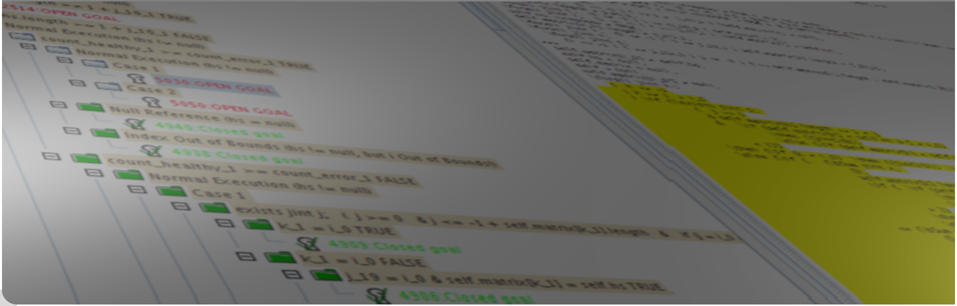
A Theory of Ordinals

Peter H. Schmitt

KIT – Institut für Theoretische Informatik



Starting Point

Many theorem provers offer support for reasoning with ordinals

- Isabelle/HOL
  Blanchette et al: Cardinals in Isabelle/HOL, ITP 2014
  Huffman: Countable Ordinals, Archive of Formal Proofs
- Coq
  Grimm: Implementation of 3 types of ordinals in Coq, RR-8407
- ACL2
  Manolios and Vroon: Ordinal Arithmetic: Algorithms and Mechanization, JAR 34(4)

Starting Point

Many theorem provers offer support for reasoning with ordinals

- Isabelle/HOL
  Blanchette et al: Cardinals in Isabelle/HOL, ITP 2014
  Huffman: Countable Ordinals, Archive of Formal Proofs
- Coq
  Grimm: Implementation of 3 types of ordinals in Coq, RR-8407
- ACL2
  Manolios and Vroon: Ordinal Arithmetic: Algorithms and Mechanization, JAR 34(4)

These efforts are based on formalizations in set theory or deal with algorithms rather than axiomatizations.

Goal of this Project

Find a first-order axiomatization of ordinal numbers in the style of Peano's axioms for natural numbers.

Peano Arithmetic

1. $\forall x(x + 1 \not\doteq 0)$
2. $\forall x \forall y(x + 1 \doteq y + 1 \rightarrow x \doteq y)$
3. $(\phi(0) \wedge \forall y(\phi(y) \rightarrow \phi(y + 1))) \rightarrow \forall x(\phi)$
   for every formula ϕ
4. $\forall x(x + 0 \doteq x)$
5. $\forall x \forall y(x + (y + 1) \doteq (x + y) + 1)$
6. $\forall x(x * 0 \doteq 0)$
7. $\forall x \forall y(x * (y + 1) \doteq (x * y) + x)$
8. $\forall x \forall y(x \geq y \leftrightarrow \exists z(x \doteq y + z))$

A New Counting Principle

$0, 1, 2, \ldots$ ω

A New Counting Principle

$$\begin{array}{ll} 0, 1, 2, \ldots & \omega \\ \omega, \omega + 1, \omega + 2, \ldots & \omega + \omega = \omega * 2 \end{array}$$

A New Counting Principle

$$
\begin{array}{ll}
0, 1, 2, \ldots & \omega \\
\omega, \omega + 1, \omega + 2, \ldots & \omega + \omega = \omega * 2 \\
\omega * 2, \omega * 3, \omega * 4, \ldots & \omega * \omega = \omega^2
\end{array}
$$

A New Counting Principle

$$\begin{array}{ll} 0, 1, 2, \ldots & \omega \\ \omega, \omega + 1, \omega + 2, \ldots & \omega + \omega = \omega * 2 \\ \omega * 2, \omega * 3, \omega * 4, \ldots & \omega * \omega = \omega^2 \\ \omega^2, \omega^3, \omega^4, \ldots & \omega^\omega \end{array}$$

