č Documentation

Louis A. Burke (lburke@labprogramming.net)

May 15, 2017

Contents

1 Introduction

č is a stack-based tacid code golf language. What sets it apart from many other golfing languages is that it cares deeply about the type of data being passed around.

By 'standard' the language performs extremely differently depending on the types of the input. However, since there is no common standard way to input a number as a natural number instead of an integer, for example, we specify our input types at compile-time. In theory the compiler could also produce sources for each possible combination of inputs, but this would be costly. The point is that while the compiler has to specify input types, this does not count as part of the source, but rather a deficiency of the compiler. An ideal č environment would have some sort of drop-down menu next to each input and would run the program on exactly the input types given.

2 Code Page

The following is the code page for č:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	⌀	¡	±	°	¬	¢	£	¤	¥	¦	¶	«	»	₥	₴	♠
1	Þ	Ð	ß	₱	ȝ	ſ	µ	ƴ	↶	↷	↯	¿	⚅	♭	♯	☨
2		!	''	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	÷
8	∀	∃	∄	∆	∇	∈	∉	∎	∏	∑	√	∞	∧	∨	∩	∪
9	∫	∴	∵	≝	≠	≡	≤	≥	⊄	⊅	⊆	⊇	₵	₿	₥	∝
A	₀	₁	₂	₃	₄	₅	₆	₇	₈	₉	₋	←	↑	→	↓	↺
B	₀	₁	₂	₃	₄	₅	₆	₇	₈	₉	□	◇	⍬	⊤	⊥	⊦
C	½	⌫	?	‼	☥	‡	•	‘	’	“	”	⋯	⋮	⋰	⋱	⁎
D	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	L	C	D	M
E	α	β	𝔇	𝔘	𝔇	𝔖	↤	↤	δ	Γ	θ	Λ	Ξ	Φ	Ψ	Ω
F	ℂ	ℍ	ℕ	ℙ	ℚ	ℝ	ℤ	π	ℯ	ℵ	ℝ	⊕	⊖	⊗	⊘	⊜

3 Execution

4 Types

4.1 Numbers

These are the numeric types available in č:

1. `integer` - Integers
2. `number` - Real numbers
3. `rational` - Rational numbers
4. `complex` - Complex numbers
5. `natural` - Natural numbers
6. `quaternion` - Quaternion numbers

4.2 Mathematical Objects

These are the mathematical types available in č:

1. `set` - Sets
2. `graph` - Graphs
3. `digraph` - Digraphs
4. `mgraph` - Multigraphs
5. `group` - Groups
6. `vector` - Vectors
7. `matrix` - Matrices

4.3 Computer Science Objects

These are the computer science types available in č:

1. `stack` - Stacks
2. `queue` - Queues
3. `string` - Strings
4. `list` - Lists
5. `tuple` - Tuples
6. `stream` - Streams

4.4 Games

These are the game types available in č:

1. `poker` - Poker cards/hands
2. `bridge` - Bridge cards/hands

4.5 Miscellaneous

These are the miscellaneous types available in č:

1. `image` - Images
2. `unit` - Standard Units
3. `grid` - Grids of things

5 Operators

Each operator is defined by a list of actions it takes on a specific argument. The argument is specified by a tuple to tuple type specification, then a short description of what the operator returns.

The actual argument tuple could be longer than that listed, in which case the function merely effects the first arguments which it needs.

Each operator will go through each valid mapping it has in order to decide what to do. Thus precedence is encoded into these listings. Some types which construct literals are not listed. Instead these types are presented in ALL CAPS to indicate that every operator has intrinsically defined actions on those types.

The special types $*$ and _ indicate anything and a single element respectively.

5.1 Null

The *Null* operator is represented by `⌀`.

1. `()` $\overset{\varnothing}{\mapsto}$ `(set)`: Returns the null set.

5.2 Top

The *Top* operator is represented by `¡`.

1. `(_,*)` $\overset{¡}{\mapsto}$ `(_)`: Returns the first argument.

5.3 Plus Minus

The *Plus Minus* operator is represented by `±`.

