Extending Church-Turing Computability

Liron Cohen

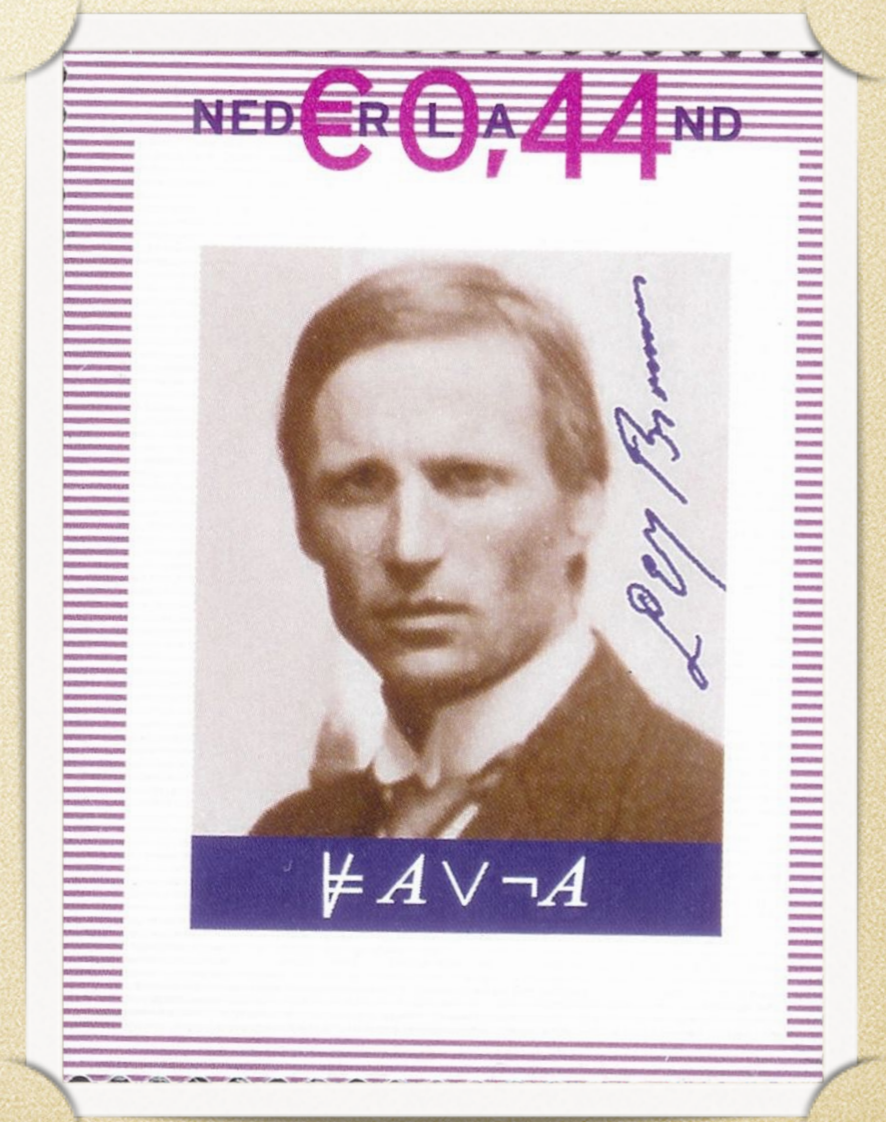
Joint work with Vincent Rahli, Mark Bickford and Robert Constable

HIM Summer School on Types, Sets and Constructions

May 2018

Broader Notion of Computability

Free choice sequences were introduced by Brouwer to capture the intuition of the continuum [Second Act of Intuitionism,1919].



"In Brouwer's case there seems to have been a nagging suspicion that unless he personally intervened to prevent it, the continuum would turn out to be discrete"