A New Counting Principle

KIT Karlsruhe Institute of Technology

$$
\begin{array}{ll}
0, 1, 2, \ldots & \omega \\
\omega, \omega + 1, \omega + 2, \ldots & \omega + \omega = \omega * 2 \\
\omega * 2, \omega * 3, \omega * 4, \ldots & \omega * \omega = \omega^2 \\
\omega^2, \omega^3, \omega^4, \ldots & \omega^\omega \\
\omega^\omega, \omega^{\omega^2}, , \omega^{\omega^3}, \ldots & \omega^{\omega^\omega}
\end{array}
$$

A New Counting Principle

$0, 1, 2, \ldots$	ω
$\omega, \omega + 1, \omega + 2, \ldots$	$\omega + \omega = \omega * 2$
$\omega * 2, \omega * 3, \omega * 4, \ldots$	$\omega * \omega = \omega^2$
$\omega^2, \omega^3, \omega^4, \ldots$	ω^ω
$\omega^\omega, \omega^{\omega^2}, , \omega^{\omega^3}, \ldots$	ω^{ω^ω}
$\omega^{\omega^\omega}, \omega^{\omega^{\omega^\omega}}, \ldots$	ϵ_0

A New Counting Principle

$$
\begin{array}{ll}
0, 1, 2, \ldots & \omega \\
\omega, \omega + 1, \omega + 2, \ldots & \omega + \omega = \omega * 2 \\
\omega * 2, \omega * 3, \omega * 4, \ldots & \omega * \omega = \omega^2 \\
\omega^2, \omega^3, \omega^4, \ldots & \omega^\omega \\
\omega^\omega, \omega^{\omega^2}, , \omega^{\omega^3}, \ldots & \omega^{\omega^\omega} \\
\omega^{\omega^\omega}, \omega^{\omega^{\omega^\omega}}, \ldots & \epsilon_0 \\
\ldots &
\end{array}
$$

The Core Theory Th^0_{Ord}

Vocabular

predicate	$n < m$	:	(Ord, Ord)
functions	$n + 1$	:	$Ord \to Ord$
	0	:	Ord
	ω	:	Ord

The Core Theory Th^0_{Ord}

Vocabular

predicate	$n < m$	:	(Ord, Ord)
functions	$n + 1$	:	$Ord \to Ord$
	0	:	Ord
	ω	:	Ord
binder	$sup_{m<b}j$	:	$Ord \times Ord \to Ord$

The Core Theory Th^0_{Ord}

Vocabular

predicate	$n < m$	:	(Ord, Ord)
functions	$n + 1$	:	$Ord \to Ord$
	0	:	Ord
	ω	:	Ord
binder	$sup_{m<b}j$	:	$Ord \times Ord \to Ord$

Examples:

$$sup_{m<\omega}(\omega + m) = \omega * 2$$

$$sup_{m<\omega}(\omega^m) = \omega^\omega$$

The Core Theory Th^0_{Ord}

Axioms

1. $\forall x, y, z(x < y \land y < z \to x < z)$ transitivity
2. $\forall x(\neg x < x)$ strict order
3. $\forall x, y(x < y \lor x \doteq y \lor y < x)$ total order
4. $\forall x(0 \leq x)$ 0 is smallest element

The Core Theory Th^0_{Ord}

Axioms

1. $\forall x, y, z(x < y \land y < z \to x < z)$ transitivity
2. $\forall x(\neg x < x)$ strict order
3. $\forall x, y(x < y \lor x \doteq y \lor y < x)$ total order
4. $\forall x(0 \leq x)$ 0 is smallest element
5. $\forall x(x < x + 1) \land \forall x, y(x < y \to x + 1 \leq y)$ successor function

The Core Theory Th^0_{Ord}

Axioms

1. $\forall x, y, z(x < y \land y < z \to x < z)$ transitivity
2. $\forall x(\neg x < x)$ strict order
3. $\forall x, y(x < y \lor x \doteq y \lor y < x)$ total order
4. $\forall x(0 \leq x)$ 0 is smallest element
5. $\forall x(x < x + 1) \land \forall x, y(x < y \to x + 1 \leq y)$ successor function
6. $0 < \omega \land \neg\exists x(\omega \doteq x + 1)$ ω is a limit ordinal
7. $\forall y(0 < y \land \forall x(x < \omega \to x + 1 < y) - > \omega \leq y)$ ω is the least limit ordinal

