Σ
A Library for Ansi Common Lisp

Christopher Mark Gore
cgore@cgore.com
http://cgore.com/programming/lisp/sigma/
https://github.com/cgore/sigma

April 24, 2013

Contents

1 Copyright **9**

2 The `Behave` Package **11**

2.1 Macros . . . 12

2.1.1 The `Behavior` Macro . . . 12

2.1.2 The `Spec` Macro . . . 12

2.1.3 The `Should` Macro . . . 12

2.1.4 The `Should-Not` Macro . . . 12

2.1.5 The `Should-Be-Null` Macro . . . 12

2.1.6 The `Should-Be-A` Macro . . . 12

2.1.7 The `Should=` Macro . . . 12

2.1.8 The `Should/=` Macro . . . 12

2.1.9 The `Should<` Macro . . . 12

2.1.10 The `Should>` Macro . . . 12

2.1.11 The `Should<=` Macro . . . 12

2.1.12 The `Should>=` Macro . . . 12

2.1.13 The `Should-Eq` Macro . . . 12

2.1.14 The `Should-Not-Eq` Macro . . . 12

2.1.15 The `Should-Eql` Macro . . . 12

2.1.16 The `Should-Not-Eql` Macro . . . 12

2.1.17 The `Should-Equal` Macro . . . 12

2.1.18 The `Should-Not-Equal` Macro . . . 12

2.1.19 The `Should-EqualP` Macro . . . 12

2.1.20 The `Should-Not-EqualP` Macro . . . 12

2.1.21 The `Should-String=` Macro . . . 12

2.1.22 The `Should-Not-String=` Macro . . . 12

2.1.23 The `Should-String/=` Macro . . . 12

2.1.24 The `Should-Not-String/=` Macro . . . 12

2.1.25 The `Should-String<` Macro . . . 12

2.1.26 The `Should-Not-String<` Macro . . . 12

2.1.27 The `Should-String>` Macro . . . 12

2.1.28 The `Should-Not-String>` Macro . . . 12

2.1.29 The `Should-String<=` Macro . . . 12

2.1.30 The `Should-Not-String<=` Macro . . . 12

2.1.31 The `Should-String>=` Macro . . . 12
2.1.32 The `Should-Not-String>=` Macro . . . 12
2.1.33 The `Should-String-Equal` Macro . . . 12
2.1.34 The `Should-Not-String-Equal` Macro . . . 12
2.1.35 The `Should-String-Not-Equal` Macro . . . 12
2.1.36 The `Should-Not-String-Not-Equal` Macro . . . 12
2.1.37 The `Should-String-LessP` Macro . . . 12
2.1.38 The `Should-Not-String-LessP` Macro . . . 12
2.1.39 The `Should-String-GreaterP` Macro . . . 12
2.1.40 The `Should-Not-String-GreaterP` Macro . . . 12
2.1.41 The `Should-String-Not-GreaterP` Macro . . . 12
2.1.42 The `Should-Not-String-Not-GreaterP` Macro . . . 12
2.1.43 The `Should-String-Not-LessP` Macro . . . 12
2.1.44 The `Should-Not-String-Not-LessP` Macro . . . 12

