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

Meeting on
Logic and Interactions
CIRM, Marseille
30 January - 2 March 2012

Peter Dybjer

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

13 February 2012

The meaning and justification of intuitionistic logic and type theory - evolution of ideas during the 20th century

Intuitionistic logic:

The meaning and justification of intuitionistic logic and type theory - evolution of ideas during the 20th century

Intuitionistic logic:

1908 BHK. A calculus of problems (Kolmogorov) or a calculus of intended constructions (Heyting)

The meaning and justification of intuitionistic logic and type theory - evolution of ideas during the 20th century

Intuitionistic logic:

1908 BHK. A calculus of problems (Kolmogorov) or a calculus of intended constructions (Heyting)

1945 Kleene. Realizability

The meaning and justification of intuitionistic logic and type theory - evolution of ideas during the 20th century

Intuitionistic logic:

1908 BHK. A calculus of problems (Kolmogorov) or a calculus of intended constructions (Heyting)

1945 Kleene. Realizability

1969 Curry-Howard. Formulas as types

The meaning and justification of intuitionistic logic and type theory - evolution of ideas during the 20th century

Intuitionistic logic:

1908 BHK. A calculus of problems (Kolmogorov) or a calculus of intended constructions (Heyting)

1945 Kleene. Realizability

1969 Curry-Howard. Formulas as types

Intuitionistic type theory:

The meaning and justification of intuitionistic logic and type theory - evolution of ideas during the 20th century

Intuitionistic logic:

1908 BHK. A calculus of problems (Kolmogorov) or a calculus of intended constructions (Heyting)

1945 Kleene. Realizability

1969 Curry-Howard. Formulas as types

Intuitionistic type theory:

1972 Martin-Löf. Intuitionistic type theory, proof theoretic properties

The meaning and justification of intuitionistic logic and type theory - evolution of ideas during the 20th century

Intuitionistic logic:

1908 BHK. A calculus of problems (Kolmogorov) or a calculus of intended constructions (Heyting)

1945 Kleene. Realizability

1969 Curry-Howard. Formulas as types

Intuitionistic type theory:

1972 Martin-Löf. Intuitionistic type theory, proof theoretic properties

1974 Aczel. Realizability

The meaning and justification of intuitionistic logic and type theory - evolution of ideas during the 20th century

Intuitionistic logic:

1908 BHK. A calculus of problems (Kolmogorov) or a calculus of intended constructions (Heyting)

1945 Kleene. Realizability

1969 Curry-Howard. Formulas as types

Intuitionistic type theory:

1972 Martin-Löf. Intuitionistic type theory, proof theoretic properties

1974 Aczel. Realizability

1979 Martin-Löf. Meaning explanations

The meaning and justification of intuitionistic logic and type theory - evolution of ideas during the 20th century

Intuitionistic logic:

1908 BHK. A calculus of problems (Kolmogorov) or a calculus of intended constructions (Heyting)

1945 Kleene. Realizability

1969 Curry-Howard. Formulas as types

Intuitionistic type theory:

1972 Martin-Löf. Intuitionistic type theory, proof theoretic properties

1974 Aczel. Realizability

1979 Martin-Löf. Meaning explanations

2009 Prawitz on Martin-Löf. Proof: epistemological or ontological concept?

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:

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

Other mathematical objects

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

Type formers of intuitionistic type theory

To interpret predicate logic with identity:

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

Other mathematical objects

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

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

Type formers of intuitionistic type theory

To interpret predicate logic with identity:

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

Other mathematical objects

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

- 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.

Judgements of Intuitionistic Type Theory

$$\Gamma \vdash A \; type$$

$$\Gamma \vdash A = A'$$

$$\Gamma \vdash a \in A$$

$$\Gamma \vdash a = a' \in A$$

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 - meaning explanations

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

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

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

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

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

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

General pattern

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

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

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

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

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

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).