1. `(number)` $\overset{\pm}{\mapsto}$ `({number})`: Returns a set containing both positive and negative versions of the number.

5.4 Degrees

The *Degrees* operator is represented by `°`.

1. `(number)` $\overset{\circ}{\mapsto}$ `(number)`: Convert from degrees to radians.

5.5 Not

The *Not* operator is represented by `¬`.

1. `(_)` $\mapsto$ `(bool)`: Logical not.

5.6 Cent

The *Cent* operator is represented by `¢`.

1. .

5.7 Pound

The *Pound* operator is represented by `£`.

1. .

5.8 Graph

The *Graph* operator is represented by `¤`.

1. .

5.9 Split

The *Split* operator is represented by `¥`.

1. .

5.10 Slit

The *Slit* operator is represented by `¦`.

1. .

5.11 Drop

The *Drop* operator is represented by ¶.

1. .

5.12 Shift Left

The *Shift Left* operator is represented by «.

1. .

5.13 Shift Right

The *Shift Right* operator is represented by ».

1. .

5.14 Convolve

The *Convolve* operator is represented by ₪.

1. .

5.15 Bet

The *Bet* operator is represented by ב.

1. .

5.16 Spade

The *Spade* operator is represented by ♠.

1. .

5.17 Bump

The *Bump* operator is represented by Þ.

1. .

5.18 Enter

The *Enter* operator is represented by Ð.

1. .

5.19 Sharp S

The *Sharp S* operator is represented by ß.

1. .

5.20 Win

The *Win* operator is represented by Þ.

1. .

5.21 Yogh

The *Yogh* operator is represented by ȝ.

1. .

5.22 Not End

The *Not End* operator is represented by ſ.

1. .

5.23 Mu

The *Mu* operator is represented by µ.

1. .

5.24 Reciprocal

The *Reciprocal* operator is represented by ⅟.

1. .

5.25 Back

The *Back* operator is represented by ↶.

1. .

5.26 Forward

The *Forward* operator is represented by ↷.

1. .

5.27 Back and Forth

The *Back and Forth* operator is represented by ↹.

1. .

5.28 Irony

The *Irony* operator is represented by ؟.

1. .

5.29 Die

The *Die* operator is represented by ⚅.

1. .

5.30 Flat

The *Flat* operator is represented by ♭.

1. .

5.31 Sharp

The *Sharp* operator is represented by `#`.

1. .

5.32 Through

The *Through* operator is represented by `†`.

1. .

5.33 Space

The *Space* operator is represented by ` `.

1. .

5.34 Bang

The *Bang* operator is represented by `!`.

1. .

5.35 Quote

The *Quote* operator is represented by `''`.

1. .

5.36 Number

The *Number* operator is represented by `#`.

1. .

5.37 Money

The *Money* operator is represented by `$`.

1. .

5.38 Percent

The *Percent* operator is represented by `%`.

1. .

5.39 And

The *And* operator is represented by `&`.

1. .

5.40 Apostrophe

The *Apostrophe* operator is represented by `'`.

1. .

5.41 Open

The *Open* operator is represented by `(`.

1. .

5.42 Close

The *Close* operator is represented by `)`.

1. .

5.43 Star

The *Star* operator is represented by `*`.

1. .

5.44 Plus

The *Plus* operator is represented by `+`.

1. .

5.45 Comma

The *Comma* operator is represented by `,`.

1. .

5.46 Minus

The *Minus* operator is represented by `-`.

1. .

5.47 Dot

The *Dot* operator is represented by `.`.

1. .

5.48 Slash

The *Slash* operator is represented by `/`.

1. .

5.49 Zero

The *Zero* operator is represented by `0`.

1. .

5.50 One

The *One* operator is represented by `1`.

1. .

5.51 Two

The *Two* operator is represented by `2`.

1. .

5.52 Three

The *Three* operator is represented by `3`.

1. .

5.53 Four

The *Four* operator is represented by `4`.

1. .

5.54 Five

The *Five* operator is represented by `5`.

