bunch

set

string

list

bunch	unpackaged	unindexed
set		
string		
list		

bunch	unpackaged	unindexed
set	packaged	unindexed
string		
list		

bunch	unpackaged	unindexed
set	packaged	unindexed
string	unpackaged	indexed
list		

bunch	unpackaged	unindexed
set	packaged	unindexed
string	unpackaged	indexed
list	packaged	indexed

String Theory

Strings are indexed sequences.

String Theory

Strings are indexed sequences.

nil	the empty string

String Theory

Strings are indexed sequences.

nil	the empty string
3	a one-item string

String Theory

Strings are indexed sequences.

nil	the empty string
3	a one-item string
3; 5; 7; 9	a four-item string

String Theory

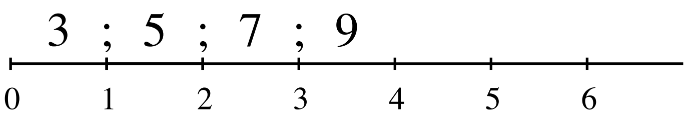
Strings are indexed sequences.

nil	the empty string
3	a one-item string
$3; 5; 7; 9$	a four-item string
$\leftrightarrow(3; 5; 7; 9) = 4$	string length operator

String Theory

Strings are indexed sequences.

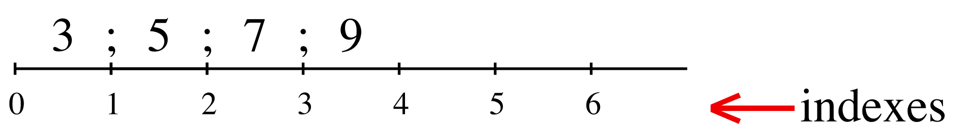
nil	the empty string
3	a one-item string
3; 5; 7; 9	a four-item string
$\leftrightarrow(3; 5; 7; 9) \ = \ 4$	string length operator



String Theory

Strings are indexed sequences.

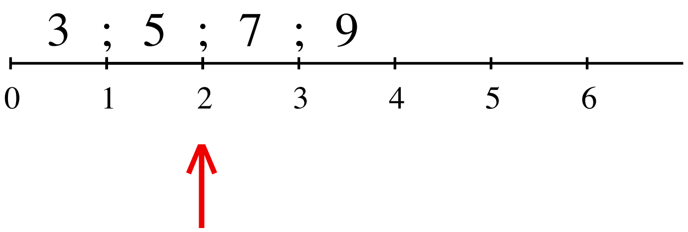
nil	the empty string
3	a one-item string
3; 5; 7; 9	a four-item string
$\leftrightarrow(3;\ 5;\ 7;\ 9)\ =\ 4$	string length operator



String Theory

Strings are indexed sequences.

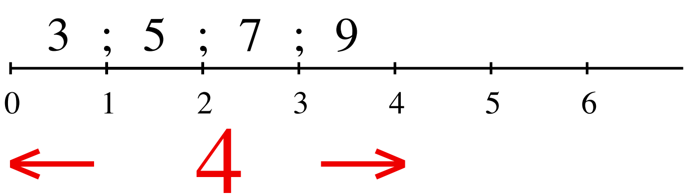
nil	the empty string
3	a one-item string
$3;\ 5;\ 7;\ 9$	a four-item string
$\leftrightarrow(3;\ 5;\ 7;\ 9)\ =\ 4$	string length operator



String Theory

Strings are indexed sequences.

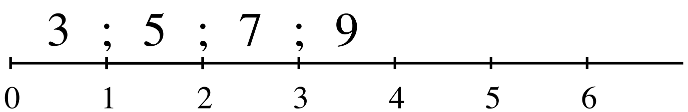
nil	the empty string
3	a one-item string
3; 5; 7; 9	a four-item string
$\leftrightarrow(3;\,5;\,7;\,9)\ =\ 4$	string length operator



String Theory

Strings are indexed sequences.

nil	the empty string
3	a one-item string
$3; 5; 7; 9$	a four-item string
$\leftrightarrow(3; 5; 7; 9) = 4$	string length operator

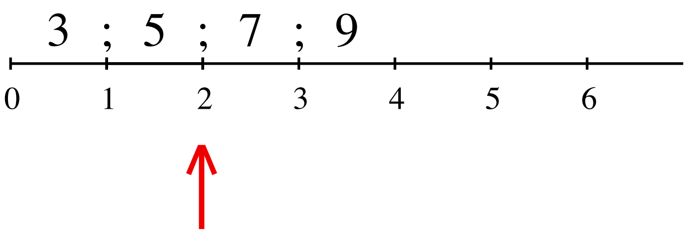


$(3; 5; 7; 9)_2 = 7$

String Theory

Strings are indexed sequences.

nil	the empty string
3	a one-item string
$3;\ 5;\ 7;\ 9$	a four-item string
$\leftrightarrow(3;\ 5;\ 7;\ 9)\ =\ 4$	string length operator



