Connecting Constructive Notions of Ordinals in Homotopy Type Theory

Nicolai Kraus Fredrik Nordvall Forsberg Chuangjie Xu

MFCS 2021, August 23–27, Tallinn/online hybrid

What are ordinals?

One answer: **Numbers** for counting/ordering:

$$0,\quad 1,\quad 2,\quad 3,\quad \dots,\quad \omega,\quad \omega+1,\quad \omega+2,\quad \dots$$

$$\omega\cdot 2,\quad \omega\cdot 2+1,\quad \dots,\quad \omega^2,\quad \dots,\quad \omega^2\cdot 3+\omega\cdot 7+13,\quad \dots,$$

$$\omega^\omega,\quad \dots,\quad \varepsilon_0=\omega^{\omega^{\omega^{\cdots}}},\quad \dots,\quad \varepsilon_{17},\quad \dots,\quad \omega_1,\quad \dots$$

What are ordinals?

One answer: **Numbers** for counting/ordering:

$$\boxed{0, \quad 1, \quad 2, \quad 3,} \quad \dots, \quad \omega, \quad \omega+1, \quad \omega+2, \quad \dots$$

$$\omega \cdot 2, \quad \omega \cdot 2 + 1, \quad \dots, \quad \omega^2, \quad \dots, \quad \omega^2 \cdot 3 + \omega \cdot 7 + 13, \quad \dots,$$

$$\omega^\omega, \quad \dots, \quad \varepsilon_0 = \omega^{\omega^{\omega^{\cdots}}}, \quad \dots, \quad \varepsilon_{17}, \quad \dots, \quad \omega_1, \quad \dots$$

What are ordinals?

One answer: **Numbers** for counting/ordering:

$$0, \quad 1, \quad 2, \quad 3, \quad \ldots, \quad \boxed{\omega, \quad \omega+1, \quad \omega+2,} \quad \ldots$$

$$\omega \cdot 2, \quad \omega \cdot 2 + 1, \quad \ldots, \quad \omega^2, \quad \ldots, \quad \omega^2 \cdot 3 + \omega \cdot 7 + 13, \quad \ldots,$$

$$\omega^\omega, \quad \ldots, \quad \varepsilon_0 = \omega^{\omega^{\omega^{\cdots}}}, \quad \ldots, \quad \varepsilon_{17}, \quad \ldots, \quad \omega_1, \quad \ldots$$

What are ordinals?

One answer: **Numbers** for counting/ordering:

$$0, \quad 1, \quad 2, \quad 3, \quad \ldots, \quad \omega, \quad \omega+1, \quad \omega+2, \quad \ldots$$

$$\boxed{\begin{array}{l} \omega\cdot 2, \quad \omega\cdot 2+1, \quad \ldots, \quad \omega^2, \quad \ldots, \quad \omega^2\cdot 3+\omega\cdot 7+13, \quad \ldots, \\ \omega^\omega, \quad \ldots, \quad \varepsilon_0=\omega^{\omega^{\omega^{\cdots}}}, \quad \ldots, \quad \varepsilon_{17}, \quad \ldots, \quad \omega_1, \quad \ldots \end{array}}$$

What are ordinals?

One answer: **Numbers** for counting/ordering:

$$0, \quad 1, \quad 2, \quad 3, \quad \ldots, \quad \omega, \quad \omega+1, \quad \omega+2, \quad \ldots$$

$$\omega\cdot 2, \quad \omega\cdot 2+1, \quad \ldots, \quad \omega^2, \quad \ldots, \quad \omega^2\cdot 3+\omega\cdot 7+13, \quad \ldots,$$

$$\omega^\omega, \quad \ldots, \quad \varepsilon_0 = \omega^{\omega^{\omega^{\cdots}}}, \quad \ldots, \quad \varepsilon_{17}, \quad \ldots, \quad \omega_1, \quad \ldots$$

Another answer: $\boxed{\textbf{Sets with an order } <}$ which is

- **transitive**: $(a < b) \to (b < c) \to (a < c)$
- **wellfounded**: every sequence $a_0 > a_1 > a_2 > a_3 > \ldots$ terminates
- and **trichotomous**: $(a < b) \vee (a = b) \vee (b < a)$
- ... or **extensional** (instead of trichotomous):
  $$(\forall a. a < b \leftrightarrow a < c) \to b = c$$

What are ordinals?

One answer: **Numbers** for counting/ordering:

$$0, \quad 1, \quad 2, \quad 3, \quad \dots, \quad \omega, \quad \omega+1, \quad \omega+2, \quad \dots$$

$$\omega\cdot 2, \quad \omega\cdot 2+1, \quad \dots, \quad \omega^2, \quad \dots, \quad \omega^2\cdot 3+\omega\cdot 7+13, \quad \dots,$$

$$\omega^\omega, \quad \dots, \quad \varepsilon_0=\omega^{\omega^{\omega^{\cdots}}}, \quad \dots, \quad \varepsilon_{17}, \quad \dots, \quad \omega_1, \quad \dots$$

Another answer: **Sets with an order** $<$ which is

- $\boxed{\textbf{transitive}\text{:} \qquad (a<b)\to(b<c)\to(a<c)}$
- **wellfounded**: every sequence $a_0 > a_1 > a_2 > a_3 > \dots$ terminates
- and **trichotomous**: $(a<b)\vee(a=b)\vee(b<a)$
- … or **extensional** (instead of trichotomous):

  $$(\forall a. a<b \leftrightarrow a<c)\to b=c$$

What are ordinals?