1. .

5.55 Six

The *Six* operator is represented by `6`.

1. .

5.56 Seven

The *Seven* operator is represented by `7`.

1. .

5.57 Eight

The *Eight* operator is represented by `8`.

1. .

5.58 Nine

The *Nine* operator is represented by `9`.

1. .

5.59 Colon

The *Colon* operator is represented by `:`.

1. .

5.60 Semi-colon

The *Semi-colon* operator is represented by `;`.

1. .

5.61 Less

The *Less* operator is represented by `<`.

1. .

5.62 Equals

The *Equals* operator is represented by `=`.

1. `(list,list)` $\overset{=}{\mapsto}$ `(list)`: List with true wherever the elements in both lists are the same.
2. `(number,number)` $\overset{=}{\mapsto}$ `(bool)`: True if the two numbers are equal.

5.63 Greater

The *Greater* operator is represented by `>`.

1. .

5.64 Query

The *Query* operator is represented by `?`.

1. .

5.65 At

The *At* operator is represented by `@`.

1. .

5.66 A

The *A* operator is represented by `A`.

1. .

5.67 B

The *B* operator is represented by `B`.

1. .

5.68 C

The *C* operator is represented by `C`.

1. .

5.69 D

The *D* operator is represented by `D`.

1. .

5.70 E

The *E* operator is represented by `E`.

1. .

5.71 F

The F operator is represented by `F`.

1. .

5.72 G

The G operator is represented by `G`.

1. .

5.73 H

The H operator is represented by `H`.

1. .

5.74 I

The I operator is represented by `I`.

1. .

5.75 J

The J operator is represented by `J`.

1. .

5.76 K

The K operator is represented by `K`.

1. .

5.77 L

The L operator is represented by `L`.

1. .

5.78 M

The M operator is represented by `M`.

1. .

5.79 N

The N operator is represented by `N`.

1. .

5.80 O

The O operator is represented by `O`.

1. .

5.81 P

The P operator is represented by `P`.

1. .

5.82 Q

The Q operator is represented by `Q`.

1. .

5.83 R

The R operator is represented by `R`.

1. .

5.84 S

The S operator is represented by `S`.

1. .

5.85 T

The T operator is represented by `T`.

1. .

5.86 U

The U operator is represented by `U`.

1. .

5.87 V

The V operator is represented by `V`.

1. .

5.88 W

The W operator is represented by `W`.

1. .

5.89 X

The X operator is represented by `X`.

1. .

5.90 Y

The Y operator is represented by `Y`.

1. .

5.91 Z

The *Z* operator is represented by `Z`.

1. .

5.92 Start

The *Start* operator is represented by `[`.

1. .

5.93 Backslash

The *Backslash* operator is represented by `\`.

1. .

5.94 End

The *End* operator is represented by `]`.

1. .

5.95 Up

The *Up* operator is represented by `^`.

1. .

5.96 Under

The *Under* operator is represented by `_`.

1. .

5.97 Tick

The *Tick* operator is represented by `` ` ``.

1. .

5.98 a

The *a* operator is represented by `a`.

1. .

5.99 b

The *b* operator is represented by `b`.

1. .

5.100 c

The *c* operator is represented by `c`.

1. .

5.101 d

The d operator is represented by `d`.

1. .

5.102 e

The e operator is represented by `e`.

1. .

5.103 f

The f operator is represented by `f`.

1. .

5.104 g

The g operator is represented by `g`.

1. .

5.105 h

The h operator is represented by `h`.

1. .

5.106 i

The i operator is represented by `i`.

1. .

5.107 j

The j operator is represented by `j`.

1. .

5.108 k

The k operator is represented by `k`.

1. .

5.109 l

The l operator is represented by `l`.

1. .

5.110 m

The m operator is represented by `m`.

1. .

5.111 n

The n operator is represented by `n`.

1. .

5.112 o

The o operator is represented by `o`.

1. .

5.113 p

The p operator is represented by `p`.

1. .

5.114 q

The q operator is represented by `q`.

1. .