The Core Theory Th^0_{Ord}

Axioms

1. $\forall x, y, z(x < y \wedge y < z \rightarrow x < z)$ transitivity
2. $\forall x(\neg x < x)$ strict order
3. $\forall x, y(x < y \vee x \doteq y \vee y < x)$ total order
4. $\forall x(0 \leq x)$ 0 is smallest element
5. $\forall x(x < x + 1) \wedge \forall x, y(x < y \rightarrow x + 1 \leq y)$ successor function
6. $0 < \omega \wedge \neg \exists x(\omega \doteq x + 1)$ ω is a limit ordinal
7. $\forall y(0 < y \wedge \forall x(x < \omega \rightarrow x + 1 < y) - > \omega \leq y)$ ω is the least limit ordinal
8. $\forall z(z < \alpha \rightarrow t[z/\lambda] \leq sup_{\lambda<\alpha} t)$ def of supremum, part 1
9. $\forall x(\forall z(z < \alpha \rightarrow t[z/\lambda] \leq x) \rightarrow sup_{\lambda<\alpha} t \leq x)$ def of supremum, part 2

The Core Theory Th^0_{Ord}

Axioms

1. $\forall x, y, z(x < y \land y < z \to x < z)$ transitivity
2. $\forall x(\neg x < x)$ strict order
3. $\forall x, y(x < y \lor x \doteq y \lor y < x)$ total order
4. $\forall x(0 \leq x)$ 0 is smallest element
5. $\forall x(x < x + 1) \land \forall x, y(x < y \to x + 1 \leq y)$ successor function
6. $0 < \omega \land \neg\exists x(\omega \doteq x + 1)$ ω is a limit ordinal
7. $\forall y(0 < y \land \forall x(x < \omega \to x + 1 < y) - > \omega \leq y)$ ω is the least limit ordinal
8. $\forall z(z < \alpha \to t[z/\lambda] \leq sup_{\lambda<\alpha} t)$ def of supremum, part 1
9. $\forall x(\forall z(z < \alpha \to t[z/\lambda] \leq x) \to sup_{\lambda<\alpha} t \leq x)$ def of supremum, part 2
10. $\forall x(\forall y(y < x \to \phi(y)) \to \phi) \to \forall x\phi$ transfinite induction

Transfinite Induction

The induction scheme

$$\forall x(\forall y(y < x \to \phi(y)) \to \phi) \to \forall x \phi$$

Transfinite Induction

The induction scheme

$$\forall x(\forall y(y < x \rightarrow \phi(y)) \rightarrow \phi) \rightarrow \forall x \phi$$

is equivalent to

Transfinite Induction

The induction scheme

$$\forall x(\forall y(y < x \to \phi(y)) \to \phi) \to \forall x \phi$$

is equivalent to

$$\begin{array}{l} \phi(0) \\ \forall x(\phi(x) \to \phi(x+1)) \\ \forall x((lim(x) \land \forall y(y < x \to \phi(y))) \to \phi(x)) \\ \to \\ \forall x \phi(x) \end{array}$$

Definitional Extension

- $\forall x(x + 0 \doteq x)$
- $\forall x, y(x + (y + 1) \doteq (x + y) + 1)$
- $\forall x, y(lim(y) \to x + sup_{\lambda<y}\lambda \doteq sup_{\lambda<y}(x + \lambda))$

Definitional Extension

- $\forall x(x + 0 \doteq x)$
- $\forall x, y(x + (y + 1) \doteq (x + y) + 1)$
- $\forall x, y(lim(y) \rightarrow x + sup_{\lambda < y}\lambda \doteq sup_{\lambda < y}(x + \lambda))$

Similarly for $x * y$ and x^y.