$(3;\ 5;\ 7;\ 9)_2\ =\ 7$

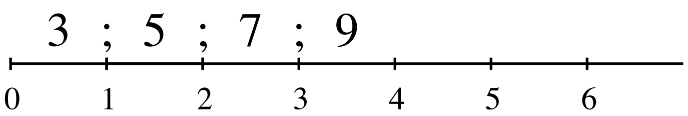
At index *n* , the number of items processed is *n*

the next item to be processed is item *n*

String Theory

Strings are indexed sequences.

nil	the empty string
3	a one-item string
$3;\ 5;\ 7;\ 9$	a four-item string
$\leftrightarrow(3;\ 5;\ 7;\ 9)\ =\ 4$	string length operator



$$(3;\ 5;\ 7;\ 9)_2\ =\ 7$$

$$(3;\ 5;\ 7;\ 9)_{2;\ 1;\ 2}\ =\ 7;\ 5;\ 7$$

Zero

Zero

John Allen Paulos:

Innumeracy: Mathematical Illiteracy and its Consequences, Hill and Wang, 1988

Beyond Numeracy, Knopf, 1991

Zero

John Allen Paulos:

Innumeracy: Mathematical Illiteracy and its Consequences, Hill and Wang, 1988

Beyond Numeracy, Knopf, 1991

0.10¢

Zero

John Allen Paulos:

Innumeracy: Mathematical Illiteracy and its Consequences, Hill and Wang, 1988

Beyond Numeracy, Knopf, 1991

0.10¢ $1.02.9

Zero

John Allen Paulos:

Innumeracy: Mathematical Illiteracy and its Consequences, Hill and Wang, 1988

Beyond Numeracy, Knopf, 1991

0.10¢ $1.02.9

There are a number of things to discuss. (But not zero things to discuss.)

Zero

John Allen Paulos:

Innumeracy: Mathematical Illiteracy and its Consequences, Hill and Wang, 1988

Beyond Numeracy, Knopf, 1991

0.10¢ $1.02.9

There are a number of things to discuss. (But not zero things to discuss.)

Subtract line A from line B; if there is no difference, write "nil".

Zero

John Allen Paulos:

Innumeracy: Mathematical Illiteracy and its Consequences, Hill and Wang, 1988

Beyond Numeracy, Knopf, 1991

0.10¢ $1.02.9

There are a number of things to discuss. (But not zero things to discuss.)

Subtract line A from line B; if there is no difference, write "nil".

keyboard, telephone: 1 2 3 4 5 6 7 8 9 0

If you need more space, use the "Comments" section on page 28.

STEP 7

How many persons who have a usual home somewhere else in Canada stayed here overnight between June 3 and 4, 1991?

○ None

OR

02 ☐ ◄ Number of persons

STEP 8

Does anyone in this household OPERATE a farm, ranch or other agricultural holding?

Other agricultural holdings include, for example: feedlots; greenhouses; mushroom houses; nurseries; fur farms; and beekeeping, sod, berry and maple syrup operations.

03 ○ No

04 ○ Yes

STEP 9

Turn the page and copy the names from Step 5 into the spaces across the top of the page.

Then continue with the questionnaire.

Note:

If there are **more than six persons** in this household, enter the first six on this questionnaire and continue on a second questionnaire. If you do not have a second questionnaire, note this in the "Comments" section on page 28. A census representative will contact you.

Long Distance Calls

Codes for Dialing Overseas

For station-to-station calls; DIAL/PRESS:
011 + Country Code + Routing Code + Local Number

For automated Calling Card™ calls; PRESS:
01 + Country Code + Routing Code + Local Number
+ (after tone) your 14 digit North American card number

For person-to-person and other types of calls; DIAL/PRESS:
01 + Country Code + Routing Code + Local Number

For countries or cities not listed; DIAL/PRESS:
"0" (zero) and ask the operator for the routing codes.

Legend:

*	Routing codes not required
TD	Approximate time difference – in hours from Eastern Standard time
NA	Time difference not applicable

These overseas codes were in effect at the time of printing this directory.

COUNTRY & COUNTRY CODE	ROUTING CODE	TD
Cameroon 237	*	+6
Chile 56		+2
Santiago	2	
Valparaiso	32	
Viña Del Mar	32	
China 86		+13
Beijing (Peking)	1	
Chengdu	28	
Shanghai	21	
Christmas Island 672		+12
Cuba 53	*	NA
Except Havana	7	
Cyprus 357		+7
Larnaca	4	
Limassol	5	
Nicosia	2	
Czech And Slovak 42		+6
Bratislava	7	
Brno	5	
Prague	2	
Denmark 45	*	+6
Ethiopia 251		+8
Addis Ababa	1	
Asmara	4	
Dire Dawa	5	
Faroe Islands 298	*	+6
Fiji Islands 679	*	+17
Finland 358		+7
Helsinki	0	
Tampere	31	

1993
1996 same

Overseas Calls

Codes for frequently called countries

For station-to-station calls: dial 011 + the country code + the routing code + the local number.

