldots, x_n\}$.
- Given an n-ary logical connective α and behaviours $\mathbf{N}_1, \ldots, \mathbf{N}_n, \mathbf{P}_1, \ldots, \mathbf{P}_n$ we define:

$$\overline{a}\langle \mathbf{N}_1, \ldots, \mathbf{N}_m \rangle := \{x_0 | \overline{a}\langle N_1, \ldots, N_m \rangle : N_i \in \mathbf{N}_i, 1 \leq i \leq m\}$$

PC: $\overline{\alpha}\langle \mathbf{N}_1, \ldots, \mathbf{N}_n \rangle := \left(\bigcup_{a \in \alpha} \overline{a}\langle \mathbf{N}_{i_1}, \ldots, \mathbf{N}_{i_m} \rangle\right)^{\perp\perp}$
where $i_1, \ldots, i_m \in \{1, \ldots, n\}$

NC: $\alpha(\mathbf{P}_1, \ldots, \mathbf{P}_n) := \overline{\alpha}\langle {\mathbf{P}_1}^{\perp}, \ldots, {\mathbf{P}_n}^{\perp} \rangle^{\perp}$

- $\left(\overline{\alpha}\langle \mathbf{N}_1, \ldots, \mathbf{N}_n \rangle\right)^{\perp} = \alpha\langle {\mathbf{N}_1}^{\perp}, \ldots, {\mathbf{N}_n}^{\perp} \rangle$.

Logical Connectives

Fix a linear order on variables: $x_0, x_1, x_2 \ldots$.

- An *n-ary logical connective* α is a finite set of negative actions $\alpha = \{a_1(\vec{x}_1), \ldots, a_n(\vec{x}_n)\}$, where $\vec{x}_1, \ldots, \vec{x}_n$ are taken over $\{x_1, \ldots, x_n\}$.
- Given an n-ary logical connective α and behaviours $\mathbf{N}_1, \ldots, \mathbf{N}_n, \mathbf{P}_1, \ldots, \mathbf{P}_n$ we define:

$$\overline{a}\langle \mathbf{N}_1, \ldots, \mathbf{N}_m \rangle := \{x_0 | \overline{a}\langle N_1, \ldots, N_m \rangle : N_i \in \mathbf{N}_i, 1 \leq i \leq m\}$$

PC: $\overline{\alpha}\langle \mathbf{N}_1, \ldots, \mathbf{N}_n \rangle := \left(\bigcup_{a \in \alpha} \overline{a}\langle \mathbf{N}_{i_1}, \ldots, \mathbf{N}_{i_m} \rangle\right)^{\perp\perp}$
where $i_1, \ldots, i_m \in \{1, \ldots, n\}$

NC: $\alpha(\mathbf{P}_1, \ldots, \mathbf{P}_n) := \overline{\alpha}\langle \mathbf{P}_1{}^{\perp}, \ldots, \mathbf{P}_n{}^{\perp} \rangle^{\perp}$

- $\left(\overline{\alpha}\langle \mathbf{N}_1, \ldots, \mathbf{N}_n \rangle\right)^{\perp} = \alpha\langle \mathbf{N}_1{}^{\perp}, \ldots, \mathbf{N}_n{}^{\perp} \rangle$.

Examples

Usual linear logic connectives can be defined by logical connectives $\parr, \&, \uparrow, \top$ below;

- $\parr := \{\wp\}$, $\bullet := \overline{\wp}$, $\otimes := \overline{\parr}$;
- $\& := \{\pi_1, \pi_2\}$, $\iota_i := \overline{\pi_i}$, $\oplus := \overline{\&}$;
- $\uparrow := \{\uparrow\}$, $\downarrow := \overline{\uparrow}$.
- $\top := \emptyset$, $\mathbf{0} = \overline{\top}$.

