Arithmetical algorithms for elementary patterns

Samuel A. Alexander[*] [†]
Department of Mathematics, the Ohio State University

March 30, 2014

Abstract

Elementary patterns of resemblance notate ordinals up to the ordinal of $\Pi^1_1 - CA_0$. We provide ordinal multiplication and exponentiation algorithms using these notations.

1 Introduction

In [4], Timothy J. Carlson used deep structural properties of the ordinal numbers [3] to settle an open problem of William Reinhardt. These properties were later organized into *elementary patterns of resemblance* [5]. In the latter paper, Carlson showed that elementary patterns notate the recursive ordinals up to an ordinal κ called the *core*. Wilken established [10] that the core is the ordinal of $KP\ell_0$ (equivalently, of $\Pi^1_1 - CA_0$), via isomorphism with notations based on collapsing functions. One way to perform pattern arithmetic, then, is to use Wilken's isomorphism, perform the arithmetic using the collapsing functions, and then reverse the isomorphism. Our goal is to establish algorithms for performing arithmetic directly on patterns, in a geometric way.

Two algorithms have already been published. In [5], Lemma 7.12 is proved constructively, implicitly yielding an algorithm for amalgamating two patterns into a single pattern. We will see shortly why this simplifies pattern arithmetic. In [7], the constructive Theorem 4.6 yields an algorithm for putting patterns in normal form.

The algorithms in the present paper are implemented in [1], an online ordinal calculator under development.

In Section 2, we review preliminaries, some of which are phrased in a new way that we hope will illuminate. This section also makes explicit the addition algorithm implicit in [5].

In Section 3 we introduce the notion of the *reach* of an ordinal below the ordinal of $\Pi^1_1 - CA_0$.

In Section 4 we give a multiplication algorithm.

In Section 5 we give algorithms for base-ω exponentiation and logarithm.

2 Preliminaries

An ordinal α is *decomposable* if $\alpha = 0$ or $\alpha = \beta+\gamma$ for some nonzero $\beta, \gamma < \alpha$. Otherwise, α is *indecomposable*. The following definition is a special case of Definition 3.7 of [5].

Definition 1. A set X of ordinals is *closed* if whenever X contains α, then X contains all the components in α's Cantor normal form decomposition.

Definition 2. Let $\mathscr{L}_0$ be the language $(0, +, \leq)$, where 0 and $\leq$ have the expected arities and $+$ is a ternary relation symbol (meant to represent the graph of a possibly non-total addition function). An *ordinal-additive structure* is an $\mathscr{L}_0$-structure that is $\mathscr{L}_0$-isomorphic to a closed set of ordinals. An element $a \neq 0$ of an ordinal-additive structure $\mathbf{A}$ is *indecomposable* if there are no $0 \neq b, c \in \mathbf{A}$ with $a = b + c$, $c < a$.

[*] Email: alexander@math.ohio-state.edu

[†] 2010 Mathematics Subject Classification: 03F15

Lemma 1. A finite $\mathscr{L}_0$-structure $\mathbf{A}$ is an ordinal-additive structure if and only if the following conditions hold.

- $\leq$ linearly orders $\mathbf{A}$ with minimum element 0.
- For all $0 \neq a \in \mathbf{A}$ there is a descending sequence $a_1, \ldots, a_m$ of $\mathbf{A}$-indecomposables with $a = a_1 + \cdots + a_m$.
- Whenever $a_1, \ldots, a_m$ and $b_1, \ldots, b_n$ are descending sequences of $\mathbf{A}$-indecomposables such that $a_1 + \cdots + a_m$ and $b_1 + \cdots + b_n$ are defined, we have
  1. $a_1 + \cdots + a_m \leq b_1 + \cdots + b_n$ iff $(a_1, \ldots, a_m) \leq_{\text{lex}} (b_1, \ldots, b_n)$.
  2. If $n \neq 0$ then either
     (a) $a_1 < b_1$ and $a_1 + \cdots + a_m + b_1 + \cdots + b_n = b_1 + \cdots + b_n$, or
     (b) $a_1 \not< b_1$ and $a_1 + \cdots + a_m + b_1 + \cdots + b_n = a_1 + \cdots + a_i + b_1 + \cdots + b_n$, where i is maximal such that $b_1 \leq a_i$.

Proof. By basic facts about ordinal arithmetic. □

In [5], the conditions of Lemma 1 (minus finiteness) define what is there called an *additive structure*.

Definition 3. (See Fig. 1) Let $\mathscr{L}_1$ be the language $(0, +, \leq, \leq_1)$, where $\leq_1$ is a binary relation symbol and $0, +, \leq$ are as in $\mathscr{L}_0$. An *additive pattern of resemblance of order one* (hereafter just a *pattern*) is a finite $\mathscr{L}_1$-structure P satisfying the following conditions.

- The $\mathscr{L}_0$ part of P is an ordinal-additive structure.
- $\leq_1$ is a reflexive transitive subrelation of $\leq$.
- $\leq_1$ is a forest respecting $\leq$, by which we mean that whenever $a \leq b \leq c$ and $a \leq_1 c$, this implies $a \leq_1 b$.
- Whenever $a <_1 b$, a is indecomposable.

Elements of the universe of a pattern are called *nodes*, or *points*, of the pattern.

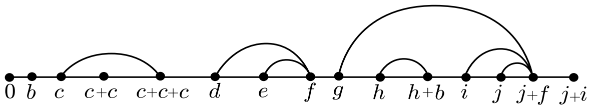


Figure 1: A pattern of resemblance. An arc with left endpoint ℓ and right endpoint r indicates that $\ell \leq_1 x$ for all $\ell \leq x \leq r$. A point that is not the left endpoint of any arc is understood to be $\not\leq_1$ any point except itself.

Having added $\leq_1$ to $\mathscr{L}_1$, we feel obliged to give an intended interpretation of $\leq_1$ on the ordinals.

Definition 4. The binary relation $\leq_1$ on *Ord* is defined by transfinite recursion so that for all $\alpha, \beta \in Ord$, $\alpha \leq_1 \beta$ iff $\alpha \leq \beta$ and $(\alpha, 0, +, \leq, \leq_1)$ is a Σ_1-elementary substructure of $(\beta, 0, +, \leq, \leq_1)$.

Lemma 2. $\leq_1$ is transitive and is a forest respecting $\leq$. More strongly, for every finite closed set $X \subseteq Ord$, $(X, 0, +, \leq, \leq_1)$ is a pattern.

Proof. Straightforward. □

More remarkably, a converse also holds.

Theorem 3. (ZF) Every pattern is $\mathscr{L}_1$-isomorphic to a closed set of ordinals.

