Tests, Games, and Martin-Löf's Meaning Explanations for Intuitionistic Type Theory

Peter Dybjer

Chalmers tekniska högskola, Göteborg
(joint work in progress with Pierre Clairambault, University of Cambridge)

Institute for Advanced Study
Princeton, NJ
30 November 2012

Intuitionistic logic and intuitionistic type theory

Intuitionistic logic:

Intuitionistic logic and intuitionistic type theory

Intuitionistic logic:

1908 **BHK.** Brouwer. Kolmogorov, a calculus of problems. Heyting, a calculus of intended constructions.

Intuitionistic logic and intuitionistic type theory

Intuitionistic logic:

1908 **BHK.** Brouwer. Kolmogorov, a calculus of problems. Heyting, a calculus of intended constructions.

1945 Kleene, realizability model.

Intuitionistic logic and intuitionistic type theory

Intuitionistic logic:

1908 **BHK.** Brouwer. Kolmogorov, a calculus of problems. Heyting, a calculus of intended constructions.

1945 Kleene, realizability model.

1969 Howard, formulas as types. De Bruijn, Automath. Lawvere, hyperdoctrines. Scott, Constructive Validity.

Intuitionistic logic and intuitionistic type theory

Intuitionistic logic:

1908 **BHK.** Brouwer. Kolmogorov, a calculus of problems. Heyting, a calculus of intended constructions.

1945 Kleene, realizability model.

1969 Howard, formulas as types. De Bruijn, Automath. Lawvere, hyperdoctrines. Scott, Constructive Validity.

Intuitionistic type theory:

Intuitionistic logic and intuitionistic type theory

Intuitionistic logic:

1908 **BHK.** Brouwer. Kolmogorov, a calculus of problems. Heyting, a calculus of intended constructions.

1945 Kleene, realizability model.

1969 Howard, formulas as types. De Bruijn, Automath. Lawvere, hyperdoctrines. Scott, Constructive Validity.

Intuitionistic type theory:

1972 Martin-Löf, intensional intuitionistic type theory, universes, proof theoretic properties.

Intuitionistic logic and intuitionistic type theory

Intuitionistic logic:

1908 **BHK.** Brouwer. Kolmogorov, a calculus of problems. Heyting, a calculus of intended constructions.

1945 Kleene, realizability model.

1969 Howard, formulas as types. De Bruijn, Automath. Lawvere, hyperdoctrines. Scott, Constructive Validity.

Intuitionistic type theory:

1972 Martin-Löf, intensional intuitionistic type theory, universes, proof theoretic properties.

1974 Aczel, realizability model.

Intuitionistic logic and intuitionistic type theory

Intuitionistic logic:

1908 **BHK.** Brouwer. Kolmogorov, a calculus of problems. Heyting, a calculus of intended constructions.

1945 Kleene, realizability model.

1969 Howard, formulas as types. De Bruijn, Automath. Lawvere, hyperdoctrines. Scott, Constructive Validity.

Intuitionistic type theory:

1972 Martin-Löf, intensional intuitionistic type theory, universes, proof theoretic properties.

1974 Aczel, realizability model.

1979 Martin-Löf, **meaning explanations**, extensional intuitionistic type theory.

Intuitionistic logic and intuitionistic type theory

Intuitionistic logic:

1908 **BHK.** Brouwer. Kolmogorov, a calculus of problems. Heyting, a calculus of intended constructions.

1945 Kleene, realizability model.

1969 Howard, formulas as types. De Bruijn, Automath. Lawvere, hyperdoctrines. Scott, Constructive Validity.

Intuitionistic type theory:

1972 Martin-Löf, intensional intuitionistic type theory, universes, proof theoretic properties.

1974 Aczel, realizability model.

1979 Martin-Löf, **meaning explanations**, extensional intuitionistic type theory.

1986 Martin-Löf, intensional intuitionistic type theory based on a logical framework (set-type distinction)

Intuitionistic Type Theory - a language for both mathematics and programming

- Full-scale framework for constructive mathematics in the style of Bishop ("ZF for intuitionism"). Others are e g
  - Myhill-Aczel constructive set theory,
  - Aczel-Feferman style type-free theories.
- A functional programming language with dependent types where all programs terminate (core of NuPRL, Coq, Agda, etc)

Type formers of intuitionistic type theory

To interpret predicate logic with identity:

$$\Pi x : A.B, \Sigma x : A.B, A + B, \mathrm{N}_0, \mathrm{N}_1, \mathrm{I}(A, a, b)$$

Other mathematical objects

$$\mathrm{N}, \mathrm{W}x : A.B, \mathrm{U}, \mathrm{U}_1, \mathrm{U}_2, \ldots$$

Type formers of intuitionistic type theory

To interpret predicate logic with identity:

$$\Pi x : A.B, \Sigma x : A.B, A + B, \mathrm{N}_0, \mathrm{N}_1, \mathrm{I}(A, a, b)$$

Other mathematical objects

$$\mathrm{N}, \mathrm{W}x : A.B, \mathrm{U}, \mathrm{U}_1, \mathrm{U}_2, \ldots$$

- Extensions with general notion of inductive definition – important for programming.

Type formers of intuitionistic type theory

To interpret predicate logic with identity:

$$\Pi x : A.B, \Sigma x : A.B, A + B, \mathrm{N}_0, \mathrm{N}_1, \mathrm{I}(A, a, b)$$

Other mathematical objects

$$\mathrm{N}, \mathrm{W}x : A.B, \mathrm{U}, \mathrm{U}_1, \mathrm{U}_2, \ldots$$

- Extensions with general notion of inductive definition – important for programming.
- Extensions into the constructive higher infinite: super universes, universe hierarchies, Mahlo universes, autonomous Mahlo universes, general inductive-recursive definitions etc.

Meaning explanations extend as well.

What are Martin-Löf's meaning explanations?

