Ordinal Exponentiation in Homotopy Type Theory

Tom de Jong[1] Nicolai Kraus[1] Fredrik Nordvall Forsberg[2] Chuangjie Xu

[1]University of Nottingham, UK
[2]University of Strathclyde, UK

Dutch Categories And Types Seminar (DutchCATS)

7 February 2025

Motivation and background

- Classically, ordinals have powerful applications as tools for e.g. establishing consistency of logical theories, proving termination of processes, and justifying induction and recursion.

Motivation and background

- Classically, ordinals have powerful applications as tools for e.g. establishing consistency of logical theories, proving termination of processes, and justifying induction and recursion.

- Can we develop the theory of ordinals constructively?
  In particular, can we give a satisfactory account of ordinal arithmetic and specifically of ordinal exponentiation?

Motivation and background

- Classically, ordinals have powerful applications as tools for e.g. establishing consistency of logical theories, proving termination of processes, and justifying induction and recursion.
- Can we develop the theory of ordinals constructively?
  In particular, can we give a satisfactory account of ordinal arithmetic and specifically of ordinal exponentiation?
- We work in homotopy type theory (HoTT) following foundational work in the **HoTT Book** and in the **TypeTopology** Agda development by Martín Escardó.

The problem(s)

- Classically, ordinal exponentiation is usually defined by **inspecting whether the exponent is zero, a successor, or a limit ordinal**:

$$
\begin{aligned}
\alpha^0 &= 1 & \qquad 0^\beta &= 0 & \text{(if } \beta \neq 0\text{)} \\
\alpha^{\beta+1} &= \alpha^\beta \times \alpha & \qquad \alpha^\lambda &= \sup_{\beta<\lambda} \alpha^\beta & \text{(if } \lambda \text{ is a limit, } \alpha \neq 0\text{)}
\end{aligned}
$$

The problem(s)

- Classically, ordinal exponentiation is usually defined by **inspecting whether the exponent is zero, a successor, or a limit ordinal**:

$$
\begin{aligned}
\alpha^0 &= 1 & 0^\beta &= 0 && (\text{if } \beta \neq 0) \\
\alpha^{\beta+1} &= \alpha^\beta \times \alpha & \alpha^\lambda &= \sup_{\beta<\lambda} \alpha^\beta && (\text{if } \lambda \text{ is a limit, } \alpha \neq 0)
\end{aligned}
$$

- Such a case distinction is only possible classically: it can be done for all ordinals if and only if excluded middle holds.

The problem(s)

- Classically, ordinal exponentiation is usually defined by **inspecting whether the exponent is zero, a successor, or a limit ordinal**:

$$\alpha^0 = 1 \qquad 0^\beta = 0 \qquad \text{(if } \beta \neq 0\text{)}$$
$$\alpha^{\beta+1} = \alpha^\beta \times \alpha \qquad \alpha^\lambda = \sup_{\beta<\lambda} \alpha^\beta \qquad \text{(if } \lambda \text{ is a limit, } \alpha \neq 0\text{)}$$

- Such a case distinction is only possible classically: it can be done for all ordinals if and only if excluded middle holds.

- Thm. (Imprecise) There is a well behaved exponentiation function on *all* ordinals if and only if excluded middle holds.

The highlights

- Working **constructively** in HoTT, we construct two well behaved ordinal exponentiation functions $\alpha^{(-)}$ with a minor condition on the base ordinal α:

The highlights

- Working **constructively** in HoTT, we construct two well behaved ordinal exponentiation functions $\alpha^{(-)}$ with a minor condition on the base ordinal α:
  - The first construction is abstract, uses **suprema of ordinals**, and is motivated by the expected equations.

    It is well defined whenever $\alpha \geq \mathbf{1}$, i.e. whenever α has a least element.

The highlights

- Working **constructively** in HoTT, we construct two well behaved ordinal exponentiation functions $\alpha^{(-)}$ with a minor condition on the base ordinal α:
  - The first construction is abstract, uses **suprema of ordinals**, and is motivated by the expected equations.

    It is well defined whenever $\alpha \geq \mathbf{1}$, i.e. whenever α has a least element.
  - The second is more concrete, **based on decreasing lists**, and a constructive version of a construction by Sierpiński based on functions with finite support.

The highlights

- Working **constructively** in HoTT, we construct two well behaved ordinal exponentiation functions $\alpha^{(-)}$ with a minor condition on the base ordinal α:
  - The first construction is abstract, uses **suprema of ordinals**, and is motivated by the expected equations.

    It is well defined whenever $\alpha \geq \mathbf{1}$, i.e. whenever α has a least element.
  - The second is more concrete, **based on decreasing lists**, and a constructive version of a construction by Sierpiński based on functions with finite support.