The meaning of hypothetical judgements (Martin-Löf 1979)

$$a \in A \ \ (x_1 \in A_1, \ldots, x_n \in A_n)$$

means that

The meaning of hypothetical judgements (Martin-Löf 1979)

$$a \in A \ \ (x_1 \in A_1, \ldots, x_n \in A_n)$$

means that

$$a(a_1, \ldots, a_n / x_1, \ldots, x_n) \in A(a_1, \ldots, a_n / x_1, \ldots, x_n)$$

provided

$$a_1 \in A_1,$$
$$\vdots$$
$$a_n \in A_n(a_1, \ldots, a_{n-1} / x_1, \ldots, x_{n-1}),$$

and, moreover,

$$a(a_1, \ldots, a_n / x_1, \ldots, x_n) = a(b_1, \ldots, b_n / x_1, \ldots, x_n) \in A(a_1, \ldots, a_n / x_1, \ldots, x_n)$$

provided

$$a_1 = b_1 \in A_1,$$
$$\vdots$$

Martin-Löf on meaning explanations

Martin-Löf on meaning explanations

Some people feel that when I have presented my meaning explanations I have said nothing.

Martin-Löf on meaning explanations

Some people feel that when I have presented my meaning explanations I have said nothing.

And this is how it should be.

Martin-Löf on meaning explanations

Some people feel that when I have presented my meaning explanations I have said nothing.

And this is how it should be.

The standard semantics should be just that: standard. It should come as no surprise.

Martin-Löf on meaning explanations

> *Some people feel that when I have presented my meaning explanations I have said nothing.*
>
> *And this is how it should be.*
>
> *The standard semantics should be just that: standard. It should come as no surprise.*

Martin-Löf at the Types Summer School in Giens 2002 (as I recall it)

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.
- Idealist vs realist; subjective vs objective; epistemic vs ontological?

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.
- Idealist vs realist; subjective vs objective; epistemic vs ontological? Is logic better off with or without reality?

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.
- Idealist vs realist; subjective vs objective; epistemic vs ontological? Is logic better off with or without reality?
- Testing, QuickCheck (Claessen and Hughes 2000), input generation. Test manual for Martin-Löf type theory. Cf Hayashi's proof animation for PX from the1980s.

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.
- Idealist vs realist; subjective vs objective; epistemic vs ontological? Is logic better off with or without reality?
- Testing, QuickCheck (Claessen and Hughes 2000), input generation. Test manual for Martin-Löf type theory. Cf Hayashi's proof animation for PX from the1980s.
- Meaning (testing) of hypothetical judgements, type equality, identity types

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.
- Idealist vs realist; subjective vs objective; epistemic vs ontological? Is logic better off with or without reality?
- Testing, QuickCheck (Claessen and Hughes 2000), input generation. Test manual for Martin-Löf type theory. Cf Hayashi's proof animation for PX from the1980s.
- Meaning (testing) of hypothetical judgements, type equality, identity types
- Meaning (testing) of functionals. Continuity, domains, games.

Is mathematics an experimental science?

Miquel 2010: The experimental effectiveness of mathematical proof:

> *We can thus argue (against Popper) that mathematics fulfills the demarcation criterion that makes mathematics an empirical science. The only specificity of mathematics is that the universal empirical hypothesis underlying mathematics is (almost) never stated explicitly.*

Truth and proof: ontological or epistemic concepts?

Prawitz 2011:

> *"More precisely, Martin-Löf makes a distinction between two senses of proofs."*

ontological: "proof object" a in $a \in A$

epistemic: "demonstration" - a tree of inferences:

$$\frac{\vdots}{\Gamma \vdash a \in A}$$

How to test categorical judgements $a \in A$?

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

    $\vdots$
  - 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.

Weak head normal form vs head normal form

Weak head normal form If $A \Rightarrow \Pi x \in B.C$, then

- if $a \Rightarrow \lambda x.c$, then test $x \in B \vdash c \in C$
- if $a \Rightarrow \mathrm{c}(b_1, \ldots, b_n)$ for some other constructor c, then the test fails.

Head normal form If $A \Rightarrow \Pi x \in B.C$, then

- test $x \in B \vdash a\,x \in C$

How to generate input?

- Easy to generate input of type N (and other algebraic data types).
- But how do we generate input of type $\mathrm{N} \to \mathrm{N}$ (and other higher types)?
  - We do not know how to generate an arbitrary $x \in \mathrm{N} \to \mathrm{N}$!
  - Key observation. Continuity tells us that x will only call finitely many arguments. Domain theory and game semantics to the rescue!