Embedding $\mathbb{N}$ in *Ord*

Add a functions symbol

$$onat : int \to Ord$$

Embedding $\mathbb{N}$ in *Ord*

Add a functions symbol

$$onat : int \to Ord$$

Axioms

1. $onat(0) \doteq 0$
2. $\forall n(0 \leq n \to onat(n+1) \doteq onat(n) + 1)$

Embedding $\mathbb{N}$ in *Ord*

Add a functions symbol

$$onat : int \rightarrow Ord$$

Axioms

1. $onat(0) \doteq 0$
2. $\forall n(0 \leq n \rightarrow onat(n+1) \doteq onat(n) + 1)$

Semantics:

$$\begin{array}{lcll} onat^{\mathcal{M}}(n) & = & \text{the ordinal } n & \text{if } n \geq 0 \\ & = & \text{undefined} & \text{otherwise} \end{array}$$

Embedding $\mathbb{N}$ in *Ord*

Derived Lemmas

3. $onat(1) \doteq 1$

Embedding $\mathbb{N}$ in *Ord*

Derived Lemmas

3. $onat(1) \doteq 1$
4. $\forall n, m(0 \leq n \wedge 0 \leq m \rightarrow onat(n + m) \doteq onat(n) + onat(m))$

Embedding $\mathbb{N}$ in *Ord*

Derived Lemmas

3. $onat(1) \doteq 1$
4. $\forall n, m(0 \leq n \wedge 0 \leq m \rightarrow onat(n+m) \doteq onat(n) + onat(m))$
5. $\forall n, m((0 \leq n \wedge 0 \leq m) \rightarrow (onat(n) \doteq onat(m) \rightarrow n \doteq m))$

Embedding $\mathbb{N}$ in *Ord*

Derived Lemmas

3. $onat(1) \doteq 1$
4. $\forall n, m(0 \leq n \wedge 0 \leq m \rightarrow onat(n + m) \doteq onat(n) + onat(m)$
5. $\forall n, m((0 \leq n \wedge 0 \leq m) \rightarrow (onat(n) \doteq onat(m) \rightarrow n \doteq m))$
6. $\forall n, m((0 \leq n \wedge 0 \leq m) \rightarrow (onat(n) < onat(m) \leftrightarrow n < m))$

Embedding $\mathbb{N}$ in *Ord*

Derived Lemmas

3. $onat(1) \doteq 1$
4. $\forall n, m(0 \leq n \land 0 \leq m \to onat(n + m) \doteq onat(n) + onat(m)$
5. $\forall n, m((0 \leq n \land 0 \leq m) \to (onat(n) \doteq onat(m) \to n \doteq m))$
6. $\forall n, m((0 \leq n \land 0 \leq m) \to (onat(n) < onat(m) \leftrightarrow n < m))$
7. $\forall n(0 \leq n \to onat(n) < \omega)$

Embedding $\mathbb{N}$ in *Ord*

Derived Lemmas

3. $onat(1) \doteq 1$
4. $\forall n, m(0 \leq n \land 0 \leq m \rightarrow onat(n+m) \doteq onat(n) + onat(m)$
5. $\forall n, m((0 \leq n \land 0 \leq m) \rightarrow (onat(n) \doteq onat(m) \rightarrow n \doteq m))$
6. $\forall n, m((0 \leq n \land 0 \leq m) \rightarrow (onat(n) < onat(m) \leftrightarrow n < m))$
7. $\forall n(0 \leq n \rightarrow onat(n) < \omega)$
8.

$$\begin{array}{rl} \forall i_1, i_2, j_1, j_2 & ((0 \leq i_1 \land 0 \leq i_2 \land 0 \leq j_1 \land 0 \leq j_2) \rightarrow \\ & \omega * onat(i_1) + onat(j_1) < \omega * onat(i_2) + onat(j_2) \\ \leftrightarrow & i_1 < i_2 \lor (i_1 \doteq i_2 \land j_1 < j_2)) \end{array}$$

Implementation

- The theory Th_{Ord} has been fully implemented on the side branch `pschmittOrd`.

Implementation

- The theory Th_{Ord} has been fully implemented on the side branch `pschmittOrd`.
- The JML parser has been extended (with the exception of `sup`). So the Th_{Ord} vocabulary can be used in JML specifications.