One answer: **Numbers** for counting/ordering:

$$0, \quad 1, \quad 2, \quad 3, \quad \ldots, \quad \omega, \quad \omega+1, \quad \omega+2, \quad \ldots$$

$$\omega \cdot 2, \quad \omega \cdot 2+1, \quad \ldots, \quad \omega^2, \quad \ldots, \quad \omega^2 \cdot 3+\omega \cdot 7+13, \quad \ldots,$$

$$\omega^\omega, \quad \ldots, \quad \varepsilon_0 = \omega^{\omega^{\omega^{\cdots}}}, \quad \ldots, \quad \varepsilon_{17}, \quad \ldots, \quad \omega_1, \quad \ldots$$

Another answer: **Sets with an order** $<$ which is

- **transitive**: $(a < b) \to (b < c) \to (a < c)$
- **wellfounded**: every sequence $a_0 > a_1 > a_2 > a_3 > \ldots$ terminates
- and **trichotomous**: $(a < b) \vee (a = b) \vee (b < a)$
- …or **extensional** (instead of trichotomous):
  $$(\forall a. a < b \leftrightarrow a < c) \to b = c$$

What are ordinals?

One answer: **Numbers** for counting/ordering:

$$0, \quad 1, \quad 2, \quad 3, \quad \ldots, \quad \omega, \quad \omega+1, \quad \omega+2, \quad \ldots$$

$$\omega \cdot 2, \quad \omega \cdot 2+1, \quad \ldots, \quad \omega^2, \quad \ldots, \quad \omega^2 \cdot 3+\omega \cdot 7+13, \quad \ldots,$$

$$\omega^\omega, \quad \ldots, \quad \varepsilon_0 = \omega^{\omega^{\omega^{\cdots}}}, \quad \ldots, \quad \varepsilon_{17}, \quad \ldots, \quad \omega_1, \quad \ldots$$

Another answer: **Sets with an order** $<$ which is

- **transitive**: $(a < b) \to (b < c) \to (a < c)$
- **wellfounded**: every sequence $a_0 > a_1 > a_2 > a_3 > \ldots$ terminates
- and **trichotomous**: $(a < b) \vee (a = b) \vee (b < a)$
- . . . or **extensional** (instead of trichotomous):
  $$(\forall a. a < b \leftrightarrow a < c) \to b = c$$

What are ordinals?

One answer: **Numbers** for counting/ordering:

$$0, \quad 1, \quad 2, \quad 3, \quad \ldots, \quad \omega, \quad \omega+1, \quad \omega+2, \quad \ldots$$

$$\omega\cdot 2, \quad \omega\cdot 2+1, \quad \ldots, \quad \omega^2, \quad \ldots, \quad \omega^2\cdot 3+\omega\cdot 7+13, \quad \ldots,$$

$$\omega^\omega, \quad \ldots, \quad \varepsilon_0 = \omega^{\omega^{\omega^{\cdots}}}, \quad \ldots, \quad \varepsilon_{17}, \quad \ldots, \quad \omega_1, \quad \ldots$$

Another answer: **Sets with an order** $<$ which is

- **transitive**: $(a<b) \to (b<c) \to (a<c)$
- **wellfounded**: every sequence $a_0 > a_1 > a_2 > a_3 > \ldots$ terminates
- and **trichotomous**: $(a<b) \vee (a=b) \vee (b<a)$
- …or **extensional** (instead of trichotomous): $(\forall a. a<b \leftrightarrow a<c) \to b=c$

What are ordinals?

One answer: **Numbers** for counting/ordering:

$0, \quad 1, \quad 2, \quad 3, \quad \ldots, \quad \omega, \quad \omega+1, \quad \omega+2, \quad \ldots$

$\omega \cdot 2, \quad \omega \cdot 2$ $\ldots,$

$\omega^\omega, \quad \ldots, \quad \varepsilon$

Another answer:

- **transitive**:
- **wellfounded** $_3 > \ldots$ terminates
- and **tricho**
- …or **extensional** (instead of trichotomous):
  $(\forall a. a < b \leftrightarrow a < c) \to b = c$

What are ordinals good for?

CS standard applications:

- proving termination of processes
  (fun example: *Hydra game*)
- justifying recursive definitions /
  (why does the Ackermann
  function terminate?)
- …

What are ordinals?