3 The `Control` Package 13
3.1 Macros . . . 14
3.1.1 The `AIf` Macro . . . 14
3.1.2 The `A?If` Macro . . . 14
3.1.3 The `AAnd` Macro . . . 14
3.1.4 The `A?And` Macro . . . 14
3.1.5 The `ALambda` Macro . . . 14
3.1.6 The `A?Lambda` Macro . . . 14
3.1.7 The `ABlock` Macro . . . 14
3.1.8 The `A?Block` Macro . . . 14
3.1.9 The `ACond` Macro . . . 14
3.1.10 The `A?Cond` Macro . . . 14
3.1.11 The `AWhen` Macro . . . 14
3.1.12 The `A?When` Macro . . . 14
3.1.13 The `AWhile` Macro . . . 14
3.1.14 The `A?While` Macro . . . 14
3.1.15 The `DeleteF` Macro . . . 14
3.1.16 The `Do-While` Macro . . . 14
3.1.17 The `Do-Until` Macro . . . 14
3.1.18 The `For` Macro . . . 14
3.1.19 The `Forever` Macro . . . 14
3.1.20 The `Multicond` Macro . . . 14
3.1.21 The `OpF` Macro . . . 14
3.1.22 The `Swap` Macro . . . 14
3.1.23 The `Swap-Unless` Macro . . . 14
3.1.24 The `Swap-When` Macro . . . 14
3.1.25 The `Until` Macro . . . 14
3.1.26 The `While` Macro . . . 14
3.2 Functions . . . 14
3.2.1 The `Compose` Function . . . 14
3.2.2 The `Conjoin` Function . . . 14

3.2.3 The `Curry` Function . . . 14
3.2.4 The `Disjoin` Function . . . 14
3.2.5 The `Function-Alias` Function . . . 14
3.2.6 The `Operator-To-Function` Function . . . 14
3.2.7 The `RCompose` Function . . . 14
3.2.8 The `RCurry` Function . . . 14
3.2.9 The `Unimplemented` Function . . . 14
3.3 Generics . . . 14
3.3.1 The `Duplicate` Generic . . . 14

4 The `Numeric IN OWN FILE` Package **15**
4.1 Macros . . . 16
4.1.1 The `DivF` Macro . . . 16
4.1.2 The `MultF` Macro . . . 16
4.2 Functions . . . 16
4.2.1 The `Bit?` Function . . . 16
4.2.2 The `Fractional-Part` Function . . . 16
4.2.3 The `Fractional-Value` Function . . . 16
4.2.4 The `Integer-Range` Function . . . 16
4.2.5 The `Nonnegative?` Function . . . 16
4.2.6 The `Nonnegative-Integer?` Function . . . 16
4.2.7 The `Positive-Integer?` Function . . . 16
4.2.8 The `Product` Function . . . 16
4.2.9 The `Sum` Function . . . 16
4.2.10 The `Unsigned-Integer?` Function . . . 16
4.3 Types . . . 16
4.3.1 The `Nonnegative-Float` Type . . . 16
4.3.2 The `Nonnegative-Integer` Type . . . 16
4.3.3 The `Positive-Float` Type . . . 16
4.3.4 The `Positive-Integer` Type . . . 16

5 The `OS` Package **17**
5.1 Functions . . . 17
5.1.1 The `Perl` Function . . . 17
5.1.2 The `Python` Function . . . 17
5.1.3 The `Read-File` Function . . . 17
5.1.4 The `Read-Lines` Function . . . 17
5.1.5 The `Ruby` Function . . . 17
5.2 Parameters . . . 17
5.2.1 The `*Perl-Path*` Parameter . . . 17
5.2.2 The `*Python-Path*` Parameter . . . 17
5.2.3 The `*Ruby-Path*` Parameter . . . 17

6 The `Probability` Package **19**
6.1 Macros . . . 19
6.1.1 The `Decaying-Probabiliity?` Macro . . . 19
6.2 Functions . . . 19
6.2.1 The `Probability?` Function . . . 19
6.3 Types . . . 19
6.3.1 The `Probability` Type . . . 19

7 The `Random` Package **21**
7.1 Macros . . . 21
7.1.1 The `NShuffle` Macro . . . 21
7.2 Functions . . . 21
7.2.1 The `Gauss` Function . . . 21
7.2.2 The `Random-Argument` Function . . . 21
7.2.3 The `Coin-Toss` Function . . . 21
7.2.4 The `Random-In-Range` Function . . . 21
7.2.5 The `Random-In-Ranges` Function . . . 21
7.2.6 The `Random-Range` Function . . . 21
7.2.7 The `Randomize-Array` Function . . . 21
7.2.8 The `Random-Array` Function . . . 21
7.3 Generics . . . 21
7.3.1 The `Random-Element` Generic . . . 21
7.3.2 The `Shuffle` Generic . . . 21