    It is well defined whenever α has a trichotomous least element: the least element a_0 is further required to satisfy the decidability condition: $\forall(x : \alpha).(a_0 < x) + (a_0 = x)$.

The highlights

- Working **constructively** in HoTT, we construct two well behaved ordinal exponentiation functions $\alpha^{(-)}$ with a minor condition on the base ordinal α:
  - The first construction is abstract, uses **suprema of ordinals**, and is motivated by the expected equations.

    It is well defined whenever $\alpha \geq \mathbf{1}$, i.e. whenever α has a least element.
  - The second is more concrete, **based on decreasing lists**, and a constructive version of a construction by Sierpiński based on functions with finite support.

    It is well defined whenever α has a trichotomous least element: the least element a_0 is further required to satisfy the decidability condition: $\forall(x : \alpha).(a_0 < x) + (a_0 = x)$.
- We show that our two constructions agree (whenever the base ordinal has a trichotomous least element).

The highlights

- Working **constructively** in HoTT, we construct two well behaved ordinal exponentiation functions $\alpha^{(-)}$ with a minor condition on the base ordinal α:
  - The first construction is abstract, uses **suprema of ordinals**, and is motivated by the expected equations.

    It is well defined whenever $\alpha \geq \mathbf{1}$, i.e. whenever α has a least element.
  - The second is more concrete, **based on decreasing lists**, and a constructive version of a construction by Sierpiński based on functions with finite support.

    It is well defined whenever α has a trichotomous least element: the least element a_0 is further required to satisfy the decidability condition: $\forall (x : \alpha).(a_0 < x) + (a_0 = x)$.
- We show that our two constructions agree (whenever the base ordinal has a trichotomous least element).
- We use this equivalence together with **univalence (representation independence)** to prove algebraic laws and decidability properties.

Commercial break

- Our paper is available on `arXiv:2501.14542`.

arXiv:2501.14542v1 [cs.LO] 24 Jan 2025

ORDINAL EXPONENTIATION IN HOMOTOPY TYPE THEORY

TOM DE JONG, NICOLAI KRAUS, FREDRIK NORDVALL FORSBERG, AND CHUANGJIE XU

Abstract. While ordinals have traditionally been studied mostly in classical frameworks, constructive ordinal theory has seen significant progress in recent years. However, a general constructive treatment of ordinal exponentiation has thus far been missing.

We present two seemingly different definitions of constructive ordinal exponentiation in the setting of homotopy type theory. The first is abstract, uses suprema of ordinals, and is solely motivated by the expected equations. The second is more concrete, based on decreasing lists, and can be seen as a constructive version of a classical construction by Sierpiński based on functions with finite support. We show that our two approaches are equivalent (whenever it makes sense to ask the question), and use this equivalence to prove algebraic laws and decidability properties of the exponential.

All our results are formalized in the proof assistant Agda.

1. Introduction

In classical mathematics and set theory, ordinals have rich and interesting structure. How much of this structure can be developed in a constructive setting, such as homotopy type theory? This is not merely a question of mathematical curiosity, as classical ordinals have powerful applications as tools for establishing consistency of logical theories [Rat07], proving termination of processes [Flo67], and justifying induction and recursion [Acz77; DS99], which all would be valuable to have available in constructive mathematics and proof assistants based on constructive type theory. There are many constructive approaches to ordinals, such as ordinal notation systems [NXG20], Brouwer trees [Ere24], or wellfounded trees with finite or countable branchings [Mar70; CLN23], to name a few. In this paper, we follow the Homotopy Type Theory Book [Uni13] and consider ordinals as order types of well ordered sets, i.e., an ordinal is a type equipped with an order relation having certain properties.

Ordinals have a theory of arithmetic that generalizes the one of the natural numbers. Classically, arithmetic operations are defined by case distinction and transfinite recursion:

$$\begin{aligned} \alpha + 0 &= \alpha \\ \alpha + (\beta + 1) &= (\alpha + \beta) + 1 \\ \alpha + \lambda &= \mathsf{sup}_{\beta < \lambda}\,(\alpha + \beta) && \text{(if } \lambda \text{ is a limit)} \end{aligned}$$

$$\begin{aligned} \alpha \times 0 &= 0 \\ \alpha \times (\beta + 1) &= (\alpha \times \beta) + \alpha \\ \alpha \times \lambda &= \mathsf{sup}_{\beta < \lambda}\,(\alpha \times \beta) && \text{(if } \lambda \text{ is a limit)} \end{aligned}$$

Key words and phrases. constructive mathematics, homotopy type theory, ordinal arithmetic, Agda formalization.

Commercial break

- Our paper is available on `arXiv:2501.14542`.

ORDINAL EXPONENTIATION IN HOMOTOPY TYPE THEORY

TOM DE JONG, NICOLAI KRAUS, FREDRIK NORDVALL FORSBERG, AND CHUANGJIE XU

Abstract. While ordinals have traditionally been studied mostly in classical frameworks, constructive ordinal theory has seen significant progress in recent years. However, a general constructive treatment of ordinal exponentiation has thus far been missing.