One answer: **Numbers** for counting/ordering:

$$0,\quad 1,\quad 2,\quad 3,\quad \ldots,\quad \omega,\quad \omega+1,\quad \omega+2,\quad \ldots$$

$$\omega\cdot 2,\quad \omega\cdot 2+1,\quad \ldots,\quad \omega^2,\quad \ldots,\quad \omega^2\cdot 3+\omega\cdot 7+13,\quad \ldots,$$

$$\omega^\omega,\quad \ldots,\quad \varepsilon_0 = \omega^{\omega^{\omega^{\cdots}}},\quad \ldots,\quad \varepsilon_{17},\quad \ldots,\quad \omega_1,\quad \ldots$$

not possible constructively

easy

Another answer: **Sets with an order** $<$ which is

- **transitive**: $(a<b) \to (b<c) \to (a<c)$
- **wellfounded**: every sequence $a_0 > a_1 > a_2 > a_3 > \ldots$ terminates
- and **trichotomous**: $(a<b) \vee (a=b) \vee (b<a)$
- ... or **extensional** (instead of trichotomous):

$$(\forall a. a<b \leftrightarrow a<c) \to b=c$$

Ordinals in dependent type theory

⇒ **Problem/feature** of a constructive setting: different definitions differ!

Three standard notions of "ordinals" in computer science:

- Cantor normal forms
- Brouwer trees
- Wellfounded & extensional & transitive orders

What's the connection? Why can we call them "ordinals"?

Developments in this paper:

(i) axiomatic framework for ordinals and ordinal arithmetic
(ii) "correct" formulation of Brouwer trees (quotient inductive-inductively)
(iii) connections between the three notions and their arithmetic operations

Ordinals in dependent type theory

Problem/feature of a constructive setting: different definitions differ!

Three standard notions of "ordinals" in computer science:

- Cantor normal forms
- Brouwer trees
- Wellfounded & extensional & transitive orders

What's the connection? Why can we call them "ordinals"?

Developments in this paper:

(i) axiomatic framework for ordinals and ordinal arithmetic

(ii) "correct" formulation of Brouwer trees (quotient inductive-inductively)

(iii) connections between the three notions and their arithmetic operations

Ordinals in dependent type theory

Problem/feature of a constructive setting: different definitions differ!

Three standard notions of "ordinals" in computer science:

- Cantor normal forms
- Brouwer trees
- Wellfounded & extensional & transitive orders

What's the connection? Why can we call them "ordinals"?

Developments in this paper:

(i) axiomatic framework for ordinals and ordinal arithmetic
(ii) "correct" formulation of Brouwer trees (quotient inductive-inductively)
(iii) connections between the three notions and their arithmetic operations

Ordinals in dependent type theory

Problem/feature of a constructive setting: different definitions differ!

Three standard notions of "ordinals" in computer science:

- Cantor normal forms
- Brouwer trees
- Wellfounded & extensional & transitive orders

What's the connection? Why can we call them "ordinals"?

Developments in this paper:

(i) axiomatic framework for ordinals and ordinal arithmetic

(ii) "correct" formulation of Brouwer trees (quotient inductive-inductively)

(iii) connections between the three notions and their arithmetic operations

Ordinals in dependent type theory

Problem/feature of a constructive setting: different definitions differ!

Three standard notions of "ordinals" in computer science:

- Cantor normal forms
- Brouwer trees
- Wellfounded & extensional & transitive orders

$\Rrightarrow$ What's the connection? Why can we call them "ordinals"?

Developments in this paper:

(i) axiomatic framework for ordinals and ordinal arithmetic
(ii) "correct" formulation of Brouwer trees (quotient inductive-inductively)
(iii) connections between the three notions and their arithmetic operations

Ordinals in dependent type theory

Problem/feature of a constructive setting: different definitions differ!

Three standard notions of “ordinals” in computer science:

- Cantor normal forms
- Brouwer trees
- Wellfounded & extensional & transitive orders

What’s the connection? Why can we call them “ordinals”?

Developments in this paper:

!

(i) axiomatic framework for ordinals and ordinal arithmetic
(ii) “correct” formulation of Brouwer trees (quotient inductive-inductively)
(iii) connections between the three notions and their arithmetic operations

What do we expect of "ordinals"?

When does $(\mathcal{O}, <)$ deserve to be called "ordinals"?

(a) **Wellfoundedness**: Every decreasing sequence terminates / can do transfinite induction.

What do we expect of “ordinals”?
When does $(\mathcal{O}, <)$ deserve to be called “ordinals”?

(a) **Wellfoundedness**: Every decreasing sequence terminates / can do transfinite induction.

(b) **Arithmetic**: Can do addition, multiplication, exponentiation, ... (But what does that mean?)

What do we expect of "ordinals"?
When does $(\mathcal{O}, <)$ deserve to be called "ordinals"?

(a) **Wellfoundedness**: Every decreasing sequence terminates / can do transfinite induction.

(b) **Arithmetic**: Can do addition, multiplication, exponentiation, ... (But what does that mean?)

