Realizing Continuity Using Stateful Computations

Liron Cohen and Vincent Rahli

February, 2023

Motivation

Continuity is a key component of intuitionistic logic

$$\forall F : \mathscr{B} \to \mathbb{N}.\ \forall \alpha : \mathscr{B}.\ \exists n : \mathbb{N}.\ \forall \beta : \mathscr{B}.$$
$$(\alpha = \beta \in \mathscr{B}_n) \to (F(\alpha) = F(\beta) \in \mathbb{N})$$

$$(\mathscr{B} = \mathbb{N}^{\mathbb{N}}\ \&\ \mathscr{B}_n = \mathbb{N}^{\mathbb{N}_n})$$

Motivation

Continuity is a key component of intuitionistic logic

$$\forall F:\mathscr{B}\to\mathbb{N}.\ \forall\alpha:\mathscr{B}.\ \exists n:\mathbb{N}.\ \forall\beta:\mathscr{B}.$$
$$(\alpha=\beta\in\mathscr{B}_n)\to(F(\alpha)=F(\beta)\in\mathbb{N})$$

$$(\mathscr{B}=\mathbb{N}^{\mathbb{N}}\ \textbf{\&}\ \mathscr{B}_n=\mathbb{N}^{\mathbb{N}_n})$$

Models exist for MLTT, System T, CTT, etc.

Motivation

Continuity is a key component of intuitionistic logic

$$\forall F:\mathscr{B}\to\mathbb{N}.\ \forall\alpha:\mathscr{B}.\ \exists n:\mathbb{N}.\ \forall\beta:\mathscr{B}.\\ (\alpha=\beta\in\mathscr{B}_n)\to(F(\alpha)=F(\beta)\in\mathbb{N})$$

($\mathscr{B}=\mathbb{N}^{\mathbb{N}}$ & $\mathscr{B}_n=\mathbb{N}^{\mathbb{N}_n}$)

Models exist for MLTT, System T, CTT, etc.

Used for example to prove that all real-valued functions on the unit interval are continuous.

Motivation

Typical methods to validate continuity:

- Forcing-based approaches (Coquand et al.)
- Models that internalize (Escardó et al.) or exhibit continuous behavior (Baillon et al.)

Motivation

Typical methods to validate continuity:

- Forcing-based approaches (Coquand et al.)
- Models that internalize (Escardó et al.) or exhibit continuous behavior (Baillon et al.)

Effectful computations: In some of these theories, the modulus can even be computed using effects (Longley)

Motivation

Typical methods to validate continuity:

- Forcing-based approaches (Coquand et al.)
- Models that internalize (Escardó et al.) or exhibit continuous behavior (Baillon et al.)

Effectful computations: In some of these theories, the modulus can even be computed using effects (Longley)

⚠ **Non-extensional** (Kreisel, Troesltra, Escardó and Xu)

For example: do $\lambda\alpha.0$ and $\lambda\alpha.\mathtt{let}\ x = \alpha(10)\ \mathtt{in}\ x - x$ have the same modulus of continuity?

This talk in 1 slide

type modality

- Beth
- Kripke
- open

$$\texttt{CTT} \longrightarrow \texttt{TT}^{\square}_{\mathcal{C}}$$

choice operator

- reference cells
- choice sequences

Realized continuity
using exceptions

Realized continuity
using stateful computations

- non-empty modality
- choices are numbers
- choices can be retrieved

(Syntactical account of the semantical methods)

$\mathrm{TT}_{\mathscr{C}}^{\square}$: A Family of Extensional Type Theories

A family of extensional type theories parameterized by
a type modality $\square$, and a choice type $\mathscr{C}$,
compatible with intuitionistic and classical principles

