Splitting the Atom of Dependent Types

...or Linear and Operational Dependent Type Theory

Matthijs Vákár

Oxford, 10 November, 2014

UNIVERSITY OF OXFORD

Our Journey

intuitionistic $\rightsquigarrow$ linear $\rightsquigarrow$ operational

Done for propositional logic, external first order quantification, (impredicative) second order quantification.

Our Journey

intuitionistic $\rightsquigarrow$ linear $\rightsquigarrow$ operational

Done for propositional logic, external first order quantification, (impredicative) second order quantification.

Subtle for internal first order quantification: dependent type theory.

Our Journey

intuitionistic $\rightsquigarrow$ linear $\rightsquigarrow$ operational

Done for propositional logic, external first order quantification, (impredicative) second order quantification.

Subtle for internal first order quantification: dependent type theory.

Table of contents

Introduction

My motivation:

- Deepen understanding of HoTT:
  - As foundation of mathematics.
  - As language for homotopy: relation to stable homotopy?
- Computational semantics for dependent types:
  - Game semantics.
  - Stepping stone: coherence space semantics.
  - Generally: models of DTT in !-co-Kleisli or !-co-EM categories.

Introduction

My motivation:

- Deepen understanding of HoTT:
  - As foundation of mathematics.
  - As language for homotopy: relation to stable homotopy?
- Computational semantics for dependent types:
  - Game semantics.
  - Stepping stone: coherence space semantics.
  - Generally: models of DTT in !-co-Kleisli or !-co-EM categories.
- Dependently Typed Quantum Computing?

Introduction

My motivation:

- Deepen understanding of HoTT:
  - As foundation of mathematics.
  - As language for homotopy: relation to stable homotopy?
- Computational semantics for dependent types:
  - Game semantics.
  - Stepping stone: coherence space semantics.
  - Generally: models of DTT in !-co-Kleisli or !-co-EM categories.
- Dependently Typed Quantum Computing?
- Find proper linear understanding of predicate logic, including identity.

Introduction

My motivation:

- Deepen understanding of HoTT:
  - As foundation of mathematics.
  - As language for homotopy: relation to stable homotopy?
- Computational semantics for dependent types:
  - Game semantics.
  - Stepping stone: coherence space semantics.
  - Generally: models of DTT in !-co-Kleisli or !-co-EM categories.
- Dependently Typed Quantum Computing?
- Find proper linear understanding of predicate logic, including identity.

Programme of research

1. Combining linear and dependent types,
   - syntactically and semantically,
   - in sufficient generality to admit models from a variety of fields.

   Matthijs Vákár, http://arxiv.org/abs/1405.0033. Submitted to FoSSaCS2015.

2. Coherence space semantics.

   With Samson Abramsky: draft ready.

3. Game semantics.

   With Radha Jagadeesan and Samson Abramsky: draft ready.

4. Stable homotopy theory.

   Planned. Related to Mike Shulman, http://www.tac.mta.ca/tac/volumes/20/18/20-18abs.html.

5. Quantum physics.

   Planned. Related to Urs Schreiber, http://arxiv.org/abs/1402.7041

Programme of research

1. Combining linear and dependent types,
   - syntactically and semantically,
   - in sufficient generality to admit models from a variety of fields.

   Matthijs Vákár, http://arxiv.org/abs/1405.0033. Submitted to FoSSaCS2015.
2. Coherence space semantics.

   With Samson Abramsky: draft ready.
3. Game semantics.

   With Radha Jagadeesan and Samson Abramsky: draft ready.
4. Stable homotopy theory.

   Planned. Related to Mike Shulman, http://www.tac.mta.ca/tac/volumes/20/18/20-18abs.html.
5. Quantum physics.

   Planned. Related to Urs Schreiber, http://arxiv.org/abs/1402.7041

Syntax

- Primarily designed as very straightforward blend of
  - (intuitionistic) dependent type theory, as in *Hofman, Syntax and semantics of dependent types*
  - intuitionistic linear type theory, as in *Barber, Dual Intuitionistic Linear Logic*.
- Very similar to Cervesato and Pfenning's LLF plus Σ-types (although developed independently). By constrast, not focussed on specific computational implementation, but meant to be general through modularity.

Syntax

- Primarily designed as very straightforward blend of
  - (intuitionistic) dependent type theory, as in *Hofman, Syntax and semantics of dependent types*
  - intuitionistic linear type theory, as in *Barber, Dual Intuitionistic Linear Logic*.
- Very similar to Cervesato and Pfenning's LLF plus Σ-types (although developed independently). By constrast, not focussed on specific computational implementation, but meant to be general through modularity.
- Very similar to upcoming Krishnaswami, Pradic, Benton, which appears to be a conservative extension.

Syntax

- Primarily designed as very straightforward blend of
  - (intuitionistic) dependent type theory, as in *Hofman, Syntax and semantics of dependent types*
  - intuitionistic linear type theory, as in *Barber, Dual Intuitionistic Linear Logic*.
- Very similar to Cervesato and Pfenning's LLF plus Σ-types (although developed independently). By constrast, not focussed on specific computational implementation, but meant to be general through modularity.
- Very similar to upcoming Krishnaswami, Pradic, Benton, which appears to be a conservative extension.

Judgements

Contexts $\Delta; \Xi$ consist of intuitionistic region Δ and linear region Ξ. (Intuitionistic and linear) types in context can depend on **intuitionistic context** to their left.