Testing functionals: an example

How to test

$$f \in \mathrm{N} \to \mathrm{N} \vdash \textit{if } f\,0 = 0 \textit{ then } f\,1 \textit{ else } f\,2 \in \mathrm{Fin}\,(\mathrm{s}(f\,1 + f\,2))$$

where

$$\begin{aligned} \mathrm{Fin}\,0 &= \mathrm{N}_0 \\ \mathrm{Fin}\,(\mathrm{s}(n)) &= \mathrm{N}_1 + \mathrm{Fin}\,n \end{aligned}$$

$$\begin{aligned} 0 + n &= n \\ \mathrm{s}(m) + n &= \mathrm{s}(m + n) \end{aligned}$$

Assume suitable encoding of numbers and

$$\mathrm{N}_0 = \mathrm{Fin}\,0 \subseteq \mathrm{Fin}\,1 \subseteq \mathrm{Fin}\,2 \subseteq \cdots \subseteq \mathrm{N}$$

Game-theoretic testing

Test $f : \mathrm{N} \to \mathrm{N} \vdash$ *if* $f\,0 = 0$ *then* $f\,1$ *else* $f\,2 \in \mathrm{Fin}(\mathrm{s}(f\,1 + f\,2))$!

Game-theoretic testing

Test $f : \mathrm{N} \rightarrow \mathrm{N} \vdash$ *if* $f\,0 = 0$ *then* $f\,1$ *else* $f\,2 \in \mathrm{Fin}(\mathrm{s}(f\,1 + f\,2))$!

- Evaluate first type and then term to hnf:

  $$f : \mathrm{N} \rightarrow \mathrm{N} \vdash \textit{ if } f\,0 = 0 \textit{ then } f\,1 \textit{ else } f\,2 \in 1 + \mathrm{Fin}(f\,1 + f\,2)$$

  We are not ready to match term constructor and type constructor because outermost constructor of term is not known: need value of $f\,0$.

Game-theoretic testing

Test $f : \mathrm{N} \rightarrow \mathrm{N} \vdash$ *if* $f0 = 0$ *then* $f1$ *else* $f2 \in \mathrm{Fin}(\mathrm{s}(f1 + f2))$!

- Evaluate first type and then term to hnf:

  $$f : \mathrm{N} \rightarrow \mathrm{N} \vdash \textit{ if } f0 = 0 \textit{ then } f1 \textit{ else } f2 \in 1 + \mathrm{Fin}(f1 + f2)$$

  We are not ready to match term constructor and type constructor because outermost constructor of term is not known: need value of $f0$.

- Generate $f0 := 0$ – opponent move.

Game-theoretic testing

Test $f : \mathrm{N} \to \mathrm{N} \vdash \; \mathit{if}\ f\,0 = 0\ \mathit{then}\ f\,1\ \mathit{else}\ f\,2 \in \mathrm{Fin}\,(\mathrm{s}(f\,1 + f\,2))$!

- Evaluate first type and then term to hnf:

$$f : \mathrm{N} \to \mathrm{N} \vdash \; \mathit{if}\ f\,0 = 0\ \mathit{then}\ f\,1\ \mathit{else}\ f\,2 \in 1 + \mathrm{Fin}\,(f\,1 + f\,2)$$

  We are not ready to match term constructor and type constructor because outermost constructor of term is not known: need value of $f\,0$.
- Generate $f\,0 := 0$ – opponent move.
- Evaluate term:

$$f : \mathrm{N} \to \mathrm{N} \vdash \; f\,1 \in 1 + \mathrm{Fin}\,(f\,1 + f\,2)$$

  Outermost term constructor still not known: need value of $f\,1$.

Game-theoretic testing

Test $f : \mathrm{N} \to \mathrm{N} \vdash$ *if* $f\,0 = 0$ *then* $f\,1$ *else* $f\,2 \in \mathrm{Fin}(\mathrm{s}(f\,1 + f\,2))$!

- Evaluate first type and then term to hnf:

$$f : \mathrm{N} \to \mathrm{N} \vdash \textit{ if } f\,0 = 0 \textit{ then } f\,1 \textit{ else } f\,2 \in 1 + \mathrm{Fin}(f\,1 + f\,2)$$

  We are not ready to match term constructor and type constructor because outermost constructor of term is not known: need value of $f\,0$.

- Generate $f\,0 := 0$ – opponent move.
- Evaluate term:

$$f : \mathrm{N} \to \mathrm{N} \vdash f\,1 \in 1 + \mathrm{Fin}(f\,1 + f\,2)$$

  Outermost term constructor still not known: need value of $f\,1$.

- Generate $f\,1 := 0$ – opponent move.