Formalized in Agda

$TT_{\mathscr{C}}^{\square}$: A Family of Extensional Type Theories

Untyped call-by-name lambda-calculus

sequent calculus

realizability semantics

Extensional

Dependent types

$\mathrm{TT}_{\mathscr{C}}^{\square}$: Syntax

Core Syntax:

$$T \in \texttt{Type} ::= \mathbb{N} \mid \mathbb{U}_i \mid \mathbf{\Pi} x{:}t.t \mid \mathbf{\Sigma} x{:}t.t \mid t = t \in t \mid t{+}t \mid \ldots$$

$$v \in \texttt{Value} ::= T \mid \star \mid \underline{n} \mid \lambda x.t \mid \langle t, t \rangle \mid \texttt{inl}(t) \mid \texttt{inr}(t) \mid \ldots$$

$$\begin{array}{lll} t \in \texttt{Term} & ::= & x \mid v \mid t\ t \mid \texttt{fix}(t) \mid \texttt{let}\ x := t\ \texttt{in}\ t \\ & & \mid \texttt{case}\ t\ \texttt{of}\ \texttt{inl}(x) \Rightarrow t\ \mathbf{|}\ \texttt{inr}(y) \Rightarrow t \\ & & \mid \texttt{let}\ x, y = t\ \texttt{in}\ t \mid \texttt{if}\ t{=}t\ \texttt{then}\ t\ \texttt{else}\ t \mid \ldots \end{array}$$

$\mathsf{TT}^{\square}_{\mathscr{C}}$: World-Based Computations

Core Operational Semantics:

$$
\begin{array}{lcllcl}
w & \vdash & (\lambda x.t_1)\ t_2 & \longmapsto & t_1[x\backslash t_2] \\
w & \vdash & \mathtt{let}\ x_1,x_2 = \langle t_1,t_2\rangle\ \mathtt{in}\ t & \longmapsto & t[x_1\backslash t_1;x_2\backslash t_2] \\
w & \vdash & \mathtt{fix}(v) & \longmapsto & v\ \mathtt{fix}(v)
\end{array}
$$

...

where $w \in \mathscr{W}$ (a poset with ordering $\sqsubseteq$)

$\mathrm{TT}_{\mathscr{C}}^{\Box}$: World-Based Computations

Core Operational Semantics:

$$
\begin{array}{lcll}
w & \vdash & (\lambda x.t_1)\ t_2 & \longmapsto & t_1[x\backslash t_2] \\
w & \vdash & \texttt{let}\ x_1, x_2 = \langle t_1, t_2 \rangle\ \texttt{in}\ t & \longmapsto & t[x_1\backslash t_1; x_2\backslash t_2] \\
w & \vdash & \texttt{fix}(v) & \longmapsto & v\ \texttt{fix}(v)
\end{array}
$$

...

where $w \in \mathscr{W}$ (a poset with ordering $\sqsubseteq$)

So far we haven't used the world

$\mathsf{TT}^{\square}_{\mathscr{C}}$: Choice Operator

Additional Components

- $\mathcal{N}$: abstract type of choice names
- $\mathscr{C}$: abstract type of choices inhabited by $\boldsymbol{\kappa_0} \neq \boldsymbol{\kappa_1}$
- a partial function: $\mathsf{choice} \in \mathscr{W} \to \mathcal{N} \to \mathscr{C}$

$\mathrm{TT}^{\square}_{\mathscr{C}}$: Choice Operator

Additional Components

- $\mathscr{N}$: abstract type of choice names
- $\mathscr{C}$: abstract type of choices inhabited by $\boldsymbol{\kappa_0} \neq \boldsymbol{\kappa_1}$
- a partial function: $\mathsf{choice} \in \mathscr{W} \to \mathscr{N} \to \mathscr{C}$

Syntax

$$v \in \texttt{Value} ::= \cdots \mid \delta \text{ (choice name)}$$

$$t \in \texttt{Term} ::= \cdots \mid !t \text{ (reading)}$$

$\mathrm{TT}_{\mathscr{C}}^{\square}$: Choice Operator

Additional Components

- $\mathcal{N}$: abstract type of choice names
- $\mathscr{C}$: abstract type of choices inhabited by $\boldsymbol{\kappa_0} \neq \boldsymbol{\kappa_1}$
- a partial function: $\mathsf{choice} \in \mathscr{W} \to \mathcal{N} \to \mathscr{C}$