We present two seemingly different definitions of constructive ordinal exponentiation in the setting of homotopy type theory. The first is abstract, uses suprema of ordinals, and is solely motivated by the expected equations. The second is more concrete, based on decreasing lists, and can be seen as a constructive version of a classical construction by Sierpiński based on functions with finite support. We show that our two approaches are equivalent (whenever it makes sense to ask the question), and use this equivalence to prove algebraic laws and decidability properties of the exponential.

All our results are formalized in the proof assistant Agda.

1. Introduction

In classical mathematics and set theory, ordinals have rich and interesting structure. How much of this structure can be developed in a constructive setting, such as homotopy type theory? This is not merely a question of mathematical curiosity, as classical ordinals have powerful applications as tools for establishing consistency of logical theories [Rat07], proving termination of processes [Flo67], and justifying induction and recursion [Acz77; DS99], which all would be valuable to have available in constructive mathematics and proof assistants based on constructive type theory. There are many constructive approaches to ordinals, such as ordinal notation systems [NXG20], Brouwer trees [Ere24], or wellfounded trees with finite or countable branchings [Mar70; CLN23], to name a few. In this paper, we follow the Homotopy Type Theory Book [Uni13] and consider ordinals as order types of well ordered sets, i.e., an ordinal is a type equipped with an order relation having certain properties.

Ordinals have a theory of arithmetic that generalizes the one of the natural numbers. Classically, arithmetic operations are defined by case distinction and transfinite recursion:

$$\alpha + 0 = \alpha$$
$$\alpha + (\beta + 1) = (\alpha + \beta) + 1$$
$$\alpha + \lambda = \mathsf{sup}_{\beta<\lambda}(\alpha + \beta) \qquad \text{(if } \lambda \text{ is a limit)}$$

$$\alpha \times 0 = 0$$
$$\alpha \times (\beta + 1) = (\alpha \times \beta) + \alpha$$
$$\alpha \times \lambda = \mathsf{sup}_{\beta<\lambda}(\alpha \times \beta) \qquad \text{(if } \lambda \text{ is a limit)}$$

Key words and phrases. constructive mathematics, homotopy type theory, ordinal arithmetic, Agda formalization.

Proposition 10 (⚙). *For ordinals α, β and γ, we have*

$$\alpha^{\beta+\gamma} = \alpha^{\beta} \times \alpha^{\gamma}.$$

Proof. We do transfinite induction on γ. Our first observation is that

$$\alpha^{\beta} \times \alpha^{\gamma} = \alpha^{\beta} \vee \mathsf{sup}_{c:\gamma}(\alpha^{\beta} \times \alpha^{\gamma\downarrow c} \times \alpha),$$

which follows from the fact that multiplication is continuous on the right (Lemma 2), noting that $\vee$ is implemented as a supremum.

Applying the induction hypothesis, we can rewrite $\alpha^{\beta} \times \alpha^{\gamma\downarrow c}$ to $\alpha^{\beta+\gamma\downarrow c}$, which is $\alpha^{(\beta+\gamma)\downarrow \mathsf{inr}\, c}$. The remaining goal thus is

$$\alpha^{\beta+\gamma} = \alpha^{\beta} \vee \mathsf{sup}_{c:\gamma}(\alpha^{(\beta+\gamma)\downarrow \mathsf{inr}\, c} \times \alpha),$$

which one gets by unfolding the definition on the left and applying antisymmetry. □

Proposition 11 (⚙). *For ordinals α, β and γ, iterated exponentiation can be calculated as follows:*

$$\left(\alpha^{\beta}\right)^{\gamma} = \alpha^{\beta\times\gamma}.$$

```
Proposition-10 : (α : Ordinal 𝓤) (β γ : Ordinal 𝓥)
               → α ^ₒ (β +ₒ γ) ＝ (α ^ₒ β) ×ₒ (α ^ₒ γ)
Proposition-10 = ^ₒ-by-+ₒ

Proposition-11 : (α : Ordinal 𝓤) (β γ : Ordinal 𝓥)
               → α ^ₒ (β ×ₒ γ) ＝ (α ^ₒ β) ^ₒ γ
Proposition-11 = ^ₒ-by-×ₒ

\end{code}

Section V. Decreasing Lists: A Constructive Formulation
           of Sierpiński's Definition

\begin{code}

Definition-12 : (α : Ordinal 𝓤) (β : Ordinal 𝓥) → 𝓤 ⊔ 𝓥 ⁺
Definition-12 α β = DecrList₂ α β
```

- All of its results are formalized in the proof assistant **Agda**. Clicking a ⚙ next to a definition, lemma, theorem, etc. in the paper takes you to its formalization.

