

Using the ordinal calculator

For ordCalc_0.1

Paul Budnik

Mountain Math Software

paul@mtnmath.com

Contents

1	Introduction	1
2	Ordinals	2
3	Syntax	2
4	Commands	3
4.1	All commands	3
4.2	Commands with options	3
5	Member functions	4
6	Comparison operators	5
7	Examples	5
7.1	Simple ordinal arithmetic	5
7.2	Comparison operators	5
7.3	Display options	6
7.4	Member functions	6

About this manual

This manual duplicates the online documentation. For a complete description of the program and the mathematics on which it is based (including references) see the paper “A Computational Approach to the Ordinal Numbers”. That document includes this manual as an appendix.

1 Introduction

Most of this manual is automatically extracted from the online documentation.

This is a command line interactive interface to a program for exploring the ordinals. It supports the Veblen function with any number of parameters and one step beyond that. Following are topics you can get more information about by entering ‘help topic’.

‘**cmds**’ – lists commands.
‘**compare**’ – describes comparison operators.
‘**members**’ – describes member functions.
‘**ordinal**’ – describes available ordinal notations.
‘**syntax**’ – describes syntax.
‘**version**’ – displays program version.

This program supports GNU ‘readline’ input line editing. You can download the program and documentation at: Mountain Math Software or at SourceForge.net (<http://www.mtnmath.com/ord> or <https://sourceforge.net/projects/ord>).

2 Ordinals

Ordinals are displayed in TeX and plain text format. (Enter ‘help opts’ to see how to control this.) The plain text format can be read. The finite ordinals are the non-negative integers. The ordinal operators are +, * and ^ for addition, multiplication and exponentiation. Exponentiation has the highest precedence. Parenthesis can be used to group subexpressions.

The ordinal of the integers, omega, is represented by the single lower case letter: ‘w’. The Veblen function is specified as ‘psi(p1,p2,...,pn)’ where n is any integer > 0. Special notations are displayed in some cases. Specifically psi(x) is displayed as w^x. psi(1,x) is displayed as epsilon(x). psi(1,0,x) is displayed as gamma(x). In all cases the displayed version can be used as input.

The largest ordinals in this implementation are specified as psi_{px}(p1,p2,...,pn). The first parameter is enclosed in brackets not parenthesis. psi_{1} is defined as the union of w, epsilon(0), gamma(0), psi(1, 0, 0, 0), psi(1, 0, 0, 0, 0), psi(1, 0, 0, 0, 0, 0), psi(1, 0, 0, 0, 0, 0, 0), ... You can access the sequence whose union is a specific ordinal using member functions. Type ‘help members’ to learn more.

3 Syntax

The syntax is that of restricted arithmetic expressions and assignment statements. The tokens are variable names, nonnegative integers and the operators: +, * and ^ (addition, multiplication and exponentiation). Comparison operators are also sup-

ported. Type ‘help comparison’ to learn about them. The letter ‘w’ is predefined as omega, the ordinal of the integers. To learn more about ordinals type ‘help ordinal’. C++ style member functions are supported with a ‘.’ separating the variable name (or expression enclosed in parenthesis) from the member function name. Enter ‘help members’ for the list of member functions.

An assignment statement or ordinal expression can be entered and it will be evaluated and displayed in normal form. Typing ‘help opts’ lists the display options. Assignment statements are stored. They can be listed (command ‘list’) and their value can be used in subsequent expressions. All statements end at the end of a line unless the last character is ‘\’. Lines can be continued indefinitely. Comments must be preceded by either ‘%’ or ‘//’.

Commands can be entered as one or more names separated by white space. File names should be enclosed in double quotes (“) if they contain any non alphanumeric characters such as dot, ‘.’. Command names can be used as variables. Enter ‘help cmds’ to get a list of commands and their functions.

4 Commands

4.1 All commands

The following commands are available:

‘examples’ – shows examples

‘exportTeX’ – exports assignments statements in TeX format

‘help’ – displays information on various topics

‘list’ – lists assignment statements

‘log’ – writes a log file (ord.log default)

‘listTeX’ – lists assignment statements in TeX format

‘logopt’ – controls the log file

‘opts’ – controls display format and other options

‘quit’ – exits the program

‘read’ – reads an input file (ord_calc.ord default)

‘save’ – saves assignment statements to a file (ord_calc.ord default)

‘yydebug’ – enables parser debugging (off option)

Type ‘help command_name’ to learn more about a specific command.

4.2 Commands with options

Following are the commands with options.