(c) **Trichotomy**: $(a < b) \vee (a = b) \vee (b < a)$

What do we expect of "ordinals"?
When does $(\mathcal{O}, <)$ deserve to be called "ordinals"?

(a) **Wellfoundedness**: Every decreasing sequence terminates / can do transfinite induction.

(b) **Arithmetic**: Can do addition, multiplication, exponentiation, ... (But what does that mean?)

(c) **Trichotomy**: $(a < b) \vee (a = b) \vee (b < a)$ not necessarily!

What do we expect of "ordinals"?
When does $(\mathcal{O}, <)$ deserve to be called "ordinals"?

(a) **Wellfoundedness**: Every decreasing sequence terminates / can do transfinite induction.

(b) **Arithmetic**: Can do addition, multiplication, exponentiation, ... (But what does that mean?)

(c) **Trichotomy**: $(a < b) \vee (a = b) \vee (b < a)$

(d) **Extensionality**: $(\forall a. a < b \leftrightarrow a < c) \to b = c$

What do we expect of "ordinals"? When does $(\mathcal{O}, <)$ deserve to be called "ordinals"?

(a) **Wellfoundedness**: Every decreasing sequence terminates / can do transfinite induction.

(b) **Arithmetic**: Can do addition, multiplication, exponentiation, ... (But what does that mean?)

(c) **Trichotomy**: $(a < b) \vee (a = b) \vee (b < a)$

(d) **Extensionality**: $(\forall a. a < b \leftrightarrow a < c) \to b = c$

(e) **Suprema/limits**: Given a sequence, we can calculate its limit.

What do we expect of "ordinals"?
When does $(\mathcal{O}, <)$ deserve to be called "ordinals"?

(a) **Wellfoundedness**: Every decreasing sequence terminates / can do transfinite induction.

(b) **Arithmetic**: Can do addition, multiplication, exponentiation, ... (But what does that mean?)

(c) **Trichotomy**: $(a < b) \vee (a = b) \vee (b < a)$

(d) **Extensionality**: $(\forall a. a < b \leftrightarrow a < c) \to b = c$

(e) **Suprema/limits**: Given a sequence, we can calculate its limit.

not necessarily!

What do we expect of “ordinals”?
When does $(\mathcal{O}, <)$ deserve to be called “ordinals”?

(a) **Wellfoundedness**: Every decreasing sequence terminates / can do transfinite induction.

(b) **Arithmetic**: Can do addition, multiplication, exponentiation, ... (But what does that mean?)

(c) **Trichotomy**: $(a < b) \vee (a = b) \vee (b < a)$

(d) **Extensionality**: $(\forall a. a < b \leftrightarrow a < c) \to b = c$

(e) **Suprema/limits**: Given a sequence, we can calculate its limit.

$\boxed{\text{(f) \textbf{Classifiability}: If } x : \mathcal{O}\text{, then } x \text{ is a zero, a successor, or a limit.}}$

What do we expect of "ordinals"?
When does $(\mathcal{O}, <)$ deserve to be called "ordinals"?

(a) **Wellfoundedness**: Every decreasing sequence terminates / can do transfinite induction.

(b) **Arithmetic**: Can do addition, multiplication, exponentiation, ... (But what does that mean?)

(c) **Trichotomy**: $(a < b) \vee (a = b) \vee (b < a)$

(d) **Extensionality**: $(\forall a.a < b \leftrightarrow a < c) \rightarrow b = c$

(e) **Suprema/limits**: Given a sequence, we can calculate its limit.

$\boxed{\text{(f) } \textbf{Classifiability}\text{: If } x : \mathcal{O}\text{, then } x \text{ is a zero, a successor, or a limit.}}$

not necessarily!

Cantor normal forms

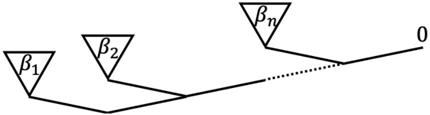
Motivation: $\alpha = \omega^{\beta_1} + \omega^{\beta_2} + \cdots + \omega^{\beta_n}$ with $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_n$

Cantor normal forms

Motivation: $\alpha = \omega^{\beta_1} + \omega^{\beta_2} + \cdots + \omega^{\beta_n}$ with $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_n$

Let $\mathcal{T}$ be the type of *unlabeled binary trees*:

$$\begin{array}{lcl} 0 & : & \mathcal{T} \\ \omega^{-} + - & : & \mathcal{T} \to \mathcal{T} \to \mathcal{T} \end{array}$$

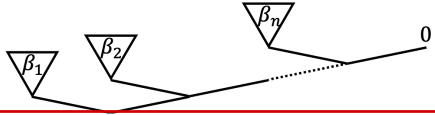
$\alpha =$ 

Cantor normal forms

Motivation: $\alpha = \omega^{\beta_1} + \omega^{\beta_2} + \cdots + \omega^{\beta_n}$ with $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_n$