$\vdash \Delta; \Xi \ \mathrm{ctxt}$	$\Delta; \Xi$ is a valid context
$\Delta; \cdot \vdash A \ \mathrm{type}$	A is a type in (intuitionistic) context Δ
$\Delta; \Xi \vdash a : A$	a is a term of type A in context $\Delta; \Xi$

Judgements

Contexts $\Delta; \Xi$ consist of intuitionistic region Δ and linear region Ξ. (Intuitionistic and linear) types in context can depend on **intuitionistic context** to their left.

$\vdash \Delta; \Xi \ \mathrm{ctxt}$	$\Delta; \Xi$ is a valid context
$\Delta; \cdot \vdash A \ \mathrm{type}$	A is a type in (intuitionistic) context Δ
$\Delta; \Xi \vdash a : A$	a is a term of type A in context $\Delta; \Xi$
$\vdash \Delta; \Xi \equiv \Delta'; \Xi' \ \mathrm{ctxt}$	$\Delta; \Xi$ and $\Delta'; \Xi'$ are judgementally equal contexts
$\Delta; \cdot \vdash A \equiv B \ \mathrm{type}$	A and B are judgementally equal types in (intuitionistic) context Δ
$\Delta; \Xi \vdash a \equiv a' : A$	a and a' are judgementally equal terms of type A in context $\Delta; \Xi$

Judgements

Contexts $\Delta; \Xi$ consist of intuitionistic region Δ and linear region Ξ. (Intuitionistic and linear) types in context can depend on **intuitionistic context** to their left.

$\vdash \Delta; \Xi \ \mathrm{ctxt}$	$\Delta; \Xi$ is a valid context
$\Delta; \cdot \vdash A \ \mathrm{type}$	A is a type in (intuitionistic) context Δ
$\Delta; \Xi \vdash a : A$	a is a term of type A in context $\Delta; \Xi$
$\vdash \Delta; \Xi \equiv \Delta'; \Xi' \ \mathrm{ctxt}$	$\Delta; \Xi$ and $\Delta'; \Xi'$ are judgementally equal contexts
$\Delta; \cdot \vdash A \equiv B \ \mathrm{type}$	A and B are judgementally equal types in (intuitionistic) context Δ
$\Delta; \Xi \vdash a \equiv a' : A$	a and a' are judgementally equal terms of type A in context $\Delta; \Xi$

Structural rules

Rules for context formation:

$$\frac{}{\cdot;\cdot\ \mathrm{ctxt}}\ \text{C-Emp}$$

$$\frac{\vdash \Delta;\Xi\ \mathrm{ctxt} \qquad \Delta;\cdot \vdash A\ \mathrm{type}}{\vdash \Delta, x : A;\Xi\ \mathrm{ctxt}}\ \text{Int-C-Ext}$$

$$\frac{\vdash \Delta;\Xi\ \mathrm{ctxt} \qquad \Delta;\cdot \vdash A\ \mathrm{type}}{\vdash \Delta;\Xi, x : A\ \mathrm{ctxt}}\ \text{Lin-C-Ext}$$

Variable/axiom rules:

$$\frac{\Delta, x : A, \Delta';\cdot\ \mathrm{ctxt}}{\Delta, x : A, \Delta';\cdot \vdash x : A}\ \text{Int-Var} \qquad \frac{\Delta; x : A\ \mathrm{ctxt}}{\Delta; x : A \vdash x : A}\ \text{Lin-Var}$$

Structural rules

Rules for context formation:

$$\frac{}{\cdot;\cdot\ \mathrm{ctxt}}\ \text{C-Emp}$$

$$\frac{\vdash \Delta;\Xi\ \mathrm{ctxt} \qquad \Delta;\cdot\vdash A\ \mathrm{type}}{\vdash \Delta, x:A;\Xi\ \mathrm{ctxt}}\ \text{Int-C-Ext}$$

$$\frac{\vdash \Delta;\Xi\ \mathrm{ctxt} \qquad \Delta;\cdot\vdash A\ \mathrm{type}}{\vdash \Delta;\Xi, x:A\ \mathrm{ctxt}}\ \text{Lin-C-Ext}$$

Variable/axiom rules:

$$\frac{\Delta, x:A, \Delta';\cdot\ \mathrm{ctxt}}{\Delta, x:A, \Delta';\cdot\vdash x:A}\ \text{Int-Var} \qquad \frac{\Delta; x:A\ \mathrm{ctxt}}{\Delta; x:A\vdash x:A}\ \text{Lin-Var}$$

Weakening: $$\frac{\Delta, \Delta'; \Xi \vdash \mathcal{J} \qquad \Delta; \cdot \vdash A \text{ type}}{\Delta, x : A, \Delta'; \Xi \vdash \mathcal{J}} \text{Int-Weak}$$

Exchange: the obvious rules in both the Int- and Lin-regions

Weakening: $$\frac{\Delta, \Delta'; \Xi \vdash \mathcal{J} \qquad \Delta; \cdot \vdash A \text{ type}}{\Delta, x : A, \Delta'; \Xi \vdash \mathcal{J}} \text{Int-Weak}$$

Exchange: the obvious rules in both the Int- and Lin-regions

$$\frac{\Delta, x : A, \Delta'; \cdot \vdash B \text{ type} \qquad \Delta; \cdot \vdash a : A}{\Delta, \Delta'[a/x]; \cdot \vdash B[a/x] \text{ type}} \text{Int-Ty-Subst}$$