Game-theoretic testing

Test $f : \mathrm{N} \to \mathrm{N} \vdash$ *if* $f\,0 = 0$ *then* $f\,1$ *else* $f\,2 \in \mathrm{Fin}(\mathrm{s}(f\,1 + f\,2))$!

- Evaluate first type and then term to hnf:

$$f : \mathrm{N} \to \mathrm{N} \vdash \textit{if } f\,0 = 0 \textit{ then } f\,1 \textit{ else } f\,2 \in 1 + \mathrm{Fin}(f\,1 + f\,2)$$

  We are not ready to match term constructor and type constructor because outermost constructor of term is not known: need value of $f\,0$.
- Generate $f\,0 := 0$ – opponent move.
- Evaluate term:

$$f : \mathrm{N} \to \mathrm{N} \vdash f\,1 \in 1 + \mathrm{Fin}(f\,1 + f\,2)$$

  Outermost term constructor still not known: need value of $f\,1$.
- Generate $f\,1 := 0$ – opponent move.
- Evaluate term:

$$f : \mathrm{N} \to \mathrm{N} \vdash 0 \in \mathrm{N}_1 + \mathrm{Fin}(f\,1 + f\,2)$$

  Test has succeeded!

Proofs, tests, and games

Testing proofs:

1980s Hayashi: proof animation for PX

Game semantics for proofs and lambda terms:

1930s Gentzen?

1950s Lorenzen: dialogue semantics

1990s Hyland and Ong; Nikau; Abramsky, Jagadeesan, and Malacaria: game semantics for PCF

Finite System T

Types

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

Terms

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

Head neutral terms

$$s \quad ::= \quad x \mid s\, a \mid \mathrm{if}\, s\, a\, a$$

Head normal forms (hnfs)

$$v \quad ::= \quad \lambda x.v \mid \mathrm{tt} \mid \mathrm{ff} \mid s$$

Head normal form relation $a \Rightarrow v$ (rules omitted)

Test of $a \in \mathrm{Bool}$

Run the closed term a!

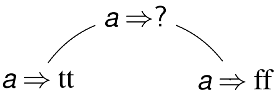
- $a \Rightarrow \mathrm{tt}$: the test succeeds.
- $a \Rightarrow \mathrm{ff}$: the test succeeds.
- $a \Rightarrow \lambda x.b$: the test fails.

Test of $a \in \mathrm{Bool}$

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:



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

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

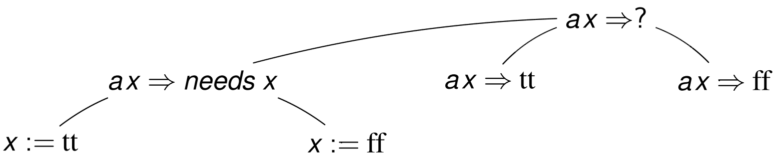
- $a\,x \Rightarrow \mathrm{tt}$: the test succeeds.
- $a\,x \Rightarrow \mathrm{ff}$: the test succeeds.
- $a\,x \Rightarrow \lambda y.b$: the test fails.
- $a\,x \Rightarrow s$ (neutral term): generate head variable $x := \mathrm{tt}$ or $x := \mathrm{ff}$.

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

Test $x \in \mathrm{Bool} \vdash ax : \mathrm{Bool}$!

- $ax \Rightarrow \mathrm{tt}$: the test succeeds.
- $ax \Rightarrow \mathrm{ff}$: the test succeeds.
- $ax \Rightarrow \lambda y.b$: the test fails.
- $ax \Rightarrow s$ (neutral term): generate head variable $x := \mathrm{tt}$ or $x := \mathrm{ff}$.

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

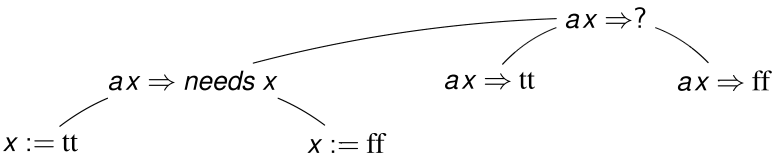


Test of $a \in \mathrm{Bool} \to \mathrm{Bool}$

Test $x \in \mathrm{Bool} \vdash ax : \mathrm{Bool}$!

- $ax \Rightarrow \mathrm{tt}$: the test succeeds.
- $ax \Rightarrow \mathrm{ff}$: the test succeeds.
- $ax \Rightarrow \lambda y.b$: the test fails.
- $ax \Rightarrow s$ (neutral term): generate head variable $x := \mathrm{tt}$ or $x := \mathrm{ff}$.