Let $\mathcal{T}$ be the type of *unlabeled binary trees*:

$$\begin{aligned} 0 \quad &: \mathcal{T} \\ \omega^{-} + - \quad &: \mathcal{T} \to \mathcal{T} \to \mathcal{T} \end{aligned}$$

$\alpha =$ 

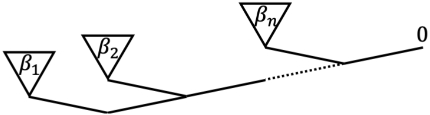
$\boxed{\text{A tree is a } \textit{Cantor normal form} \text{ if } \beta_1 \geq \beta_2 \geq \cdots \geq \beta_n \text{ (}\textit{lexicographical order}\text{).}}$

Cantor normal forms

Motivation: $\alpha = \omega^{\beta_1} + \omega^{\beta_2} + \cdots + \omega^{\beta_n}$ with $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_n$

Let $\mathcal{T}$ be the type of *unlabeled binary trees*:

$$\begin{aligned} 0 \quad &: \mathcal{T} \\ \omega^{-} + - \quad &: \mathcal{T} \to \mathcal{T} \to \mathcal{T} \end{aligned}$$

$\alpha =$ 

A tree is a *Cantor normal form* if $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_n$ (*lexicographical order*).

Cannot calculate limits of sequences, but everything else works
– including continuity of arithmetic operations!

Brouwer trees (a.k.a. Brouwer ordinal trees)

How about this inductive type $\mathcal{O}$ of Brouwer trees?

$$\mathsf{zero} : \mathcal{O} \qquad \mathsf{succ} : \mathcal{O} \to \mathcal{O} \qquad \mathsf{sup} : (\mathbb{N} \to \mathcal{O}) \to \mathcal{O}$$

Brouwer trees (a.k.a. Brouwer ordinal trees)

How about this inductive type $\mathcal{O}$ of Brouwer trees?

$$\mathsf{zero} : \mathcal{O} \qquad \mathsf{succ} : \mathcal{O} \to \mathcal{O} \qquad \boxed{\mathsf{sup} : (\mathbb{N} \to \mathcal{O}) \to \mathcal{O}}$$

Brouwer trees (a.k.a. Brouwer ordinal trees)

How about this inductive type $\mathcal{O}$ of Brouwer trees?

$$\mathsf{zero} : \mathcal{O} \qquad \mathsf{succ} : \mathcal{O} \to \mathcal{O} \qquad \mathsf{sup} : (\mathbb{N} \to \mathcal{O}) \to \mathcal{O}$$

Problem: $\mathsf{sup}(0, 1, 2, 3, \ldots) \neq \mathsf{sup}(1, 2, 3, \ldots)$

How to fix this without losing wellfoundedness, validity of arithmetic operations, and so on?

Brouwer trees quotient inductive-inductively

(cubical Agda)

```
data Brw where
  zero  : Brw
  succ  : Brw → Brw
  limit : (f : ℕ → Brw) → {f↑ : increasing f} → Brw
  bisim : ∀ f {f↑} g {g↑} →
          f ≈ g →
          limit f {f↑} ≡ limit g {g↑}
  trunc : isSet Brw

data _≤_ where
  ≤-zero      : ∀ {x} → zero ≤ x
  ≤-trans     : ∀ {x y z} → x ≤ y → y ≤ z → x ≤ z
  ≤-succ-mono : ∀ {x y} → x ≤ y → succ x ≤ succ y
  ≤-cocone    : ∀ {x} f {f↑ k} → (x ≤ f k) → (x ≤ limit f {f↑})
  ≤-limiting  : ∀ f {f↑ x} → ((k : ℕ) → f k ≤ x) → limit f {f↑} ≤ x
  ≤-trunc     : ∀ {x y} → isProp (x ≤ y)
```

Brouwer trees quotient inductive-inductively

(cubical Agda)

```
data Brw where
  zero  : Brw
  succ  : Brw → Brw
  limit : (f : ℕ → Brw) → {f↑ : increasing f} → Brw
  bisim : ∀ f {f↑} g {g↑} →
          f ≈ g →
          limit f {f↑} ≡ limit g {g↑}
  trunc : isSet Brw

data _≤_ where
  ≤-zero      : ∀ {x} → zero ≤ x
  ≤-trans     : ∀ {x y z} → x ≤ y → y ≤ z → x ≤ z
  ≤-succ-mono : ∀ {x y} → x ≤ y → succ x ≤ succ y
  ≤-cocone    : ∀ {x} f {f↑ k} → (x ≤ f k) → (x ≤ limit f {f↑})
  ≤-limiting  : ∀ f {f↑ x} → ((k : ℕ) → f k ≤ x) → limit f {f↑} ≤ x
  ≤-trunc     : ∀ {x y} → isProp (x ≤ y)
```