$\wp, \bullet$ binary names, $\pi_i, \iota_i, \uparrow, \downarrow$ unary names.

$$
\begin{array}{rcl@{\qquad}rcl}
\mathbf{N} \otimes \mathbf{M} & = & \bullet\langle \mathbf{N}, \mathbf{M}\rangle^{\perp\perp} & \mathbf{P} \parr \mathbf{Q} & = & \bullet\langle \mathbf{P}^{\perp}, \mathbf{Q}^{\perp}\rangle^{\perp} \\
\mathbf{N} \oplus \mathbf{M} & = & (\iota_1\langle \mathbf{N}\rangle \cup \iota_2\langle \mathbf{M}\rangle)^{\perp\perp} & \mathbf{P} \& \mathbf{Q} & = & \iota_1\langle \mathbf{P}^{\perp}\rangle^{\perp} \cap \iota_2\langle \mathbf{Q}^{\perp}\rangle^{\perp} \\
\downarrow\mathbf{N} & = & \downarrow\langle \mathbf{N}\rangle^{\perp\perp} & \uparrow\mathbf{P} & = & \downarrow\langle \mathbf{P}^{\perp}\rangle^{\perp} \\
\mathbf{1} & = & \downarrow\langle \top\rangle^{\perp\perp} & \perp & = & \downarrow\langle \top\rangle^{\perp}
\end{array}
$$

Examples

Usual linear logic connectives can be defined by logical connectives $\parr, \&, \uparrow, \top$ below;

- $\parr := \{\wp\}$, $\bullet := \overline{\wp}$, $\otimes := \overline{\parr}$;
- $\& := \{\pi_1, \pi_2\}$, $\iota_i := \overline{\pi_i}$, $\oplus := \overline{\&}$;
- $\uparrow := \{\uparrow\}$, $\downarrow := \overline{\uparrow}$.
- $\top := \emptyset$, $\mathbf{0} = \overline{\top}$.

$\wp, \bullet$ binary names, $\pi_i, \iota_i, \uparrow, \downarrow$ unary names.

$$
\begin{array}{rcl@{\qquad}rcl}
\mathbf{N} \otimes \mathbf{M} & = & \bullet\langle \mathbf{N}, \mathbf{M}\rangle^{\perp\perp} & \mathbf{P} \parr \mathbf{Q} & = & \bullet\langle \mathbf{P}^{\perp}, \mathbf{Q}^{\perp}\rangle^{\perp} \\
\mathbf{N} \oplus \mathbf{M} & = & (\iota_1\langle \mathbf{N}\rangle \cup \iota_2\langle \mathbf{M}\rangle)^{\perp\perp} & \mathbf{P} \,\&\, \mathbf{Q} & = & \iota_1\langle \mathbf{P}^{\perp}\rangle^{\perp} \cap \iota_2\langle \mathbf{Q}^{\perp}\rangle^{\perp} \\
\downarrow\mathbf{N} & = & \downarrow\langle \mathbf{N}\rangle^{\perp\perp} & \uparrow\mathbf{P} & = & \downarrow\langle \mathbf{P}^{\perp}\rangle^{\perp} \\
\mathbf{1} & = & \downarrow\langle \top\rangle^{\perp\perp} & \perp & = & \downarrow\langle \top\rangle^{\perp}
\end{array}
$$

Logical behaviours and semantical sequents

Logical behaviours: *inductively* defined by

$$\mathbf{P} ::= \overline{\alpha}\langle \mathbf{N}_1, \ldots, \mathbf{N}_n \rangle \quad \mathbf{N} ::= \alpha(\mathbf{P}_1, \ldots, \mathbf{P}_n)$$

- $P \models x_1 : \mathbf{P}_1, x_2 : \mathbf{P}_2$ if $\mathrm{fv}(P) \subseteq \{x_1, x_2\}$ and $P[N_1/x_1, N_n/x_2] = \maltese$ for any $N_1 \in \mathbf{P}_1^{\perp}$, $N_2 \in \mathbf{P}_2^{\perp}$.
- $N \models x : \mathbf{P}, \mathbf{N}$ if $\mathrm{fv}(N) \subseteq \{x\}$ and $P[N[M/x]/x_0] = \maltese$ for any $M \in \mathbf{P}^{\perp}$, $P \in \mathbf{N}^{\perp}$.
- $P \models x_0 : \mathbf{P}$ iff $P \in \mathbf{P}$.