Ordinals in HoTT

- An ordinal is a type α with a binary proposition-valued relation $<$ on α that is transitive, extensional and wellfounded.

Ordinals in HoTT

- An ordinal is a type α with a binary proposition-valued relation $<$ on α that is transitive, extensional and wellfounded.

  Extensionality says that two elements are equal if and only if they have the same predecessors: $x = y$ if and only if $\forall (u : \alpha). u < x \leftrightarrow u < y$.

Ordinals in HoTT

- An ordinal is a type α with a binary proposition-valued relation $<$ on α that is transitive, extensional and wellfounded.

  Extensionality says that two elements are equal if and only if they have the same predecessors: $x = y$ if and only if $\forall(u : \alpha).u < x \leftrightarrow u < y$.

  Wellfoundedness is defined via an inductive accessibility predicate but is equivalent to transfinite induction: for any type family P over α and $x : \alpha$, we have $(\forall(y : \alpha).\, y < x \to P\, y) \longrightarrow P\, x$.

- Examples of ordinals include $\mathbf{0}, \mathbf{1}, \mathbb{N}$ and the type $\mathsf{List}_<(\alpha)$ of decreasing lists over any ordinal α.

The ordinal of (small) ordinals

- A fundamental fact is that for any ordinal α and $a : \alpha$, the initial segment

$$\alpha \downarrow a :\equiv \Sigma(x : \alpha).\, x < a$$

  is again an ordinal.

The ordinal of (small) ordinals

- A fundamental fact is that for any ordinal α and $a : \alpha$, the initial segment

$$\alpha \downarrow a :\equiv \Sigma(x : \alpha).\, x < a$$

  is again an ordinal.

Key idea: characterize ordinals by describing their initial segments.

The ordinal of (small) ordinals

- A fundamental fact is that for any ordinal α and $a : \alpha$, the initial segment

$$\alpha \downarrow a :\equiv \Sigma(x : \alpha).\, x < a$$

  is again an ordinal.

  Key idea: characterize ordinals by describing their initial segments.

- Setting

$$\alpha < \beta :\equiv \Sigma(b : \beta).\, \alpha = \beta \downarrow b$$

  makes the type Ord of (small) ordinals into an ordinal itself.

  For proving extensionality, we use **univalence**.

The ordinal of (small) ordinals

- A fundamental fact is that for any ordinal α and $a : \alpha$, the initial segment

$$\alpha \downarrow a :\equiv \Sigma(x : \alpha).\, x < a$$

  is again an ordinal.

Key idea: characterize ordinals by describing their initial segments.

- Setting

$$\alpha < \beta :\equiv \Sigma(b : \beta).\, \alpha = \beta \downarrow b$$

  makes the type Ord of (small) ordinals into an ordinal itself.

  For proving extensionality, we use **univalence**.

  In particular, we can define operations on Ord by transfinite induction.

The ordinal of (small) ordinals

- A fundamental fact is that for any ordinal α and $a : \alpha$, the initial segment

$$\alpha \downarrow a :\equiv \Sigma(x : \alpha).\, x < a$$

is again an ordinal.

Key idea: characterize ordinals by describing their initial segments.

- Setting

$$\alpha < \beta :\equiv \Sigma(b : \beta).\, \alpha = \beta \downarrow b$$

makes the type Ord of (small) ordinals into an ordinal itself.
For proving extensionality, we use **univalence**.

In particular, we can define operations on Ord by transfinite induction.

- Moreover, Ord is a poset with

$$\alpha \leq \beta :\equiv \Sigma(f : \alpha \to \beta).\, \forall(a : A).\, \alpha \downarrow a = \beta \downarrow f\, a.$$

Preliminary constructions of ordinals

- Given ordinals α and β, their sum is given by the type $\alpha + \beta$ and putting everything in the left component below anything in the right component.

$$(\alpha + \beta) \downarrow \mathsf{inl}\, a = \alpha \downarrow a \quad \text{and} \quad (\alpha + \beta) \downarrow \mathsf{inr}\, b = \alpha + (\beta \downarrow b)$$

Preliminary constructions of ordinals

- Given ordinals α and β, their sum is given by the type $\alpha + \beta$ and putting everything in the left component below anything in the right component.

$$(\alpha + \beta) \downarrow \mathsf{inl}\, a = \alpha \downarrow a \quad \text{and} \quad (\alpha + \beta) \downarrow \mathsf{inr}\, b = \alpha + (\beta \downarrow b)$$

- Given ordinals α and β, their product is given by the type $\alpha \times \beta$ **ordered lexicographically with the right component taking precedence**.

$$(\alpha \times \beta) \downarrow (a, b) = \alpha \times (\beta \downarrow b) + (\alpha \downarrow a)$$