Substitution: $$\frac{\Delta, x : A, \Delta'; \Xi \vdash b : B \qquad \Delta; \cdot \vdash a : A}{\Delta, \Delta'[a/x]; \Xi[a/x] \vdash b[a/x] : B[a/x]} \text{Int-Tm-Subst}$$

$$\frac{\Delta; \Xi, x : A \vdash b : B \qquad \Delta; \Xi' \vdash a : A}{\Delta; \Xi, \Xi' \vdash b[a/x] : B} \text{Lin-Tm-Subst}$$

Weakening: $$\frac{\Delta, \Delta'; \Xi \vdash \mathcal{J} \qquad \Delta; \cdot \vdash A \text{ type}}{\Delta, x : A, \Delta'; \Xi \vdash \mathcal{J}} \text{Int-Weak}$$

Exchange: the obvious rules in both the Int- and Lin-regions

Substitution:

$$\frac{\Delta, x : A, \Delta'; \cdot \vdash B \text{ type} \qquad \Delta; \cdot \vdash a : A}{\Delta, \Delta'[a/x]; \cdot \vdash B[a/x] \text{ type}} \text{Int-Ty-Subst}$$

$$\frac{\Delta, x : A, \Delta'; \Xi \vdash b : B \qquad \Delta; \cdot \vdash a : A}{\Delta, \Delta'[a/x]; \Xi[a/x] \vdash b[a/x] : B[a/x]} \text{Int-Tm-Subst}$$

$$\frac{\Delta; \Xi, x : A \vdash b : B \qquad \Delta; \Xi' \vdash a : A}{\Delta; \Xi, \Xi' \vdash b[a/x] : B} \text{Lin-Tm-Subst}$$

and all the obvious rules for judgemental equality...

Weakening: $$\frac{\Delta, \Delta'; \Xi \vdash \mathcal{J} \qquad \Delta; \cdot \vdash A \text{ type}}{\Delta, x : A, \Delta'; \Xi \vdash \mathcal{J}} \text{ Int-Weak}$$

Exchange: the obvious rules in both the Int- and Lin-regions

Substitution:

$$\frac{\Delta, x : A, \Delta'; \cdot \vdash B \text{ type} \qquad \Delta; \cdot \vdash a : A}{\Delta, \Delta'[a/x]; \cdot \vdash B[a/x] \text{ type}} \text{ Int-Ty-Subst}$$

$$\frac{\Delta, x : A, \Delta'; \Xi \vdash b : B \qquad \Delta; \cdot \vdash a : A}{\Delta, \Delta'[a/x]; \Xi[a/x] \vdash b[a/x] : B[a/x]} \text{ Int-Tm-Subst}$$

$$\frac{\Delta; \Xi, x : A \vdash b : B \qquad \Delta; \Xi' \vdash a : A}{\Delta; \Xi, \Xi' \vdash b[a/x] : B} \text{ Lin-Tm-Subst}$$

and all the obvious rules for judgemental equality...

Logical rules

Optional natural deduction style (F-, I-, E-, C-, and U-) rules for

- standard type formers from linear logic, in each intuitionistic context: I, $\otimes$, $\multimap$, $!$, $\top$, $\&$, 0, $\oplus$,
- (multiplicative) linear variants of Σ-, Π-, and Id-types from dependent type theory,

with all the commutative conversions one would expect.

Σ, Π, and Id are the ones that require some thought.

Logical rules

Optional natural deduction style (F-, I-, E-, C-, and U-) rules for

- standard type formers from linear logic, in each intuitionistic context: I, $\otimes$, $\multimap$, $!$, $\top$, $\&$, 0, $\oplus$,
- (multiplicative) linear variants of Σ-, Π-, and Id-types from dependent type theory,

with all the commutative conversions one would expect.

Σ, Π, and Id are the ones that require some thought.

Obvious additive Σ-types and Id-types 'boring' as forced to be intuitionistic.

Logical rules

Optional natural deduction style (F-, I-, E-, C-, and U-) rules for

- standard type formers from linear logic, in each intuitionistic context: I, $\otimes$, $\multimap$, $!$, $\top$, $\&$, 0, $\oplus$,
- (multiplicative) linear variants of Σ-, Π-, and Id-types from dependent type theory,

with all the commutative conversions one would expect.

Σ, Π, and Id are the ones that require some thought.

Obvious additive Σ-types and Id-types 'boring' as forced to be intuitionistic.

Multiplicative Σ-types:

$$\frac{\Delta, x : A; \cdot \vdash B \text{ type}}{\Delta; \cdot \vdash \Sigma_{!x:!A} B \text{ type}} \; \Sigma\text{-F}$$

$$\frac{\Delta; \cdot \vdash a : A \qquad \Delta; \Xi \vdash b : B[a/x]}{\Delta; \Xi \vdash !a \otimes b : \Sigma_{!x:!A} B} \; \Sigma\text{-I}$$

$$\frac{\begin{array}{c}\Delta; \cdot \vdash C \text{ type} \\ \Delta; \Xi \vdash t : \Sigma_{!x:!A} B \\ \Delta, x : A; \Xi', y : B \vdash c : C\end{array}}{\Delta; \Xi, \Xi' \vdash \text{let } t \text{ be } !x \otimes y \text{ in } c : C} \; \Sigma\text{-E}$$

and the obvious C- and U- rules.

Multiplicative Π-types:

$$\frac{\Delta, x : A; \cdot \vdash B \text{ type}}{\Delta; \cdot \vdash \Pi_{!x:!A} B \text{ type}} \ \Pi\text{-F}$$

$$\frac{\Delta, x : A; \Xi \vdash b : B}{\Delta; \Xi \vdash \lambda_{!x:!A} b : \Pi_{!x:!A} B} \ \Pi\text{-I} \qquad \frac{\Delta; \cdot \vdash a : A \qquad \Delta; \Xi \vdash f : \Pi_{!x:!A} B}{\Delta; \Xi \vdash f(!a) : B[a/x]} \ \Pi\text{-E}$$

and the obvious C- and U-rules.