8 The `Sequence` Package **23**
8.1 Macros . . . 24
8.1.1 The `Arefable?` Macro . . . 24
8.1.2 The `NConcF` Macro . . . 24
8.1.3 The `Nthable?` Macro . . . 24
8.1.4 The `Set-NthCdr` Macro . . . 24
8.2 Functions . . . 24
8.2.1 The `Array-Values` Function . . . 24
8.2.2 The `Nth-From-End` Function . . . 24
8.2.3 The `Sequence?` Function . . . 24
8.2.4 The `Empty-Sequence?` Function . . . 24
8.2.5 The `Join-Symbol-To-All-Following` Function . . . 24
8.2.6 The `Join-Symbol-To-All-Preceeding` Function . . . 24
8.2.7 The `List-To-Vector` Function . . . 24
8.2.8 The `Set-Equal` Function . . . 24
8.2.9 The `Simple-Vector-To-List` Function . . . 24
8.2.10 The `Sort-Order` Function . . . 24
8.2.11 The `The-Last` Function . . . 24
8.2.12 The `Vector-To-List` Function . . . 24
8.3 Generics . . . 24
8.3.1 The `Best` Generic . . . 24
8.3.2 The `Minimum` Generic . . . 24

8.3.3 The Minimum? Generic . . . 24
8.3.4 The Maximum Generic . . . 24
8.3.5 The Maximum? Generic . . . 24
8.3.6 The Sort-On Generic . . . 24
8.3.7 The Slice Generic . . . 24
8.3.8 The Split Generic . . . 24
8.3.9 The Worst Generic . . . 24

9 The String Package **25**
9.1 Functions . . . 25
9.1.1 The Character-Range Function . . . 25
9.1.2 The Character-Ranges Function . . . 25
9.1.3 The Escape-Tildes Function . . . 26
9.1.4 The Replace-Char Function . . . 26
9.1.5 The StrCat Function . . . 26
9.1.6 The StrMult Function . . . 26
9.1.7 The String-Join Function . . . 26
9.1.8 The Stringify Function . . . 26
9.1.9 The To-String Function . . . 26
9.2 Methods . . . 26
9.2.1 The Split Methods . . . 26

10 The Time-Series Package **27**
10.1 Macros . . . 27
10.1.1 The Snap-Index Macro . . . 27
10.2 Functions . . . 27
10.2.1 The Array-Raster-Line Function . . . 27
10.2.2 The Distance Function . . . 27
10.2.3 The Norm Function . . . 27
10.2.4 The Raster-Line Function . . . 27
10.2.5 The Similar-Points? Function . . . 27
10.2.6 The Time-Series? Function . . . 27
10.2.7 The Time-Multiseries? Function . . . 27
10.2.8 The TMSref Function . . . 27
10.2.9 The TMS-Dimensions Function . . . 27
10.2.10 The TMS-Raster-Line Function . . . 27
10.2.11 The TMS-Values Function . . . 27
10.3 Types . . . 27
10.3.1 The Time-Multiseries Type . . . 27

11 The Truth Package **29**
11.1 Functions . . . 29
11.1.1 The [?] Function . . . 29
11.1.2 The Toggle Function . . . 29
11.2 Generics . . . 29
11.2.1 The ? Generic . . . 29

12 The `Sigma` Package **31**
12.1 Variables 31
12.1.1 The `*Sigma-Packages*` Variable 31
12.2 Functions 31
12.2.1 The `Use-All-Sigma` Function 31

Chapter 1

Copyright

Copyright © 2005 – 2013, Christopher Mark Gore,
Soli Deo Gloria,
All rights reserved.

