The Taylor Series Part I

Fun And Games With Ordinals

Chris Taylor

Ordinals

Definition 1

Recall that an *ordinal* is a set x that is both transitive and well-ordered by $\in$. That is, x is an ordinal if:

1. if $x \in y$ and $y \in z$ then $z \in x$ (transitivity), and
2. every non-empty subset of x has a minimum element (well-ordered).

Ordinals

Definition 1

Recall that an *ordinal* is a set x that is both transitive and well-ordered by $\in$. That is, x is an ordinal if:

1. if $x \in y$ and $y \in z$ then $z \in x$ (transitivity), and
2. every non-empty subset of x has a minimum element (well-ordered).

Definition 2

Let α be an ordinal. The *successor* of α is the ordinal $S(\alpha) = \alpha \cup \{\alpha\}$. If $\alpha = S(\beta)$ for some ordinal β then α is a *successor ordinal*, and otherwise it is a *limit ordinal*.

Ordinals

Definition 1

Recall that an *ordinal* is a set x that is both transitive and well-ordered by $\in$. That is, x is an ordinal if:

1. if $x \in y$ and $y \in z$ then $z \in x$ (transitivity), and
2. every non-empty subset of x has a minimum element (well-ordered).

Definition 2

Let α be an ordinal. The *successor* of α is the ordinal $S(\alpha) = \alpha \cup \{\alpha\}$. If $\alpha = S(\beta)$ for some ordinal β then α is a *successor ordinal*, and otherwise it is a *limit ordinal*.

Examples will appear in chalk.

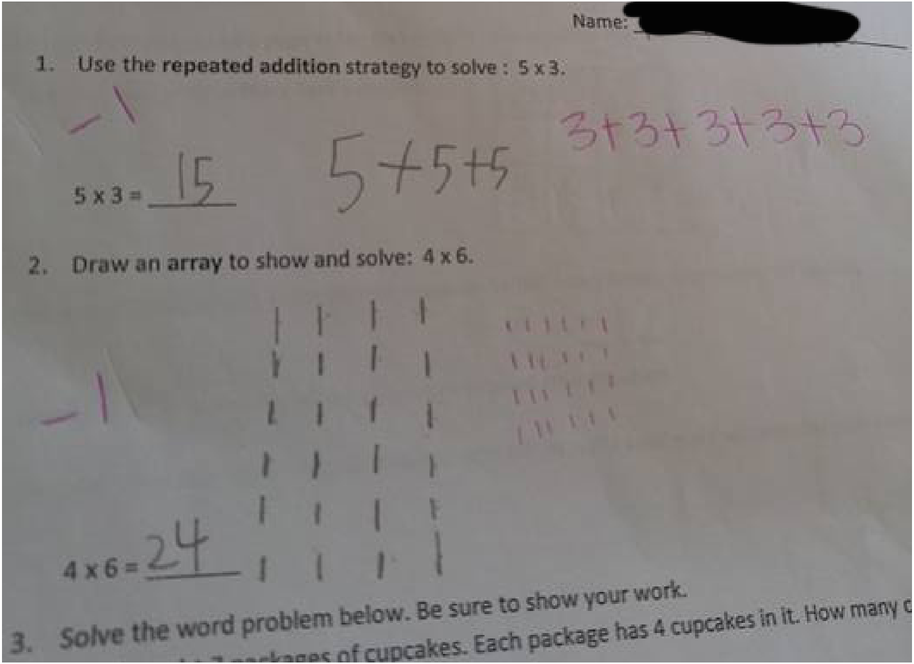
Ordinal arithmetic

Definition 3

$$\beta + \alpha = \begin{cases} \beta & \text{if } \alpha = 0 \\ S(\beta + \gamma) & \text{if } \alpha = S(\gamma) \\ \bigcup\{\beta + \gamma \mid \gamma < \alpha\} & \text{otherwise.} \end{cases}$$

$$\beta \cdot \alpha = \begin{cases} 0 & \text{if } \alpha = 0 \\ \beta \cdot \gamma + \beta & \text{if } \alpha = S(\gamma) \\ \bigcup\{\beta \cdot \gamma \mid \gamma < \alpha\} & \text{otherwise.} \end{cases}$$

$$\beta^\alpha = \begin{cases} 1 & \text{if } \alpha = 0 \\ \beta^\gamma \cdot \beta & \text{if } \alpha = S(\gamma) \\ \bigcup\{\beta^\gamma \mid \gamma < \alpha\} & \text{otherwise.} \end{cases}$$