5.115 r

The r operator is represented by `r`.

1. .

5.116 s

The s operator is represented by `s`.

1. .

5.117 t

The t operator is represented by `t`.

1. .

5.118 u

The u operator is represented by `u`.

1. .

5.119 v

The v operator is represented by `v`.

1. .

5.120 w

The w operator is represented by `w`.

1. .

5.121 x

The *x* operator is represented by `x`.

1. .

5.122 y

The *y* operator is represented by `y`.

1. .

5.123 z

The *z* operator is represented by `z`.

1. .

5.124 Begin

The *Begin* operator is represented by `{`.

1. .

5.125 Pipe

The *Pipe* operator is represented by `|`.

1. .

5.126 End

The *End* operator is represented by `}`.

1. .

5.127 Tilde

The *Tilde* operator is represented by `~`.

1. .

5.128 Divide

The *Divide* operator is represented by `÷`.

1. .

5.129 Universal

The *Universal* operator is represented by `∀`.

1. .

5.130 Existential

The *Existential* operator is represented by `∃`.

1. .

5.131 Non Existential

The *Non Existential* operator is represented by `∄`.

1. .

5.132 Increment

The *Increment* operator is represented by `∆`.

1. .

5.133 Decrement

The *Decrement* operator is represented by `∇`.

1. .

5.134 In

The *In* operator is represented by `∈`.

1. `(_,set)` $\overset{\in}{\mapsto}$ `(_,set)`: Coerces the top element to an element of the set.

5.135 Not In

The *Not In* operator is represented by `∉`.

1. .

5.136 Box

The *Box* operator is represented by `∎`.

1. .

5.137 Product

The *Product* operator is represented by `∏`.

1. .

5.138 Sum

The *Sum* operator is represented by `∑`.

1. .

5.139 Root

The *Root* operator is represented by `√`.

1. .

5.140 Infinity

The *Infinity* operator is represented by `∞`.

1. .

5.141 Conjunction

The *Conjunction* operator is represented by `∧`.

1. .

5.142 Disjunction

The *Disjunction* operator is represented by `∨`.

1. .

5.143 Intersection

The *Intersection* operator is represented by `∩`.

1. .

5.144 Union

The *Union* operator is represented by `∪`.

1. .

5.145 Integral

The *Integral* operator is represented by `∫`.

1. .

5.146 Therefore

The *Therefore* operator is represented by `∴`.

1. .

5.147 Because

The *Because* operator is represented by `∵`.

1. .

5.148 Define

The *Define* operator is represented by `≝`.

1. .

5.149 Not Equals

The *Not Equals* operator is represented by `≠`.

1. .

5.150 Exact Equals

The *Exact Equals* operator is represented by `≡`.

1. `(_,_)` $\overset{\equiv}{\mapsto}$ `(bool)`: True if the two elements on top of the stack are identical, false otherwise.

5.151 Not Greater

The *Not Greater* operator is represented by ≰.

1. .

5.152 Not Less

The *Not Less* operator is represented by ≱.

1. .

5.153 Make Greater

The *Make Greater* operator is represented by ≯.

1. .

5.154 Make Less

The *Make Less* operator is represented by ≮.

1. .

5.155 Subset

The *Subset* operator is represented by ⊆.

1. .

5.156 Superset

The *Superset* operator is represented by ⊇.

1. .

5.157 Make Superset

The *Make Superset* operator is represented by ⊉.

1. .

5.158 Make Subset

The *Make Subset* operator is represented by ⊈.

1. .

5.159 Join

The *Join* operator is represented by ⋈.

1. .

5.160 Proportional

The *Proportional* operator is represented by ∝.

1. .

5.161 Super Zero

The *Super Zero* operator is represented by `0`.

1. .

5.162 Super One

The *Super One* operator is represented by `1`.

1. .

5.163 Super Two

The *Super Two* operator is represented by `2`.

1. .

5.164 Super Three

The *Super Three* operator is represented by `3`.

1. .

5.165 Super Four

The *Super Four* operator is represented by `4`.