Implementation

- The theory Th_{Ord} has been fully implemented on the side branch `pschmittOrd`.
- The JML parser has been extended
  (with the exception of `sup`).
  So the Th_{Ord} vocabulary can be used in JML specifications.
- The Java to KeY parser has been extended to cover the Th_{Ord} vocabulary.
  So it can be used in assignments to ghost fields etc.

How strong is Th_{Ord}?

Proved some 182 lemmas from relevant text books.
Some examples:

How strong is Th_{Ord}?

Proved some 182 lemmas from relevant text books.
Some examples:

1. $\forall\lambda(\lambda < \alpha_1 \rightarrow \exists\lambda'(\lambda' < \alpha_2 \wedge t_1 \leq t_2[\lambda'/\lambda]))\wedge$
   $\forall\lambda(\lambda < \alpha_2 \rightarrow \exists\lambda'(\lambda' < \alpha_1 \wedge t_2 \leq t_1[\lambda'/\lambda])) \quad \rightarrow$
   $sup_{\lambda<\alpha_1} t_1 \doteq sup_{\lambda<\alpha_2} t_2$

How strong is Th_{Ord}?

Proved some 182 lemmas from relevant text books.
Some examples:

1. $\forall\lambda(\lambda < \alpha_1 \rightarrow \exists\lambda'(\lambda' < \alpha_2 \wedge t_1 \leq t_2[\lambda'/\lambda]))\wedge$
   $\forall\lambda(\lambda < \alpha_2 \rightarrow \exists\lambda'(\lambda' < \alpha_1 \wedge t_2 \leq t_1[\lambda'/\lambda])) \quad \rightarrow$
   $sup_{\lambda<\alpha_1} t_1 \doteq sup_{\lambda<\alpha_2} t_2$
2. $(\alpha * \beta) * \gamma \doteq \alpha * (\beta * \gamma)$

How strong is Th_{Ord}?

Proved some 182 lemmas from relevant text books.
Some examples:

1. $\forall\lambda(\lambda < \alpha_1 \to \exists\lambda'(\lambda' < \alpha_2 \land t_1 \leq t_2[\lambda'/\lambda]))\land$
   $\forall\lambda(\lambda < \alpha_2 \to \exists\lambda'(\lambda' < \alpha_1 \land t_2 \leq t_1[\lambda'/\lambda])) \quad \to$
   $sup_{\lambda<\alpha_1} t_1 \doteq sup_{\lambda<\alpha_2} t_2$
2. $(\alpha * \beta) * \gamma \doteq \alpha * (\beta * \gamma)$
3. $\alpha * (\beta + \gamma) \doteq \alpha * \beta + \alpha * \gamma$

How strong is Th_{Ord}?

Proved some 182 lemmas from relevant text books.
Some examples:

1. $\forall\lambda(\lambda < \alpha_1 \rightarrow \exists\lambda'(\lambda' < \alpha_2 \wedge t_1 \leq t_2[\lambda'/\lambda]))\wedge$
   $\forall\lambda(\lambda < \alpha_2 \rightarrow \exists\lambda'(\lambda' < \alpha_1 \wedge t_2 \leq t_1[\lambda'/\lambda])) \rightarrow$
   $sup_{\lambda<\alpha_1} t_1 \doteq sup_{\lambda<\alpha_2} t_2$
2. $(\alpha * \beta) * \gamma \doteq \alpha * (\beta * \gamma)$
3. $\alpha * (\beta + \gamma) \doteq \alpha * \beta + \alpha * \gamma$
4. $\alpha < \beta \rightarrow \gamma + \alpha < \gamma + \beta$

How strong is Th_{Ord}?