Multiplicative Id-types:

$$\frac{\Delta; \cdot \vdash a : A \qquad \Delta; \cdot \vdash a' : A}{\Delta; \cdot \vdash \mathrm{Id}_{!A}(a, a') \text{ type}}\ \text{Id-F}$$

$$\frac{\Delta; \cdot \vdash a : A}{\Delta; \cdot \vdash \mathrm{refl}_{!a} : \mathrm{Id}_{!A}(a, a)}\ \text{Id-I}$$

$$\frac{\begin{array}{l}\Delta, x : A, x' : A; \cdot \vdash D \text{ type} \\ \Delta, z : A; \Xi \vdash d : D[z/x, z/x'] \\ \Delta; \cdot \vdash a : A \\ \Delta; \cdot \vdash a' : A \\ \Delta; \Xi' \vdash p : \mathrm{Id}_{!A}(a, a')\end{array}}{\Delta; \Xi[a/z], \Xi' \vdash \text{let } (a, a', p) \text{ be } (z, z, \mathrm{refl}_{!z}) \text{ in } d : D[a/x, a'/x']}\ \text{Id-E}$$

and the obvious C- and U-rules.

Some metatheorems

Theorem (Consistency)

The full calculus with all logical rules is consistent, both as a logic and type theory, as we have (several) non-trivial models.

Theorem (Interdefinable connectives)

If $x : A$ is not free in B, then

$$\Sigma_{!x:!A} B = !A \otimes B,$$

$$\Pi_{!x:!A} B = !A \multimap B.$$

In particular,

$$\Sigma_{!x:!A} I = !A.$$

Theorem (Interdefinable connectives)

If $x : A$ is not free in B, then

$$\Sigma_{!x:!A} B = !A \otimes B,$$

$$\Pi_{!x:!A} B = !A \multimap B.$$

In particular,

$$\Sigma_{!x:!A} I = !A.$$

If 2 is a type of Booleans with dependent elimination rule, then

$$\Sigma_{!x:!2} B = B(\mathrm{tt}) \oplus B(\mathrm{ff}),$$

$$\Pi_{!x:!2} B = B(\mathrm{tt}) \& B(\mathrm{ff}).$$

Theorem (Interdefinable connectives)

If $x : A$ is not free in B, then

$$\Sigma_{!x:!A} B = !A \otimes B,$$

$$\Pi_{!x:!A} B = !A \multimap B.$$

In particular,

$$\Sigma_{!x:!A} I = !A.$$

If 2 is a type of Booleans with dependent elimination rule, then

$$\Sigma_{!x:!2} B = B(\mathrm{tt}) \oplus B(\mathrm{ff}),$$

$$\Pi_{!x:!2} B = B(\mathrm{tt}) \& B(\mathrm{ff}).$$

Categorical Semantics

First sound and complete categorical semantics for linear dependent types. It fits in with existing traditions.

Designed to be a mixture of (c.f. syntax)

- Benton's linear-non-linear adjunction [semantics for linear types],
- Split comprehension categories, viewed as indexed categories [semantics for dependent types].

Categorical Semantics

First sound and complete categorical semantics for linear dependent types. It fits in with existing traditions.

Designed to be a mixture of (c.f. syntax)

- Benton's linear-non-linear adjunction [semantics for linear types],
- Split comprehension categories, viewed as indexed categories [semantics for dependent types].

Recall: semantics of (intuitionistic) linear types

syntax	*semantics*
structural core	symmetric multicategory $\mathcal{D}$
I- and $\otimes$-types	$\mathcal{D}$ equivalent to symmetric monoidal category
$\multimap$-types	$\mathcal{D}$ symmetric monoidal closed
$\top$- and &-types	finite products in $\mathcal{D}$
0- and $\oplus$-types	finite distributive coproducts in $\mathcal{D}$
!-types	linear exponential comonad† $(!, \mathrm{der}, \delta)$ on $\mathcal{D}$,

† i.e. ! is a comonad that arises from a linear-non-linear adjunction: monoidal adjunction to a cartesian monoidal category.

$$(\mathcal{C}, 1, \times) \underset{M}{\overset{L}{\rightleftarrows}} (\mathcal{D}, I, \otimes) \quad (L \dashv M)$$

Recall: semantics of (intuitionistic) linear types

syntax	*semantics*
structural core	symmetric multicategory $\mathcal{D}$
I- and $\otimes$-types	$\mathcal{D}$ equivalent to symmetric monoidal category
$\multimap$-types	$\mathcal{D}$ symmetric monoidal closed
$\top$- and &-types	finite products in $\mathcal{D}$
0- and $\oplus$-types	finite distributive coproducts in $\mathcal{D}$
!-types	linear exponential comonad† $(!, \mathrm{der}, \delta)$ on $\mathcal{D}$,

† i.e. ! is a comonad that arises from a linear-non-linear adjunction: monoidal adjunction to a cartesian monoidal category.

$$(\mathcal{C}, 1, \times) \underset{M}{\overset{L}{\rightleftarrows}} (\mathcal{D}, I, \otimes) \quad (L \dashv M)$$