Proof. See Lemma 6.1 and Theorem 5.9 (1-2) of [5]. Carlson's proof of Lemma 6.1 uses the full force of ZF, in the form of the reflection principle. □

Theorem 3 is also obtained in [10].

For the remainder of the paper, we assume full ZF so that we may use Theorem 3. We will not use the following lemma, but it will motivate a later definition.

Lemma 4. For all $\alpha \leq \beta \in Ord$, $\alpha \leq_1 \beta$ if and only if for every finite $X \subseteq \alpha$ and finite $Y \subseteq [\alpha, \beta)$, there is $X < \widetilde{Y} < \alpha$ such that $X \cup \widetilde{Y} \cong_{\mathscr{L}_1} X \cup Y$.

Proof. Straightforward; for a proof, see Gunnar Wilken's dissertation [8]. □

2.1 Pattern Syntax and Semantics

Nodes of a pattern notate ordinals in two ways. One is concrete and constructive, the other is abstract. Carlson showed that both are equivalent. Identifying the two, one can learn much about the ordinals.

Definition 5. A *pointed pattern* is a pair (P, x) where P is a pattern and $x \in P$.

Definition 6. A pointed pattern (P, x) is said to *semantically notate* the ordinal α if, writing P^* for the lexicominimal closed set of ordinals $\mathscr{L}_1$-isomorphic to P, α corresponds to x under the isomorphism.

Example 5. By the exercise atop p. 21 of [5] (see also [9]), ω^ω is the least ordinal α such that[1] $\alpha \leq_1 \alpha + 1$; ϵ_0 is the least nonzero ordinal α such that $\alpha \leq_1 \alpha+\alpha$; and Γ_0 is the least nonzero ordinal $\alpha \leq_1 \alpha^2$; it follows that the nodes in Fig. 2 notate 0, 1, ω^ω, $\omega^\omega + 1$, ϵ_0, $\epsilon_0+\epsilon_0$, Γ_0, Γ_0^2, and $\Gamma_0^2+\Gamma_0$, in that order. (In the case of the last three nodes, this takes some more work to establish, but it can be done using the aforementioned exercise.)

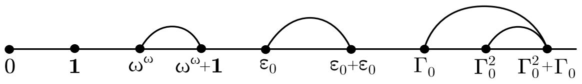


Figure 2: A pattern of resemblance with its nodes labeled by the ordinals they semantically notate.

We will also define what it means for (P, x) to syntactically notate an ordinal; this requires more machinery.

Lemma 6. (See Fig. 3) Suppose P is a pattern and $a_1, \ldots, a_{n+1}$ is a descending sequence of P-indecomposables such that $a_1 + \cdots + a_n$ is defined but $a_1 + \cdots + a_{n+1}$ is not. There is an extension of P to a pattern P^+ such that

- $a_1 + \cdots + a_{n+1}$ is defined in P^+.
- $a_1 + \cdots + a_{n+1}$ is the unique element of $P^+ \backslash P$.
- For every $x < a_1 + \cdots + a_{n+1}$, $x <_1 a_1 + \cdots + a_{n+1}$ iff $x <_1 y$ for some $y > a_1 + \cdots + a_{n+1}$.

Any two such extensions are $\mathscr{L}_1$-isomorphic over P. Such an extension is called a *simple additive extension* of P.

Proof. Straightforward (Lemma 4.5 of [5]). □

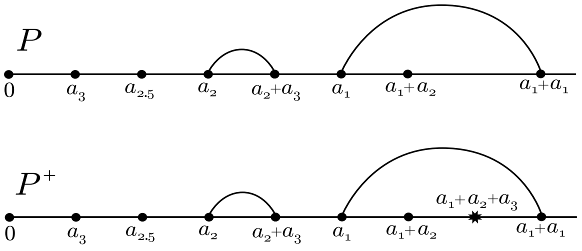


Figure 3: Pattern P^+ is a simple additive extension of pattern P.

By a *subpattern* of a pattern P, we mean an $\mathscr{L}_1$-substructure of P. A subpattern Q of P is *closed* (with respect to P) if whenever $a +^P b \in Q$, this implies $a \in Q$ and $b \in Q$.

Lemma 7. (Compare Lemma 4) (See Fig. 4) Let P be a pattern, $a, b \in P$, $a <_1 b$, and let $Y \subseteq [a, b)^P$ be such that $[0, a)^P \cup Y$ is a closed subpattern of P. There is a pattern P^+, of which P is a subpattern, such that, writing $\widetilde{Y}$ for $P^+ \backslash P$, we have:

- $[0, a)^P < \widetilde{Y} < a$
- $[0, a)^P \cup Y \cong_{\mathscr{L}_1} [0, a)^P \cup \widetilde{Y}$.
- Whenever $y \in \widetilde{Y}$ and $y \leq_1 z$ then $z \in \widetilde{Y}$.

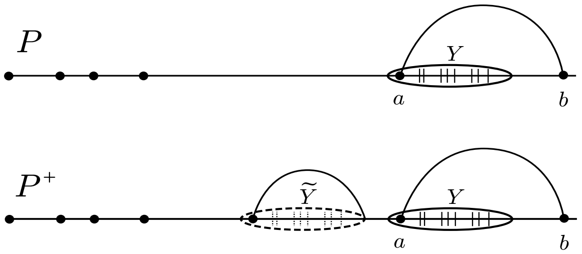


Figure 4: Downward reflection.

[1] Bès showed [2] that for $\alpha \in Ord$, the elementary theory of $(\alpha, \leq, \cdot)$ is decidable precisely if $\alpha < \omega^\omega$. Thus in loose geometric terms, we can say that the elementary theory of ordinals under multiplication becomes badly behaved precisely when the first "arc" appears.

Furthermore, any two candidates for P^+ are $\mathscr{L}_1$-isomorphic over P. Such a P^+ is said to be *obtained from P by reflecting Y below a* (or simply to be *obtained from P by reflection*).

Proof. See Lemma 4.8 of [5]. □

Definition 7. (Immediate Extension) Let P, Q be patterns. We call Q an *immediate extension* of P if either $Q = P$, or Q is a simple additive extension of P, or Q is obtained from P by reflection.

Definition 8. (Fair Sequences) Suppose P_n ($n \in \mathbb{N}$) is a sequence of patterns, each an immediate extension of the previous. Let $P_\infty = \cup_n P_n$. We say the P_n form a *fair sequence* for P_0 if the following conditions hold.

1. $+$ is a total function on P_∞. In other words, for every $i \in \mathbb{N}$ and every descending sequence $a_1, \ldots, a_m$ of indecomposables in P_i, there is $j \geq i$ such that $a_1 + \cdots + a_m$ is defined in P_j.
2. For every $i \in \mathbb{N}$, every $a, b \in P_i$ with $a <_1 b$, and every $Y \subseteq [a, b)^{P_i}$ such that $[0, a)^{P_i} \cup Y$ is a closed subpattern of P_i, there is some $j \geq i$ such that P_{j+1} is obtained from P_j by reflecting Y below a.