Meaning explanations. Also called

> *direct semantics, intuitive semantics, standard semantics, syntactico-semantical approach*
>
> *"pre-mathematical" as opposed to "meta-mathematical":*

References:

- *Constructive Mathematics and Computer Programming*, LMPS 1979;
- *Intuitionistic Type Theory*, Bibliopolis, 1984;
- *Philosophical Implications of Type Theory*, Firenze lectures 1987.

Before 1979: normalization proofs, but no meaning explanations.

Natural numbers - computation to canonical form

Start with *untyped computation system* with notion of computation of *closed* expression to *canonical form* (whnf) $a \Rightarrow v$.

$$\mathrm{N} \Rightarrow \mathrm{N}$$

$$0 \Rightarrow 0 \qquad\qquad \mathrm{s}(a) \Rightarrow \mathrm{s}(a)$$

$$\frac{c \Rightarrow 0 \quad d \Rightarrow v}{\mathrm{R}(c,d,e) \Rightarrow v} \qquad\qquad \frac{c \Rightarrow \mathrm{s}(a) \quad e(a,\mathrm{R}(a,d,e)) \Rightarrow v}{\mathrm{R}(c,d,e) \Rightarrow v}$$

Natural numbers - meaning explanations

$$\frac{A \Rightarrow \mathrm{N}}{A\ type}$$

$$\frac{A \Rightarrow \mathrm{N} \quad A' \Rightarrow \mathrm{N}}{A = A'}$$

$$\frac{A \Rightarrow \mathrm{N} \quad a \Rightarrow 0}{a : A} \qquad \frac{A \Rightarrow \mathrm{N} \quad a \Rightarrow \mathrm{s}(b) \quad b : \mathrm{N}}{a : A}$$

$$\frac{A \Rightarrow \mathrm{N} \quad a \Rightarrow 0 \quad a' \Rightarrow 0}{a = a' : A} \qquad \frac{A \Rightarrow \mathrm{N} \quad a \Rightarrow \mathrm{s}(b) \quad a' \Rightarrow \mathrm{s}(b') \quad b = b' : \mathrm{N}}{a = a' : A}$$

Natural numbers - meaning explanations

$$\frac{A \Rightarrow \mathrm{N}}{A \; type}$$

$$\frac{A \Rightarrow \mathrm{N} \quad A' \Rightarrow \mathrm{N}}{A = A'}$$

$$\frac{A \Rightarrow \mathrm{N} \quad a \Rightarrow 0}{a : A} \qquad \frac{A \Rightarrow \mathrm{N} \quad a \Rightarrow \mathrm{s}(b) \quad b : \mathrm{N}}{a : A}$$

$$\frac{A \Rightarrow \mathrm{N} \quad a \Rightarrow 0 \quad a' \Rightarrow 0}{a = a' : A} \qquad \frac{A \Rightarrow \mathrm{N} \quad a \Rightarrow \mathrm{s}(b) \quad a' \Rightarrow \mathrm{s}(b') \quad b = b' : \mathrm{N}}{a = a' : A}$$

How to understand these rules, meta-mathematically (realizability) or pre-mathematically (meaning explanations)?

General pattern

$$\frac{A \Rightarrow \mathrm{C}(a_1, \ldots, a_m) \quad \cdots}{A \ type}$$

$$\frac{A \Rightarrow \mathrm{C}(a_1, \ldots, a_m) \quad A' \Rightarrow \mathrm{C}(a'_1, \ldots, a'_m) \quad \cdots}{A = A'}$$

where C is an m-place type constructor $(\mathrm{N}, \Pi, \Sigma, \mathrm{I}, \mathrm{U}$, etc), and

$$\frac{A \Rightarrow \mathrm{C}(a_1, \ldots, a_m) \quad a \Rightarrow \mathrm{c}(b_1, \ldots, b_n) \quad \cdots}{a : A}$$

$$\frac{A \Rightarrow \mathrm{C}(a_1, \ldots, a_m) \quad a \Rightarrow \mathrm{c}(b_1, \ldots, b_n) \quad a' \Rightarrow \mathrm{c}(b'_1, \ldots, b'_n) \quad \cdots}{a = a' : A}$$

where c is an n-place term constructor for the m-place type constructor C ($0, \mathrm{s}$ for N; λ for Π; $\mathrm{N}, \Pi, \ldots$ for U; etc).

Universe (ła Russell) - meaning explanations

$$\frac{A \Rightarrow \mathrm{U}}{A\ type}$$

$$\frac{A \Rightarrow \mathrm{U} \quad A' \Rightarrow \mathrm{U}}{A = A'}$$

$$\frac{A \Rightarrow \mathrm{U} \quad a \Rightarrow \mathrm{N}}{a : A} \qquad \frac{A \Rightarrow \mathrm{U} \quad a \Rightarrow \Pi x : B.C \quad B : \mathrm{U} \quad x : B \vdash C : \mathrm{U}}{a : A} \quad \cdots$$

$$\frac{A \Rightarrow \mathrm{U} \quad a \Rightarrow \mathrm{N} \quad a' \Rightarrow \mathrm{N}}{a = a' : A}$$

$$\frac{A \Rightarrow \mathrm{U} \quad a \Rightarrow \Pi x : B.C \quad a' \Rightarrow \Pi x : B'.C' \quad B = B' : \mathrm{U} \quad x : B \vdash C = C' : \mathrm{U}}{a = a' : A} \, .$$

(Remark: equality of types means "same elements and same equal elements" in Martin-Löf 1979)

The meaning of hypothetical judgments

$$x_1 : A_1, \ldots, x_n : A_n \vdash a : A$$

means that

The meaning of hypothetical judgments

$$x_1 : A_1, \ldots, x_n : A_n \vdash a : A$$

means that

$$a[a_1, \ldots, a_n / x_1, \ldots, x_n) : A[a_1, \ldots, a_n / x_1, \ldots, x_n]$$

provided

$$\begin{aligned} &a_1 : A_1, \\ &\vdots \\ &a_n : A_n[a_1, \ldots, a_{n-1} / x_1, \ldots, x_{n-1}], \end{aligned}$$

and, moreover,

$$a[a_1, \ldots, a_n / x_1, \ldots, x_n] = a[b_1, \ldots, b_n / x_1, \ldots, x_n] : A[a_1, \ldots, a_n / x_1, \ldots, x_n]$$

provided

$$\begin{aligned} &a_1 = b_1 : A_1, \\ &\vdots \end{aligned}$$

However ...