Recall: semantics of dependent types

Use an equivalent of categories with families:

syntax	*semantics*
structural core	(strict) indexed cartesian multicategory
	$\cdot \in \mathcal{C}^{op} \xrightarrow{\mathcal{I}} \mathrm{CMultCat} \quad (-\{f\} := \mathcal{I}(f))$
	with *fully faithful* comprehension† $(\mathbf{p}, \mathbf{v})$
1- and $\times$-types	$\mathcal{I}$ factoring over CMCat
	(i.e. $\mathcal{I}$ indexed cartesian monoidal category)
$\rightarrow$-types	$\mathcal{I}$ factoring over CCCat

Recall: semantics of dependent types

Use an equivalent of categories with families:

syntax	*semantics*
structural core	(strict) indexed cartesian multicategory
	$\cdot \in \mathcal{C}^{op} \xrightarrow{\mathcal{I}} \mathrm{CMultCat} \quad (-\{f\} := \mathcal{I}(f))$
	with *fully faithful* comprehension† $(\mathbf{p}, \mathbf{v})$
1- and $\times$-types	$\mathcal{I}$ factoring over CMCat
	(i.e. $\mathcal{I}$ indexed cartesian monoidal category)
$\to$-types	$\mathcal{I}$ factoring over CCCat
(extensional) Σ-types	left adjoints to $-\{\mathbf{p}\}$ satisfying
	Beck-Chevalley and Frobenius reciprocity
(extensional) Π-types	right adjoints to $-\{\mathbf{p}\}$ satisfying BC
(extensional) Id-types	left adjoints to $-\{\mathrm{diag}\}$ satisfying BC

Recall: semantics of dependent types

Use an equivalent of categories with families:

syntax	*semantics*
structural core	(strict) indexed cartesian multicategory
	$\cdot \in \mathcal{C}^{op} \xrightarrow{\mathcal{I}} \mathrm{CMultCat} \quad (-\{f\} := \mathcal{I}(f))$
	with *fully faithful* comprehension† $(\mathbf{p}, \mathbf{v})$
1- and $\times$-types	$\mathcal{I}$ factoring over CMCat
	(i.e. $\mathcal{I}$ indexed cartesian monoidal category)
$\to$-types	$\mathcal{I}$ factoring over CCCat
(extensional) Σ-types	left adjoints to $-\{\mathbf{p}\}$ satisfying
	Beck-Chevalley and Frobenius reciprocity
(extensional) Π-types	right adjoints to $-\{\mathbf{p}\}$ satisfying BC
(extensional) Id-types	left adjoints to $-\{\mathrm{diag}\}$ satisfying BC

†: we say $\mathcal{C}^{op} \xrightarrow{\mathcal{I}} \mathrm{CMultCat}$ satisfies *comprehension axiom* if for all $\Delta \in \mathrm{ob}(\mathcal{C})$, $A \in \mathrm{ob}(\mathcal{I}(\Delta))$

$$(\mathcal{C}/\Delta)^{op} \longrightarrow \mathrm{Set}$$

$$f \longmapsto \mathcal{I}(\mathrm{dom}(f))(\cdot, A\{f\})$$

is representable: $\mathcal{I}(\mathrm{dom}(f))(\cdot, A\{f\}) \xrightarrow{\cong} \mathcal{C}/\Delta(f, \mathbf{p}_{\Delta,A})$.

Call *fully faithful* if $A \mapsto \mathbf{p}_{\Delta,A}$ defines fully faithful functor.

†: we say $\mathcal{C}^{op} \xrightarrow{\mathcal{I}} \mathrm{CMultCat}$ satisfies *comprehension axiom* if for all $\Delta \in \mathrm{ob}(\mathcal{C})$, $A \in \mathrm{ob}(\mathcal{I}(\Delta))$

$$(\mathcal{C}/\Delta)^{op} \longrightarrow \mathrm{Set}$$

$$f \longmapsto \mathcal{I}(\mathrm{dom}(f))(\cdot, A\{f\})$$

is representable: $\mathcal{I}(\mathrm{dom}(f))(\cdot, A\{f\}) \xrightarrow{\cong} \mathcal{C}/\Delta(f, \mathbf{p}_{\Delta,A})$.

Call *fully faithful* if $A \mapsto \mathbf{p}_{\Delta,A}$ defines fully faithful functor.

Equivalently, by Yoneda, we have

- a representing object $\mathbf{p}_{\Delta,A}$ (projection)
- a universal element $\mathbf{v}_{\Delta,A}$ (diagonal).

†: we say $\mathcal{C}^{op} \xrightarrow{\mathcal{I}} \mathrm{CMultCat}$ satisfies *comprehension axiom* if for all $\Delta \in \mathrm{ob}(\mathcal{C})$, $A \in \mathrm{ob}(\mathcal{I}(\Delta))$

$$(\mathcal{C}/\Delta)^{op} \longrightarrow \mathrm{Set}$$

$$f \longmapsto \mathcal{I}(\mathrm{dom}(f))(\cdot, A\{f\})$$

is representable: $\mathcal{I}(\mathrm{dom}(f))(\cdot, A\{f\}) \xrightarrow{\cong} \mathcal{C}/\Delta(f, \mathbf{p}_{\Delta,A})$.

Call *fully faithful* if $A \mapsto \mathbf{p}_{\Delta,A}$ defines fully faithful functor.

Equivalently, by Yoneda, we have

- a representing object $\mathbf{p}_{\Delta,A}$ (projection)
- a universal element $\mathbf{v}_{\Delta,A}$ (diagonal).