Syntax

$$v \in \mathtt{Value} ::= \cdots \mid \delta \text{ (choice name)}$$

$$t \in \mathtt{Term} ::= \cdots \mid !t \text{ (reading)}$$

Operational Semantics

$$w \vdash !\delta \mapsto \mathsf{choice}(w, \delta)$$

$\mathsf{TT}_{\mathscr{C}}^{\square}$: Inference Rules

Standard ETT rules:

$$\frac{\Gamma, x : A \vdash b : B[x] \qquad \Gamma \vdash \star : (A \in \mathbb{U}_i)}{\Gamma \vdash \lambda x.b : \mathbf{\Pi} a{:}A.B[a]} \qquad \cdots$$

$\mathrm{TT}_{\mathscr{C}}^{\square}$: Inference Rules

Standard ETT rules:

$$\frac{\Gamma, x : A \vdash b : B[x] \qquad \Gamma \vdash \star : (A \in \mathbb{U}_i)}{\Gamma \vdash \lambda x.b : \mathbf{\Pi} a{:}A.B[a]} \qquad \cdots$$

+ LEM for some $\square$ modalities (e.g., Beth)

$\mathrm{TT}_{\mathscr{C}}^{\Box}$: Inference Rules

Standard ETT rules:

$$\frac{\Gamma, x : A \vdash b : B[x] \qquad \Gamma \vdash \star : (A \in \mathbb{U}_i)}{\Gamma \vdash \lambda x.b : \mathbf{\Pi} a{:}A.B[a]} \qquad \cdots$$

+ LEM for some $\Box$ modalities (e.g., Beth)

+ $\neg$LEM for some $\Box$ modalities (e.g., Open)

$TT_{\mathscr{C}}^{\Box}$: Realizability semantics

An inductive relation that expresses type equality

$$w \vDash T_1 \equiv T_2$$

A recursive function that expresses equality in a type

$$w \vDash a \equiv b \in T$$

$\mathrm{TT}^{\square}_{\mathscr{C}}$: Realizability semantics

An inductive relation that expresses type equality

$$w \vDash T_1 \equiv T_2$$

A recursive function that expresses equality in a type

$$w \vDash a \equiv b \in T$$

For example (product types):

$$w \vDash \mathbf{\Pi} x_1{:}A_1.B_1 \equiv \mathbf{\Pi} x_2{:}A_2.B_2$$
$$\Longleftrightarrow$$
$$\forall w' \sqsupseteq w. w' \vDash A_1 \equiv A_2 \;\land$$
$$\forall w' \sqsupseteq w. \forall a_1, a_2.\; w' \vDash a_1 \equiv a_2 \in A_1 \Rightarrow w' \vDash B_1[x_1 \backslash a_1] \equiv B_2[x_2 \backslash a_2]$$

$TT_{\mathscr{C}}^{\Box}$: Modalities

An abstract modality on (the semantics of) types: $\Box$

$\mathrm{TT}_{\mathscr{C}}^{\square}$: Modalities

An abstract modality on (the semantics of) types: $\square$

Forcing interpretation: $\square_w(w'.w' \vDash T) \rightarrow w \vDash T$

$TT_{\mathscr{C}}^{\square}$: Modalities

An abstract modality on (the semantics of) types: $\square$

Forcing interpretation: $\square_w(w'.w' \vDash T) \to w \vDash T$

Properties (where $(w:\mathscr{W}),(P,Q:\mathscr{P}_w)$**):**

monotonicity of $\square$	$\forall w' \sqsupseteq w.\square_w P \to \square_{w'} P$
K, distribution axiom	$\square_w(P \to Q) \to \square_w P \to \square_w Q$
$C4$, i.e., $\square\square \to \square$	$\square_w(w'.\square_{w'} P) \to \square_w P$
$\forall \to \square$	$\forall_w^{\sqsubseteq}(P) \to \square_w P$
T, reflexivity axiom	$\forall(P:\mathbb{P}).\square_w(w'.P) \to P$