However ...

The meaning of the identity type (Martin-Löf 1984, *Intuitionistic Type Theory*, p 32)

> *We now have to explain how to form canonical elements of* $\mathrm{I}(A, a, b)$*. The standard way to know that* $\mathrm{I}(A, a, b)$ *is true is to have* $a = b : A$*. Thus the introduction rule is simply: if* $a = b : A$*, then there is a canonical proof* r *of* $\mathrm{I}(A, a, b)$*. Here* r *does not depend on* a, b *or* A*; it does not matter what canonical element* $\mathrm{I}(A, a, b)$ *has when* $a = b : A$*, as long as it has one.*

Extensional intuitionistic type theory

We can then validate

I-*elimination*

$$\frac{c : \mathrm{I}(A, a, b)}{a = b : A}$$

I-*equality*

$$\frac{c : \mathrm{I}(A, a, b)}{c = \mathrm{r} : \mathrm{I}(A, a, b)}$$

which lead to:

- *extensional* intuitionistic type theory, with function extensionality
- non-normalizing terms (although $a : A$ implies that a and A have *canonical* form for *closed* terms).
- undecidable judgments, e g, it is not decidable whether $a : A$ even if we know that A *type*.

Extensional intuitionistic type theory

We can then validate

I-*elimination*

$$\frac{c : \mathrm{I}(A, a, b)}{a = b : A}$$

I-*equality*

$$\frac{c : \mathrm{I}(A, a, b)}{c = \mathrm{r} : \mathrm{I}(A, a, b)}$$

which lead to:

- *extensional* intuitionistic type theory, with function extensionality
- non-normalizing terms (although $a : A$ implies that a and A have *canonical* form for *closed* terms).
- undecidable judgments, e g, it is not decidable whether $a : A$ even if we know that A *type*.

Hence, Martin-Löf rejected the above rules of I-elimination and equality in 1986.

Extensional intuitionistic type theory

We can then validate

I-*elimination*

$$\frac{c : \mathrm{I}(A, a, b)}{a = b : A}$$

I-*equality*

$$\frac{c : \mathrm{I}(A, a, b)}{c = \mathrm{r} : \mathrm{I}(A, a, b)}$$

which lead to:

- *extensional* intuitionistic type theory, with function extensionality
- non-normalizing terms (although $a : A$ implies that a and A have *canonical* form for *closed* terms).
- undecidable judgments, e g, it is not decidable whether $a : A$ even if we know that A *type*.

Hence, Martin-Löf rejected the above rules of I-elimination and equality in 1986. (But they remained valid at Cornell.)

Extensional meaning explanations for intensional type theory?

Possible attitudes:

Extensional meaning explanations for intensional type theory?

Possible attitudes:

- The meaning explanations provide *necessary*, but *not sufficient* criteria for validity of judgments. Judgments should also be *decidable* (a *global* condition).

Extensional meaning explanations for intensional type theory?

Possible attitudes:

- The meaning explanations provide *necessary*, but *not sufficient* criteria for validity of judgments. Judgments should also be *decidable* (a *global* condition).
- The judgments of extensional type theory are all valid, but we may sometimes prefer intensional type theory for pragmatic reasons.

Extensional meaning explanations for intensional type theory?

Possible attitudes:

- The meaning explanations provide *necessary*, but *not sufficient* criteria for validity of judgments. Judgments should also be *decidable* (a *global* condition).
- The judgments of extensional type theory are all valid, but we may sometimes prefer intensional type theory for pragmatic reasons.
- We need alternative *meaning explanations based on the normalization of open terms*, rather than just closed terms. Cf Martin-Löf 2009.

Proof animation

Proof animation

Knuth 1977:

> *Beware of bugs in the above code; I have only proved it correct, not tried it.*

Proof animation

Knuth 1977:

> *Beware of bugs in the above code; I have only proved it correct, not tried it.*

What about?

> *Beware of bugs in the above proof; I have only followed inference rules, not run it.*

Proof animation

Knuth 1977:

> *Beware of bugs in the above code; I have only proved it correct, not tried it.*

What about?

> *Beware of bugs in the above proof; I have only followed inference rules, not run it.*

Hayashi: *proof animation*.

Proof animation

Knuth 1977:

> *Beware of bugs in the above code; I have only proved it correct, not tried it.*

What about?

> *Beware of bugs in the above proof; I have only followed inference rules, not run it.*

Hayashi: *proof animation*.
This is where the buck stops.

Proof animation

Knuth 1977:

> *Beware of bugs in the above code; I have only proved it correct, not tried it.*

What about?

> *Beware of bugs in the above proof; I have only followed inference rules, not run it.*

Hayashi: *proof animation*.
This is where the buck stops.
Judgments can be refuted and corroborated by running tests (cf Popper). The validity of a judgment should be *local*; it should not refer to the formal system of which it is a part.

Testing categorical judgments $a : A$

Compute the canonical form of A and a!

- If $A \Rightarrow \mathrm{N}$, then
  - if $a \Rightarrow 0$, then the test is successful.
  - if $a \Rightarrow \mathrm{s}(b)$, then test whether $b : \mathrm{N}$.
  - if $a \Rightarrow \mathrm{c}(b_1, \ldots, b_n)$ for some other constructor c (including λ), then the test fails.
- If $A \Rightarrow \mathrm{U}$
  - if $a \Rightarrow \mathrm{N}$, then the test is successful.
  - if $a \Rightarrow \Pi x : B.C$, then test whether $B : \mathrm{U}$ and $x : B \vdash C : \mathrm{U}$.