Name: [redacted]

1. Use the repeated addition strategy to solve : 5 x 3.

-1

5 x 3 = 15   5+5+5   3+3+3+3+3

2. Draw an array to show and solve: 4 x 6.

-1

4 x 6 = 24

3. Solve the word problem below. Be sure to show your work.

... 7 packages of cupcakes. Each package has 4 cupcakes in it. How many c

Cantor Normal Form

Definition 4

Let ϵ_0 denote the smallest ordinal α such that $\omega^\alpha = \alpha$.

Cantor Normal Form

Definition 4

Let ϵ_0 denote the smallest ordinal α such that $\omega^\alpha = \alpha$.

Theorem (Cantor Normal Form)

Let α be an ordinal. Then there exists natural numbers $n_1, \ldots, n_k$ and ordinals $\beta \geq \alpha_1 > \cdots > \alpha_k \geq 0$ such that

$$\beta = \omega^{\alpha_1} \cdot n_1 + \cdots + \omega^{\alpha_k} \cdot n_k.$$

Moreover, if $\beta < \epsilon_0$ then $\alpha_1 \neq \beta$.

Peano Arithmetic

Definition 5 (First-order Peano arithmetic)

- Language: $\langle +, \cdot, 0 \rangle$.

Peano Arithmetic

Definition 5 (First-order Peano arithmetic)

- Language: $\langle +, \cdot, 0 \rangle$.
- Axioms:
  1. $0 \neq S(x)$.
  2. $S(x) = S(y) \implies x = y$.
  3. $x + 0 = x$.
  4. $x + S(y) = S(x + y)$.
  5. $x \cdot 0 = 0$.
  6. $x \cdot S(y) = x \cdot y + x$.

Peano Arithmetic

Definition 5 (First-order Peano arithmetic)

- Language: $\langle +, \cdot, 0 \rangle$.
- Axioms:
  1. $0 \neq S(x)$.
  2. $S(x) = S(y) \implies x = y$.
  3. $x + 0 = x$.
  4. $x + S(y) = S(x + y)$.
  5. $x \cdot 0 = 0$.
  6. $x \cdot S(y) = x \cdot y + x$.
- Induction schema:

$$\forall \overline{y}\ (\varphi(0, \overline{y}) \wedge \forall x\ (\varphi(x, \overline{y}) \to \varphi(S(x), \overline{y})) \to \forall x\ \varphi(x, \overline{y})).$$

  for all formulas φ.

Recursive functions

Definition 6 (Initial functions)

The *initial functions* are the following three classes of functions.

Recursive functions

Definition 6 (Initial functions)

The *initial functions* are the following three classes of functions.

1. The *constant functions*. For all $n, k \geq 0$, the function $f\colon \mathbb{N}^k \to \mathbb{N}$ given by
$$f(x_1, \ldots, x_k) = n.$$

Recursive functions

Definition 6 (Initial functions)

The *initial functions* are the following three classes of functions.

1. The *constant functions*. For all $n, k \geq 0$, the function $f\colon \mathbb{N}^k \to \mathbb{N}$ given by
$$f(x_1, \ldots, x_k) = n.$$
2. The *successor function* $S\colon \mathbb{N} \to \mathbb{N}$, seen before.

Recursive functions

Definition 6 (Initial functions)

The *initial functions* are the following three classes of functions.

1. The *constant functions*. For all $n, k \geq 0$, the function $f\colon \mathbb{N}^k \to \mathbb{N}$ given by
$$f(x_1, \ldots, x_k) = n.$$
2. The *successor function* $S\colon \mathbb{N} \to \mathbb{N}$, seen before.
3. The *projection functions*. For every $i, k \geq 1$ such that $i \leq k$, the function $\pi_i^k\colon \mathbb{N}^k \to N$ given by
$$\pi_i^k(x_1, \ldots, x_k) = x_i.$$

Recursive functions

Definition 7 (Primitive recursion)

A function is *primitive recursive* if it is an initial function, or if it arises from applications of *composition* and *primitive recursion* to primitive recursive functions.