Fig. 5 shows the beginning of a fair sequence for a pattern whose nodes notate $\{0, 1, \omega^\omega, \omega^\omega + 1\}$. A fair sequence may be thought of as an attempt, starting with P_0, to generate as many nodes as possible using machinery from Definition 7.

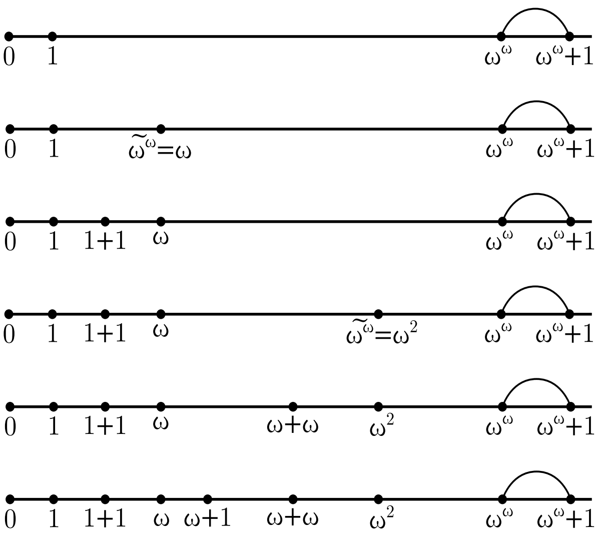


Figure 5: The first few patterns in a fair sequence.

Theorem 8. Let P be a pattern.

1. If P_n ($n \in \mathbb{N}$) is a fair sequence for P, then $P_\infty = \cup_n P_n$ is well founded by $\leq$, i.e., it has no infinite strictly decreasing sequence of nodes.
2. If $x \in P$ then for any two fair sequences P_n, P'_n for P, the order type of the nodes below x in P_∞ is equal to the order type of the nodes below x in P'_∞.

Proof. Follows from Lemma 5.8 of [5]. □

Definition 9. (Compare Definition 6) A pointed pattern (P, x) is said to *syntactically notate* the ordinal α if for some (equivalently for every) fair sequence P_n for P, α is the order type of the nodes below x in $P_\infty = \cup_n P_n$.

We have developed enough machinery to state a theorem saying that pointed patterns syntactically notate the same ordinals as they semantically notate. But first we will develope a little more machinery in order to state an even broader theorem.

Definition 10. Let P, Q be patterns such that P is a subpattern of Q.

- We say P *syntactically exactly generates* Q if Q can be obtained from P by a finite sequence of immediate extensions.
- We say P *syntactically generates* Q if P syntactically exactly generates a pattern Q' in which Q is a subpattern.
- Let $\iota : P \to Q$ be inclusion. Let P^*, Q^* be the lexicominimal closed *Ord*-substructures isomorphic to P and Q respectively, with isomorphisms $\phi : P \to P^*$ and $\psi : Q \to Q^*$. We say P *semantically generates* Q if P contains Q's maximum indecomposable (or $P = Q$ if Q has no indecomposables) and $\phi = \psi \circ \iota$.

Theorem 9. (Syntax-Semantics Equivalence)

1. If pointed pattern (P, x) semantically notates α and syntactically notates α', then $\alpha = \alpha'$.

2. If P is a subpattern of Q then P semantically generates Q if and only if P syntactically generates Q.

Proof. Follows from the way Lemmas 5.7 and 5.8 and Theorem 5.9 of [5] are proved. □

Definition 11. If (P, x) is a pointed pattern, $(P, x)^*$ shall denote the ordinal that is semantically (equivalently, syntactically) notated by (P, x). In case P is clear from context, we may write x^* for the same thing. If $P \subseteq Q$, we say P *generates* Q if P semantically (equivalently, syntactically) generates Q.

Theorem 10. The ordinals notated by pointed patterns are precisely the ordinals below the ordinal of $\Pi^1_1 - CA_0$, or equivalently below the ordinal of $KP\ell_0$.

Proof. See Wilken [10]. □

Theorem 11. (The Interval Theorem) Suppose P is a pattern, $x < y \in P$. Let $P = P_0, P_1, \ldots$ be a fair sequence for P, let $P_\infty = \cup_i P_i$. The ordinals in the interval $(x^*, y^*)^{Ord}$ are precisely the ordinals $\{(P_n, z)^* : z \in (x, y)^{P_n}$ for some $n \in \mathbb{N}\}$.

Proof. First suppose $z \in (x, y)^{P_n}$ for some $n \in \mathbb{N}$. Since P generates P_n,

$$x^* = (P, x)^* = (P_n, x)^* < (P_n, z)^* < (P_n, y)^* = (P, y)^* = y^*,$$

so $(P_n, z)^* \in (x^*, y^*)^{Ord}$.

We will prove the converse by contradiction. Let $Z = \{(P_n, z)^* : z \in (x, y)^{P_n}$ for some $n \in \mathbb{N}\}$ and assume $(x^*, y^*) \not\subseteq Z$. Let $\alpha \in (x^*, y^*)\backslash Z$ be minimal and let $\beta \in Z \cup \{y^*\}$ be minimal such that $\beta > \alpha$. There is some $n \in \mathbb{N}$ and $z \in P_n$ with $z^* = \beta$. By minimality of α and β, the set $Z' = \{(P_m, t)^* : t \in (x, z)^{P_m}$ for some $m \geq n\}$ is exactly (x^*, α). By the syntactic definition of z^*, it follows that $z^* = \alpha$, a contradiction. □

2.2 Amalgamation

The following (semantic) definition differs from the (syntactic) Definition 6.5 of [5], but the two are easily seen to be equivalent in light of Theorem 9.

Definition 12. (See Fig. 6) Assume P_1, P_2 are patterns. An *amalgamation* of P_1 and P_2 is a pattern $\widehat{P}$ along with embeddings $\phi_i : P_i \to \widehat{P}$ such that, writing $\widehat{P_i}$ for $\phi_i(P_i)$ and $\widehat{x}$ for $\phi_i(x)$,

- $\widehat{P} = \widehat{P_1} \cup \widehat{P_2}$.
- $\forall i \in \{1, 2\}$, $\forall x \in P_i$, $(P_i, x)^* = (\widehat{P}, \widehat{x})^*$.

Theorem 12. There is an algorithm that takes patterns P, Q as input and outputs an amalgamation of P and Q.