8729 Lower Marine Road, Saint Jacob, Illinois 62281 USA.
Web: `http://cgore.com`
Email: `cgore@cgore.com`

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Christopher Mark Gore nor the names of other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright holder or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

Chapter 2

The Behave Package

2.1 Macros

2.1.1 The Behavior Macro

2.1.2 The Spec Macro

2.1.3 The Should Macro

2.1.4 The Should-Not Macro

2.1.5 The Should-Be-Null Macro

2.1.6 The Should-Be-A Macro

2.1.7 The Should= Macro

2.1.8 The Should/= Macro

2.1.9 The Should< Macro

2.1.10 The Should> Macro

2.1.11 The Should<= Macro

2.1.12 The Should>= Macro

2.1.13 The Should-Eq Macro

2.1.14 The Should-Not-Eq Macro

2.1.15 The Should-Eql Macro

2.1.16 The Should-Not-Eql Macro

2.1.17 The Should-Equal Macro

2.1.18 The Should-Not-Equal Macro

2.1.19 The Should-EqualP Macro

2.1.20 The Should-Not-EqualP Macro

2.1.21 The Should-String= Macro

2.1.22 The Should-Not-String= Macro

2.1.23 The Should-String/= Macro

Chapter 3

The Control Package

3.1 Macros

3.1.1 The AIf Macro

3.1.2 The A?If Macro

3.1.3 The AAnd Macro

3.1.4 The A?And Macro

3.1.5 The ALambda Macro

3.1.6 The A?Lambda Macro

3.1.7 The ABlock Macro

3.1.8 The A?Block Macro

3.1.9 The ACond Macro

3.1.10 The A?Cond Macro

3.1.11 The AWhen Macro

3.1.12 The A?When Macro

3.1.13 The AWhile Macro

3.1.14 The A?While Macro

3.1.15 The DeleteF Macro

3.1.16 The Do-While Macro

3.1.17 The Do-Until Macro

3.1.18 The For Macro

3.1.19 The Forever Macro

3.1.20 The Multicond Macro

3.1.21 The OpF Macro

3.1.22 The Swap Macro

3.1.23 The Swap-Unless Macro

Chapter 4

The Numeric IN OWN FILE Package

4.1 Macros

4.1.1 The DivF Macro

4.1.2 The MultF Macro

4.2 Functions

4.2.1 The Bit? Function

4.2.2 The Fractional-Part Function

4.2.3 The Fractional-Value Function

4.2.4 The Integer-Range Function

4.2.5 The Nonnegative? Function

4.2.6 The Nonnegative-Integer? Function

4.2.7 The Positive-Integer? Function

4.2.8 The Product Function

4.2.9 The Sum Function

4.2.10 The Unsigned-Integer? Function

4.3 Types

4.3.1 The Nonnegative-Float Type

4.3.2 The Nonnegative-Integer Type

4.3.3 The Positive-Float Type

4.3.4 The Positive-Integer Type

Chapter 5

The OS Package

5.1 Functions

5.1.1 The Perl Function

5.1.2 The Python Function

5.1.3 The Read-File Function

5.1.4 The Read-Lines Function

5.1.5 The Ruby Function

5.2 Parameters

5.2.1 The *Perl-Path* Parameter

5.2.2 The *Python-Path* Parameter

5.2.3 The *Ruby-Path* Parameter

Chapter 6

The `Probability` Package

6.1 Macros

6.1.1 The `Decaying-Probabiliity?` Macro

6.2 Functions

6.2.1 The `Probability?` Function

6.3 Types

6.3.1 The `Probability` Type

Chapter 7

The `Random` Package

7.1 Macros

7.1.1 The `NShuffle` Macro

7.2 Functions

7.2.1 The `Gauss` Function

7.2.2 The `Random-Argument` Function

7.2.3 The `Coin-Toss` Function

7.2.4 The `Random-In-Range` Function

7.2.5 The `Random-In-Ranges` Function

7.2.6 The `Random-Range` Function

7.2.7 The `Randomize-Array` Function

7.2.8 The `Random-Array` Function

7.3 Generics