Recursive functions

Definition 7 (Primitive recursion)

A function is *primitive recursive* if it is an initial function, or if it arises from applications of *composition* and *primitive recursion* to primitive recursive functions.

1. *Composition*. If $h\colon \mathbb{N}^m \to \mathbb{N}$ and $g_1, \ldots, g_m\colon \mathbb{N}^k \to \mathbb{N}$ are all functions, then $h \circ (g_1, \ldots, g_m)\colon \mathbb{N}^k \to \mathbb{N}$ is given by
$$h \circ (g_1, \ldots, g_m)(\overline{x}) = h(g_1(\overline{x}), \ldots, g_m(\overline{x})).$$

Recursive functions

Definition 7 (Primitive recursion)

A function is *primitive recursive* if it is an initial function, or if it arises from applications of *composition* and *primitive recursion* to primitive recursive functions.

1. *Composition*. If $h\colon \mathbb{N}^m \to \mathbb{N}$ and $g_1, \ldots, g_m\colon \mathbb{N}^k \to \mathbb{N}$ are all functions, then $h \circ (g_1, \ldots, g_m)\colon \mathbb{N}^k \to \mathbb{N}$ is given by
$$h \circ (g_1, \ldots, g_m)(\overline{x}) = h(g_1(\overline{x}), \ldots, g_m(\overline{x})).$$
2. *Primitive recursion*. If $g\colon \mathbb{N}^k \to \mathbb{N}$ and $h\colon N^{k+2} \to \mathbb{N}$ are functions then the primitive recursion operator produces the function $\rho_{g,h}\colon N^{k+1} \to \mathbb{N}$, defined by
$$\rho_{g,h}(0, \overline{x}) = g(\overline{x})$$
$$\rho_{g,h}(y+1, \overline{x}) = h(y, \rho_{g,h}(y, \overline{x}), \overline{x}).$$

Primitive recursive functions

The following functions are all primitive recursive:

1. Addition: $x + y$.
2. Multiplication: $x \cdot y$.
3. Exponentiation: x^y.
4. Quotient w.r.t. q: if $x = kq + r$ returns kq.
5. Remainder w.r.t. q: if $x = kq + r$ returns r.
6. Predecessor: if $x > 0$ return $x - 1$, otherwise return 0.
7. Restricted subtraction: $x \dot{-} y = \max\{0, x - y\}$
8. p_i: the $i + 1$-st prime number.
9. Primality: if x is prime return 1, else 0.
10. $\text{cap}(x) = \begin{cases} 1 & \text{if x > 0,} \\ 0 & \text{otherwise.} \end{cases}$

Primitive recursive functions

The following functions are all primitive recursive:

1. Addition: $x + y$.
2. Multiplication: $x \cdot y$.
3. Exponentiation: x^y.
4. Quotient w.r.t. q: if $x = kq + r$ returns kq.
5. Remainder w.r.t. q: if $x = kq + r$ returns r.
6. Predecessor: if $x > 0$ return $x - 1$, otherwise return 0.
7. Restricted subtraction: $x \dot{-} y = \max\{0, x - y\}$
8. p_i: the $i + 1$-st prime number.
9. Primality: if x is prime return 1, else 0.
10. $\text{cap}(x) = \begin{cases} 1 & \text{if x > 0,} \\ 0 & \text{otherwise.} \end{cases}$

...and much more!

Recursive functions

Definition 8 (μ-recursion)

A partial function is μ-*recursive* if it is an initial function, or if it arises from applications of composition, primitive recursion and *minimisation* to μ-recursive functions.

Recursive functions

Definition 8 (μ-recursion)

A partial function is *μ-recursive* if it is an initial function, or if it arises from applications of composition, primitive recursion and *minimisation* to μ-recursive functions.

1. *Minimisation*. Let $f : \mathbb{N}^{k+1} \to \mathbb{N}$ be a *proper* function. Then minisation produces the function μ_f, defined by

$$\mu_f(x_1, \ldots, x_k) = \begin{cases} \min\{z \mid f(z, x_1, \ldots, x_k) = 0\} & \text{if it exists} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Recursive relations

Henceforth, we will just say *recursive* instead of μ-recursive.