Preliminary constructions of ordinals

- Given ordinals α and β, their sum is given by the type $\alpha + \beta$ and putting everything in the left component below anything in the right component.

$$(\alpha + \beta) \downarrow \mathsf{inl}\, a = \alpha \downarrow a \quad \text{and} \quad (\alpha + \beta) \downarrow \mathsf{inr}\, b = \alpha + (\beta \downarrow b)$$

- Given ordinals α and β, their product is given by the type $\alpha \times \beta$ **ordered lexicographically with the right component taking precedence**.

$$(\alpha \times \beta) \downarrow (a, b) = \alpha \times (\beta \downarrow b) + (\alpha \downarrow a)$$

- Given a family of ordinals $F_\bullet : I \to \mathsf{Ord}$, we can construct its supremum $\sup F_\bullet$.

Preliminary constructions of ordinals

- Given ordinals α and β, their sum is given by the type $\alpha + \beta$ and putting everything in the left component below anything in the right component.

$$(\alpha + \beta) \downarrow \mathsf{inl}\, a = \alpha \downarrow a \quad \text{and} \quad (\alpha + \beta) \downarrow \mathsf{inr}\, b = \alpha + (\beta \downarrow b)$$

- Given ordinals α and β, their product is given by the type $\alpha \times \beta$ **ordered lexicographically with the right component taking precedence**.

$$(\alpha \times \beta) \downarrow (a, b) = \alpha \times (\beta \downarrow b) + (\alpha \downarrow a)$$

- Given a family of ordinals $F_\bullet : I \to \mathsf{Ord}$, we can construct its supremum $\sup F_\bullet$.

  In particular we have maps $[i, -] : F_i \leq \sup F_\bullet$ such that for any $y : \sup F_\bullet$ there exists $i : I$ and $x : F_i$ with

$$y = [i, x] \quad \text{and} \quad \sup F_\bullet \downarrow y = F_i \downarrow x.$$

Abstract exponentiation

- Inspired by the classical definition (and the no-go theorem), we now wish to construct, for $\alpha \geq \mathbf{1}$, an operation $\alpha^{(-)}$ satisfying the **specification**:

$$\begin{aligned} \alpha^{0} &= \mathbf{1} \\ \alpha^{\beta+\mathbf{1}} &= \alpha^{\beta} \times \alpha \\ \alpha^{\mathsf{sup}_{i:I} F_i} &= \sup_{i:I}(\alpha^{F_i}) \qquad \text{(if } I \text{ is inhabited)} \end{aligned}$$

Abstract exponentiation

- Inspired by the classical definition (and the no-go theorem), we now wish to construct, for $\alpha \geq \mathbf{1}$, an operation $\alpha^{(-)}$ satisfying the **specification**:

$$\begin{aligned} \alpha^{0} &= \mathbf{1} \\ \alpha^{\beta+\mathbf{1}} &= \alpha^{\beta} \times \alpha \\ \alpha^{\sup_{i:I} F_i} &= \sup_{i:I}(\alpha^{F_i}) && \text{(if } I \text{ is inhabited)} \end{aligned}$$

- Lemma For every ordinal β we have $\beta = \sup_{b:\beta}((\beta \downarrow b) + \mathbf{1})$.

Abstract exponentiation

- Inspired by the classical definition (and the no-go theorem), we now wish to construct, for $\alpha \geq \mathbf{1}$, an operation $\alpha^{(-)}$ satisfying the **specification**:

$$\alpha^{0} = \mathbf{1}$$
$$\alpha^{\beta+\mathbf{1}} = \alpha^{\beta} \times \alpha$$
$$\alpha^{\sup_{i:I} F_i} = \sup_{i:I}(\alpha^{F_i}) \qquad \text{(if } I \text{ is inhabited)}$$

- Lemma For every ordinal β we have $\beta = \sup_{b:\beta}((\beta \downarrow b) + \mathbf{1})$.

- **Idea**: *If* we had α^{β}, then

$$\alpha^{\beta} = \alpha^{\sup_{b:\beta}(\beta \downarrow b)+\mathbf{1}} = \mathbf{1} \vee \sup_{b:\beta} \alpha^{(\beta \downarrow b)+\mathbf{1}} = \mathbf{1} \vee \sup_{b:\beta} \left(\alpha^{\beta \downarrow b} \times \alpha\right).$$

Abstract exponentiation

- Inspired by the classical definition (and the no-go theorem), we now wish to construct, for $\alpha \geq \mathbf{1}$, an operation $\alpha^{(-)}$ satisfying the **specification**:

$$\alpha^0 = \mathbf{1}$$
$$\alpha^{\beta+\mathbf{1}} = \alpha^\beta \times \alpha$$
$$\alpha^{\sup_{i:I} F_i} = \sup_{i:I}(\alpha^{F_i}) \qquad \text{(if } I \text{ is inhabited)}$$