Duplication/ $\bigwedge$

Any positive logical behaviour satisfies:

Duplicability: $P[x_0/x_1, x_0/x_2] \models x_0 : \mathbf{P} \iff P \models x_1 : \mathbf{P}, x_2 : \mathbf{P}$

Any negative logical behaviour satisfies:

Closure under $\bigwedge$: $N, M \in \mathbf{N} \iff N \wedge M \in \mathbf{N}$

$N = \sum a(\vec{x}).P \quad M = \sum a(\vec{x}).Q \quad N \wedge M = \sum a(\vec{x}).P \wedge Q.$

Duplication/ $\bigwedge$

Any positive logical behaviour satisfies:

Duplicability: $P[x_0/x_1, x_0/x_2] \models x_0 : \mathbf{P} \iff P \models x_1 : \mathbf{P}, x_2 : \mathbf{P}$

Any negative logical behaviour satisfies:

Closure under $\bigwedge$: $N, M \in \mathbf{N} \iff N \wedge M \in \mathbf{N}$

$N = \sum a(\vec{x}).P \quad M = \sum a(\vec{x}).Q \quad N \wedge M = \sum a(\vec{x}).P \wedge Q.$

About internal completeness (I)

- A purely monistic, local notion of completeness.
- A direct description of the elements in behaviours (built by logical connectives) without using the orthogonality and without referring to any proof system.

Internal completeness holds for negative logical connectives:

$$\alpha(\mathbf{P}_1, \ldots, \mathbf{P}_n) = \{\textstyle\sum_\alpha a(\vec{x}).P_a : P_a \models x_{i_1} : \mathbf{P}_{i_1}, \ldots x_{i_m} : \mathbf{P}_{i_m}\}$$

- P_b can be arbitrary when $b(\vec{x}) \notin \alpha$.
- We have a lot of garbage...

$$\begin{aligned} \mathbf{P}_1 \,\&\, \mathbf{P}_2 &= \{\pi_1(x_1).P_1 + \pi_2(x_2).P_2 + \cdots : \; P_i \models x_i : \mathbf{P}_i\} \\ &= \{\pi_1(x_0).P_1 + \pi_2(x_0).P_2 + \cdots : \; P_i \in \mathbf{P}_i\} \end{aligned}$$

irrelevant components of the sum are suppressed by $\cdots$
Up to *incarnation* (i.e. removal of irrelevant part), $\mathbf{P}_1 \& \mathbf{P}_2$, which has been defined by *intersection*, is isomorphic to the cartesian product of $\mathbf{P}_1$ and $\mathbf{P}_2$: a phenomenon called *mystery of incarnation*.

About internal completeness (I)

- A purely monistic, local notion of completeness.
- A direct description of the elements in behaviours (built by logical connectives) without using the orthogonality and without referring to any proof system.

Internal completeness holds for negative logical connectives:

$$\alpha(\mathbf{P}_1, \ldots, \mathbf{P}_n) = \{\sum_\alpha a(\vec{x}).P_a : P_a \models x_{i_1} : \mathbf{P}_{i_1}, \ldots x_{i_m} : \mathbf{P}_{i_m}\}$$

- P_b can be arbitrary when $b(\vec{x}) \notin \alpha$.
- We have a lot of garbage...

$$\begin{array}{rcl} \mathbf{P}_1 \,\&\, \mathbf{P}_2 & = & \{\pi_1(x_1).P_1 + \pi_2(x_2).P_2 + \cdots : \; P_i \models x_i : \mathbf{P}_i\} \\ & = & \{\pi_1(x_0).P_1 + \pi_2(x_0).P_2 + \cdots : \; P_i \in \mathbf{P}_i\} \end{array}$$

irrelevant components of the sum are suppressed by $\cdots$ Up to *incarnation* (i.e. removal of irrelevant part), $\mathbf{P}_1 \& \mathbf{P}_2$, which has been defined by *intersection*, is isomorphic to the cartesian product of $\mathbf{P}_1$ and $\mathbf{P}_2$: a phenomenon called *mystery of incarnation*.

About internal completeness (II)

For positive logical behaviours, it only holds (in that simple form) for *linear* and *deterministic designs*.

- Because any logical positive behaviour is *built* on linear and deterministic designs...
- But we want to take repetitions into account!