    $\vdots$
  - if $a \Rightarrow \mathrm{c}(b_1, \ldots, b_n)$ for some c which is not a constructor for small sets, then the test fails.

Testing hypothetical judgments

To test

$$x_1 : A_1, \ldots, x_n : A_n \vdash a : A$$

we need to *generate* arbitrary elements $a_1 : A_1, \ldots, a_n : A_n$ and test the categorical judgment

$$a[a_1, \ldots, a_n / x_1, \ldots, x_n] : A[a_1, \ldots, a_n / x_1, \ldots, x_n]$$

How to do this?

- No problem if $A_i \Rightarrow \mathrm{N}$. Generate $x_i := 0, \mathrm{s}(0), \ldots$.
- No problem if $A_i \Rightarrow \mathrm{I}(\mathrm{N}, m, n)$ for closed m, n. Generate r if $m = n : \mathrm{N}$.
- But what if $A_i \Rightarrow \mathrm{N} \rightarrow \mathrm{N}$?
- And what if $A_i \Rightarrow \mathrm{I}(\mathrm{N} \rightarrow \mathrm{N}, f, g)$?

The meaning of higher order functions

- We cannot generate arbitrary input $f : \mathrm{N} \to \mathrm{N}$; higher order functions have been claimed to be *impredicative*!

The meaning of higher order functions

- We cannot generate arbitrary input $f : \mathrm{N} \to \mathrm{N}$; higher order functions have been claimed to be *impredicative*!
- To test $f : \mathrm{N} \to \mathrm{N} \vdash a : N$ we make use of continuity: f will only call finitely many arguments.

The meaning of higher order functions

- We cannot generate arbitrary input $f : \mathrm{N} \to \mathrm{N}$; higher order functions have been claimed to be *impredicative*!
- To test $f : \mathrm{N} \to \mathrm{N} \vdash a : N$ we make use of continuity: f will only call finitely many arguments.
- Use *domain theory*: generate *neighborhoods/finite elements* of f?

The meaning of higher order functions

- We cannot generate arbitrary input $f : \mathrm{N} \to \mathrm{N}$; higher order functions have been claimed to be *impredicative*!
- To test $f : \mathrm{N} \to \mathrm{N} \vdash a : N$ we make use of continuity: f will only call finitely many arguments.
- Use *domain theory*: generate *neighborhoods/finite elements* of f?
- Use *game semantics*: lazily generate opponent *strategies* for f, e g, following Hyland and Ong's *arena games* with *innocent strategies*.

The meaning of higher order functions

- We cannot generate arbitrary input $f : \mathrm{N} \to \mathrm{N}$; higher order functions have been claimed to be *impredicative*!
- To test $f : \mathrm{N} \to \mathrm{N} \vdash a : N$ we make use of continuity: f will only call finitely many arguments.
- Use *domain theory*: generate *neighborhoods/finite elements* of f?
- Use *game semantics*: lazily generate opponent *strategies* for f, e g, following Hyland and Ong's *arena games* with *innocent strategies*.
- A *refinement* of Martin-Löf's meaning explanations, where input generation plays a dual role to output computation. Technically, game semantics rather than realizability semantics.

Generating elements of identity types

Define meaning of identity type by induction on the type structure:

$$
\begin{array}{rcl}
\mathrm{I}(\mathrm{N}, m, n) & = & \mathrm{T}(m =_{\mathrm{N}} n) \; - \textit{compute recursive function} \\
\mathrm{I}(\Pi x : A.B, f, g) & = & \Pi x : A.\mathrm{I}(B, f(x), g(x)) \\
& \vdots &
\end{array}
$$

We can then define $\mathrm{r}_N, \mathrm{r}_{\Pi x:A.B}$, etc and validate the rules of I-elimination and I-equality in extensional type theory:

$$
\frac{c : \mathrm{I}(\mathrm{N}, m, n)}{m = n : \mathrm{N}} \qquad \frac{c : \mathrm{I}(\mathrm{N}, m, n)}{c = \mathrm{r}_{\mathrm{N}} : \mathrm{I}(\mathrm{N}, m, n)}
$$

$$
\frac{c : \mathrm{I}(\Pi x : A.B, f, g)}{f = g : \Pi x : A.B} \qquad \frac{c : \mathrm{I}(\Pi x : A.B, f, g)}{c = \mathrm{r}_{\Pi x:A.B} : \mathrm{I}(\Pi x : A.B, f, g)}
$$

Identity on universe?

What is $\mathrm{I}(\mathrm{U}, A, B)$? Freedom of choice.

Identity on universe?

What is $\mathrm{I}(\mathrm{U}, A, B)$? Freedom of choice.

Extensional equality of codes. The elements of $\mathrm{I}(\mathrm{U}, A, B)$ are generated in the following way:

$$\mathrm{r}_{\mathrm{UN}} : \mathrm{I}(\mathrm{U}, \mathrm{N}, \mathrm{N}) \qquad \frac{c : \mathrm{I}(\mathrm{U}, A, A') \quad x : A \vdash d : \mathrm{I}(\mathrm{U}, B, B')}{\mathrm{r}_{\mathrm{U}\Pi}(c, d) : \mathrm{I}(\mathrm{U}, \Pi x : A.B, \Pi x : A'.B')} \qquad \ldots$$

Validates I-elimination and I-equality of extensional type theory for U.

Identity on universe?

What is $\mathrm{I}(\mathrm{U}, A, B)$? Freedom of choice.