$TT_{\mathscr{C}}^{\Box}$: Modalities

An abstract modality on (the semantics of) types: $\Box$

Forcing interpretation: $\Box_w(w'.w' \vDash T) \rightarrow w \vDash T$

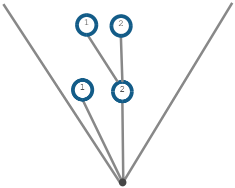
Properties (where $(w : \mathscr{W}), (P, Q : \mathscr{P}_w)$**):**

monotonicity of $\Box$	$\forall w' \sqsupseteq w.\Box_w P \rightarrow \Box_{w'} P$
K, distribution axiom	$\Box_w(P \rightarrow Q) \rightarrow \Box_w P \rightarrow \Box_w Q$
$C4$, i.e., $\Box\Box \rightarrow \Box$	$\Box_w(w'.\Box_{w'} P) \rightarrow \Box_w P$
$\forall \rightarrow \Box$	$\forall_w^{\sqsubseteq}(P) \rightarrow \Box_w P$
T, reflexivity axiom	$\forall (P : \mathbb{P}).\Box_w(w'.P) \rightarrow P$

Enough to prove standard properties of the type system: consistency, symmetry, transitivity, etc.

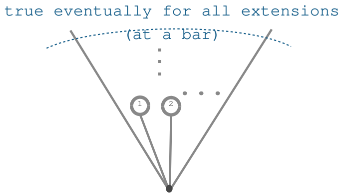
TT$^{\square}_{\mathscr{C}}$: Examples of Modalities

Kripke modality



$w \vDash T \iff$
$\forall w_1 \sqsupseteq w. w_1 \vDash T$
(modality: $\square_K(T)$)

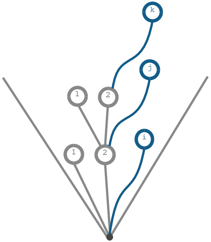
Beth modality



$w \vDash T \iff$
$\exists b \in \mathrm{bar}(w). \forall w_1 \in b.$
$\forall w_2 \sqsupseteq w_1. w_2 \vDash T$

(modality: $\square_B(T)$)

Open modality



$w \vDash T \iff$
$\forall w_1 \sqsupseteq w. \exists w_2 \sqsupseteq w_1.$
$\forall w_3 \sqsupseteq w_2. w_2 \vDash T$
(modality: $\square_O(T)$)

$\mathsf{TT}_{\mathscr{C}}^{\square}$: Modalities from Coverings

Modalities can be derived from **coverings**

$\mathrm{TT}_{\mathscr{C}}^{\square}$: Modalities from Coverings

Modalities can be derived from **coverings**

Opens: $\mathscr{O} := \mathscr{W} \rightarrow \mathbb{P}$ (sets of worlds)

$\mathsf{TT}^{\square}_{\mathscr{C}}$: Modalities from Coverings

Modalities can be derived from **coverings**

Opens: $\mathscr{O} := \mathscr{W} \to \mathbb{P}$ (sets of worlds)

Predicates on opens: $\mathsf{Covering} := \mathscr{W} \to \mathscr{O} \to \mathbb{P}$

$\mathsf{TT}^{\Box}_{\mathscr{C}}$: Modalities from Coverings

Modalities can be derived from **coverings**

Opens: $\mathscr{O} := \mathscr{W} \to \mathbb{P}$ (sets of worlds)

Predicates on opens: $\mathsf{Covering} := \mathscr{W} \to \mathscr{O} \to \mathbb{P}$

$C \in \mathsf{Covering}$ is a **covering** if:

- it is closed under binary intersections, union & subsets
- it contains the top element
- its elements are non-empty

$TT^{\Box}_{\mathscr{C}}$: Modalities from Coverings

Modalities can be derived from **coverings**

Opens: $\mathscr{O} := \mathscr{W} \rightarrow \mathbb{P}$ (sets of worlds)

Predicates on opens: $\mathsf{Covering} := \mathscr{W} \rightarrow \mathscr{O} \rightarrow \mathbb{P}$