Proved some 182 lemmas from relevant text books.
Some examples:

1. $\forall\lambda(\lambda < \alpha_1 \rightarrow \exists\lambda'(\lambda' < \alpha_2 \wedge t_1 \leq t_2[\lambda'/\lambda]))\wedge$
   $\forall\lambda(\lambda < \alpha_2 \rightarrow \exists\lambda'(\lambda' < \alpha_1 \wedge t_2 \leq t_1[\lambda'/\lambda])) \rightarrow$
   $sup_{\lambda<\alpha_1} t_1 \doteq sup_{\lambda<\alpha_2} t_2$
2. $(\alpha * \beta) * \gamma \doteq \alpha * (\beta * \gamma)$
3. $\alpha * (\beta + \gamma) \doteq \alpha * \beta + \alpha * \gamma$
4. $\alpha < \beta \rightarrow \gamma + \alpha < \gamma + \beta$
5. $\alpha < \beta \leftrightarrow \exists\gamma(\alpha + \gamma \doteq \beta)$

How strong is Th_{Ord}?

Proved some 182 lemmas from relevant text books.
Some examples:

1. $\forall\lambda(\lambda < \alpha_1 \rightarrow \exists\lambda'(\lambda' < \alpha_2 \wedge t_1 \leq t_2[\lambda'/\lambda]))\wedge$
   $\forall\lambda(\lambda < \alpha_2 \rightarrow \exists\lambda'(\lambda' < \alpha_1 \wedge t_2 \leq t_1[\lambda'/\lambda])) \rightarrow$
   $sup_{\lambda<\alpha_1} t_1 \doteq sup_{\lambda<\alpha_2} t_2$
2. $(\alpha * \beta) * \gamma \doteq \alpha * (\beta * \gamma)$
3. $\alpha * (\beta + \gamma) \doteq \alpha * \beta + \alpha * \gamma$
4. $\alpha < \beta \rightarrow \gamma + \alpha < \gamma + \beta$
5. $\alpha < \beta \leftrightarrow \exists\gamma(\alpha + \gamma \doteq \beta)$
6. $\omega \leq \alpha \rightarrow \exists\lambda\exists n(\alpha = \lambda + n \wedge \lim(\lambda) \wedge n < \omega)$

How strong is Th_{Ord}?

Proved some 182 lemmas from relevant text books.
Some examples:

1. $\forall\lambda(\lambda < \alpha_1 \rightarrow \exists\lambda'(\lambda' < \alpha_2 \wedge t_1 \leq t_2[\lambda'/\lambda]))\wedge$
   $\forall\lambda(\lambda < \alpha_2 \rightarrow \exists\lambda'(\lambda' < \alpha_1 \wedge t_2 \leq t_1[\lambda'/\lambda])) \quad \rightarrow$
   $sup_{\lambda<\alpha_1} t_1 \doteq sup_{\lambda<\alpha_2} t_2$
2. $(\alpha * \beta) * \gamma \doteq \alpha * (\beta * \gamma)$
3. $\alpha * (\beta + \gamma) \doteq \alpha * \beta + \alpha * \gamma$
4. $\alpha < \beta \rightarrow \gamma + \alpha < \gamma + \beta$
5. $\alpha < \beta \leftrightarrow \exists\gamma(\alpha + \gamma \doteq \beta)$
6. $\omega \leq \alpha \rightarrow \exists\lambda\exists n(\alpha = \lambda + n \wedge \lim(\lambda) \wedge n < \omega)$
7. $\lim(\lambda) \rightarrow \exists\beta(\lambda \doteq \omega * \beta)$

How strong is Th_{Ord}?

Proved some 182 lemmas from relevant text books.
Some examples:

1. $\forall\lambda(\lambda < \alpha_1 \to \exists\lambda'(\lambda' < \alpha_2 \wedge t_1 \leq t_2[\lambda'/\lambda]))\wedge$
   $\forall\lambda(\lambda < \alpha_2 \to \exists\lambda'(\lambda' < \alpha_1 \wedge t_2 \leq t_1[\lambda'/\lambda])) \;\; \to$
   $sup_{\lambda<\alpha_1} t_1 \doteq sup_{\lambda<\alpha_2} t_2$
2. $(\alpha * \beta) * \gamma \doteq \alpha * (\beta * \gamma)$
3. $\alpha * (\beta + \gamma) \doteq \alpha * \beta + \alpha * \gamma$
4. $\alpha < \beta \to \gamma + \alpha < \gamma + \beta$
5. $\alpha < \beta \leftrightarrow \exists\gamma(\alpha + \gamma \doteq \beta)$
6. $\omega \leq \alpha \to \exists\lambda\exists n(\alpha = \lambda + n \wedge \lim(\lambda) \wedge n < \omega)$
7. $\lim(\lambda) \to \exists\beta(\lambda \doteq \omega * \beta)$
8. $(\alpha \neq 0 \wedge \beta < \omega \wedge \lim(\gamma)) \to (\alpha + \beta) * \gamma \doteq \alpha * \gamma$