Command `'examples'` – shows examples.
It has one parameter with the following options.
`'arith'` – show a simple ordinal arithmetic example.
`'compare'` – show some compare examples.
`'display'` – show display options examples.
`'member'` – show member function examples.

Command `'logopt'` – controls the log file.
It has one parameter with the following options.
`'flush'` – flush log file.
`'stop'` – stop logging.

Command `'opts'` – controls display format and other options.
It has one parameter with the following options.
`'both'` – display ordinals in both plain text and TeX formats.
`'tex'` – display ordinals in TeX format only.
`'text'` – display ordinals in plain text format only (default).

5 Member functions

Every ordinal (except 0) is the union of smaller ordinals. Every limit ordinal is the union of an infinite sequence of smaller ordinals. Member functions allow access to to a sequence of these smaller ordinals. One can specify how many elements of this sequence to display or get the value of a specific instance of the sequence. For a limit ordinal, the sequence displayed, were it extended to infinity and its union taken, that union would equal the original ordinal.

The syntax for a member function begins with either an ordinal name (from an assignment statement) or an ordinal expression enclosed in parenthesis. This is followed by a dot (.) and then the member function name and its parameters enclosed in parenthesis. The format is `'ordinal_name.memberFunction(p)'` where p may be optional.

The member functions are:

`'limitElt'` – 'evaluates to specified (no default) limit element'
`'listLimitElts'` – 'lists specified (default 10) limit elements'

6 Comparison operators

Any two ordinals or ordinal expressions can be compared using the operators: $<$, $<=$, $>$, $>=$ and $==$. The result of the comparison is the text either TRUE or FALSE. Comparison operators have lower precedence than ordinal operators.

7 Examples

In the examples a line that begins with the standard prompt 'ordCalc>' contains user input. All other lines contain program output

To select an examples type 'examples' followed by one of the following options.

'arith' – show a simple ordinal arithmetic example

'compare' – show some compare examples

'display' – show display options examples

'member' – show member function examples

7.1 Simple ordinal arithmetic

```
ordCalc>a=w^w
Assigning (w^(w)) to 'a'.
ordCalc>b=w*w
Assigning (w^(2)) to 'b'.
ordCalc>c=a+b
Assigning (w^(w)) + (w^(2)) to 'c'.
ordCalc>d=b+a
Assigning (w^(w)) to 'd'.
```

7.2 Comparison operators

```
ordCalc>psi(1,0,0) == gamma(0)
TRUE
ordCalc>psi(1,w) == epsilon(w)
TRUE
ordCalc>w^w < psi(1)
FALSE
ordCalc>psi(1)
Normal form: w
```

7.3 Display options

```
ordCalc>a=w^(w^w)
Assigning (w^((w^(w)))) to 'a'.
ordCalc>b=epsilon(a)
Assigning epsilon((w^((w^(w))))) to 'b'.
ordCalc>c=gamma(b)
Assigning gamma(epsilon((w^((w^(w))))) to 'c'.
ordCalc>list
a = (w^((w^(w))))
b = epsilon((w^((w^(w)))))
c = gamma(epsilon((w^((w^(w)))))
ordCalc>opts tex
ordCalc>list
a = \omega{}^{\omega{}^{\omega{}}}
b = \epsilon_{\omega{}^{\omega{}^{\omega{}}}}
c = \varphi( 1, 0, \epsilon_{\omega{}^{\omega{}^{\omega{}}}})
ordCalc>opts both
ordCalc>list
a = (w^((w^(w))))
a = \omega{}^{\omega{}^{\omega{}}}
b = epsilon((w^((w^(w)))))
b = \epsilon_{\omega{}^{\omega{}^{\omega{}}}}
c = gamma(epsilon((w^((w^(w)))))
c = \varphi( 1, 0, \epsilon_{\omega{}^{\omega{}^{\omega{}}}})
```

7.4 Member functions

```
ordCalc>a=psi(1,0,0,0,0)
Assigning psi(1, 0, 0, 0, 0) to 'a'.
ordCalc>a.listLimitElts(3)
```

```
First 3 limitElements for psi(1, 0, 0, 0, 0)
le(0) = 0
le(1) = psi(1, 0, 0, 0)
le(2) = psi(psi(1, 0, 0, 0) + 1, 0, 0, 0)
End limitElements
```

```
Normal form: psi(1, 0, 0, 0, 0)
ordCalc>b=a.limitElt(6)
Assigning psi(psi(psi(psi(psi(psi(1, 0, 0, 0) + 1, 0, 0, 0) + 1, 0, 0, 0)
+ 1, 0, 0, 0) + 1, 0, 0, 0) + 1, 0, 0, 0) to 'b'.
```