$C \in \mathsf{Covering}$ is a **covering** if:

- it is closed under binary intersections, union & subsets
- it contains the top element
- its elements are non-empty

Any covering $C \in \mathsf{Covering}$ can be turned into a modality $\Box$

$TT^{\Box}_{\mathscr{C}}$: Modalities from Coverings

Modalities can be derived from **coverings**

Opens: $\mathscr{O} := \mathscr{W} \to \mathbb{P}$ (sets of worlds)

Predicates on opens: $\mathsf{Covering} := \mathscr{W} \to \mathscr{O} \to \mathbb{P}$

$C \in \mathsf{Covering}$ is a **covering** if:

- it is closed under binary intersections, union & subsets
- it contains the top element
- its elements are non-empty

Any covering $C \in \mathsf{Covering}$ can be turned into a modality $\Box$

For example, Kripke, Beth, Open coverings

Continuity – Functions in $\mathbb{N}^{\mathscr{B}}$ only need initial segments

Continuity – Functions in $\mathbb{N}^{\mathscr{B}}$ only need initial segments

Continuity axiom for numbers:

$$\forall F : \mathbb{N}^{\mathscr{B}}.\ \forall \alpha : \mathscr{B}.\ \exists n : \mathbb{N}.\ \forall \beta : \mathscr{B}.\ \alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Continuity – Functions in $\mathbb{N}^{\mathscr{B}}$ only need initial segments

Continuity axiom for numbers:

$$\forall F : \mathbb{N}^{\mathscr{B}}.\ \forall \alpha : \mathscr{B}.\ \exists n : \mathbb{N}.\ \forall \beta : \mathscr{B}.\ \alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Uniform continuity theorem ($f \in [\alpha, \beta] \rightarrow \mathbb{R}$):

$$\forall \epsilon > 0. \exists \delta > 0. \forall x, y : [\alpha, \beta].\ |x - y| \leq \delta \rightarrow |f(x) - f(y)| \leq \epsilon$$

Continuity – Functions in $\mathbb{N}^{\mathscr{B}}$ only need initial segments

Continuity axiom for numbers:

$$\forall F : \mathbb{N}^{\mathscr{B}}.\ \forall \alpha : \mathscr{B}.\ \exists n : \mathbb{N}.\ \forall \beta : \mathscr{B}.\ \alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Uniform continuity theorem ($f \in [\alpha, \beta] \rightarrow \mathbb{R}$):

$$\forall \epsilon > 0. \exists \delta > 0. \forall x, y : [\alpha, \beta].\ |x - y| \leq \delta \rightarrow |f(x) - f(y)| \leq \epsilon$$

False (Kreisel 62, Troelstra 77, Escardó & Xu 2015):

$$\mathbf{\Pi} F{:}\mathscr{B} \rightarrow \mathbb{N}. \mathbf{\Pi} \alpha{:}\mathscr{B}. \mathbf{\Sigma} n{:}\mathbb{N}. \mathbf{\Pi} \beta{:}\mathscr{B}. \alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Continuity – Functions in $\mathbb{N}^{\mathscr{B}}$ only need initial segments

Continuity axiom for numbers:

$$\forall F : \mathbb{N}^{\mathscr{B}}.\ \forall \alpha : \mathscr{B}.\ \exists n : \mathbb{N}.\ \forall \beta : \mathscr{B}.\ \alpha =_{\mathscr{B}_n} \beta \to F(\alpha) =_{\mathbb{N}} F(\beta)$$

Uniform continuity theorem ($f \in [\alpha, \beta] \to \mathbb{R}$):

$$\forall \epsilon > 0. \exists \delta > 0. \forall x, y : [\alpha, \beta].\ |x - y| \leq \delta \to |f(x) - f(y)| \leq \epsilon$$