Extensional equality of codes. The elements of $\mathrm{I}(\mathrm{U}, A, B)$ are generated in the following way:

$$\mathrm{r}_{\mathrm{UN}} : \mathrm{I}(\mathrm{U}, \mathrm{N}, \mathrm{N}) \qquad \frac{c : \mathrm{I}(\mathrm{U}, A, A') \quad x : A \vdash d : \mathrm{I}(\mathrm{U}, B, B')}{\mathrm{r}_{\mathrm{U}\Pi}(c, d) : \mathrm{I}(\mathrm{U}, \Pi x : A.B, \Pi x : A'.B')} \qquad \cdots$$

Validates I-elimination and I-equality of extensional type theory for U.

Isomorphism. Let

$$\mathrm{I}(\mathrm{U}, A, B) = A \cong_{\mathrm{U}} B$$

Validates univalence axiom for U, but not I-elimination (J) of intensional type theory for this type.

Groupoid model?

Groupoid model of intensional type theory with univalence for U in extensional type theory (conjecture)? Harper and Licata 2012, *Canonicity for 2-Dimensional Type Theory*

> *An alternative to our current work would be to parallel this approach, and define a groupoid interpretation into extensional type theory, and thereby inherit equality from the meta-language. The benefit of the approach we take here is that it provides a more direct description of the equational theory, presenting it directly in terms of the source language.*

Finite System T

Types

$$A \quad ::= \quad \mathrm{Bool} \mid A \to A$$

Terms

$$a \quad ::= \quad x \mid a\,a \mid \lambda x.a \mid \mathrm{tt} \mid \mathrm{ff} \mid \mathrm{if}\,a\,a\,a$$

Innocent head normal forms of type $A_1 \to \cdots \to A_m \to \mathrm{Bool}$

$$\lambda x_1 \cdots x_m.\mathrm{tt}$$
$$\lambda x_1 \cdots x_m.\mathrm{ff}$$
$$\lambda x_1 \cdots x_m.\mathrm{if}\,(x\,a_1\,\ldots\,a_n)\,a\,a'$$

generate one level of Curien's PCF Böhm trees.

Test of $a : \mathrm{Bool}$

Run the closed term a!

- $a \Rightarrow \mathrm{tt}$: the test succeeds.
- $a \Rightarrow \mathrm{ff}$: the test succeeds.
- $a \Rightarrow \lambda x.b$: the test fails.

Test of $a : \mathrm{Bool}$

Run the closed term a!

- $a \Rightarrow \mathrm{tt}$: the test succeeds.
- $a \Rightarrow \mathrm{ff}$: the test succeeds.
- $a \Rightarrow \lambda x.b$: the test fails.

The arena for Bool:

$$\begin{tikzcd}
 & a \Rightarrow ? \arrow[dl, dash, bend right] \arrow[dr, dash, bend left] & \\
a \Rightarrow \mathrm{tt} & & a \Rightarrow \mathrm{ff}
\end{tikzcd}$$

Test of $a : \mathrm{Bool} \longrightarrow \mathrm{Bool}$

Test $x : \mathrm{Bool} \vdash a\,x : \mathrm{Bool}$!

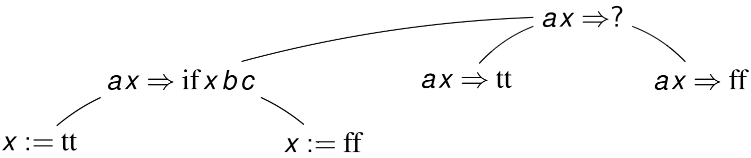
- $a\,x \Rightarrow \mathrm{tt}$: the test succeeds.
- $a\,x \Rightarrow \mathrm{ff}$: the test succeeds.
- $a\,x \Rightarrow \mathrm{if}\, x\, b\, c$: generate head variable
  - either $x := \mathrm{tt}$, and test $x : \mathrm{Bool} \vdash b : \mathrm{Bool}$ or
  - or $x := \mathrm{ff}$, and test $x : \mathrm{Bool} \vdash c : \mathrm{Bool}$
- otherwise, the test fails

Test of $a : \mathrm{Bool} \longrightarrow \mathrm{Bool}$

Test $x : \mathrm{Bool} \vdash a\,x : \mathrm{Bool}$!

- $a\,x \Rightarrow \mathrm{tt}$: the test succeeds.
- $a\,x \Rightarrow \mathrm{ff}$: the test succeeds.
- $a\,x \Rightarrow \mathrm{if}\,x\,b\,c$: generate head variable
  - either $x := \mathrm{tt}$, and test $x : \mathrm{Bool} \vdash b : \mathrm{Bool}$ or
  - or $x := \mathrm{ff}$, and test $x : \mathrm{Bool} \vdash c : \mathrm{Bool}$
- otherwise, the test fails

Arena for $\mathrm{Bool} \longrightarrow \mathrm{Bool}$:

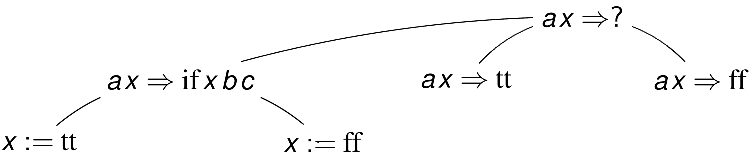


Test of $a : \mathrm{Bool} \longrightarrow \mathrm{Bool}$

Test $x : \mathrm{Bool} \vdash a\,x : \mathrm{Bool}$!