Theorem 9

Let $f\colon \mathbb{N}^n \to \mathbb{N}$ be recursive. Then there is a formula $\varphi(y, x_1, \ldots, x_n)$ in the language of P such that

1. *$f(x_1, \ldots, x_n) = y$ implies $\mathbb{N} \models \varphi(y, x_1, \ldots, x_n)$, and*
2. *$f(x_1, \ldots, x_n) \neq y$ implies $\mathbb{N} \models \neg\varphi(y, x_1, \ldots, x_n)$.*

Recursive relations

Henceforth, we will just say *recursive* instead of μ-recursive.

Theorem 9

Let $f\colon \mathbb{N}^n \to \mathbb{N}$ *be recursive. Then there is a formula* $\varphi(y, x_1, \ldots, x_n)$ *in the language of* P *such that*

1. $f(x_1, \ldots, x_n) = y$ *implies* $\mathbb{N} \models \varphi(y, x_1, \ldots, x_n)$*, and*
2. $f(x_1, \ldots, x_n) \neq y$ *implies* $\mathbb{N} \models \neg\varphi(y, x_1, \ldots, x_n)$.

Definition 10

An n-ary relation R is *recursive* if there exists a recursive function $f_R\colon \mathbb{N}^n \to \mathbb{N}$ such that

$$f_R(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } (x_1, \ldots, x_n) \in R \\ 0 & \text{otherwise.} \end{cases}$$

Recursive relations

Example 11

Let R and S be n-ary recursive relations.

$$f_R(x_1, \ldots, x_n) \cdot f_S(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } (x_1, \ldots, x_n) \in R \cap S \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{cap}(f_R(x_1, \ldots, x_n) + f_S(x_1, \ldots, x_n)) = \begin{cases} 1 & \text{if } (x_1, \ldots, x_n) \in R \cup S \\ 0 & \text{otherwise,} \end{cases}$$

$$1 \dot{-} f_R(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } (x_1, \ldots, x_n) \notin R \\ 0 & \text{otherwise.} \end{cases}$$

The usual ordering on $\mathbb{N}$ is also recursive, since

$$\text{cap}(y \dot{-} x) = \begin{cases} 1 & \text{if } x < y \\ 0 & \text{otherwise.} \end{cases}$$

Recursive ordinals

Definition 12

An ordinal α is said to be a *recursive ordinal* if there is a recursive binary relation $R_\alpha \subseteq \mathbb{N}^2$ with order type α.

Recursive ordinals

Definition 12

An ordinal α is said to be a *recursive ordinal* if there is a recursive binary relation $R_\alpha \subseteq \mathbb{N}^2$ with order type α.

Corollary 13

Let α be a recursive ordinal. Then there is a formula $\varphi(x, y)$ in the language of P such that

$$(x, y) \in R_\alpha \iff \mathbb{N} \models \varphi(x, y).$$

Recursive ordinals

Definition 12

An ordinal α is said to be a *recursive ordinal* if there is a recursive binary relation $R_\alpha \subseteq \mathbb{N}^2$ with order type α.

Corollary 13

Let α be a recursive ordinal. Then there is a formula $\varphi(x, y)$ in the language of P such that

$$(x, y) \in R_\alpha \iff \mathbb{N} \models \varphi(x, y).$$

Fact 14

Every ordinal less than or equal to ϵ_0 is recursive.

Transfinite induction

Definition 15

Let α be a recursive ordinal. We let $\mathrm{TI}(\alpha)$ denote the axiom schema

$$[\forall x(\forall y(yR_\alpha x \implies \varphi(y)) \implies \varphi(x)] \implies \forall x\varphi(x),$$

which we call *transfinite induction up to* α.

Transfinite induction

Definition 15

Let α be a recursive ordinal. We let $\mathrm{TI}(\alpha)$ denote the axiom schema

$$[\forall x(\forall y(yR_\alpha x \implies \varphi(y)) \implies \varphi(x)] \implies \forall x\varphi(x),$$

which we call *transfinite induction up to* α.