Semantics of linear dependent types

Nothing surprising here...

syntax	*semantics*
structural core	(strict) indexed symmetric multicategory
	$\cdot \in \mathcal{C}^{op} \xrightarrow{\mathcal{L}} \mathrm{SMultCat} \quad (-\{f\} := \mathcal{L}(f))$
	with comprehension† $(\mathbf{p}, \mathbf{v})$
I- and $\otimes$-types	$\mathcal{L}$ factoring over SMCat
	(i.e. $\mathcal{L}$ indexed symmetric monoidal category)
$\multimap$-types	$\mathcal{L}$ factoring over SMCCat

Semantics of linear dependent types

Nothing surprising here...

syntax	*semantics*
structural core	(strict) indexed symmetric multicategory
	$\cdot \in \mathcal{C}^{op} \xrightarrow{\mathcal{L}} \mathrm{SMultCat} \quad (-\{f\} := \mathcal{L}(f))$
	with comprehension† $(\mathbf{p}, \mathbf{v})$
I- and $\otimes$-types	$\mathcal{L}$ factoring over SMCat
	(i.e. $\mathcal{L}$ indexed symmetric monoidal category)
$\multimap$-types	$\mathcal{L}$ factoring over SMCCat
$\top$- and $\&$-types	$\mathcal{L}$ factoring over CMCat
0- and $\oplus$-types	$\mathcal{L}$ factoring over $\mathrm{distr.coCMCat}$

Semantics of linear dependent types

Nothing surprising here...

syntax	*semantics*
structural core	(strict) indexed symmetric multicategory
	$\cdot \in \mathcal{C}^{op} \xrightarrow{\mathcal{L}} \mathrm{SMultCat} \quad (-\{f\} := \mathcal{L}(f))$
	with comprehension† $(\mathbf{p}, \mathbf{v})$
I- and $\otimes$-types	$\mathcal{L}$ factoring over SMCat
	(i.e. $\mathcal{L}$ indexed symmetric monoidal category)
$\multimap$-types	$\mathcal{L}$ factoring over SMCCat
$\top$- and &-types	$\mathcal{L}$ factoring over CMCat
0- and $\oplus$-types	$\mathcal{L}$ factoring over $\mathrm{distr.coCMCat}$
(ext.) mult. Σ-types	left adjoints to $-\{\mathbf{p}\}$ satisfying BC and Frob.
(ext.) mult. Π-types	right adjoints to $-\{\mathbf{p}\}$ satisfying BC
(ext.) mult. Id-types	left adjoints to $-\{\mathrm{diag}\}$ satisfying BC

Semantics of linear dependent types

Nothing surprising here...

syntax	*semantics*
structural core	(strict) indexed symmetric multicategory
	$\cdot \in \mathcal{C}^{op} \xrightarrow{\mathcal{L}} \mathrm{SMultCat} \quad (-\{f\} := \mathcal{L}(f))$
	with comprehension† $(\mathbf{p}, \mathbf{v})$
I- and $\otimes$-types	$\mathcal{L}$ factoring over SMCat
	(i.e. $\mathcal{L}$ indexed symmetric monoidal category)
$\multimap$-types	$\mathcal{L}$ factoring over SMCCat
$\top$- and &-types	$\mathcal{L}$ factoring over CMCat
0- and $\oplus$-types	$\mathcal{L}$ factoring over $\mathrm{distr.coCMCat}$
(ext.) mult. Σ-types	left adjoints to $-\{\mathbf{p}\}$ satisfying BC and Frob.
(ext.) mult. Π-types	right adjoints to $-\{\mathbf{p}\}$ satisfying BC
(ext.) mult. Id-types	left adjoints to $-\{\mathrm{diag}\}$ satisfying BC

...except

syntax	*semantics*
!-types	comprehension induces unique linear exponential comonad on each fibre $\mathcal{L}(\Delta)$.

Recall, comprehension defines morphism of indexed categories onto $\mathcal{I} \subset_{\text{full}} \mathcal{C}/-$ (equivalence earlier; now monoidal adjunction!)

$$\mathcal{L}(\Delta) \xrightarrow{M_\Delta} \mathcal{I}(\Delta)$$

$$A \longmapsto \mathbf{p}_{\Delta,A}$$

(+ obvious definition on morphisms)

...except

syntax	*semantics*
!-types	comprehension induces unique linear exponential comonad on each fibre $\mathcal{L}(\Delta)$.

Recall, comprehension defines morphism of indexed categories onto $\mathcal{I} \subset_{\text{full}} \mathcal{C}/-$ (equivalence earlier; now monoidal adjunction!)

$$\mathcal{L}(\Delta) \xrightarrow{M_\Delta} \mathcal{I}(\Delta) \qquad A \longmapsto \mathbf{p}_{\Delta,A}$$

(+ obvious definition on morphisms)

!-types iff this has strong monoidal left adjoint L_Δ:
linear-non-linear adjunction $L \dashv M : \mathcal{L} \longrightarrow \mathcal{I}$ of indexed categories.

...except

syntax	*semantics*
!-types	comprehension induces unique linear exponential comonad on each fibre $\mathcal{L}(\Delta)$.

Recall, comprehension defines morphism of indexed categories onto $\mathcal{I} \subset_{\text{full}} \mathcal{C}/-$ (equivalence earlier; now monoidal adjunction!)

$$\begin{array}{ccc} \mathcal{L}(\Delta) & \xrightarrow{M_\Delta} & \mathcal{I}(\Delta) \\ A & \longmapsto & \mathbf{p}_{\Delta,A} \end{array}$$

(+ obvious definition on morphisms)

!-types iff this has strong monoidal left adjoint L_Δ:
linear-non-linear adjunction $L \dashv M : \mathcal{L} \longrightarrow \mathcal{I}$ of indexed categories.