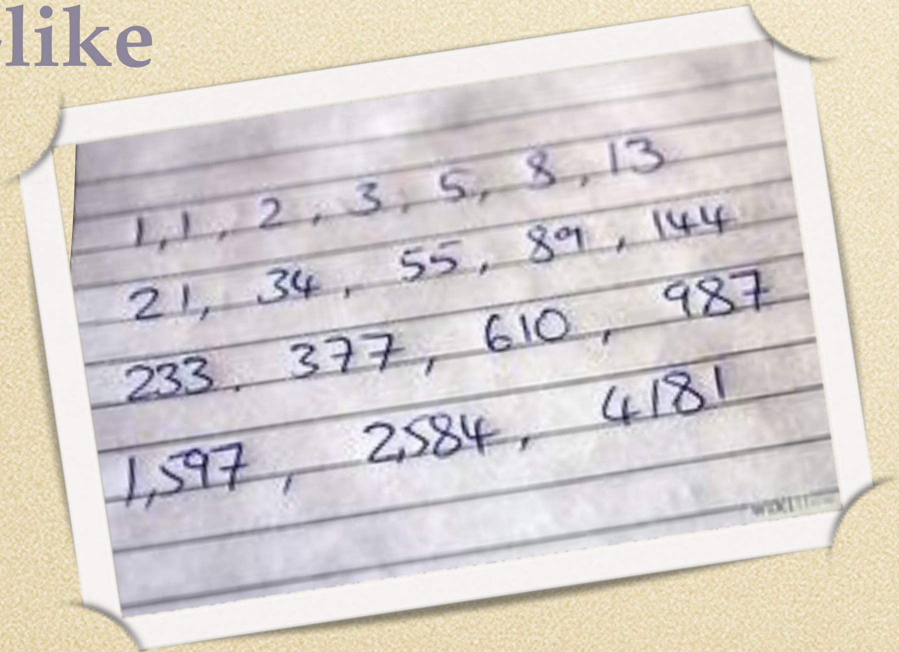
- Bishop, 1985

Free Choice Sequences

"new mathematical entities… in the form of infinitely proceeding sequences $p_1, p_2, \ldots$, whose terms are chosen more or less freely from mathematical entities previously acquired . . ."

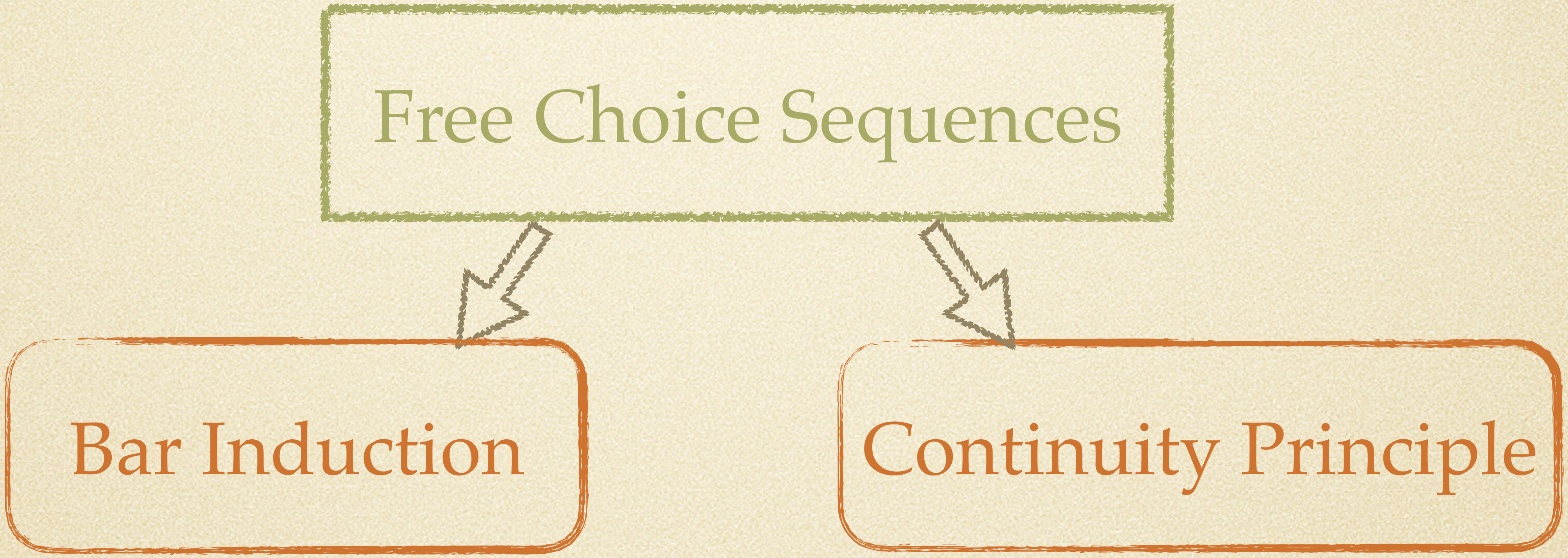
- Brouwer, 1919

Law-like



Law-less (Free)

Some Consequences



Requires the existence of some non-recursive functions

Entails a restriction on the behavior of all non-recursive functions

Litrature

- D. Bridges and F. Richman. Varieties of Constructive Mathematics. Cambridge University Press, 1987.
- T. Coquand and B. Mannaa. The Independence of Markov's Principle in Type Theory. In: FSCD 2016. Vol. 52. LIPIcs, 17:1–17:18.
- D. van Dalen. An interpretation of intuitionistic analysis. In: Annals of mathematical logic 13.1. 1978, pp. 1–43.
- M. H. Escardó and C. Xu. The Inconsistency of a Brouwerian Continuity Principle with the Curry-Howard Interpretation. In: TLCA 2015. Vol. 38. LIPIcs, pp. 153–164.
- J. R. Moschovakis. An intuitionistic theory of lawlike, choice and lawless sequences. In: Logic Colloquium'90, Helsinki. 1993, pp. 191–209.
- M. Rathjen. A note on Bar Induction in Constructive Set Theory. In: Math. Log. Q. 52.3. 2006, pp. 253–258.
- A. S. Troelstra. Choice sequences: a chapter of intuitionistic mathematics. Clarendon Press Oxford, 1977.