Theorem 16 (Gentzen, 1936)

If $P \vdash \mathrm{TI}(\epsilon_0)$ *then P can prove its own consistency.*

Transfinite induction

Definition 15

Let α be a recursive ordinal. We let $\mathrm{TI}(\alpha)$ denote the axiom schema

$$[\forall x(\forall y(yR_\alpha x \implies \varphi(y)) \implies \varphi(x)] \implies \forall x\varphi(x),$$

which we call *transfinite induction up to* α.

Theorem 16 (Gentzen, 1936)

If $P \vdash \mathrm{TI}(\epsilon_0)$ *then P can prove its own consistency.*

Theorem 17 (Gödel's second incompleteness theorem)

If P can prove its own consistency then P is inconsistent.

Gentzen's Theorem

Gentzen defines a notion of "reduction procedure" for proofs in Peano arithmetic. For a given proof, such a procedure produces a tree of proofs, with the given one serving as the root of the tree, and the other proofs being, in a sense, "simpler" than the given one. This increasing simplicity is formalized by attaching an ordinal less than ϵ_0 to every proof, and showing that, as one moves down the tree, these ordinals get smaller with every step. He then shows that if there were a proof of a contradiction, the reduction procedure would result in an infinite descending sequence of ordinals below ϵ_0, produced by a primitive recursive operation on proofs corresponding to a quantifier-free formula. (*"Gentzen's consistency proof", Wikipedia*)

True but not provable

ACCESSIBLE INDEPENDENCE RESULTS FOR PEANO ARITHMETIC

LAURIE KIRBY AND JEFF PARIS

Recently some interesting first-order statements independent of Peano Arithmetic (P) have been found. Here we present perhaps the first which is, in an informal sense, purely number-theoretic in character (as opposed to metamathematical or combinatorial). The methods used to prove it, however, are combinatorial. We also give another independence result (unashamedly combinatorial in character) proved by the same methods.

Some notation

Definition 18

Let n be a natural number. The *hereditary base b notation* of n is defined recursively. If $n < b$, then write n as n. Otherwise, write n in base b, with all the exponents also written in hereditary base b notation.

Some notation

Definition 18

Let n be a natural number. The *hereditary base b notation* of n is defined recursively. If $n < b$, then write n as n. Otherwise, write n in base b, with all the exponents also written in hereditary base b notation.

Example 19

If $n = 25$ and $b = 2$, we have

$$25 = 2^4 + 2^3 + 1.$$

Some notation

Definition 18

Let n be a natural number. The *hereditary base b notation* of n is defined recursively. If $n < b$, then write n as n. Otherwise, write n in base b, with all the exponents also written in hereditary base b notation.

Example 19

If $n = 25$ and $b = 2$, we have

$$25 = 2^{2^2} + 2^3 + 1.$$

Some notation

Definition 18

Let n be a natural number. The *hereditary base b notation* of n is defined recursively. If $n < b$, then write n as n. Otherwise, write n in base b, with all the exponents also written in hereditary base b notation.

Example 19

If $n = 25$ and $b = 2$, we have

$$25 = 2^{2^2} + 2^{2+1} + 1.$$

Some notation

Definition 18

Let n be a natural number. The *hereditary base b notation* of n is defined recursively. If $n < b$, then write n as n. Otherwise, write n in base b, with all the exponents also written in hereditary base b notation.

Example 19

If $n = 25$ and $b = 2$, we have

$$25 = 2^{2^2} + 2^{2+1} + 1.$$

If $n = 45955$ and $b = 3$, we have

$$45955 = 3^9 \cdot 2 + 3^8 + 3^3 + 1$$

Some notation

Definition 18

Let n be a natural number. The *hereditary base b notation* of n is defined recursively. If $n < b$, then write n as n. Otherwise, write n in base b, with all the exponents also written in hereditary base b notation.

Example 19

If $n = 25$ and $b = 2$, we have

$$25 = 2^{2^2} + 2^{2+1} + 1.$$

If $n = 45955$ and $b = 3$, we have

$$45955 = 3^{3^2} \cdot 2 + 3^8 + 3^3 + 1$$

Some notation

Definition 18

Let n be a natural number. The *hereditary base b notation* of n is defined recursively. If $n < b$, then write n as n. Otherwise, write n in base b, with all the exponents also written in hereditary base b notation.