False (Kreisel 62, Troelstra 77, Escardó & Xu 2015):

$$\mathbf{\Pi} F{:}\mathscr{B} \to \mathbb{N}. \mathbf{\Pi} \alpha{:}\mathscr{B}. \mathbf{\Sigma} n{:}\mathbb{N}. \mathbf{\Pi} \beta{:}\mathscr{B}. \alpha =_{\mathscr{B}_n} \beta \to F(\alpha) =_{\mathbb{N}} F(\beta)$$

Consistent with CTT if truncated (MSCS'17):

$$\mathbf{\Pi} F{:}\mathbb{N}^{\mathscr{B}}. \mathbf{\Pi} \alpha{:}\mathscr{B}. \downharpoonleft \mathbf{\Sigma} n{:}\mathbb{N}. \mathbf{\Pi} \beta{:}\mathscr{B}. \alpha =_{\mathscr{B}_n} \beta \to F(\alpha) =_{\mathbb{N}} F(\beta)$$

Continuity – Computing moduli of continuity

$$\mathbf{\Pi} F{:}\mathbb{N}^{\mathscr{B}}.\mathbf{\Pi}\alpha{:}\mathscr{B}.{\downharpoonleft}\mathbf{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi}\beta{:}\mathscr{B}.\alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Continuity – Computing moduli of continuity

$$\mathbf{\Pi} F{:}\mathbb{N}^{\mathscr{B}}.\mathbf{\Pi}\alpha{:}\mathscr{B}.\downharpoonleft\mathbf{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi}\beta{:}\mathscr{B}.\alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Consistent with CTT if truncated (MSCS'17):

Continuity – Computing moduli of continuity

$$\mathbf{\Pi} F{:}\mathbb{N}^{\mathscr{B}}.\mathbf{\Pi}\alpha{:}\mathscr{B}.\downharpoonleft\mathbf{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi}\beta{:}\mathscr{B}.\alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Consistent with CTT if truncated (MSCS'17):

Essence: Test whether a seq. α of length n is long enough

Continuity – Computing moduli of continuity

$$\mathbf{\Pi} F{:}\mathbb{N}^{\mathscr{B}}.\mathbf{\Pi}\alpha{:}\mathscr{B}.\downarrow\mathbf{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi}\beta{:}\mathscr{B}.\alpha =_{\mathscr{B}_n} \beta \to F(\alpha) =_{\mathbb{N}} F(\beta)$$

Consistent with CTT if truncated (MSCS'17):

Essence: Test whether a seq. α of length n is long enough

Effectful computations following Longley's method:

```
let exception e in
(F (fun x ⇒ if x < n
              then α(x)
              else raise e);
 true) handle e ⇒ false
```

Continuity – Computing moduli of continuity

$$\mathbf{\Pi} F{:}\mathbb{N}^{\mathscr{B}}.\mathbf{\Pi}\alpha{:}\mathscr{B}.\rfloor\mathbf{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi}\beta{:}\mathscr{B}.\alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Consistent with CTT if truncated (MSCS'17):

Essence: Test whether a seq. α of length n is long enough

Effectful computations following Longley's method:

```
let exception e in
(F (fun x ⇒ if x < n
              then α(x)
              else raise e);
 true) handle e ⇒ false
```

Plus a loop until the modulus of continuity is reached

Continuity – Computing moduli of continuity

$$\mathbf{\Pi} F{:}\mathbb{N}^{\mathscr{B}}.\mathbf{\Pi}\alpha{:}\mathscr{B}.{\color{red}\rfloor\mathbf{\Sigma}} n{:}\mathbb{N}.\mathbf{\Pi}\beta{:}\mathscr{B}.\alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Continuity – Computing moduli of continuity

$$\mathbf{\Pi} F{:}\mathbb{N}^{\mathscr{B}}.\mathbf{\Pi}\alpha{:}\mathscr{B}.\rfloor\boldsymbol{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi}\beta{:}\mathscr{B}.\alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Also consistent with $\mathsf{TT}^{\square}_{\mathscr{C}}$ (CSL'23):