7.3.1 The `Random-Element` Generic

7.3.2 The `Shuffle` Generic

Chapter 8

The Sequence Package

8.1 Macros

8.1.1 The Arefable? Macro

8.1.2 The NConcF Macro

8.1.3 The Nthable? Macro

8.1.4 The Set-NthCdr Macro

8.2 Functions

8.2.1 The Array-Values Function

8.2.2 The Nth-From-End Function

8.2.3 The Sequence? Function

8.2.4 The Empty-Sequence? Function

8.2.5 The Join-Symbol-To-All-Following Function

8.2.6 The Join-Symbol-To-All-Preceeding Function

8.2.7 The List-To-Vector Function

8.2.8 The Set-Equal Function

8.2.9 The Simple-Vector-To-List Function

8.2.10 The Sort-Order Function

8.2.11 The The-Last Function

8.2.12 The Vector-To-List Function

8.3 Generics

8.3.1 The Best Generic

8.3.2 The Minimum Generic

8.3.3 The Minimum? Generic

8.3.4 The Maximum Generic

Chapter 9

The `String` Package

The `String` package contains useful tools for working with strings.

9.1 Functions

9.1.1 The `Character-Range` Function

The `character-range` function returns a list of characters from the *start* to the *end* character. Note that this is returning a list, not a string.

Syntax

`(character-range` *start end*`)` $\Longrightarrow$ `'(`*start ... end*`)`

Arguments and Values

Start The character to start the range with, inclusive.

End The character to end the range with, inclusive.

Examples

```
(character-range #\a #\e) ⟹ '(#\a #\b #\c #\d #\e)
(character-range #\e #\a) ⟹ '(#\a #\b #\c #\d #\e)
```

9.1.2 The `Character-Ranges` Function

The `character-ranges` function is a convenience wrapper for `character-range` function, concatenating several calls and making the resultant list contain only unique instances.

Syntax

`(character-ranges` $start_1$ end_1 ... $\Longrightarrow$ `'(`$character_1$ ...`)`

Arguments and Values

Start$_n$ The character to start the nth range with, inclusive.

End$_n$ The character to end the nth range with, inclusive.

Examples

```
(character-ranges #\a #\c #\x #\z) ⟹ '(#\a #\b #\c #\x #\y #\z)
(character-ranges #\a #\c #\a #\c) ⟹ '(#\a #\b #\c)
```

9.1.3 The `Escape-Tildes` Function

9.1.4 The `Replace-Char` Function

9.1.5 The `StrCat` Function

9.1.6 The `StrMult` Function

9.1.7 The `String-Join` Function

9.1.8 The `Stringify` Function

9.1.9 The `To-String` Function

9.2 Methods

9.2.1 The `Split` Methods

Chapter 10

The `Time-Series` Package

10.1 Macros

10.1.1 The `Snap-Index` Macro

10.2 Functions

10.2.1 The `Array-Raster-Line` Function

10.2.2 The `Distance` Function

10.2.3 The `Norm` Function

10.2.4 The `Raster-Line` Function

10.2.5 The `Similar-Points?` Function

10.2.6 The `Time-Series?` Function

10.2.7 The `Time-Multiseries?` Function

10.2.8 The `TMSref` Function

10.2.9 The `TMS-Dimensions` Function

10.2.10 The `TMS-Raster-Line` Function

10.2.11 The `TMS-Values` Function

10.3 Types

10.3.1 The `Time-Multiseries` Type

Chapter 11

The `Truth` Package

11.1 Functions

11.1.1 The `[?]` Function

11.1.2 The `Toggle` Function

11.2 Generics

11.2.1 The `?` Generic

Chapter 12

The `Sigma` Package

12.1 Variables

12.1.1 The `*Sigma-Packages*` Variable

12.2 Functions

12.2.1 The `Use-All-Sigma` Function