Dependent Seely Isomorphisms?

Theorem (Type Formers in $\mathcal{I}$ and Dependent Seely Isomorphisms)

The intuitionistic type formers in $\mathcal{I}$ relate to the linear ones in $\mathcal{L}$ as follows (where $L \dashv M$ induces $!$):

$$\Sigma_{!A}!B \cong L(\Sigma_{MA}MB) \qquad M\Pi_{!B}C \cong \Pi_{MB}MC$$
$$\mathrm{Id}_{!A}(!B) \cong L\mathrm{Id}_{MA}(MB).$$

$\mathcal{I}$ supports Σ- respectively Id-types iff we have "additive" Σ- resp. Id-types, that is $\Sigma^{\&}_{A}B, \mathrm{Id}^{\&}_{A}(B) \in \mathrm{ob}(\mathcal{L})$ s.t.

$$M\Sigma^{\&}_{A}B \cong \Sigma_{MA}MB \quad \text{and hence} \quad !\Sigma^{\&}_{A}B \cong \Sigma^{\otimes}_{!A}!B \quad \text{resp.}$$
$$M\mathrm{Id}^{\&}_{A}(B) \cong \mathrm{Id}_{MA}(MB) \quad \text{and hence} \quad !\mathrm{Id}^{\&}_{A}(B) \cong \mathrm{Id}^{\otimes}_{!A}(!B).$$

In this situation when modelling DTT in co-Kleisli category $\mathcal{L}(\cdot)_!$.

Dependent Seely Isomorphisms?

Theorem (Type Formers in $\mathcal{I}$ and Dependent Seely Isomorphisms)

The intuitionistic type formers in $\mathcal{I}$ relate to the linear ones in $\mathcal{L}$ as follows (where $L \dashv M$ induces $!$):

$$\Sigma_{!A}!B \cong L(\Sigma_{MA}MB) \qquad M\Pi_{!B}C \cong \Pi_{MB}MC$$

$$\mathrm{Id}_{!A}(!B) \cong L\mathrm{Id}_{MA}(MB).$$

$\mathcal{I}$ supports Σ- respectively Id-types iff we have "additive" Σ- resp. Id-types, that is $\Sigma^{\&}_{A}B, \mathrm{Id}^{\&}_{A}(B) \in \mathrm{ob}(\mathcal{L})$ s.t.

$$M\Sigma^{\&}_{A}B \cong \Sigma_{MA}MB \quad \text{and hence} \quad !\Sigma^{\&}_{A}B \cong \Sigma^{\otimes}_{!A}!B \quad \text{resp.}$$

$$M\mathrm{Id}^{\&}_{A}(B) \cong \mathrm{Id}_{MA}(MB) \quad \text{and hence} \quad !\mathrm{Id}^{\&}_{A}(B) \cong \mathrm{Id}^{\otimes}_{!A}(!B).$$

In this situation when modelling DTT in co-Kleisli category $\mathcal{L}(\cdot)_!$.

Dependent Seely Isomorphisms?

Theorem (Type Formers in $\mathcal{I}$ and Dependent Seely Isomorphisms)

The intuitionistic type formers in $\mathcal{I}$ relate to the linear ones in $\mathcal{L}$ as follows (where $L \dashv M$ induces $!$):

$$\Sigma_{!A}!B \cong L(\Sigma_{MA}MB) \qquad M\Pi_{!B}C \cong \Pi_{MB}MC$$

$$\mathrm{Id}_{!A}(!B) \cong L\mathrm{Id}_{MA}(MB).$$

$\mathcal{I}$ supports Σ- respectively Id-types iff we have "additive" Σ- resp. Id-types, that is $\Sigma^{\&}_{A}B, \mathrm{Id}^{\&}_{A}(B) \in \mathrm{ob}(\mathcal{L})$ s.t.

$$M\Sigma^{\&}_{A}B \cong \Sigma_{MA}MB \quad \text{and hence} \quad !\Sigma^{\&}_{A}B \cong \Sigma^{\otimes}_{!A}!B \quad \text{resp.}$$

$$M\mathrm{Id}^{\&}_{A}(B) \cong \mathrm{Id}_{MA}(MB) \quad \text{and hence} \quad !\mathrm{Id}^{\&}_{A}(B) \cong \mathrm{Id}^{\otimes}_{!A}(!B).$$

In this situation when modelling DTT in co-Kleisli category $\mathcal{L}(\cdot)_{!}$.

Soundness & completeness

Theorem (Soundness & Completeness)

The semantics presented is both sound and complete.

Theorem (Failure of Definability)

In line with the tradition of categorical semantics of dependent types, definability fails. This choice was made to fit in smoothly with convention.

Corollary (Restoring Definability)

By a slight modification, either by extending the syntax or restricting the semantics, though, we can easily obtain the situation of a real internal language.

Cofree type dependency

Theorem

The forgetful functor $\mathrm{SMCat}^{\mathrm{Set}^{op}}_{\mathrm{compr}} \xrightarrow{\mathrm{ev}_1} \mathrm{SMCat}$ *has a right adjoint* $\mathrm{Fam} : \mathcal{V} \mapsto \mathrm{Cat}(-, \mathcal{V})$.