Brouwer trees quotient inductive-inductively

(cubical Agda)

```
data Brw where
  zero  : Brw
  succ  : Brw → Brw
  limit : (f : ℕ → Brw) → {f↑ : increasing f} → Brw
  bisim : ∀ f {f↑} g {g↑} →
          f ≈ g →
          limit f {f↑} ≡ limit g {g↑}
  trunc : isSet Brw

data _≤_ where
  ≤-zero      : ∀ {x} → zero ≤ x
  ≤-trans     : ∀ {x y z} → x ≤ y → y ≤ z → x ≤ z
  ≤-succ-mono : ∀ {x y} → x ≤ y → succ x ≤ succ y
  ≤-cocone    : ∀ {x} f {f↑ k} → (x ≤ f k) → (x ≤ limit f {f↑})
  ≤-limiting  : ∀ f {f↑ x} → ((k : ℕ) → f k ≤ x) → limit f {f↑} ≤ x
  ≤-trunc     : ∀ {x y} → isProp (x ≤ y)
```

Brouwer trees quotient inductive-inductively

(cubical Agda)

```
data Brw where
  zero  : Brw
  succ  : Brw → Brw
  limit : (f : ℕ → Brw) → {f↑ : increasing f} → Brw
  bisim : ∀ f {f↑} g {g↑} →
          f ≈ g →
          limit f {f↑} ≡ limit g {g↑}
  trunc : isSet Brw

data _≤_ where
  ≤-zero      : ∀ {x} → zero ≤ x
  ≤-trans     : ∀ {x y z} → x ≤ y → y ≤ z → x ≤ z
  ≤-succ-mono : ∀ {x y} → x ≤ y → succ x ≤ succ y
  ≤-cocone    : ∀ {x} f {f↑ k} → (x ≤ f k) → (x ≤ limit f {f↑})
  ≤-limiting  : ∀ f {f↑ x} → ((k : ℕ) → f k ≤ x) → limit f {f↑} ≤ x
  ≤-trunc     : ∀ {x y} → isProp (x ≤ y)
```

Brouwer trees quotient inductive-inductively

(cubical Agda)

```
data Brw where
  zero  : Brw
  succ  : Brw → Brw
  limit : (f : ℕ → Brw) → {f↑ : increasing f} → Brw
  bisim : ∀ f {f↑} g {g↑} →
          f ≈ g →
          limit f {f↑} ≡ limit g {g↑}
  trunc : isSet Brw

data _≤_ where
  ≤-zero      : ∀ {x} → zero ≤ x
  ≤-trans     : ∀ {x y z} → x ≤ y → y ≤ z → x ≤ z
  ≤-succ-mono : ∀ {x y} → x ≤ y → succ x ≤ succ y
  ≤-cocone    : ∀ {x} f {f↑ k} → (x ≤ f k) → (x ≤ limit f {f↑})
  ≤-limiting  : ∀ f {f↑ x} → ((k : ℕ) → f k ≤ x) → limit f {f↑} ≤ x
  ≤-trunc     : ∀ {x y} → isProp (x ≤ y)
```

Brouwer trees quotient inductive-inductively

(cubical Agda)

```
data Brw where
  zero  : Brw
  succ  : Brw → Brw
  limit : (f : ℕ → Brw) → {f↑ : increasing f} → Brw
  bisim : ∀ f {f↑} g {g↑} →
          f ≈ g →
          limit f {f↑} ≡ limit g {g↑}
  trunc : isSet Brw

data _≤_ where
  ≤-zero      : ∀ {x} → zero ≤ x
  ≤-trans     : ∀ {x y z} → x ≤ y → y ≤ z → x ≤ z
  ≤-succ-mono : ∀ {x y} → x ≤ y → succ x ≤ succ y
  ≤-cocone    : ∀ {x} f {f↑ k} → (x ≤ f k) → (x ≤ limit f {f↑})
  ≤-limiting  : ∀ f {f↑ x} → ((k : ℕ) → f k ≤ x) → limit f {f↑} ≤ x
  ≤-trunc     : ∀ {x y} → isProp (x ≤ y)
```

Brouwer trees quotient inductive-inductively

(cubical Agda)

```
data Brw where
  zero  : Brw
  succ  : Brw → Brw
  limit : (f : ℕ → Brw) → {f↑ : increasing f} → Brw
  bisim : ∀ f {f↑} g {g↑} →
          f ≈ g →
          limit f {f↑} ≡ limit g {g↑}
  trunc : isSet Brw

data _≤_ where
  ≤-zero      : ∀ {x} → zero ≤ x
  ≤-trans     : ∀ {x y z} → x ≤ y → y ≤ z → x ≤ z
  ≤-succ-mono : ∀ {x y} → x ≤ y → succ x ≤ succ y
  ≤-cocone    : ∀ {x} f {f↑ k} → (x ≤ f k) → (x ≤ limit f {f↑})
  ≤-limiting  : ∀ f {f↑ x} → ((k : ℕ) → f k ≤ x) → limit f {f↑} ≤ x
  ≤-trunc     : ∀ {x y} → isProp (x ≤ y)
```