- $a\,x \Rightarrow \mathrm{tt}$: the test succeeds.
- $a\,x \Rightarrow \mathrm{ff}$: the test succeeds.
- $a\,x \Rightarrow \mathrm{if}\, x\, b\, c$: generate head variable
  - either $x := \mathrm{tt}$, and test $x : \mathrm{Bool} \vdash b : \mathrm{Bool}$ or
  - or $x := \mathrm{ff}$, and test $x : \mathrm{Bool} \vdash c : \mathrm{Bool}$
- otherwise, the test fails

Arena for $\mathrm{Bool} \longrightarrow \mathrm{Bool}$:



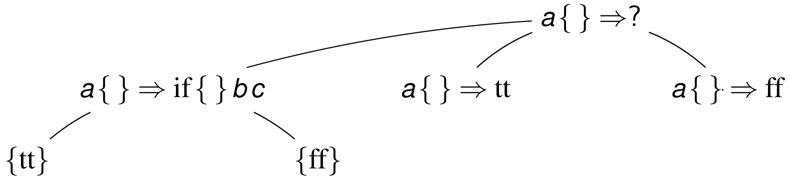
Correspondence with game semantics. Move in arbitrary innocent well-bracketed opponent strategy $x : \mathrm{Bool}$. Only head occurrence of x is instantiated at the first stage. Repetition of moves possible.

Test of $a : \mathrm{Bool} \to \mathrm{Bool}$

Test $\{\,\} : \mathrm{Bool} \vdash a\{\,\} : \mathrm{Bool}$!

- $a\{\,\} \Rightarrow \mathrm{tt}$: the test succeeds.
- $a\{\,\} \Rightarrow \mathrm{ff}$: the test succeeds.
- $a\{\,\} \Rightarrow \mathrm{if}\{\,\}\,b\,c$: generate head variable
  - either $\{\,\} := \mathrm{tt}$, and test $\{\,\} : \mathrm{Bool} \vdash b : \mathrm{Bool}$ or
  - or $\{\,\} := \mathrm{ff}$, and test $\{\,\} : \mathrm{Bool} \vdash c : \mathrm{Bool}$
- otherwise, the test fails.

Arena for $\mathrm{Bool} \to \mathrm{Bool}$:



Correspondence with game semantics. Move in arbitrary innocent well-bracketed opponent strategy $\{\,\} : \mathrm{Bool}$. Only head occurrence of $\{\,\}$ is instantiated at the first stage. Repetition of moves possible.

Test of $a : (\mathrm{Bool} \to \mathrm{Bool}) \to \mathrm{Bool}$

Test $\{\ \} : \mathrm{Bool} \to \mathrm{Bool} \vdash a\{\ \} : \mathrm{Bool}$.

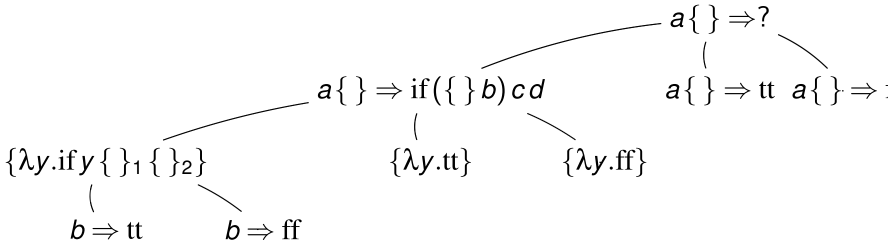
- $a\{\ \} \Rightarrow \mathrm{tt}$: the test succeeds.
- $a\{\ \} \Rightarrow \mathrm{ff}$: the test succeeds.
- $a\{\ \} \Rightarrow \mathrm{if}\,(\{\ \}\,b)\,c\,d$. See below.
- otherwise, the test fails.

Test of $a : (\mathrm{Bool} \to \mathrm{Bool}) \to \mathrm{Bool}$

Test $\{\,\} : \mathrm{Bool} \to \mathrm{Bool} \vdash a\{\,\} : \mathrm{Bool}$.

- $a\{\,\} \Rightarrow \mathrm{tt}$: the test succeeds.
- $a\{\,\} \Rightarrow \mathrm{ff}$: the test succeeds.
- $a\{\,\} \Rightarrow \mathrm{if}(\{\,\}\,b)\,c\,d$. See below.
- otherwise, the test fails.

Arena for $(\mathrm{Bool} \to \mathrm{Bool}) \to \mathrm{Bool}$:



Playing $x\ (x\ \mathrm{tt})$ versus $\lambda y.\mathrm{if}\ y\ \mathrm{ff}\ \mathrm{tt}$

$$(\mathrm{Bool} \quad \to \quad \mathrm{Bool}) \quad \to \quad \mathrm{Bool}$$

$x\ (x\ \mathrm{tt}) \Rightarrow ?$

$x\ (x\ \mathrm{tt}) \Rightarrow \underline{x}\ (x\ \mathrm{tt})$

$x\ (x\ \mathrm{tt}) := ?$

$x\ \mathrm{tt} \Rightarrow \underline{x}\ \mathrm{tt}$

$x\ \mathrm{tt} := ?$

$\mathrm{tt} \Rightarrow \mathrm{tt}$

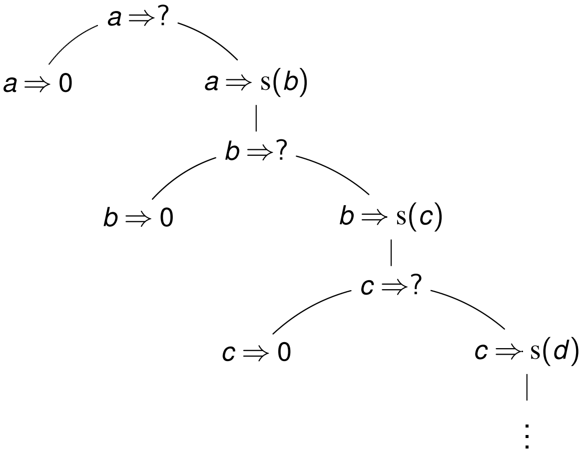
$x\ \mathrm{tt} := \mathrm{ff}$

$x\ \mathrm{tt} \Rightarrow \mathrm{ff}$

$x\ (x\ \mathrm{tt}) := \mathrm{tt}$

$x\ (x\ \mathrm{tt}) \Rightarrow \mathrm{tt}$

The arena for lazy N



Terms of intuitionistic type theory ($\mathrm{Bool}, \mathrm{N}, \Sigma, \Pi, \mathrm{U}$-fragment)