Proof. The algorithm is implicit in the constructive proof of Lemma 7.12 of [5]. □

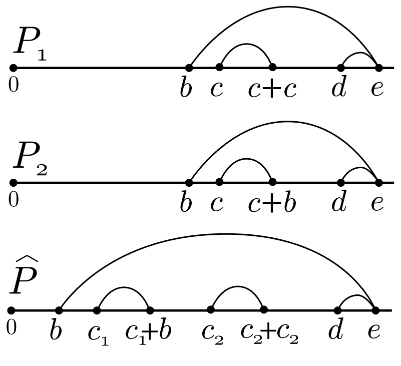


Figure 6: Amalgamation.

Theorem 12 is implemented in the Patterns of Resemblance Ordinal Calculator [1] as the *amalgamate* command.

2.3 Preliminary Arithmetic Algorithms

Proposition 13. (Comparison Algorithm) The following algorithm, taking input two pointed patterns (P,x) and (Q,y), decides whether or not $(P,x)^* \le (Q,y)^*$ and whether or not $(P,x)^* \le_1 (Q,y)^*$.

1. Using Theorem 12, compute an amalgamation A of P and Q, along with inclusions $\widehat{P}$ and $\widehat{Q}$ of P and Q in A.

2. If $\widehat{x} \le \widehat{y}$, report that $(P,x)^* \le (Q,y)^*$. Otherwise, report $(P,x)^* > (Q,y)^*$.

3. If $\widehat{x} \le_1 \widehat{y}$, report that $(P,x)^* \le_1 (Q,y)^*$. Otherwise, report $(P,x)^* \not\le_1 (Q,y)^*$.

Proof. Immediate by Theorem 12 and Definition 12. □

Proposition 14. (Addition Algorithm) The following algorithm, taking input two pointed patterns (P,x) and (Q,y), outputs a pointed pattern (R,z) such that $(P,x)^* + (Q,y)^* = (R,z)^*$.

1. Using Theorem 12, compute an amalgamation A of P and Q, along with inclusions $\widehat{P}$ and $\widehat{Q}$ of P and Q in A.

2. If $\widehat{x} + \widehat{y}$ is defined in A, output $(A, \widehat{x} + \widehat{y})$ and stop.

3. Otherwise, use Lemma 6 (repeatedly, if needed) to compute a sequence $A = A_1, \ldots, A_n$ of simple additive extensions such that $\widehat{x} + \widehat{y}$ is defined in A_n. Output $(A_n, \widehat{x} + \widehat{y})$.

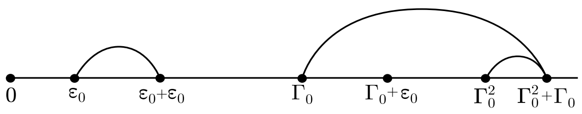


Figure 7: Result of adding Γ_0 and ϵ_0.

Proof. If the algorithm halts on line 2, its accuracy is immediate by Theorem 12 and Definition 12. Suppose the algorithm halts on line 3. Since A syntactically generates A_n, by Syntax-Semantics Equivalence, A semantically generates A_n, thus $(A_n, \widehat{x})^* = (A, \widehat{x})^* = (P,x)^*$ and $(A_n, \widehat{y})^* = (A, \widehat{y})^* = (Q,y)^*$, and the algorithm's accuracy follows. □

Comparison ($\le$) and addition are implemented in the Patterns of Resemblance Ordinal Calculator [1] as the *compare* and *add* commands.

3 Reach and Index

In this section we introduce two notions that play a crucial role in the arithmetic of patterns: reach and index.

Lemma 15. The indecomposable ordinals are precisely the ordinals of the form ω^α.

Proof. Basic ordinal arithmetic. □

Lemma 16. A nonzero ordinal α is an epsilon number if and only if $\alpha \le_1 \alpha + \alpha$.

Proof. Follows from Theorem 4.16 of [9]. □

Definition 13. (Reach) If P is a pattern, a *reach* is a finite formal descending sum of P-indecomposables. If $x \in P$ is indecomposable, we say $\mathrm{reach}(x) = a_1 + \cdots + a_n$ if $a_1 + \cdots + a_n$ is the lexicomaximal reach such that $x + a_1 + \cdots + a_n$ is defined and $x \le_1 x + a_1 + \cdots + a_n$. In case there is no $y > 0^P$ such that $x \le_1 x + y$, $\mathrm{reach}(x) = 0^P$ (we identify 0^P with the empty formal descending sum). We also define $\mathrm{minreach}(x)$ to be $\min\{x, \mathrm{reach}(x)\}$. If r_1 and r_2 are reaches, write $r_1 < r_2$ to indicate that r_1 precedes r_2 lexicographically. If $r = a_1 + \cdots + a_n$, we write $(P,r)^*$ (or just r^* if P is clear from context) for $a_1^* + \cdots + a_n^*$.

Lemma 17. If P is a pattern and $x \in P$, the following are equivalent.

1. x is the leftmost indecomposable node of P, and $\mathrm{reach}(x) = 0$.

2. $x^* = 1$.

Proof. Let $P_0, P_1, \ldots$ be a fair sequence for P.

($1 \Rightarrow 2$) Given (1), it is clear that there is no way (neither by simple additive extensions nor by downward reflections) to add a point between 0^P and x. The order type of $[0,x)^{\cup P_i}$ is the order type of $\{0\}$, namely 1.

($2 \Rightarrow 1$) If $x^* = 1$ then at no step in $P_0, P_1, \ldots$ is there a point between 0^P and x. In particular, x is the leftmost nonzero node of P, hence indecomposable. If $\mathrm{reach}(x) > 0$, the fair sequence would be obligated, at some step, to reflect $[x, x + \mathrm{reach}(x))$ below x; this does not happen, so $\mathrm{reach}(x) = 0$. □

Based on Lemma 17, if P is a pattern, we write 1^P for the leftmost indecomposable of P, provided it has reach 0. Otherwise (or if P has no indecomposable), 1^P is undefined.

Lemma 18. For (P, x) a pointed pattern, the following are equivalent.

1. $\mathrm{minreach}(x) = x$.
2. $x \le_1 x + x$.
3. x^* is an epsilon number.

Proof. By Lemma 16. □

Lemma 19. If (P, x) is a pointed pattern and x is decomposable, then $\mathrm{reach}(x) = 0$.

Proof. The definition of a pattern (Definition 3) forbids $x <_1 y$ when x is decomposable. □

Definition 14. The *index* of an ordinal α, written $\mathrm{Index}(\alpha)$, is the order type of the indecomposable ordinals below α. If (P, x) is a pointed pattern, we will abuse notation and write $\mathrm{Index}(P, x)$ (or even simply $\mathrm{Index}(x)$ if P is clear) for $\mathrm{Index}(x^*)$.

Lemma 20. If α is an indecomposable ordinal, then $\alpha = \omega^{\mathrm{Index}(\alpha)}$.