About internal completeness (II)

For positive logical behaviours, it only holds (in that simple form) for *linear* and *deterministic designs*.

- Because any logical positive behaviour is *built* on linear and deterministic designs...
- But we want to take repetitions into account!

About internal completeness (II)

For positive logical behaviours, it only holds (in that simple form) for *linear* and *deterministic designs*.

- Because any logical positive behaviour is *built* on linear and deterministic designs...
- But we want to take repetitions into account!

Proofs and Models

- A **proof** is a design in which all the conjunctions are unary. In other words, a proof is a deterministic and $\maltese$-free design.
- A **model** is an atomic linear design (in which conjunctions of arbitrary cardinality may occur).

Proof-system

$$\frac{M_{i_1} \vdash \Gamma, \mathbf{N}_{i_1} \quad \dots \quad M_{i_m} \vdash \Gamma, \mathbf{N}_{i_m} \quad (z : \overline{\alpha}\langle \mathbf{N}_1, \dots, \mathbf{N}_n \rangle \in \Gamma)}{z|\overline{a}\langle M_{i_1}, \dots, M_{i_m} \rangle \vdash \Gamma} \; (\overline{\alpha}, \overline{a})$$

$$\frac{\{P_a \vdash \Gamma, \vec{x}_a : \vec{\mathbf{P}}_a\}_{a \in \alpha}}{\sum a(\vec{x}).P_a \vdash \Gamma, \alpha(\mathbf{P}_1, \dots, \mathbf{P}_n)} \; (\alpha) \qquad \frac{P \vdash \Gamma, z : \mathbf{P} \quad N \vdash \Gamma, \mathbf{P}^{\perp}}{P[N/z] \vdash \Gamma} \; (cut)$$

where:

- In the rule $(\overline{\alpha}, \overline{a})$, $a \in \alpha$, $ar(a) = m$, and $i_1, \dots, i_m \in \{1, \dots, n\}$.
- In (α), $\vec{x}_a : \vec{\mathbf{P}}_a$ stands for $x_{i_1} : \mathbf{P}_{i_1}, \dots, x_{i_m} : \mathbf{P}_{i_m}$.

Notice that:

- Structural rules (weakening and contraction/duplication) are implicit.

Proof-system

$$\frac{M_{i_1} \vdash \Gamma, \mathbf{N}_{i_1} \quad \ldots \quad M_{i_m} \vdash \Gamma, \mathbf{N}_{i_m} \quad (z : \overline{\alpha}\langle \mathbf{N}_1, \ldots, \mathbf{N}_n \rangle \in \Gamma)}{z|\overline{a}\langle M_{i_1}, \ldots, M_{i_m} \rangle \vdash \Gamma} \; (\overline{\alpha}, \overline{a})$$

$$\frac{\{P_a \vdash \Gamma, \vec{x}_a : \vec{\mathbf{P}}_a\}_{a \in \alpha}}{\sum a(\vec{x}).P_a \vdash \Gamma, \alpha(\mathbf{P}_1, \ldots, \mathbf{P}_n)} \; (\alpha) \qquad \frac{P \vdash \Gamma, z : \mathbf{P} \quad N \vdash \Gamma, \mathbf{P}^{\perp}}{P[N/z] \vdash \Gamma} \; (cut)$$

where:

- In the rule $(\overline{\alpha}, \overline{a})$, $a \in \alpha$, $ar(a) = m$, and $i_1, \ldots, i_m \in \{1, \ldots, n\}$.
- In (α), $\vec{x}_a : \vec{\mathbf{P}}_a$ stands for $x_{i_1} : \mathbf{P}_{i_1}, \ldots, x_{i_m} : \mathbf{P}_{i_m}$.

Notice that:

- Structural rules (weakening and contraction/duplication) are implicit.