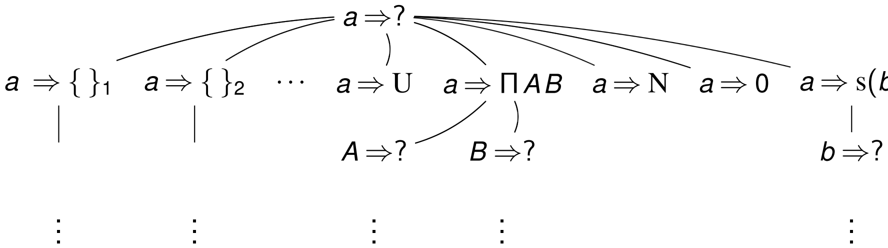
$$
\begin{array}{rcl}
a & ::= & x \mid a\,a \mid \lambda x.a \\
& & \mid \mathrm{tt} \mid \mathrm{ff} \mid 0 \mid \mathrm{s}\,a \mid (a, a) \\
& & \mid \mathrm{Bool} \mid \mathrm{N} \mid \Sigma\,a\,a \mid \Pi\,a\,a \mid \mathrm{U} \\
& & \mid \mathrm{if}\,a\,a\,a \mid \mathrm{natrec}\,a\,a\,a \mid \mathrm{fst}\,a \mid \mathrm{snd}\,a \mid \mathrm{urec}\,a\,a\,a\,a\,a
\end{array}
$$

Russell-style universe with universe recursion

Testing judgments as game playing

- computation to weak head normal form (player move)
- instantiation of variables (opponent move)
- matching constructors of terms with constructors of types (checking rules of the game)

The arena for Intuitionistic type theory



Transition system for testing

Configurations

$$\langle \mathcal{J}, \tau \rangle$$

where

- $\mathcal{J}$ is a judgment
- Formally τ is a *channel environment* - a partial association between channels (metavariables) and *atomic normal forms*
- a *channel* has a type and a context $\Gamma \vdash c : A$
- an *atomic normal form* records *one* move qua one level of unfolding of a Böhm tree corresponding to an opponent strategy.

Initial state of transition system

If $\mathcal{J}$ is a categorical (closed) judgment, then the initial state is

$$\langle \mathcal{J}, \emptyset \rangle$$

If $\mathcal{J}$ is a hypothetical judgment, then the initial channel environment τ_Γ is obtained by replacing all variables in the context Γ by channels of appropriate types:

$$\langle \mathcal{J}, \tau_\Gamma \rangle$$

Final states for test of typing judgment

Final states correspond to introduction rules without premises ("axioms")

$$\langle \mathrm{tt} : \mathrm{Bool}, \tau \rangle$$
$$\langle \mathrm{ff} : \mathrm{Bool}, \tau \rangle$$
$$\langle 0 : \mathrm{N}, \tau \rangle$$
$$\langle \mathrm{Bool} : \mathrm{U}, \tau \rangle$$
$$\langle \mathrm{N} : \mathrm{U}, \tau \rangle$$

Transitions corresponding to introduction rules

Transitions corresponding to introduction rules with premises

$$
\begin{array}{rcl}
\langle \mathrm{s}(a) : \mathrm{N}, \tau \rangle & \longrightarrow & \langle a : \mathrm{N}, \tau \rangle \\
\langle (a, b) : \Sigma A B, \tau \rangle & \longrightarrow & \langle a : A, \tau \rangle \\
\langle (a, b) : \Sigma A B, \tau \rangle & \longrightarrow & \langle b : B a, \tau \rangle \\
\langle \lambda x.b : \Pi A B, \tau \rangle & \longrightarrow & \langle b[x = c] : B c, \tau \cup \{\vdash c : A\} \rangle \\
\langle \Sigma A B : \mathrm{U}, \tau \rangle & \longrightarrow & \langle A : \mathrm{U}, \tau \rangle \\
\langle \Sigma A B : \mathrm{U}, \tau \rangle & \longrightarrow & \langle B c : \mathrm{U}, \tau \cup \{\vdash c : A\} \rangle \\
\langle \Pi A B : \mathrm{U}, \tau \rangle & \longrightarrow & \langle A : \mathrm{U}, \tau \rangle \\
\langle \Pi A B : \mathrm{U}, \tau \rangle & \longrightarrow & \langle B c : \mathrm{U}, \tau \cup \{\vdash c : A\} \rangle
\end{array}
$$

Transition from "generative evaluation"

Generative evaluation $\langle a, \tau_1 \rangle \Longrightarrow \langle v, \tau_2 \rangle$ combines computation of an expression a to canonical form v with instantiation of channels by extending the channel environment τ_1 to τ_2. We have the rule

$$\frac{\langle A, \tau_1 \rangle \Longrightarrow \langle V, \tau_2 \rangle \qquad \langle a, \tau_2 \rangle \Longrightarrow \langle v, \tau_2 \rangle}{\langle a : A, \tau_1 \rangle \longrightarrow \langle v : V, \tau_3 \rangle}$$

- v is a canonical term former: outermost form is term constructor
- V is a canonical type former: outermost form is type constructor

Pure computation to canonical form

Pure computation to canonical form

$$\frac{a \Rightarrow v}{\langle a, \tau \rangle \Longrightarrow \langle v, \tau \rangle}$$

Channel instantiation and generation in head contexts

Head contexts

$$HC[\,] ::= [\,] \mid [\,]\,a \mid \mathrm{if}\,[\,]\,a\,a \mid \mathrm{natrec}\,[\,]\,a\,a \mid \mathrm{urec}\,[\,]\,a\,a\,a\,a$$