Type formers in $\mathrm{Fam}(\mathcal{V})$*:*

Σ-*types*	$\mathcal{V}$ *small coproducts that distribute over* $\otimes$
Π-*types*	$\mathcal{V}$ *small products*
Id-*types*	$\mathcal{V}$ *with initial object (*$\mathcal{V}$ *also has 1*$\Rightarrow$*only if)*
$\multimap$-*types*	$\mathcal{V}$ *monoidal closed (note* $\otimes$ *then distributes)*
0- *and* $\oplus$-*types*	$\mathcal{V}$ *finite distributive coproducts*
$\top$- *and* &-*types*	$\mathcal{V}$ *finite products*
!-*types*	$\mathcal{V}$ *has small distributive sums of I*

Cofree type dependency

Theorem

The forgetful functor $\mathrm{SMCat}^{\mathrm{Set}^{op}}_{\mathrm{compr}} \xrightarrow{\mathrm{ev}_1} \mathrm{SMCat}$ *has a right adjoint* $\mathrm{Fam} : \mathcal{V} \mapsto \mathrm{Cat}(-, \mathcal{V})$.

Type formers in $\mathrm{Fam}(\mathcal{V})$*:*

Σ-*types*	$\mathcal{V}$ *small coproducts that distribute over* $\otimes$
Π-*types*	$\mathcal{V}$ *small products*
Id-*types*	$\mathcal{V}$ *with initial object (*$\mathcal{V}$ *also has 1⇒only if)*
$\multimap$-*types*	$\mathcal{V}$ *monoidal closed (note* $\otimes$ *then distributes)*
0- *and* $\oplus$-*types*	$\mathcal{V}$ *finite distributive coproducts*
$\top$- *and* &-*types*	$\mathcal{V}$ *finite products*
!-*types*	$\mathcal{V}$ *has small distributive sums of I*

Dependency in Coherence Spaces and Games

Theorem (Work with Samson Abramsky and Radha Jagadeesan)

The usual models of linear logic in coherence spaces and games come with a completely natural notion of dependent type, as well as Σ-, Π-, *and* Id-*type (extensional if total functions).*

Comprehension is not satisfied, however, over co-Kleisli category, as we don't have additive Σ-*types* $\Sigma^{\&}_{A} B$.

Dependency in Coherence Spaces and Games

Theorem (Work with Samson Abramsky and Radha Jagadeesan)

The usual models of linear logic in coherence spaces and games come with a completely natural notion of dependent type, as well as Σ-, Π-, and Id-type (extensional if total functions).

Comprehension is not satisfied, however, over co-Kleisli category, as we don't have additive Σ-types $\Sigma^{\&}_{A} B$.

2 (equivalent) options to get an actual model:

- *use (fragment of) co-Eilenberg-Moore as category of contexts;*
- *add extra structure to objects so we get additive Σ-types and get model over modified co-Kleisli.*

Dependency in Coherence Spaces and Games

Theorem (Work with Samson Abramsky and Radha Jagadeesan)

The usual models of linear logic in coherence spaces and games come with a completely natural notion of dependent type, as well as Σ-, Π-, *and* Id-*type (extensional if total functions).*

Comprehension is not satisfied, however, over co-Kleisli category, as we don't have additive Σ-*types* $\Sigma^{\&}_{A} B$.

2 (equivalent) options to get an actual model:

- *use (fragment of) co-Eilenberg-Moore as category of contexts;*
- *add extra structure to objects so we get additive* Σ-*types and get model over modified co-Kleisli.*

Further directions

Questions:

- Intensional ILDTT? Lack of dependent E-rules a problem?
- Should we expect natural models in !-co-Kleisli categories? If not, should we re-evaluate its status as canonical model of intuitionistic logic?

Further directions

Questions:

- Intensional ILDTT? Lack of dependent E-rules a problem?
- Should we expect natural models in !-co-Kleisli categories? If not, should we re-evaluate its status as canonical model of intuitionistic logic?
- Can we make sense of additive Σ- and Id-types syntactically?

Further directions

Questions:

- Intensional ILDTT? Lack of dependent E-rules a problem?
- Should we expect natural models in !-co-Kleisli categories? If not, should we re-evaluate its status as canonical model of intuitionistic logic?
- Can we make sense of additive Σ- and Id-types syntactically?
- Relationship with linear predicate logic?

Further directions

Questions:

- Intensional ILDTT? Lack of dependent E-rules a problem?
- Should we expect natural models in !-co-Kleisli categories? If not, should we re-evaluate its status as canonical model of intuitionistic logic?
- Can we make sense of additive Σ- and Id-types syntactically?
- Relationship with linear predicate logic?

Further directions

Further models:

- Models related to quantum theory.

Hope

We can construct a model of ILDTT to represent quantum information theory parametrised by classical information theory.

- Models in stable homotopy theory.

Hope

Spectra parametrised over spaces form a model of (intensional) ILDTT, with $\Sigma, \Pi, \mathrm{Id}, !, I, \otimes, \multimap, \top, \&, 0, \oplus$-types. Here, $! = \Sigma^\infty \Omega^\infty$. Connections with the Goodwillie calculus?

Further directions

Further models:

- Models related to quantum theory.

Hope

We can construct a model of ILDTT to represent quantum information theory parametrised by classical information theory.

- Models in stable homotopy theory.

Hope

Spectra parametrised over spaces form a model of (intensional) ILDTT, with $\Sigma, \Pi, \mathrm{Id}, !, I, \otimes, \multimap, \top, \&, 0, \oplus$*-types. Here,* $! = \Sigma^\infty \Omega^\infty$. *Connections with the Goodwillie calculus?*