Proof. Follows from Lemma 15. □

Corollary 21. For any nonzero ordinal α, α is an epsilon number if and only if $\mathrm{Index}(\alpha) = \alpha$.

If α is an ordinal, we will sometimes write $\exp(\alpha)$ for ω^α. If (P, x) is a pointed pattern, we will abuse notation and write $\exp(P, x)$ (or even just $\exp(x)$) for $\exp(x^*)$. If $r = a_1 + \cdots + a_n$ is a reach in P, we will increase our abuse and write $\exp(r)$ for $\exp(r^*)$. Index, reach, and exp are related by the following theorem.

Theorem 22. Suppose P is a pattern, $x < y \in P$, x is indecomposable, and there is no indecomposable strictly between x and y in P. Then $\mathrm{Index}(y) = \mathrm{Index}(x) + \exp(\mathrm{minreach}(y))$.

Proof. Let $r = \mathrm{minreach}(y)$. We induct on r^*. Let $P = P_0, P_1, \ldots$ be a fair sequence for P and let $P_\infty = \cup_i P_i$. By the Interval Theorem (Theorem 11), the indecomposables in $[x^*, y^*)$ are exactly the ordinals $\{z^* : z \in [x, y)^{P_\infty} \text{ is indecomposable}\}$.

Case 1: $r^* = 0$. Hence $\mathrm{reach}(y) = 0$. There are no indecomposable $z \in (x, y)^{P_\infty}$: if there were, there would be such a $z \in (x, y)^{P_{n+1}}$ for some minimal n, but the only way to add (via immediate extension) an indecomposable to $(x, y)^{P_n}$ (given that there are no indecomposables already in $(x, y)^{P_n}$) would be to reflect some set X below y, impossible when $\mathrm{reach}(y) = 0$. Hence, since x itself is indecomposable, $\mathrm{Index}(y) = \mathrm{Index}(x) + 1 = \mathrm{Index}(x) + \exp(0)$.

Case 2: $r^* = y^*$. By Lemma 18, y^* is an epsilon number. Thus

$$\begin{aligned}
\mathrm{Index}(y) &= y^* && \text{(Corollary 21)}\\
&= \exp(y) && (y^* \text{ is an epsilon number})\\
&= \mathrm{Index}(x) + \exp(y) && (\text{Clearly } \mathrm{Index}(x) < y^*)\\
&= \mathrm{Index}(x) + \exp(\mathrm{minreach}(y)). && \text{(Lemma 18)}
\end{aligned}$$

Case 3: $0 < r^* < y^*$. So $\mathrm{reach}(y) = r$. Let

$$R = \{n > 0 \,:\, P_n \text{ is obtained from } P_{n-1} \text{ by reflecting a set below } y\}.$$

For each $n \in R$, let $z_{n1}, \ldots, z_{nk_n}$ be (see Fig. 8) the indecomposables (in ascending order) in $P_n \backslash P_{n-1}$, with

$$\mathrm{minreach}(z_{n1}) = r_{n1},\ \ \ldots,\ \mathrm{minreach}(z_{nk_n}) = r_{nk_n}.$$

Evidently each $r_{n\ell} < r$. Observe that for any two consecutive $m < n \in R$, z_{mk_m} and z_{n1} are consecutive indecomposables in P_n.

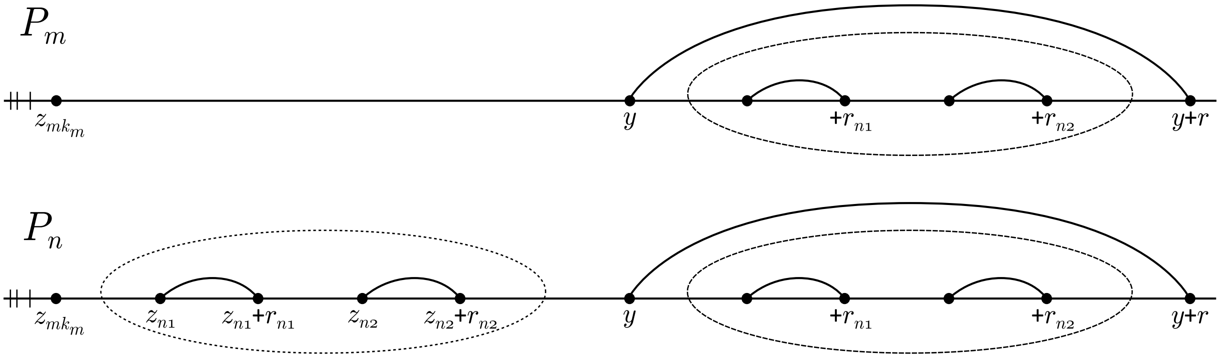


Figure 8: Each $r_{n\ell} < r$, providing a foothold for transfinite induction.

By k_n applications of the induction hypothesis,

$$\mathrm{Index}(z_{nk_n}) = \mathrm{Index}(z_{mk_m}) + \exp(r_{n1}) + \cdots + \exp(r_{nk_n}).$$

If $m \neq \min(R)$, we may repeat the above argument to unravel $\mathrm{Index}(z_{mk_m})$; if $m = \min(R)$, we may repeat the above argument substituting x for z_{mk_m}. Thus

$$\mathrm{Index}(z_{nk_n}) = \mathrm{Index}(x) + \sum_{\substack{m \in R \\ m \leq n}} \sum_{\ell=1}^{k_m} \exp(r_{m\ell}).$$

Since the $z_\bullet$ are cofinal below y in P_∞, each indecomposable ordinal below y^* is accounted for in the above sum for some n large enough. Thus the order type of indecomposables in $[0, y^*)$ is of the form $\mathrm{Index}(x) + \sum_p \exp(r_p)$ where every $r_p < r$. The fact that the sequence $P_0, P_1, \ldots$ is fair implies that the r_p are themselves cofinal below r, thus $\sum_p \exp(r_p) = \exp(r)$. □

In the definition of syntactic exact generation, we recalled two rules for generating new patterns from old: simple additive extension, and downward reflection. Carlson proved these are exhaustive, in the sense of Theorem 9. We now introduce a third rule, *minreach insertion*, useful for arithmetic.