1. .

5.166 Super Five

The *Super Five* operator is represented by `5`.

1. .

5.167 Super Six

The *Super Six* operator is represented by `6`.

1. .

5.168 Super Seven

The *Super Seven* operator is represented by `7`.

1. .

5.169 Super Eight

The *Super Eight* operator is represented by `8`.

1. .

5.170 Super Nine

The *Super Nine* operator is represented by `9`.

1. .

5.171 Super Minus

The *Super Minus* operator is represented by ⁻.

1. .

5.172 Left

The *Left* operator is represented by ←.

1. .

5.173 Up

The *Up* operator is represented by ↑.

1. .

5.174 Right

The *Right* operator is represented by →.

1. .

5.175 Down

The *Down* operator is represented by ↓.

1. .

5.176 Loop

The *Loop* operator is represented by ↺.

1. .

5.177 Sub 0

The *Sub 0* operator is represented by ₀.

1. .

5.178 Sub 1

The *Sub 1* operator is represented by ₁.

1. .

5.179 Sub 2

The *Sub 2* operator is represented by ₂.

1. .

5.180 Sub 3

The *Sub 3* operator is represented by ₃.

1. .

5.181 Sub 4

The *Sub 4* operator is represented by `₄`.

1. .

5.182 Sub 5

The *Sub 5* operator is represented by `₅`.

1. .

5.183 Sub 6

The *Sub 6* operator is represented by `₆`.

1. .

5.184 Sub 7

The *Sub 7* operator is represented by `₇`.

1. .

5.185 Sub 8

The *Sub 8* operator is represented by `₈`.

1. .

5.186 Sub 9

The *Sub 9* operator is represented by `₉`.

1. .

5.187 Necessarily

The *Necessarily* operator is represented by `□`.

1. .

5.188 Possibly

The *Possibly* operator is represented by `◇`.

1. .

5.189 Impossibly

The *Impossibly* operator is represented by `∯`.

1. .

5.190 True

The *True* operator is represented by `⊤`.

1. .

5.191 False

The *False* operator is represented by `⊥`.

1. .

5.192 Proves

The *Proves* operator is represented by `⊦`.

1. .

5.193 Half

The *Half* operator is represented by `½`.

1. .

5.194 Delete

The *Delete* operator is represented by `⌫`.

1. .

5.195 Stop

The *Stop* operator is represented by `?`.

1. .

5.196 Primorial

The *Primorial* operator is represented by `‼`.

1. .

5.197 Dagger

The *Dagger* operator is represented by `†`.

1. .

5.198 Double Dagger

The *Double Dagger* operator is represented by `‡`.

1. .

5.199 Bullet

The *Bullet* operator is represented by `•`.

1. .

5.200 Begin Function

The *Begin Function* operator is represented by `‘`.

1. `()` $\overset{‘}{\mapsto}$ `(FUNCTION_DEFINITION)`: Returns an empty function definition.

5.201 End Function

The *End Function* operator is represented by `’`.

1. `(FUNCTION_DEFINITION)` $\overset{’}{\mapsto}$ `(function)`: Compiles the definition of a function to an actual callable function.

5.202 Begin String

The *Begin String* operator is represented by `“`.

1. .

5.203 End String

The *End String* operator is represented by `”`.

1. .

5.204 Flatten

The *Flatten* operator is represented by `⋯`.

1. .

5.205 Expand

The *Expand* operator is represented by `⋮`.

1. .

5.206 Ascending

The *Ascending* operator is represented by `⋰`.

1. `(list)` $\overset{⋰}{\mapsto}$ `(list)`: Sorts the list in ascending order.

5.207 Descending

The *Descending* operator is represented by `⋱`.

1. `(list)` $\overset{⋱}{\mapsto}$ `(list)`: Sorts the list in descending order.

5.208 Graph

The *Graph* operator is represented by `⁜`.

1. .

5.209 One

The *One* operator is represented by `I`.

1. .

5.210 Two

The *Two* operator is represented by `II`.

1. .