- Lemma For every ordinal β we have $\beta = \sup_{b:\beta}((\beta \downarrow b) + \mathbf{1})$.
- **Idea**: *If* we had α^β, then

$$\alpha^\beta = \alpha^{\sup_{b:\beta}(\beta\downarrow b)+\mathbf{1}} = \mathbf{1} \vee \sup_{b:\beta} \alpha^{(\beta\downarrow b)+\mathbf{1}} = \mathbf{1} \vee \sup_{b:\beta} \left(\alpha^{\beta\downarrow b} \times \alpha\right).$$

- Def. Define abstract exponentiation α^β by transfinite induction in Ord on β as

$$\alpha^\beta :\equiv \sup_{x:\mathbf{1}+\beta} \begin{cases} \text{inl} \star \mapsto \mathbf{1} \\ \text{inr}\, b \mapsto \alpha^{\beta\downarrow b} \times \alpha \end{cases}$$

Properties of abstract exponentiation

- Def. (repeated) Abstract exponentiation α^β is given by transfinite induction on β:

$$\alpha^\beta :\equiv \sup_{x:\mathbf{1}+\beta} \begin{cases} \mathsf{inl}\,\star \mapsto \mathbf{1} \\ \mathsf{inr}\,b \mapsto \alpha^{\beta\downarrow b} \times \alpha \end{cases}$$

- Thm. This transfinite construction satisfies the specification as well as

$$\alpha^{\beta+\gamma} = \alpha^\beta \times \alpha^\gamma \quad \text{and} \quad \alpha^{\beta\times\gamma} = \left(\alpha^\beta\right)^\gamma.$$

Properties of abstract exponentiation

- Def. (repeated) Abstract exponentiation α^β is given by transfinite induction on β:

$$\alpha^\beta :\equiv \sup_{x:\mathbf{1}+\beta} \begin{cases} \mathsf{inl}\ \star \mapsto \mathbf{1} \\ \mathsf{inr}\ b \mapsto \alpha^{\beta\downarrow b} \times \alpha \end{cases}$$

- Thm. This transfinite construction satisfies the specification as well as

$$\alpha^{\beta+\gamma} = \alpha^\beta \times \alpha^\gamma \quad \text{and} \quad \alpha^{\beta\times\gamma} = \left(\alpha^\beta\right)^\gamma.$$

- Using the characterization of initial segments of suprema and products, we have for $a : \alpha$, $b : \beta$ and $e : \alpha^{\beta\downarrow b}$ that

$$\alpha^\beta \downarrow [\mathsf{inr}\ b, (e, a)] = \alpha^{\beta\downarrow b} \times (\alpha \downarrow a) + (\alpha^{\beta\downarrow b} \downarrow e).$$

Concrete exponentiation

- Sierpiński classically constructs α^β using the set of functions $f : \beta \to \alpha$ such that f has **finite support**: i.e. $f\ b$ is not the least element a_0 only finitely many times.

Concrete exponentiation

- Sierpiński classically constructs α^β using the set of functions $f : \beta \to \alpha$ such that f has **finite support**: i.e. $f\ b$ is not the least element a_0 only finitely many times.

- Constructively well behaved version: represent a function with finite support as a **list of (output, input) pairs**, ordered decreasingly in the input-component to ensure uniqueness of the representation.

Concrete exponentiation

- Sierpiński classically constructs α^β using the set of functions $f : \beta \to \alpha$ such that f has **finite support**: i.e. $f\ b$ is not the least element a_0 only finitely many times.

- Constructively well behaved version: represent a function with finite support as a **list of (output, input) pairs**, ordered decreasingly in the input-component to ensure uniqueness of the representation.

  The least element $a_0 : \alpha$ should not be an output, so we consider

$$\alpha_{>0} :\equiv \Sigma(a : \alpha).\, a > a_0$$

Concrete exponentiation

- Sierpiński classically constructs α^β using the set of functions $f : \beta \to \alpha$ such that f has **finite support**: i.e. $f\ b$ is not the least element a_0 only finitely many times.

- Constructively well behaved version: represent a function with finite support as a **list of (output, input) pairs**, ordered decreasingly in the input-component to ensure uniqueness of the representation.

  The least element $a_0 : \alpha$ should not be an output, so we consider

$$\alpha_{>0} :\equiv \Sigma(a : \alpha).\, a > a_0$$

  and define concrete exponentiation as

$$\exp(\alpha, \beta) :\equiv \Sigma(\ell : \mathrm{List}(\alpha_{>0} \times \beta)).\, \ell \text{ is decreasing in the } \beta\text{-component.}$$

Concrete exponentiation

- Sierpiński classically constructs α^β using the set of functions $f : \beta \to \alpha$ such that f has **finite support**: i.e. $f\ b$ is not the least element a_0 only finitely many times.