How strong is Th_{Ord}?

Proved some 182 lemmas from relevant text books.
Some examples:

1. $\forall\lambda(\lambda < \alpha_1 \rightarrow \exists\lambda'(\lambda' < \alpha_2 \wedge t_1 \leq t_2[\lambda'/\lambda]))\wedge$
   $\forall\lambda(\lambda < \alpha_2 \rightarrow \exists\lambda'(\lambda' < \alpha_1 \wedge t_2 \leq t_1[\lambda'/\lambda])) \quad \rightarrow$
   $sup_{\lambda<\alpha_1} t_1 \doteq sup_{\lambda<\alpha_2} t_2$
2. $(\alpha * \beta) * \gamma \doteq \alpha * (\beta * \gamma)$
3. $\alpha * (\beta + \gamma) \doteq \alpha * \beta + \alpha * \gamma$
4. $\alpha < \beta \rightarrow \gamma + \alpha < \gamma + \beta$
5. $\alpha < \beta \leftrightarrow \exists\gamma(\alpha + \gamma \doteq \beta)$
6. $\omega \leq \alpha \rightarrow \exists\lambda\exists n(\alpha = \lambda + n \wedge \lim(\lambda) \wedge n < \omega)$
7. $\lim(\lambda) \rightarrow \exists\beta(\lambda \doteq \omega * \beta)$
8. $(\alpha \neq 0 \wedge \beta < \omega \wedge \lim(\gamma)) \rightarrow (\alpha + \beta) * \gamma \doteq \alpha * \gamma$
9. $\lim(\alpha) \wedge 0 < \beta \wedge 0 < \gamma \wedge 0 < m < \omega \rightarrow$
   $\neg\lim(\gamma) \wedge (\alpha^\beta * m)^\gamma \doteq \alpha^{\beta*\gamma} * m \vee$
   $\lim(\gamma) \wedge (\alpha^\beta * m)^\gamma \doteq \alpha^{\beta*\gamma}$

A Termination Proof

```
class CClass { int x,y;
/*@ normal_behaviour  requires  0<=x && 0<=y; @*/
void  method() {
 /*@ loop_invariant  x>=0 && y>=0;
   @ decreases
     \ord_add(\ord_times(\omega,\onat(x)),\onat(y))
   @*/
 while (x>0 || y>0) { g();  } }
 /*@ normal_behaviour  ensures
   @    (x == \old(x)-1 && x>=0 &&y > 0) ||
   @    (x == \old(x)   && y == \old(y)-1 && y>=0);
   @*/
    void g(){ }
}
```

A Termination Proof

```
class CClass { int x,y;
/*@ normal_behaviour  requires  0<=x && 0<=y; @*/
void  method() {
 /*@ loop_invariant  x>=0 && y>=0;
   @ decreases
     \ord_add(\ord_times(\omega,\onat(x)),\onat(y))
   @*/
 while (x>0 || y>0) { g();  } }
 /*@ normal_behaviour  ensures
   @    (x == \old(x)-1 && x>=0 &&y > 0) ||
   @    (x == \old(x)    && y == \old(y)-1 && y>=0);
   @*/
    void g(){ }
}
```

pretty printed variant: $\omega * x + y$.

Future Work

Prove termination of the Goodstein sequences.

This is a well-known problem that cannot be proved in Peano arithmetic.

Future Work

Prove termination of the Goodstein sequences.

This is a well-known problem that cannot be proved in Peano arithmetic.
A proof in Th_{Ord} is in progress.

THE END