5.211 Three

The *Three* operator is represented by 𝕀𝕀.

1. .

5.212 Four

The *Four* operator is represented by 𝕀𝕍.

1. .

5.213 Five

The *Five* operator is represented by 𝕍.

1. .

5.214 Six

The *Six* operator is represented by 𝕍𝕀.

1. .

5.215 Seven

The *Seven* operator is represented by 𝕍𝕀𝕀.

1. .

5.216 Eight

The *Eight* operator is represented by 𝕍𝕀𝕀𝕀.

1. .

5.217 Nine

The *Nine* operator is represented by 𝕀𝕏.

1. .

5.218 Ten

The *Ten* operator is represented by 𝕏.

1. .

5.219 Eleven

The *Eleven* operator is represented by 𝕏𝕀.

1. .

5.220 Twelve

The *Twelve* operator is represented by 𝕏𝕀𝕀.

1. .

5.221 Fifty

The *Fifty* operator is represented by 𝙻.

1. .

5.222 Hundred

The *Hundred* operator is represented by 𝙲.

1. .

5.223 Five Hundred

The *Five Hundred* operator is represented by 𝙳.

1. .

5.224 Thousand

The *Thousand* operator is represented by 𝙼.

1. .

5.225 Alpha

The *Alpha* operator is represented by 𝛼.

1. .

5.226 Beta

The *Beta* operator is represented by 𝛽.

1. .

5.227 Five Thousand

The *Five Thousand* operator is represented by 𝔻.

1. .

5.228 Ten Thousand

The *Ten Thousand* operator is represented by 𝕄.

1. .

5.229 Fifty Thousand

The *Fifty Thousand* operator is represented by 𝔻.

1. .

5.230 Hundred Thousand

The *Hundred Thousand* operator is represented by 𝕊.

1. .

5.231 Pop

The *Pop* operator is represented by `↤`.

1. .

5.232 Push

The *Push* operator is represented by `⇤`.

1. .

5.233 Delta

The *Delta* operator is represented by `δ`.

1. .

5.234 Gamma

The *Gamma* operator is represented by `Γ`.

1. .

5.235 Theta

The *Theta* operator is represented by `θ`.

1. .

5.236 Lambda

The *Lambda* operator is represented by `∧`.

1. .

5.237 Xi

The *Xi* operator is represented by `Ξ`.

1. .

5.238 Phi

The *Phi* operator is represented by `Φ`.

1. .

5.239 Psi

The *Psi* operator is represented by `Ψ`.

1. .

5.240 Omega

The *Omega* operator is represented by `Ω`.

1. .

5.241 Complex

The *Complex* operator is represented by $\mathbb{C}$.

1. .

5.242 Quaternion

The *Quaternion* operator is represented by $\mathbb{H}$.

1. .

5.243 Natural

The *Natural* operator is represented by $\mathbb{N}$.

1. .

5.244 Prime

The *Prime* operator is represented by $\mathbb{P}$.

1. `(natural)` $\overset{\mathbb{P}}{\mapsto}$ `(bool)`: Check if the number is prime.

5.245 Rational

The *Rational* operator is represented by $\mathbb{Q}$.

1. .

5.246 Real

The *Real* operator is represented by $\mathbb{R}$.

1. .

5.247 Integer

The *Integer* operator is represented by $\mathbb{Z}$.

1. .

5.248 Pi

The *Pi* operator is represented by π.

1. .

5.249 Euler

The *Euler* operator is represented by e.

1. .

5.250 Aleph

The *Aleph* operator is represented by $\aleph$.

1. .

5.251 Prescribe

The *Prescribe* operator is represented by ⍈.

1. .

5.252 Oplus

The *Oplus* operator is represented by ⊕.

1. .

5.253 Ominus

The *Ominus* operator is represented by ⊖.

1. .

5.254 Otimes

The *Otimes* operator is represented by ⊗.

1. .

5.255 Odivide

The *Odivide* operator is represented by ⊘.

1. .

5.256 Oequals

The *Oequals* operator is represented by ⊜.

1. .

6 Examples