- Constructively well behaved version: represent a function with finite support as a **list of (output, input) pairs**, ordered decreasingly in the input-component to ensure uniqueness of the representation.

  The least element $a_0 : \alpha$ should not be an output, so we consider

$$\alpha_{>0} :\equiv \Sigma(a : \alpha).\, a > a_0$$

  and define concrete exponentiation as

$$\exp(\alpha, \beta) :\equiv \Sigma(\ell : \mathsf{List}(\alpha_{>0} \times \beta)).\, \ell \text{ is decreasing in the } \beta\text{-component.}$$

- We require a **trichotomous least element** a_0, i.e. a_0 satisfies $(a_0 < x) + (a_0 = x)$ for all $x : \alpha$, to ensure that $\alpha_{>0}$ and hence $\exp(\alpha, \beta)$ is an ordinal.

Properties of abstract and concrete exponentiation

- Thm. Concrete exponentiation satisfies the specification.

  Some other algebraic properties proved tedious to establish rigorously.

Properties of abstract and concrete exponentiation

- Thm. Concrete exponentiation satisfies the specification.

  Some other algebraic properties proved tedious to establish rigorously.

- Thm. Concrete exponentiation **obviously preserves decidability properties**, e.g. if α and β have decidable equality, then so does $\mathsf{exp}(\alpha, \beta)$.

  This is not at all obvious for abstract exponentiation.

Properties of abstract and concrete exponentiation

- Thm. Concrete exponentiation satisfies the specification.

  Some other algebraic properties proved tedious to establish rigorously.

- Thm. Concrete exponentiation **obviously preserves decidability properties**, e.g. if α and β have decidable equality, then so does $\exp(\alpha, \beta)$.

  This is not at all obvious for abstract exponentiation.

- Thankfully, abstract and concrete exponentiation agree!*

  So we can **transfer properties** from one construction to the other and make use of their particular advantages.

* When the base ordinal has a trichotomous least element

Proving that abstract and concrete exponentiation agree

- The key idea is to characterize initial segments.
- Recall that $\alpha^\beta \downarrow [\mathsf{inr}\, b, (e, a)] = \alpha^{\beta \downarrow b} \times (\alpha \downarrow a) + (\alpha^{\beta \downarrow b} \downarrow e)$.

Proving that abstract and concrete exponentiation agree

- The key idea is to characterize initial segments.
- Recall that $\alpha^{\beta} \downarrow [\mathsf{inr}\ b, (e, a)] = \alpha^{\beta \downarrow b} \times (\alpha \downarrow a) + (\alpha^{\beta \downarrow b} \downarrow e)$.
- For concrete exponentiation we can prove

  $$\exp(\alpha, \beta) \downarrow ((a, b) :: \iota_b\, \ell) \;=\; \exp(\alpha, \beta \downarrow b) \times (\alpha \downarrow a) + \exp(\alpha, \beta \downarrow b) \downarrow \ell$$

  where $\iota_b : \exp(\alpha, \beta \downarrow b) \hookrightarrow \exp(\alpha, \beta)$ is the obvious inclusion.

  Notice the **similarity to the above equation**!

Proving that abstract and concrete exponentiation agree

- The key idea is to characterize initial segments.
- Recall that $\alpha^\beta \downarrow [\mathrm{inr}\, b, (e, a)] = \alpha^{\beta \downarrow b} \times (\alpha \downarrow a) + (\alpha^{\beta \downarrow b} \downarrow e)$.
- For concrete exponentiation we can prove

$$\exp(\alpha, \beta) \downarrow ((a, b) :: \iota_b\, \ell) \;=\; \exp(\alpha, \beta \downarrow b) \times (\alpha \downarrow a) + \exp(\alpha, \beta \downarrow b) \downarrow \ell$$

  where $\iota_b : \exp(\alpha, \beta \downarrow b) \hookrightarrow \exp(\alpha, \beta)$ is the obvious inclusion.

  Notice the **similarity to the above equation**!

- A proof by transfinite induction in Ord on β then shows:

  Thm. For α with a trichotomous least element and β we have $\exp(\alpha, \beta) = \alpha^\beta$.

Proving that abstract and concrete exponentiation agree

- The key idea is to characterize initial segments.
- Recall that $\alpha^{\beta} \downarrow [\mathsf{inr}\, b, (e, a)] = \alpha^{\beta \downarrow b} \times (\alpha \downarrow a) + (\alpha^{\beta \downarrow b} \downarrow e)$.
- For concrete exponentiation we can prove

  $$\exp(\alpha, \beta) \downarrow ((a, b) :: \iota_b\, \ell) \;=\; \exp(\alpha, \beta \downarrow b) \times (\alpha \downarrow a) + \exp(\alpha, \beta \downarrow b) \downarrow \ell$$

  where $\iota_b : \exp(\alpha, \beta \downarrow b) \hookrightarrow \exp(\alpha, \beta)$ is the obvious inclusion.