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



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

Test of $a \in (\mathrm{Bool} \rightarrow \mathrm{Bool}) \rightarrow \mathrm{Bool}$

Test $x \in \mathrm{Bool} \rightarrow \mathrm{Bool} \vdash a\,x \in \mathrm{Bool}$.

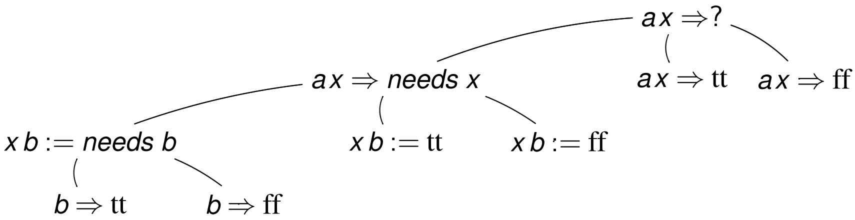
- $a\,x \Rightarrow \mathrm{tt}$: the test succeeds.
- $a\,x \Rightarrow \mathrm{ff}$: the test succeeds.
- $a\,x \Rightarrow \lambda y.b$: the test fails.
- $a\,x \Rightarrow s$ (neutral term). Evaluation is blocked by head variable x.

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

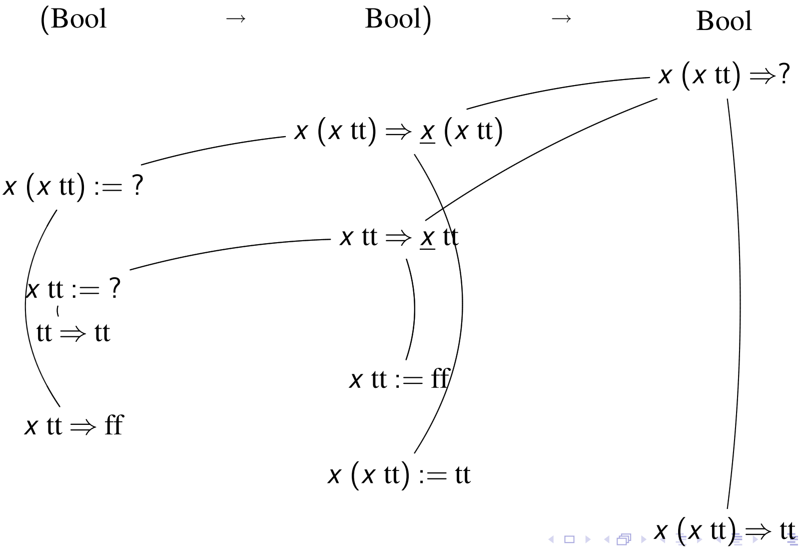
Test $x \in \mathrm{Bool} \to \mathrm{Bool} \vdash a\,x \in \mathrm{Bool}$.

- $a\,x \Rightarrow \mathrm{tt}$: the test succeeds.
- $a\,x \Rightarrow \mathrm{ff}$: the test succeeds.
- $a\,x \Rightarrow \lambda y.b$: the test fails.
- $a\,x \Rightarrow s$ (neutral term). Evaluation is blocked by head variable x.

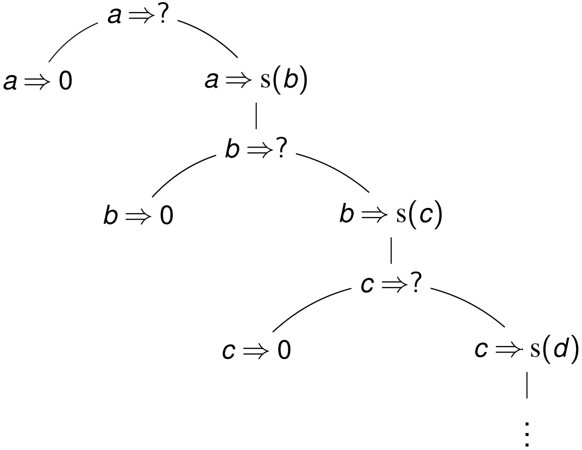
Arena for $(\mathrm{Bool} \to \mathrm{Bool}) \to \mathrm{Bool}$:



A play for $x \in \mathrm{Bool} \rightarrow \mathrm{Bool} \vdash x\ (x\ \mathrm{tt}) \in \mathrm{Bool}$