Example

$$\frac{M_1 \vdash \Gamma, \mathbf{N}_1 \quad M_2 \vdash \Gamma, \mathbf{N}_2 \quad (z : \mathbf{N}_1 \otimes \mathbf{N}_2 \in \Gamma)}{z | \bullet \langle M_1, M_2 \rangle \vdash \Gamma} \ (\otimes, \bullet)$$

$$\frac{M \vdash \Gamma, \mathbf{N}_i \quad (z : \mathbf{N}_1 \oplus \mathbf{N}_2 \in \Gamma)}{z | \iota_i \langle M \rangle \vdash \Gamma} \ (\oplus, \iota_i)$$

$$\frac{P \vdash \Gamma, x_1 : \mathbf{P}_1, x_2 : \mathbf{P}_2}{\wp(x_1, x_2).P + \cdots \vdash \Gamma, \mathbf{P}_1 \parr \mathbf{P}_2} \ (\parr)$$

$$\frac{P_1 \vdash \Gamma, x_1 : \mathbf{P}_1 \quad P_2 \vdash \Gamma, x_2 : \mathbf{P}_2}{\pi_1(x_1).P_1 + \pi_2(x_2).P_2 + \cdots \vdash \Gamma, \mathbf{P}_1 \,\&\, \mathbf{P}_2} \ (\&)$$

Theorem (Soundness)

$$P \vdash \mathbf{P} \Longrightarrow P \models x : \mathbf{P}.$$

The proof is given by induction on the depth of the type derivation $P \vdash \mathbf{P}$.

Theorem (Completeness (for proofs))

If P is a proof:

$$P \models x : \mathbf{P} \Longrightarrow P \vdash \mathbf{P}.$$

Likewise for negative logical behaviours.

Theorem (Soundness)

$$P \vdash \mathbf{P} \Longrightarrow P \models x : \mathbf{P}.$$

The proof is given by induction on the depth of the type derivation $P \vdash \mathbf{P}$.

Theorem (Completeness (for proofs))

If P is a proof:

$$P \models x : \mathbf{P} \Longrightarrow P \vdash \mathbf{P}.$$

Likewise for negative logical behaviours.

Sketch of the proof

- Analogous to Schütte's proof of Gödel's completeness. We consider the statement:

$$P \nvdash \mathbf{P} \Longrightarrow P \not\models x : \mathbf{P}.$$

  1. Given an unprovable sequent $\vdash \mathbf{P}$, find an open branch in the cut-free proof search tree.
  2. From the open branch, build a *countermodel* M in which $\mathbf{P}$ is false.

- The *countermodel* is here an atomic linear design in which conjunctions of arbitrary cardinality may occur. We can *explicitly* construct the countermodel.
- König Lemma is here essential.
- Closure under $\bigwedge$ of $\mathbf{P}^{\perp}$ is essential to prove that the countermodel belongs to $\mathbf{P}^{\perp}$.

Sketch of the proof

- Analogous to Schütte's proof of Gödel's completeness. We consider the statement:

$$P \nvdash \mathbf{P} \Longrightarrow P \not\models x : \mathbf{P}.$$

  1. Given an unprovable sequent $\vdash \mathbf{P}$, find an open branch in the cut-free proof search tree.
  2. From the open branch, build a *countermodel* M in which $\mathbf{P}$ is false.

- The *countermodel* is here an atomic linear design in which conjunctions of arbitrary cardinality may occur. We can *explicitly* construct the countermodel.
- König Lemma is here essential.
- Closure under $\bigwedge$ of $\mathbf{P}^{\perp}$ is essential to prove that the countermodel belongs to $\mathbf{P}^{\perp}$.

Corollaries

Downward Löwenheim-Skolem Let P be a proof and $\mathbf{P}$ a logical behaviour. If $P \notin \mathbf{P}$, then there is a *countable* model $M \in \mathbf{P}^{\perp}$ such that $P \not\perp M$ (M is countable in the sense that it consists of countably many actions $\neq \Omega$).

Finite model property If P is linear, there is a finite (and deterministic) model $M \in \mathbf{P}^{\perp}$ such that $P \not\perp M$.

Conclusions

- Gödel's completeness revisited in terms of ludics.
- We have enlighten the duality between proofs and models.
- We can give an explicit construction of a countermodel to any wrong proof attempt.

Related works

- Gödel's incompleteness theorem.
- Recursive types (Melliès-Vouillon 05).

Thank you!

Questions?

Thank you!

Questions?