Brouwer trees quotient inductive-inductively

(cubical Agda)

```
data Brw where
  zero  : Brw
  succ  : Brw → Brw
  limit : (f : ℕ → Brw) → {f↑ : increasing f} → Brw
  bisim : ∀ f {f↑} g {g↑} →
          f ≈ g →
          limit f {f↑} ≡ limit g {g↑}
  trunc : isSet Brw

data _≤_ where
  ≤-zero      : ∀ {x} → zero ≤ x
  ≤-trans     : ∀ {x y z} → x ≤ y → y ≤ z → x ≤ z
  ≤-succ-mono : ∀ {x y} → x ≤ y → succ x ≤ succ y
  ≤-cocone    : ∀ {x} f {f↑ k} → (x ≤ f k) → (x ≤ limit f {f↑})
  ≤-limiting  : ∀ f {f↑ x} → ((k : ℕ) → f k ≤ x) → limit f {f↑} ≤ x
  ≤-trunc     : ∀ {x y} → isProp (x ≤ y)
```

Not trichotomous ($a < b$ undecidable), everything else works – notably wellfoundedness and arithmetic.

Brouwer trees quotient inductive-inductively

(cubical Agda)

```
data Brw where
  zero  : Brw
  succ  : Brw → Brw
  limit : (f : ℕ → Brw) → {f↑ : increasing f} → Brw
  bisim : ∀ f {f↑} g {g↑} →
          f ≈ g →
          limit f {f↑} ≡ limit g {g↑}
  trunc : isSet Brw

data _≤_ where
  ≤-zero      : ∀ {x} → zero ≤ x
  ≤-trans     : ∀ {x y z} → x ≤ y → y ≤ z → x ≤ z
  ≤-succ-mono : ∀ {x y} → x ≤ y → succ x ≤ succ y
  ≤-cocone    : ∀ {x} f {f↑ k} → (x ≤ f k) → (x ≤ limit f {f↑})
  ≤-limiting  : ∀ f {f↑ x} → ((k : ℕ) → f k ≤ x) → limit f {f↑} ≤ x
  ≤-trunc     : ∀ {x y} → isProp (x ≤ y)
```

Caveat: Cannot define $a \cdot \mathrm{limit}(f_i) = \mathrm{limit}(a \cdot f_i)$

Not trichotomous ($a < b$ undecidable), everything else works – notably wellfoundedness and arithmetic.

Extensional wellfounded orders

Definition of type Ord:

Pairs $(X : \mathsf{Type}, \prec : X \to X \to \mathsf{Prop})$ such that $\prec$ is transitive, extensional, wellfounded.

$(X, \prec_X) \leq (Y, \prec_Y)$ is given by:

A *monotone* function $f : X \to Y$
such that: if $y \prec_Y f\,x$, then there is $x_0 \prec_X x$ such that $f\,x_0 = y$.

Extensional wellfounded orders

Definition of type Ord:

Pairs $(X : \mathsf{Type}, \prec : X \to X \to \mathsf{Prop})$ such that $\prec$ is transitive, extensional, wellfounded.

$(X, \prec_X) \leq (Y, \prec_Y)$ is given by:

A *monotone* function $f : X \to Y$
such that: if $y \prec_Y f\,x$, then there is $x_0 \prec_X x$ such that $f\,x_0 = y$.

Extensional wellfounded orders

Definition of type Ord:

Pairs $(X : \text{Type}, \prec : X \to X \to \text{Prop})$ such that $\prec$ is transitive, extensional, wellfounded.

$(X, \prec_X) \leq (Y, \prec_Y)$ is given by:

A *monotone* function $f : X \to Y$
such that: if $y \prec_Y f\, x$, then there is $x_0 \prec_X x$ such that $f\, x_0 = y$.

$(\text{Ord}, <)$ is extensional and wellfounded, we has addition and multiplication, we can calculate limits.

Extensional wellfounded orders

Definition of type Ord:

Pairs $(X : \mathsf{Type}, \prec : X \to X \to \mathsf{Prop})$ such that $\prec$ is transitive, extensional, wellfounded.

$(X, \prec_X) \leq (Y, \prec_Y)$ is given by:

A *monotone* function $f : X \to Y$
such that: if $y \prec_Y f\,x$, then there is $x_0 \prec_X x$ such that $f\,x_0 = y$.

$(\mathsf{Ord}, <)$ is extensional and wellfounded, we has addition and multiplication, we can calculate limits.

Unsurprisingly, nothing is decidable. E.g. deciding whether $x : \mathsf{Ord}$ is a successor implies LEM (in the HoTT sense).