  Notice the **similarity to the above equation**!
- A proof by transfinite induction in Ord on β then shows:

  Thm. For α with a trichotomous least element and β we have $\exp(\alpha, \beta) = \alpha^{\beta}$.
- The equality $\exp(\alpha, \beta) = \alpha^{\beta}$ induces a function $\exp(\alpha, \beta) \to \alpha^{\beta}$ which we show to coincide with a natural **denotation** map that captures the intuition that a list in $\exp(\alpha, \beta)$ is a concrete representation of an abstract element of α^{β}.

Wrapping up

- We presented two constructively well behaved ordinal exponentiation functions for base ordinals with a least element, and showed them to be equivalent in case the base ordinal has a trichotomous least element.

Wrapping up

- We presented two constructively well behaved ordinal exponentiation functions for base ordinals with a least element, and showed them to be equivalent in case the base ordinal has a trichotomous least element.
- Thanks to **univalence** we can **transfer various results**, such as algebraic laws and decidability properties, from one construction to the other.

Wrapping up

- We presented two constructively well behaved ordinal exponentiation functions for base ordinals with a least element, and showed them to be equivalent in case the base ordinal has a trichotomous least element.
- Thanks to **univalence** we can **transfer various results**, such as algebraic laws and decidability properties, from one construction to the other.
- In the paper we further mark the **limits of a constructive treatment** by presenting no-go theorems that show the law of excluded middle to be equivalent to certain statements about ordinal exponentiation.

Wrapping up

- We presented two constructively well behaved ordinal exponentiation functions for base ordinals with a least element, and showed them to be equivalent in case the base ordinal has a trichotomous least element.
- Thanks to **univalence** we can **transfer various results**, such as algebraic laws and decidability properties, from one construction to the other.
- In the paper we further mark the **limits of a constructive treatment** by presenting no-go theorems that show the law of excluded middle to be equivalent to certain statements about ordinal exponentiation.
- Natural question: can we **fuse the two constructions and use quotiented lists** to define ordinal exponentiation for base ordinals that do not necessarily have a trichotomous least element?

Wrapping up

- We presented two constructively well behaved ordinal exponentiation functions for base ordinals with a least element, and showed them to be equivalent in case the base ordinal has a trichotomous least element.
- Thanks to **univalence** we can **transfer various results**, such as algebraic laws and decidability properties, from one construction to the other.
- In the paper we further mark the **limits of a constructive treatment** by presenting no-go theorems that show the law of excluded middle to be equivalent to certain statements about ordinal exponentiation.
- Natural question: can we **fuse the two constructions and use quotiented lists** to define ordinal exponentiation for base ordinals that do not necessarily have a trichotomous least element?

Thank you!

`arXiv:2501.14542`

References

[1] Peter Aczel. 'An introduction to inductive definitions'. In: *Handbook of Mathematical Logic*. Ed. by Jon Barwise. Vol. 90. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company, 1977, pp. 739–782. DOI: 10.1016/S0049-237X(08)71120-0.

[2] Peter Dybjer and Anton Setzer. 'A Finite Axiomatization of Inductive-Recursive Definitions'. In: *Typed Lambda Calculi and Applications. 4th International Conferene, TLCA'99, L'Aquila, Italy, April 1999, Proceedings*. Ed. by Jean-Yves Girard. Vol. 1581. Lecture Notes in Computer Science. Springer, 1999, pp. 129–146. DOI: 10.1007/3-540-48959-2_11.

[3] Martín Hötzel Escardó et al. 'Ordinals in univalent type theory in Agda notation'. Agda development, HTML rendering available at: https://www.cs.bham.ac.uk/~mhe/TypeTopology/Ordinals.index.html. Since 2018. URL: https://github.com/martinescardo/TypeTopology.

[4] Robert W. Floyd. 'Assigning meanings to programs'. In: *Mathematical Aspects of Computer Science*. Ed. by J. T. Schwartz. Vol. 19. Proceedings of Symposia in Applied Mathematics. American Mathematical Society, 1967, pp. 19–32. DOI: 10.1090/psapm/019/0235771.

[5] Nicolai Kraus, Fredrik Nordvall Forsberg and Chuangjie Xu. 'Type-Theoretic Approaches to Ordinals'. In: *Theoretical Computer Science* 957 (2023). DOI: 10.1016/j.tcs.2023.113843. arXiv: 2208.03844 [cs.LO].

[6] Michael Rathjen. 'The art of ordinal analysis'. In: *Proceedings of the International Congress of Mathematicians. Madrid 2006*. Vol. 2. European Mathematical Society Publishing House, 2007, pp. 45–69. DOI: 10.4171/022-2/3.

[7] Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: https://homotopytypetheory.org/book, 2013.