Our Goal

Extend the constructive type theory implemented by the Nuprl proof assistant to support Brouwer's broader sense of computability through the embedding of choice sequences

↓

A new intuitionistic type theory — BITT

Key implementation features

In the theory:

Free choice sequence =

Finite, yet unbounded, non-deterministic sequence

open-ended

Library

"state"

In the model:

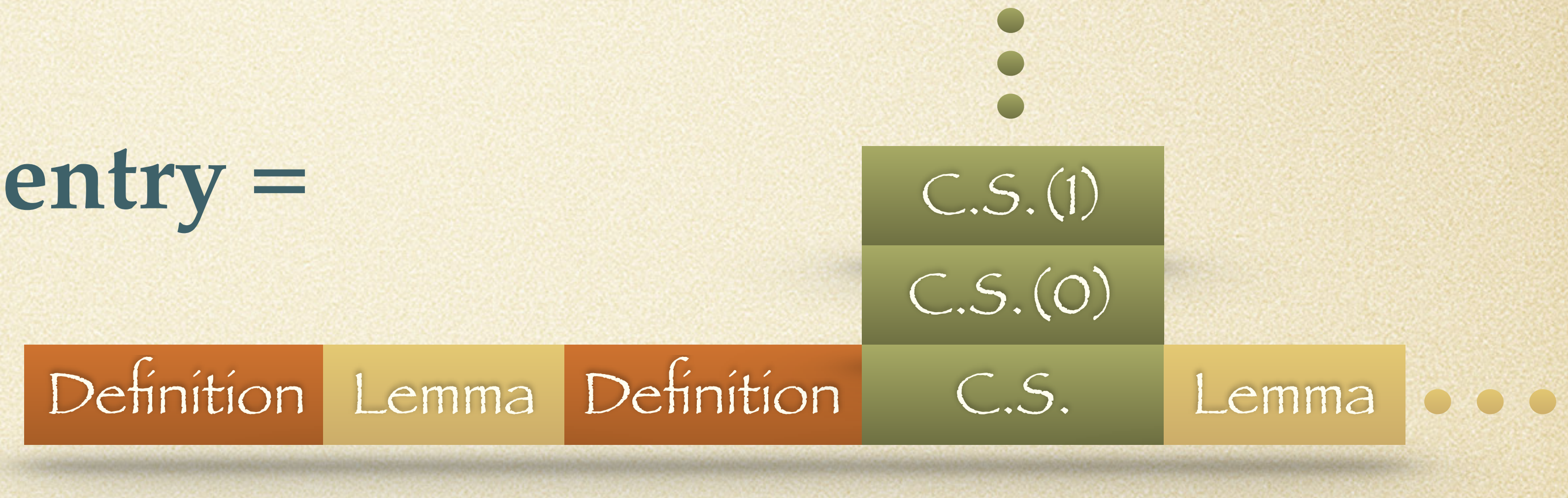
A Beth-style semantics → possible worlds = Library extensions

Nuprl in a Nutshell

- Implements constructive type theory (CTT)
  - extensional
  - with dependent types
  - with partial functions
- Types are interpreted as PERs on terms

The Library

Choice sequence entry = a list of terms



+

a restriction

law — $\lambda n,t.\exists i:\mathbb{N}.t=i$

restriction predicate (+proof of totality) — $\lambda n,t.\text{if } n \leq |l| \text{ then } t=l[n] \text{ else } \exists i:\mathbb{N}.t=i$

- Adding a value to a choice sequence entails proving that the restriction of the sequence is satisfied.
  - For decidable restrictions, this can be done automatically.

Name Spaces

- A mechanism for enforcing certain restrictions.

a choice sequence name: string (raw name)

\+

constraint (name space)

number

0 - choice sequence of numbers

<“a”,0>

list of numbers

choice sequence of numbers, extending the list

<“a”,[2,4,1]>

Extended Computation System

$csn \in \text{CSName}$	$::=$	$\langle s, space \rangle$	C.S. name
$s \in \text{RawCSName}$			
$space \in \text{Space}$	$::=$	$n \mid [n_1, \ldots, n_k]$	C.S. name space
$v \in \text{Value}$	$::=$	$\ldots \mid \text{seq}(csn)$	C.S.
$vt \in \text{Type}$	$::=$	$\ldots \mid \text{Free}(n)$	C.S. type
$t \in \text{Term}$	$::=$	$\ldots \mid \text{if } t_1 = t_2 \text{ then } t_3 \text{ else } t_4$	C.S. equality