Connections between the notions

decidable | partially decidable | undecidable

$$\boxed{\mathsf{Cnf}} \xhookrightarrow{\mathsf{CtoB}} \boxed{\mathsf{Brw}} \xhookrightarrow{\mathsf{BtoO}} \boxed{\mathsf{Ord}}$$

$(\omega^a + b) \mapsto \omega^{\mathsf{CtoB}(a)} + \mathsf{CtoB}(b)$

$A \mapsto \Sigma(Y : \mathsf{Brw}).Y < A$

- injective
- preserves and reflects $<$, $\leq$
- commutes with $+$, $*$, ω^x
- bounded (by ϵ_0)

- injective
- preserves $<$, $\leq$
- over-approximates $+$, $*$:
  $\mathsf{BtoO}(x + y) \geq \mathsf{BtoO}(x) + \mathsf{BtoO}(y)$
- commutes with limits
  (but not successors)
- BtoO is a simulation $\Rightarrow$ WLPO
- LEM $\Rightarrow$ BtoO is a simulation
- bounded (by Brw)

Connections between the notions

$$\begin{array}{ccccc} \text{decidable} & & \text{partially decidable} & & \text{undecidable} \\ \boxed{\mathsf{Cnf}} & \xhookrightarrow{\mathsf{CtoB}} & \boxed{\mathsf{Brw}} & \xhookrightarrow{\mathsf{BtoO}} & \boxed{\mathsf{Ord}} \end{array}$$

$(\omega^a + b) \mapsto \omega^{\mathsf{CtoB}(a)} + \mathsf{CtoB}(b)$

$A \mapsto \Sigma(Y : \mathsf{Brw}).Y < A$

- injective
- preserves and reflects $<$, $\leq$
- commutes with $+$, $*$, ω^x
- bounded (by ϵ_0)

- injective
- preserves $<$, $\leq$
- over-approximates $+$, $*$: $\mathsf{BtoO}(x+y) \geq \mathsf{BtoO}(x) + \mathsf{BtoO}(y)$
- commutes with limits (but not successors)
- BtoO is a simulation $\Rightarrow$ WLPO
- LEM $\Rightarrow$ BtoO is a simulation
- bounded (by Brw)

Connections between the notions

$$\begin{array}{ccccc} \text{decidable} & & \text{partially decidable} & & \text{undecidable} \\ \boxed{\mathsf{Cnf}} & \xhookrightarrow{\mathsf{CtoB}} & \boxed{\mathsf{Brw}} & \xhookrightarrow{\mathsf{BtoO}} & \boxed{\mathsf{Ord}} \end{array}$$

$(\omega^a + b) \mapsto \omega^{\mathsf{CtoB}(a)} + \mathsf{CtoB}(b)$

$A \mapsto \Sigma(Y : \mathsf{Brw}).Y < A$

- injective
- preserves and reflects $<$, $\leq$
- commutes with $+$, $*$, ω^x
- bounded (by ϵ_0)

- injective
- preserves $<$, $\leq$
- over-approximates $+$, $*$:
  $\mathsf{BtoO}(x+y) \geq \mathsf{BtoO}(x) + \mathsf{BtoO}(y)$
- commutes with limits
  (but not successors)
- BtoO is a simulation $\Rightarrow$ WLPO
- LEM $\Rightarrow$ BtoO is a simulation
- bounded (by Brw)

Connections between the notions

$$\underset{\text{decidable}}{\boxed{\text{Cnf}}} \xhookrightarrow[(\omega^a + b) \mapsto \omega^{\text{CtoB}(a)} + \text{CtoB}(b)]{\text{CtoB}} \underset{\text{partially decidable}}{\boxed{\text{Brw}}} \xhookrightarrow[A \mapsto \Sigma(Y : \text{Brw}).Y < A]{\text{BtoO}} \underset{\text{undecidable}}{\boxed{\text{Ord}}}$$

- injective
- preserves and reflects $<$, $\leq$
- commutes with $+$, $*$, ω^x
- bounded (by ϵ_0)

- injective
- preserves $<$, $\leq$
- over-approximates $+$, $*$:
  $\text{BtoO}(x + y) \geq \text{BtoO}(x) + \text{BtoO}(y)$
- commutes with limits
  (but not successors)
- BtoO is a simulation $\Rightarrow$ WLPO
- LEM $\Rightarrow$ BtoO is a simulation
- bounded (by Brw)

Connections between the notions

$$\begin{array}{ccccc} \text{decidable} & & \text{partially decidable} & & \text{undecidable} \\ \boxed{\mathsf{Cnf}} & \xhookrightarrow[(\omega^a + b) \mapsto \omega^{\mathsf{CtoB}(a)} + \mathsf{CtoB}(b)]{\mathsf{CtoB}} & \boxed{\mathsf{Brw}} & \xhookrightarrow[A \mapsto \Sigma(Y : \mathsf{Brw}).Y < A]{\mathsf{BtoO}} & \boxed{\mathsf{Ord}} \end{array}$$

- injective
- preserves and reflects $<$, $\leq$
- commutes with $+$, $*$, ω^x
- bounded (by ϵ_0)

- injective
- preserves $<$, $\leq$
- over-approximates $+$, $*$: $\mathsf{BtoO}(x + y) \geq \mathsf{BtoO}(x) + \mathsf{BtoO}(y)$
- commutes with limits (but not successors)
- BtoO is a simulation $\Rightarrow$ WLPO
- LEM $\Rightarrow$ BtoO is a simulation
- bounded (by Brw)