Example 19

If $n = 25$ and $b = 2$, we have

$$25 = 2^{2^2} + 2^{2+1} + 1.$$

If $n = 45955$ and $b = 3$, we have

$$45955 = 3^{3^2} \cdot 2 + 3^{3\cdot 2+2} + 3^3 + 1$$

Goodstein's sequence

Definition 20

Let $\mathrm{Rep}(x, a, b)$ denote the result of writing x in hereditary base a notation, then replacing every a with a b.

Example 21

$$25 = 2^{2^2} + 2^{2+1} + 1, \text{so}$$

Goodstein's sequence

Definition 20

Let $\text{Rep}(x, a, b)$ denote the result of writing x in hereditary base a notation, then replacing every a with a b.

Example 21

$$25 = 2^{2^2} + 2^{2+1} + 1, \text{so}$$
$$\text{Rep}(25, 2, 3) = 3^{3^3} + 3^{3+1} + 1 = 7625597485069,$$

Goodstein's sequence

Definition 20

Let $\mathrm{Rep}(x, a, b)$ denote the result of writing x in hereditary base a notation, then replacing every a with a b.

Example 21

$$25 = 2^{2^2} + 2^{2+1} + 1, \text{so}$$
$$\mathrm{Rep}(25, 2, 3) = 3^{3^3} + 3^{3+1} + 1 = 7625597485069,$$
$$45955 = 3^9 \cdot 2 + 3^8 + 3^3 + 1, \text{so}$$

Goodstein's sequence

Definition 20

Let $\mathrm{Rep}(x, a, b)$ denote the result of writing x in hereditary base a notation, then replacing every a with a b.

Example 21

$$\begin{aligned} 25 &= 2^{2^2} + 2^{2+1} + 1, \text{so} \\ \mathrm{Rep}(25, 2, 3) &= 3^{3^3} + 3^{3+1} + 1 = 7625597485069, \\ 45955 &= 3^9 \cdot 2 + 3^8 + 3^3 + 1, \text{so} \\ \mathrm{Rep}(45955, 3, 5) &= 5^{5^2} \cdot 2 + 5^{5\cdot 2+2} + 5^5 + 1 = 596046447998050001. \end{aligned}$$

Goodstein's sequence

Definition 22

Let $f\colon \mathbb{N} \to \mathbb{N}$ be a non-decreasing function. We say that f is *valid* if $f(0) \geq 2$. For a valid function f, define the *Goodstein sequence for f starting at n*, written G^f_n, as follows:

Goodstein's sequence

Definition 22

Let $f: \mathbb{N} \to \mathbb{N}$ be a non-decreasing function. We say that f is *valid* if $f(0) \geq 2$. For a valid function f, define the *Goodstein sequence for f starting at n*, written G_n^f, as follows:

$$G_n^f(0) = n$$
$$G_n^f(k+1) = \mathrm{Rep}(G_n^f(k), f(k), f(k+1)) - 1.$$

Goodstein's sequence

Definition 22

Let $f : \mathbb{N} \to \mathbb{N}$ be a non-decreasing function. We say that f is *valid* if $f(0) \geq 2$. For a valid function f, define the *Goodstein sequence for f starting at n*, written G_n^f, as follows:

$$G_n^f(0) = n$$
$$G_n^f(k+1) = \mathrm{Rep}(G_n^f(k), f(k), f(k+1)) - 1.$$

Theorem 23

Let f be a valid function. Then for all n there exists r such that $G_n^f(r) = 0$.

True but not provable

Theorem 24 (Kirby & Paris, 1982)

Let $f(x) = x + 2$. *Then*

$$\mathbb{N} \models \forall m \exists r \; G_n^f(r) = 0,$$

but

$$P \nvdash \forall m \exists r \; G_n^f(r) = 0.$$

True but not provable

Theorem 24 (Kirby & Paris, 1982)

Let $f(x) = x + 2$. *Then*

$$\mathbb{N} \models \forall m \exists r \; G_n^f(r) = 0,$$

but

$$P \nvdash \forall m \exists r \; G_n^f(r) = 0.$$

For $n = 4$, the required r is

$$3 \cdot 2^{402,653,211} - 3.$$