Using old instantiation $c = a' : \tau$

$$\frac{a \Rightarrow HC[c] \qquad \langle HC[a'], \tau_1\rangle \Longrightarrow \langle v, \tau_2\rangle}{\langle a, \tau_1\rangle \Longrightarrow \langle v, \tau_2\rangle}$$

Creating new instantiation $c : A : \tau$ and $a' : ANF(\Gamma, V)$

$$\frac{a \Rightarrow HC[c] \qquad \langle A, \tau_1\rangle \Longrightarrow \langle V, \tau_2\rangle \qquad \langle HC[a'], \tau_2\rangle \Longrightarrow \langle v, \tau_3\rangle}{\langle a, \tau_1\rangle \Longrightarrow \langle v, \tau_3\rangle}$$

Atomic normal forms (opponent's moves)

- $\mathrm{tt}, \mathrm{ff} : \mathit{ANF}(\Gamma, \mathrm{Bool})$.
- $0 : \mathit{ANF}(\Gamma, \mathrm{N})$, $\mathrm{s}(c) : \mathit{ANF}(\Gamma, \mathrm{N})$, where $\Gamma \vdash c : \mathrm{N}$ is a new channel.
- $(c, d) : \mathit{ANF}(\Gamma, \Sigma\, A\, B)$ where $\Gamma \vdash c : A$ and $\Gamma \vdash d : B\, c$ are new channels.
- $\lambda x.c : \mathit{ANF}(\Gamma, \Pi\, A\, B)$, where $\Gamma, x : A \vdash c : B\, x$ is a new channel.
- $\mathrm{Bool}, \mathrm{N} : \mathit{ANF}(\Gamma, \mathrm{U})$, $\Sigma\, c\, d, \Pi\, c\, d : \mathit{ANF}(\Gamma, \mathrm{U})$ where $c : \Gamma \vdash \mathrm{U}$ and $\Gamma \vdash d\, c : \mathrm{U}$ are new channels.

Atomic normal forms (neutral case)

- $C_{\Gamma \vdash B}[x_i : A_i] : ANF(\Gamma, B)$, where $x_i : A_i : \Gamma$.

We use a new auxiliary atomic normal form $C_{\Gamma \vdash B}[a : A]$, which will expand to "neutral" atomic normal forms, when the the canonical form of the type A_i is determined.

Expansion of neutral terms for $\mathrm{Bool}, \mathrm{N}$, and U

$C_{\Gamma \vdash B}[a : A]$ is an atomic normal form which depends on A and a.

$$\frac{\langle A, \tau_1 \rangle \Rightarrow \langle \mathrm{Bool}, \tau_2 \rangle \quad \langle \mathrm{if}\, a\, c\, d, \tau_2 \cup \{\Gamma \vdash c : B, \Gamma \vdash d : B\} \rangle \Longrightarrow \langle v, \tau_3 \rangle}{\langle C_{\Gamma \vdash B}[a : A], \tau_1 \rangle \Longrightarrow \langle v, \tau_3 \rangle}$$

$$\frac{\langle A, \tau_1 \rangle \Rightarrow \langle \mathrm{N}, \tau_2 \rangle \quad \langle \mathrm{natcase}\, a\, c\, d, \tau_2 \cup \{\Gamma \vdash c : B, \Gamma \vdash d : \mathrm{N} \to B\} \rangle \Longrightarrow \langle v, \tau_3 \rangle}{\langle C_{\Gamma \vdash B}[a : A], \tau_1 \rangle \Longrightarrow \langle v, \tau_3 \rangle}$$

$$\frac{\langle A, \tau_1 \rangle \Rightarrow \langle \mathrm{U}, \tau_2 \rangle \quad \langle \mathrm{ucase}\, a\, c\, d\, e\, f, \tau_2 \cup \cdots \rangle \Longrightarrow \langle v, \tau_3 \rangle}{\langle C_{\Gamma \vdash B}[a : A], \tau_1 \rangle \Longrightarrow \langle v, \tau_3 \rangle}$$

Expansion of neutral terms for Σ and Π

Σ:

$$\frac{\langle A, \tau_1 \rangle \Rightarrow \langle \Sigma\, D\, E, \tau_2 \rangle \quad \langle C_{\Gamma \vdash B}[\mathrm{fst}\, a : D], \tau_2 \rangle \Longrightarrow \langle v, \tau_3 \rangle}{\langle C_{\Gamma \vdash B}[a : A], \tau_1 \rangle \Longrightarrow \langle v, \tau_3 \rangle}$$

$$\frac{\langle A, \tau_1 \rangle \Rightarrow \langle \Sigma\, D\, E, \tau_2 \rangle \quad \langle C_{\Gamma \vdash B}[\mathrm{snd}\, a : D\,(\mathrm{fst}\, a)], \tau_2 \rangle \Longrightarrow \langle v, \tau_3 \rangle}{\langle C_{\Gamma \vdash B}[a : A], \tau_1 \rangle \Longrightarrow \langle v, \tau_3 \rangle}$$

Π:

$$\frac{\langle A, \tau_1 \rangle \Rightarrow \langle \Pi\, D\, E, \tau_2 \rangle \quad \langle C_{\Gamma \vdash B}[a\, c : E\, c], \tau_2 \cup \{c : \Gamma \vdash D\} \rangle \Longrightarrow \langle v, \tau_3 \rangle}{\langle C_{\Gamma \vdash B}[a : A], \tau_1 \rangle \Longrightarrow \langle v, \tau_3 \rangle}$$

Summary

- The test manual determines the meaning of judgments, including equality judgments.
- Tests can corroborate or refute judgments.
- Tests with functional input leads us to games: input generation corresponds to playing opponent strategy.
- New interpretation of hypothetical judgments, type equality, and identity types.
- Rules of extensional type theory of Martin-Löf 1979 can be justified – work in progress to construct a game model of extensional type theory based on the test manual.