Theorem 23. (Minreach Insertion) (See Fig. 9) Let P be a pattern, $x \in P$, and let $r = a_1 + \cdots + a_n$ be a reach ($a_i \in P$ indecomposable). Suppose that for every $z > x$ in P with $\mathrm{minreach}(z) = r$, there is $x < z' < z$ in P with $\mathrm{minreach}(z') > r$. Let $y \geq x$ be the largest node in P such that for all $x < y' \leq y$ in P, $\mathrm{minreach}(y') < r$. Let P^+ be obtained from P by inserting $n+1$ new points

$$z,\ z + a_1,\ z + a_1 + a_2,\ \ldots,\ z + a_1 + \cdots + a_n,$$

in that order (with respect to $\leq$), directly after y; let $\leq_1^{P^+}$ extend $\leq_1^P$ so that $\mathrm{reach}(z) = r$ and so that for all $u < z$ and all $v \in [z, z+r]$, $u \leq_1 v$ iff $u \leq_1 w$ for some $w > z + r$. Then P^+ is a pattern, $\mathrm{minreach}(z) = r$, and for every $q \in P$, $(P, q)^* = (P^+, q)^*$.

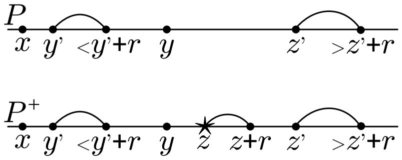


Figure 9: Minreach Insertion.

Proof. That P^+ is a pattern is straightforward. We divide the rest of the proof into two claims.

Claim 1 minreach$(z) = r$.

Assume not. By construction, reach$(z) = r$. Thus $r = a_1 + \cdots + a_n > z$. Since z is an indecomposable unequal to a_1, this implies $a_1 > z$. In particular, y is not the last point in P. Let y_2 be the next point after y in P. Since $a_1 > z$ and $z > y$, $y_2 \leq a_1$.

Case 1: $r > a_1$. Then minreach$(y_2) \leq y_2 \leq a_1 < r$, violating maximality of y.

Case 2: $r = a_1 =$ minreach(y_2). This violates the supposition of the theorem: $y_2 > x$ is an element of P with minreach r and no element of $(x, y_2)^P$ has minreach $> r$.

Case 3: $r = a_1 \neq$ minreach(y_2). By maximality of y, minreach$(y_2) \geq r$, so minreach$(y_2) > r$. This implies $y_2 \geq r = a_1$, so $y_2 = a_1$. We have minreach$(r) =$ minreach$(a_1) =$ minreach$(y_2) > r$, absurd since minreach$(r) \leq r$.

Claim 2 For all $q \in P$, $(P, q)^* = (P^+, q)^*$.

Fix $q \in P$, and let $P^+ = P_0, P_1, \ldots$ be a fair sequence for P^+.

Case 1: $z > \max(P)$. Because of how we defined $\leq_1^{P^+}$, for all $q \in P$, $q \not\leq_1^{P^+} z$. For all $i \in \mathbb{N}$, if P_{i+1} is formed from P_i by an immediate extension involving any part of P_i that is $> \max(P)$, it is clear (by considering simple additive extension and downward reflection separately) that the new nodes in $P_{i+1} \backslash P_i$ will also lie above $\max(P)$. Thus, all the points $\leq \max(P)$ in $\cup_i P_i$ could just as well be created by a fair sequence for P itself. By Syntax-Semantics Equivalence, since $q \in P$, this implies $(P, q)^* = (P^+, q)^*$.

Case 2 (See Fig. 10): $z \leq \max(P)$. So $z < \max(P)$ since $z \notin P$. Let y_2 be the next point in P after y (so $y < z < y_2$). By choice of y, minreach$(y_2) \geq r$, in fact by the theorem's hypothesis, minreach$(y_2) > r$. So certainly reach$(y_2) > r$. We will show P generates P^+, so that $(P^+, q)^* = (P, q)^*$.

Let P' be obtained by adding (as needed) $y_2 + a_1$, $\ldots$, $y_2 + r$ to P. Then P exactly generates P' via simple additive extensions. Let $Y = \{y_2, y_2 + a_1, \ldots, y_2 + r\}$. Since reach$(y_2) > r$, $Y \subseteq [y_2, y_2 + \text{reach}(y_2))^{P'}$. Thus we may reflect Y below y_2 to exactly generate a new pattern P^{++}. By construction, the copy $\widetilde{Y}$ of Y that we add below y_2 is isomorphic to—and so we can assume equal to—the points we add to P in the construction of P^+. So P^+ is a subpattern of P^{++}. Since P exactly generates P^{++}, this shows that P generates P^+. □

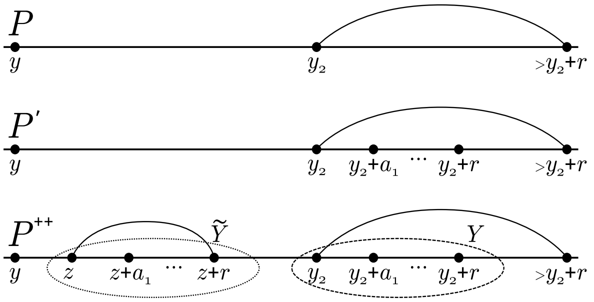


Figure 10: Emulating minreach insertion.

Corollary 24. There is an algorithm that takes as input a pattern P such that 1^P is undefined and outputs a pattern P' such that

1. $1^{P'}$ is defined.

2. $P' \backslash P = \{1^{P'}\}$.

3. If P has at least one nonzero element, then P generates P'.

Proof. By Lemma 17, and Theorem 23 with $x = 0^P$ and $r = 0$. □

4 Multiplication

In the proofs below, we intend to get our hands dirty with minreaches, so the following notation will be useful.

Notation 15. If P is a pattern, $x \in P$, we will write $x^\frown$ for minreach(x).

Definition 16. The *node multiplication algorithm* is as follows.

1. Input: Two pointed patterns (P, x) and (P, y) such that 1^P is defined and x, y are indecomposable.
2. Let ℓ ("left") be 1^P and let r ("right") be x. Let P_+ be a copy of P.

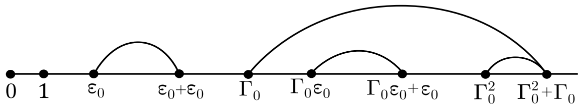


Figure 11: Result of multiplying Γ_0 by ϵ_0.

3. If $\ell < y$ then
   (a) Let ℓ_+ be the next indecomposable in P after ℓ.
   (b) Using Lemma 23 to enlarge P_+ if necessary (without changing which ordinals its nodes notate), let $r_+ > r$ be a P_+-indecomposable such that $r_+^{\frown} = \ell_+^{\frown}$ and such that $q^{\frown} < \ell_+^{\frown}$ for all indecomposable $r < q < r_+$ in P_+.
   (c) Let $\ell = \ell_+$, let $r = r_+$, and goto 3.
4. Output (P_+, r).

Theorem 25. Let P, x, y be as in Definition 16. The node multiplication algorithm halts on input (P, x), (P, y). If (P^+, r) is its output, then $r^* = x^* y^*$.

Proof. The algorithm halts because in line 3a, we let ℓ_+ be the next indecomposable in P, not in P_+. There are only finitely many indecomposables in P (and we never enlarge P), so eventually $\ell = y$.