The arena for lazy N



Intuitionistic type theory (the $\mathrm{N}, \Pi, \mathrm{U}$-fragment)

Terms (including types)

$$a \quad ::= \quad x \mid a\,a \mid \lambda x.a \mid 0 \mid \mathrm{s}(a) \mid \mathrm{R}\,a\,a\,a \mid \Pi x \in a.a \mid \mathrm{N} \mid \mathrm{U}$$

Head neutral terms

$$s \quad ::= \quad x \mid s\,a \mid \mathrm{R}\,s\,a\,a$$

Head normal forms

$$v \quad ::= \quad \lambda x.v \mid 0 \mid \mathrm{s}(a) \mid \Pi x \in a.a \mid \mathrm{N} \mid \mathrm{U} \mid s$$

Test of hypothetical judgement: case N

To test

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

we compute hnf of A.

- $A \Rightarrow \mathrm{N}$. Then we compute hnf of a.
  - $a \Rightarrow 0$: success.
  - $a \Rightarrow \mathrm{s}(b)$. Test

    $$x_1 \in A_1, \ldots, x_n \in A_n \vdash b \in \mathrm{N}$$

  - $a \Rightarrow s$ (neutral term). Play strategy for head variable $x_i \in A_i$. First we must generate the outermost type constructor of A_i, that is, we must test

    $$x_1 \in A_1, \ldots, x_{i-1} \in A_{i-1} \vdash A_i \; type$$

    This computation may require playing strategies for other variables $x_j \in A_j$ for $j < i$, etc.
  - If the hnf of a is another constructor (including λ) then we fail.

Test of hypothetical judgement: case Π

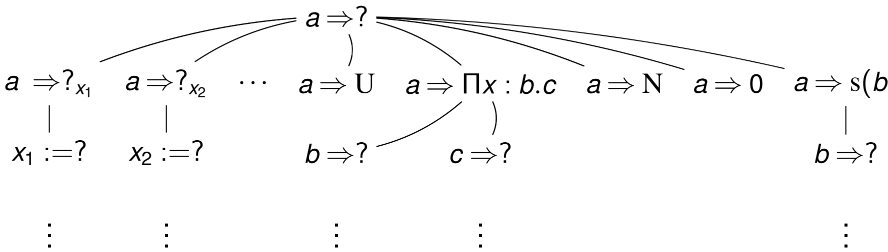
- $A \Rightarrow \Pi x \in B.C$. Test

$$x_1 \in A_1, \ldots, x_n \in A_n, x \in B \vdash a\,x \in C$$

Test of hypothetical judgement: neutral case

- $A \Rightarrow s$ (neutral term).
  - As before we play the strategy for the head variable $x_i \in A_i$ of s, remembering (by innocence) what was played before.
  - The computation of A_i can itself enforce playing strategies for other variables $x_j \in A_j$ for $j < i$. Etc.
  - When we have generated (played) enough of the variables $x_1 \in A_1, \ldots, x_n \in A_n$, then we compute hnf a, generating, if necessary more of the variables, respecting innocence.

The arena for untyped expressions



Summary. A question.

Summary

- The test manual determines the meaning of judgements (including equality judgements).
- Tests can corroborate or refute judgements (Popper).
- Tests with functional input leads us into games: input generation corresponds to playing innocent, well-bracketed opponent strategy.
- Meaning of hypothetical judgements, type equality, and identity types differs from Martin-Löf 1979.
- Nevertheless, rules of extensional type theory of Martin-Löf 1979 are justified, but in a different way.

Summary. A question.

Summary

- The test manual determines the meaning of judgements (including equality judgements).
- Tests can corroborate or refute judgements (Popper).
- Tests with functional input leads us into games: input generation corresponds to playing innocent, well-bracketed opponent strategy.
- Meaning of hypothetical judgements, type equality, and identity types differs from Martin-Löf 1979.
- Nevertheless, rules of extensional type theory of Martin-Löf 1979 are justified, but in a different way.

A question

- Meaning explanations for impredicative intuitionistic type theory (System F, Calculus of Constructions) in terms of tests?

Innocence

In an innocent player strategy the next move is uniquely determined by the *player view*. This view ignores two kinds of moves:

- those between an opponent question and its justifying player question;
- those between an opponent answer and its corresponding opponent question.

Formally:

$$P - view([]) \quad = \quad [$$