Continuity – Computing moduli of continuity

$$\mathbf{\Pi} F{:}\mathbb{N}^{\mathscr{B}}.\mathbf{\Pi}\alpha{:}\mathscr{B}.\downharpoonleft\mathbf{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi}\beta{:}\mathscr{B}.\alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Also consistent with $\mathsf{TT}^{\square}_{\mathscr{C}}$ (CSL'23):

Essence: Moduli of continuity can be computed in one go using reference-like operators

Continuity – Computing moduli of continuity

$$\mathbf{\Pi} F{:}\mathbb{N}^{\mathscr{B}}.\mathbf{\Pi}\alpha{:}\mathscr{B}.{\color{red}\downharpoonleft\mathbf{\Sigma}} n{:}\mathbb{N}.\mathbf{\Pi}\beta{:}\mathscr{B}.\alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Also consistent with $\mathbf{TT}^{\square}_{\mathscr{C}}$ (CSL'23):

Essence: Moduli of continuity can be computed in one go using reference-like operators

Again following Longley's method:

```
let r = ref 0 in
F (fun x ⇒ (if x > !r then r := x); α(x));
!r + 1
```

Continuity – Computing moduli of continuity

$$\mathbf{\Pi} F{:}\mathbb{N}^{\mathscr{B}}.\mathbf{\Pi}\alpha{:}\mathscr{B}.\downarrow\mathbf{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi}\beta{:}\mathscr{B}.\alpha =_{\mathscr{B}_n} \beta \rightarrow F(\alpha) =_{\mathbb{N}} F(\beta)$$

Also consistent with $\mathbf{TT}^{\square}_{\mathscr{C}}$ (CSL'23):

Essence: Moduli of continuity can be computed in one go using reference-like operators

Again following Longley's method:

```
let r = ref 0 in
F (fun x ⇒ (if x > !r then r := x); α(x));
!r + 1
```

More straightforward; No need for a loop

Continuity – Purity

Different moduli in extensions:

- $\lambda\alpha.\alpha(!\delta);0$
- α might get applied to 0 in w_1
- and to 1 in $w_2 \sqsupseteq w_1$

Continuity – Purity

Different moduli in extensions:

- $\lambda\alpha.\alpha(!\delta);0$
- α might get applied to 0 in w_1
- and to 1 in $w_2 \sqsupseteq w_1$

? Are impure functions continuous?

Continuity – Purity

Different moduli in extensions:

- $\lambda\alpha.\alpha(!\delta);0$
- α might get applied to 0 in w_1
- and to 1 in $w_2 \sqsupseteq w_1$

? Are impure functions continuous?

? Can the modulus of continuity inhabit a variant of $\mathbb{N}$ where numbers are allowed to change in extensions?

Continuity – Purity

We require here functions to be pure ($\mathbf{\Pi_p}$):

Theorem (Continuity Principle)

The following continuity principle, is valid w.r.t. the above semantics:

$$\mathbf{\Pi_p} F{:}\mathscr{B} \to \mathbb{N}.\mathbf{\Pi_p}\alpha{:}\mathscr{B}.\downharpoonleft\mathbf{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi_p}\beta{:}\mathscr{B}.$$
$$(\alpha = \beta \in \mathscr{B}_n) \to (F(\alpha) = F(\beta) \in \mathbb{N})$$

and is inhabited by the above computation, denoted $\texttt{mod}(F, \alpha)$

Continuity – Further Additional Components

Further Additional Components

- a function: $\texttt{update} \in \mathscr{W} \to \mathscr{N} \to \mathscr{C} \to \mathscr{W}$
- $\texttt{namefree}(t)$ states that t does not contain choices

Continuity – Further Additional Components

Further Additional Components

- a function: $\texttt{update} \in \mathscr{W} \to \mathscr{N} \to \mathscr{C} \to \mathscr{W}$
- $\texttt{namefree}(t)$ states that t does not contain choices