Operational Semantics

$$\text{if seq}(csn_1)\text{=seq}(csn_2) \text{ then } t_1 \text{ else } t_2 \mapsto_{lib} \begin{cases} t_1 & csn_1 = csn_2 \\ t_2 & csn_1 \neq csn_2 \end{cases}$$

$$\text{seq}(csn)(i) \mapsto_{lib} cs[i] \qquad \text{when } cs[i] \text{ is defined in } lib$$

The computation rules explicitly depend on the library

Beth-Style Semantics

- The notion of truth has to also explicitly depend on the library.
- The libraries behave as the worlds in the possible-world semantics, and in any particular library the semantic is induced by the realizability semantics.

$$\exists x . a(100) = x$$

Should be valid

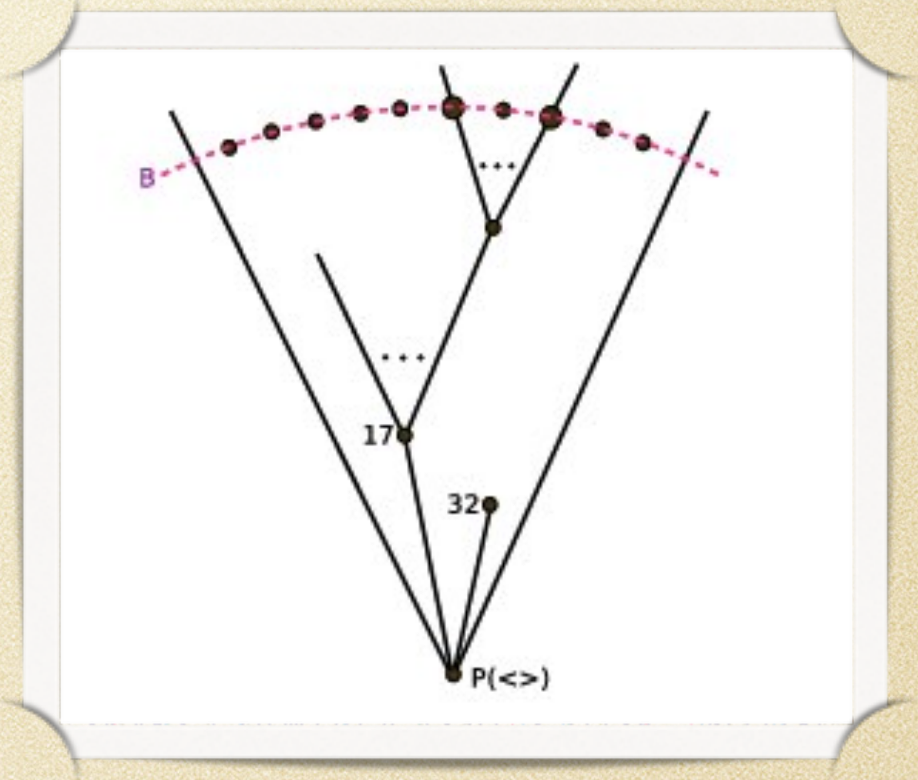
Not valid
in Kripke Semantics

Need to use
Beth Semantics

In Beth models objects only
need to "eventually" exists

Bars

- Bar of a library = a collection of libraries covering all possible extensions of the library.
- Operations on bars:
  - intersecting bars
  - raising bars
  - collapsing/expanding bars



- Types are interpreted as PERs on closed terms that need only exist in a bar of the library.

Type System

Building BITT

- Define operators that interpret the standard type constructors.
- Adding new constructors:
  - BAR — assigns meaning to types at a library, if they are defined in some bar.
  - FREE — assigns meaning to new Free(n) types.
- Define an hierarchy of universes by closing under the type constructors.
- BITT is the collection of all universes closed under the type constructors.

BITT satisfies ⇒ symmetry, transitivity,…
BITT satisfies ⇒ monotonicity + locality

The Axioms

- For any finite list of values there is a choice sequence that extends it.

$$\forall n:\mathbb{N}.\forall f:B_{\mathbb{N}}.\exists a:Free(0).f =_{B_{\mathbb{N}}} a$$

- Decidability of equality.

$$\forall a,b:Free(0).a \simeq b \vee \neg a \simeq b$$

- Axiom of Open Data

$$\forall a:Free(0).\phi(a) \Rightarrow \downarrow \exists n:\mathbb{N}.\forall b:Free(0).a =_{\mathbb{N}_n} b \Rightarrow \phi(b)$$

  - If ϕ holds for a choice sequence, then it has a finite initial segment, l, s.t. ϕ holds for all choice sequences that extend l.*