For the rest of the proof, I claim that every time the algorithm hits line 3, $r^* = x^* \ell^*$. The algorithm halts and outputs (P_+, r) when $\ell = y$ so this will prove the theorem. The claim certainly holds the first time we hit line 3, when $\ell = 1^P$ and $r = x$.

Suppose that $r^* = x^* \ell^*$ when we hit line 3 for the nth time. Let P', r_+, ℓ_+ be the values of the variables when we hit line 3 for the $(n+1)$th time, we will show $r_+^* = x^* \ell_+^*$.

Let $r = r_1, \ldots, r_k = r_+$ list the indecomposables, in order, from r to r_+ in P'. By repeated applications of Theorem 22,

$$\text{Index}(r_+) = \text{Index}(r_1) + \exp(r_2^{\frown}) + \cdots + \exp(r_k^{\frown}).$$

By choice of $r_+ = r_k$, $r_i^{\frown} < r_k^{\frown}$ for $i = 2, \ldots, k-1$. Thus by ordinal arithmetic,

$$\begin{aligned} \text{Index}(r_+) &= \text{Index}(r_1) + \exp(r_k^{\frown}) \\ &= \text{Index}(r) + \exp(r_+^{\frown}). \end{aligned}$$

Now we compute:

$$\begin{aligned}
x^* \ell_+^* &= x^* \exp(\text{Index}(\ell_+)) && \text{(Lemma 20)} \\
&= x^* \exp(\text{Index}(\ell) + \exp(\ell_+^{\frown})) && \text{(Theorem 22)} \\
&= x^* \exp(\text{Index}(\ell)) \exp(\exp(\ell_+^{\frown})) && \text{(Ordinal arithmetic)} \\
&= x^* \ell^* \exp(\exp(\ell_+^{\frown})) && \text{(Lemma 20)} \\
&= r^* \exp(\exp(\ell_+^{\frown})) && \text{(By assumption } x^*\ell^* = r^*) \\
&= r^* \exp(\exp(r_+^{\frown})) && \text{(By choice of } r_+ \text{ in line 3b of Def. 16)} \\
&= \exp(\text{Index}(r) + \exp(r_+^{\frown})) && \text{(Lemma 20 and ordinal arithmetic)} \\
&= \exp(\text{Index}(r_+)) && \text{(By the above discussion)} \\
&= r_+^*, && \text{(Lemma 20)}
\end{aligned}$$

as desired. □

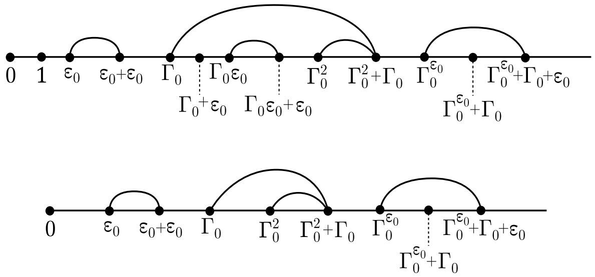
Corollary 26. (The Multiplication Algorithm) Suppose (P, x) and (Q, y) are pointed patterns. The following algorithm can be used to compute (a pattern that notates) $x^* y^*$.

1. If $x = 0^P$ or $y = 0^Q$, output $(\{0\}, 0)$ and stop. If $y = 1^Q$ (see Lemma 17), output (P, x) and stop.
2. Using Theorem 12 and Corollary 24, ensure that $P = Q$ and that 1^P is defined.
3. If $y = b_1 + \cdots + b_n$ is decomposable ($b_1, \ldots, b_n$ a decreasing sequence of indecomposables), recursively compute pointed patterns $(B_1, z_1), \ldots, (B_n, z_n)$ notating $x^* b_1^*, \ldots, x^* b_n^*$ respectively. Use Theorem 12 to ensure $B_1 = \cdots = B_n$ (call it B). If necessary, extend B via simple additive extensions so that $z_1 + \cdots + z_n$ is defined. Output $(B, z_1 + \cdots + z_n)$ and stop.
4. If $x = a_1 + \cdots + a_n$ is decomposable ($a_1, \ldots, a_n$ a descending sequence of indecomposables), recursively output a pointed pattern that notates $a_1^* y^*$ and stop.
5. If this step is reached, x and y are both indecomposable. Employ the node multiplication algorithm (Definition 16).

Proof. Elementary ordinal arithmetic and Theorem 25. □

The multiplication algorithm is implemented in [1] via the *mult* command.

5 Exponentiation



0
1
ε_0
$\varepsilon_0+\varepsilon_0$
Γ_0
$\Gamma_0\varepsilon_0$
Γ_0^2
$\Gamma_0^2+\Gamma_0$
$\Gamma_0^{\varepsilon_0}$
$\Gamma_0^{\varepsilon_0}+\Gamma_0+\varepsilon_0$
$\Gamma_0+\varepsilon_0$
$\Gamma_0\varepsilon_0+\varepsilon_0$
$\Gamma_0^{\varepsilon_0}+\Gamma_0$
0
ε_0
$\varepsilon_0+\varepsilon_0$
Γ_0
Γ_0^2
$\Gamma_0^2+\Gamma_0$
$\Gamma_0^{\varepsilon_0}$
$\Gamma_0^{\varepsilon_0}+\Gamma_0+\varepsilon_0$
$\Gamma_0^{\varepsilon_0}+\Gamma_0$

Figure 12: Above, the result of computing $\Gamma_0^{\epsilon_0} = \exp(\Gamma_0\epsilon_0)$. Below, the reduction to normal form (i.e., the simplification) using an algorithm related to [7].

Definition 17. The (base ω) *exponentiation algorithm* and the *index algorithm* are defined simultaneously in terms of one another as follows. Both take as input a pointed pattern (P, x) (by Corollary 24 we can assume 1^P is defined).

- (Exponentiation Algorithm)
  1. If $x = 0^P$, output $(\{0, 1\}, 1)$ and stop. If $x = 1^P$, output $(\{0, 1, \omega\}, \omega)$ (or any other fixed notation for ω) and stop.
  2. If x^* is an epsilon number (see Lemma 16), output (P, x) and stop.
  3. If x is decomposable, say $x = a_1 + \cdots + a_n$ where the a_i are decreasing indecomposables, recursively compute pointed patterns notating $\exp(a_1), \ldots, \exp(a_n)$, use Corollary 26 to output their product, and stop.
  4. Use the index algorithm (below) to compute a pointed pattern (Q, y) notating $\mathrm{Index}(x)$. Using Theorem 12, ensure $P = Q$. Abuse notation and identify y and $\mathrm{Index}(x)$.
  5. Using Lemma 23 to enlarge P if needed, let z be a node in P such that $z^{\frown} = \mathrm{Index}(x)$ and $z_0^{\frown} < \mathrm{Index}(x)$ for all $z_0 < z$. Output (P, z) and stop.
- (Index Algorithm)
  1. If $x = 0^P$ or $x = 1^P$, output $(\{0\}, 0)$ and stop.
  2. If x^* is an epsilon number (see Lemma 16), output (P, x) and stop.
  3. Using simple additive extensions if necessary, ensure P contains $\mathrm{reach}(x)$ as a point.
  4. Using the index algorithm (recursively) and the exponentiation algorithm (above), along with the addition algorithm (Proposition 14), output a pointed pattern notating $\mathrm{Index}(x_0) + \exp(\mathrm{reach}(x))$, where x_0 is the largest P-indecomposable less than x.