Syntax

$$
\begin{array}{rcl}
t \in \texttt{Term} & ::= & \cdots \boxed{\mid \boldsymbol{\nu} x.t} \boxed{\mid \texttt{choose}(t_1, t_2)} \\
& & \mid \texttt{if } t_1 < t_2 \texttt{ then } t_3 \texttt{ else } t_4 \mid t_1 + t_2 \\
T \in \texttt{Type} & ::= & \cdots \boxed{\mid \texttt{pure}} \mid t_1 \cap t_2 \mid \downharpoonleft t
\end{array}
$$

Continuity – Further Additional Components

Further Additional Components

- a function: $\mathsf{update} \in \mathscr{W} \to \mathscr{N} \to \mathscr{C} \to \mathscr{W}$
- $\mathtt{namefree}(t)$ states that t does not contain choices

Syntax

$$
\begin{array}{rl}
t \in \mathtt{Term} & ::= \cdots \boxed{\mid \boldsymbol{\nu} x.t} \boxed{\mid \mathtt{choose}(t_1, t_2)} \\
& \mid \mathtt{if}\ t_1\ <\ t_2\ \mathtt{then}\ t_3\ \mathtt{else}\ t_4 \mid t_1 + t_2 \\
T \in \mathtt{Type} & ::= \cdots \boxed{\mid \mathtt{pure}} \mid t_1 \cap t_2 \mid \downharpoonleft t
\end{array}
$$

Operational Semantics

$$w, \mathsf{update}(w, \delta, t) \vdash \mathtt{choose}(\delta, t) \mapsto \star$$

Continuity – Proof Steps

Step 1 (The Modulus is a Number)

If $\mathtt{namefree}(F)$, $\mathtt{namefree}(\alpha)$, $w \vDash F \in \mathbb{N}^{\mathscr{B}}$, *and* $w \vDash \alpha \in \mathscr{B}$, *for some world* w, *then* $w \vDash \mathtt{mod}(F, \alpha) \in \mathbb{N}$

Continuity – Proof Steps

Step 1 (The Modulus is a Number)

If $\mathtt{namefree}(F)$, $\mathtt{namefree}(\alpha)$, $w \vDash F \in \mathbb{N}^{\mathscr{B}}$, *and* $w \vDash \alpha \in \mathscr{B}$, *for some world* w, *then* $w \vDash \mathtt{mod}(F,\alpha) \in \mathbb{N}$

Step 2 (The Modulus is the Highest Number)

If $w, w' \vdash \mathtt{mod}(F,\alpha) \mapsto^* \underline{n}$ *such that* $\mathtt{mod}(F,\alpha)$ *generates a fresh name* δ, *then for any world* w_0 *occurring along this computation, it must be that* $\mathsf{choice}(w_0,\delta) \leq \mathsf{choice}(w',\delta)$.

Continuity – Proof Steps

Step 1 (The Modulus is a Number)

If $\mathtt{namefree}(F)$, $\mathtt{namefree}(\alpha)$, $w \vDash F \in \mathbb{N}^{\mathscr{B}}$, *and* $w \vDash \alpha \in \mathscr{B}$, *for some world* w, *then* $w \vDash \mathtt{mod}(F,\alpha) \in \mathbb{N}$

Step 2 (The Modulus is the Highest Number)

If $w, w' \vdash \mathtt{mod}(F,\alpha) \mapsto^* \underline{n}$ *such that* $\mathtt{mod}(F,\alpha)$ *generates a fresh name* δ, *then for any world* w_0 *occurring along this computation, it must be that* $\mathsf{choice}(w_0,\delta) \leq \mathsf{choice}(w',\delta)$.

Step 3 (The Modulus is the Modulus)

If $w \vDash \alpha \equiv \beta \in \mathscr{B}_n$ *then* $w \vDash F(\alpha) \equiv F(\beta) \in \mathbb{N}$.

Summary

$\boxed{\text{TT}^{\square}_{\mathscr{C}}\text{: a type theory to program with effects}}$

$\square \in \{\mathit{Kripke}, \mathit{Beth}, \mathit{Open}\}$
$\mathscr{C} \in \{\mathit{Ref}, \mathit{CS}\}$

$\boxed{\text{Simple reference-based computation of continuity}}$

What about impure functions?

Questions?