Theorem 27. Let (P, x) be a pointed pattern (notating ordinal α). The exponentiation algorithm and the index algorithm halt on input (P, x). Call their outputs (Q, y) and (R, z), respectively. Then $y^* = \omega^\alpha$ and $z^* = \mathrm{Index}(\alpha)$.

Proof. By induction on α. If $\alpha \leq 1$, the theorem is clear, assume $\alpha > 1$, so $x > 1^P$.

Claim 1 The index algorithm halts on (P, x) and its output notates $\mathrm{Index}(\alpha)$.

If x^* is an epsilon number, the claim follows by Lemma 21. Assume not. Then by Lemma 21, $\mathrm{reach}(x) < x$ (so $\mathrm{reach}(x)^* < \alpha$). By induction, the exponentiation algorithm halts on input $(P, \mathrm{reach}(x))$ and its output notates $\exp(\mathrm{reach}(x))$. Let x_0 be the largest P-indecomposable less than x (x_0 exists since $x > 1^P$). By induction, the index algorithm halts on input (P, x_0) and its output notates $\mathrm{Index}(x_0)$. Thus, the index algorithm halts on (P, x). By construction, its output notates $\mathrm{Index}(x_0) + \exp(\mathrm{reach}(x))$, the same as $\mathrm{Index}(x_0) + \exp(x^\frown)$ (since $\mathrm{reach}(x) < x$). By Theorem 22, $\mathrm{Index}(x_0) + \exp(x^\frown) = \mathrm{Index}(x)$ as desired.

Claim 2 The exponentiation algorithm halts on (P, x) and its output notates ω^α.

Case 1: α is decomposable, hence so is x, write $x = a_1 + \cdots + a_n$, (a_i) a decreasing sequence of P-indecomposables, each $a_i < x$, so $a_i^* < \alpha$. By induction, the exponentiation algorithm behaves correctly on each input (P, a_i), and the claim follows by basic ordinal arithmetic.

Case 2: α is an epsilon number. Then the claim follows by Lemma 21.

Case 3: α is a non-epsilon indecomposable. By Claim 1, the index algorithm halts on input (P, x) and its output notates $\mathrm{Index}(x)$. Thus the z described in line 5 of the exponentiation algorithm really does have the properties it is constructed to have. Let $1^P = z_1, \ldots, z_k = z$ list the indecomposables in P (after possibly enlarging it as described in line 5), in ascending order. By repeated applications of Lemma 22,

$$\mathrm{Index}(z) = \mathrm{Index}(1^P) + \exp(z_2^\frown) + \cdots + \exp(z_{k-1}^\frown) + \exp(z^\frown).$$

By choice of z, $z_i^\frown < z^\frown$ for all $2 \leq i < k$, so by ordinal arithmetic,

$$\begin{aligned}
\mathrm{Index}(z) &= \mathrm{Index}(1^P) + \exp(z^\frown) \\
&= 0 + \exp(z^\frown) \\
&= \exp(\mathrm{Index}(x)). && \text{(Since } z^\frown = \mathrm{Index}(x)\text{)}
\end{aligned}$$

Thus

$$\begin{aligned}
z^* &= \exp(\mathrm{Index}(z)) && \text{(Lemma 20 applied to } z^*\text{)} \\
&= \exp(\exp(\mathrm{Index}(x))) \\
&= \exp(x), && \text{(Lemma 20 applied to } x^*\text{)}
\end{aligned}$$

as desired. □

Corollary 28. (An algorithm for the base-ω logarithm) Given a pointed pattern (P, x) (by Corollary 24 we may assume 1^P is defined), if $x^* = \omega^\alpha$ for some α, then the following algorithm will output a pointed pattern notating α, and otherwise the following algorithm will output an error message.

1. If x is not indecomposable, output an error message and stop.
2. Output a pattern notating $\mathrm{Index}(x)$ (using the index algorithm) and stop.

Proof. By Theorem 27 and Lemma 20. □

Base-ω exponentiation and logarithm are implemented (and automatically simplified) in [1] via the *exp* and *log* commands.

6 Future work

In future work, we would like to publish algorithms for the epsilon function $\alpha \mapsto \epsilon_\alpha$, the Gamma function $\alpha \mapsto \Gamma_\alpha$, the Veblen function $(\alpha, \beta) \mapsto \varphi\alpha\beta$, and other ordinal arithmetical functions of interest. We would also like to give algorithms in terms of second-order patterns of resemblance [6].

References

[1] Alexander, S. (2012) The Patterns of Resemblance Ordinal Calculator. `http://www.xamuel.com/patterns/`

[2] Bès, A. (2002) Decidability and definability results related to the elementary theory of ordinal multiplication. *Fundamenta Mathematicae* **171**, 197–211.

[3] Carlson, T.J. (1999) Ordinal arithmetic and Σ_1 elementarity. *Archive for Mathematical Logic*, **38**, 449–460.

[4] Carlson, T.J. (2000) Knowledge, machines, and the consistency of Reinhardt's strong mechanistic thesis. *Annals of Pure and Applied Logic*, **105**, 51–82.

[5] Carlson, T.J. (2001) Elementary patterns of resemblance. *Annals of Pure and Applied Logic* **108**(1), 19–77.

[6] Carlson, T.J. (2009) Patterns of resemblance of order 2. *Annals of Pure and Applied Logic* **158**(1–2), 90–124.

[7] Carlson, T.J. & Wilken, G. (2012) Normal forms for elementary patterns. *Journal of Symbolic Logic* **77**(1), 174–194.

[8] Wilken, G. (2004) Σ_1-elementarity and Skolem hull operators. Thesis (Dissertation), University of Münster.

[9] Wilken, G. (2006) The Bachmann-Howard Structure in Terms of Σ_1-Elementarity. *Arch. Math. Logic* **45**(7), 807–829.

[10] Wilken, G. (2007) Assignment of ordinals to elementary patterns of resemblance. *Journal of Symbolic Logic* **72**(2), 704–720.