For person-to-person and other types of calls: dial 01 + the country code + the routing code + the local number.

For Automated Billing Service calls: dial 01 + the country code + the routing code + the local number, and (after the tone) the Automated Billing Service will tell you what steps to take to complete your call.

Dial 0 (zero):
- to obtain a number you don't know.
- to obtain credit for unsatisfactory calls, e.g. when you reach a wrong number.

Legend:
- * Routing codes not required.
- TD Approximate time difference in hours from Eastern Standard time
- = Time difference not applicable.

These overseas codes were in effect at the time of printing this directory.

COUNTRY & ROUTING CODE	TD
Algeria 213	+6
Alger 2	
Constantine 4	
Oran 6	
Argentina 54	+2
Buenos Aires 1	
Córdoba 51	
Rosario 41	
Australia 61	+16
Adelaide 8	
Brisbane 7	
Canberra 6	
Melbourne 3	

COUNTRY & ROUTING CODE	TD
Brazil 55	+2
Belo Horizonte 31	
Brasilia 61	
Pôrto Alegre 51	
Recife 81	
Rio de Janeiro 21	
Salvador 71	
São Paulo 11	
Chile 56	+1
Santiago 2	
Valparaíso 32	
China 86	+13
Beijing (Peking) 1	
Shanghai 21	

COUNTRY & ROUTING CODE	TD
Denmark 45 *	+6
Egypt 20	+7
Alexandria 3	
Cairo 2	
Port Said 66	
El Salvador 503 *	−1
Finland 358	+7
Helsinki 0	
Tampere 31	
Turku (Åbo) 21	

COUNTRY & ROUTING CODE	TD
Greece 30	+7
Athens 1	
Iráklion Kritis 81	
Kavalla 51	
Lárissa 41	
Pátrai 61	
Thessaloniki 31	
Volos 421	
Guatemala 502	−1
Escuintla 9	
Guatemala City 2	
Quetzaltenango 9	
Haiti 509 *	=

1997-8

1998-9

OVERSEAS CALLS

Legend for Country and City Code Table

0 No time difference

★ City codes not required.

TD Approximate time difference in hours from Eastern Standard time

FREQUENTLY CALLED PLACES

COUNTRY & CITY CODE	TD	COUNTRY & CITY CODE	TD	COUNTRY & CITY CODE	TD	COUNTRY & CITY CODE	TD
Algeria 213 Alger 2 Constantine 4	+6	**Bangladesh 880** Chittagong 31 Khulna 41	+11	**Chile 56** Santiago 2 Valparaíso 32	+1	**Costa Rica 506** ★	–1
Argentina 54 Buenos Aires 1 Córdoba 51	+2	**Belgium 32** Antwerpen 3 Brussels 2 Gent 9	+6	**China 86** Beijing (Peking) 10 Shanghai 21 Tianjim (Tientsin) 22	+13	**Croatia 385**	+6
Australia 61 Adelaide 8 Canberra 6 Melbourne 3 Sydney 2	+16	**Bosnia–Herzegovina 387**	+6	**C.I.S. 7** (Commonwealth of Independent States) Moscow 095 St. Petersburg 812	+8	**Cyprus 357** Larnaca 4 Limassol 5	+7
Austria 43 Innsbruck 512 Salzburg 662 Vienna 1	+6	**Brazil 55** Brasilia 61 Pôrto Alegre 51 Rio de Janeiro 21 São Paulo 11	+2	**Colombia 57** Bogotá 1 Medellín 4	0	**Czech & Slovak 42** Bratislava 7 Prague 2	+6
						Denmark 45 ★	+6

For countries or cities *NOT* listed dial “0” (zero) and ask the operator for the appropriate codes.
These overseas codes were in effect at the time of printing this directory.

Zero

Measuring must start at 0.

Zero

Measuring must start at 0. Counting is measuring.

Zero

Measuring must start at 0. Counting is measuring.

An octave is an interval of 8. What interval is 2 octaves?

Zero

Measuring must start at 0. Counting is measuring.

An octave is an interval of 8. What interval is 2 octaves? It's 15.

Zero

Measuring must start at 0. Counting is measuring.

An octave is an interval of 8. What interval is 2 octaves? It's 15.

How many years from July 1 in year X to July 1 in year Y?

Zero

Measuring must start at 0. Counting is measuring.

An octave is an interval of 8. What interval is 2 octaves? It's 15.

How many years from July 1 in year X to July 1 in year Y? Y–X years?

Zero

Measuring must start at 0. Counting is measuring.

An octave is an interval of 8. What interval is 2 octaves? It's 15.

How many years from July 1 in year X to July 1 in year Y? Y–X years?

Fortran 1955 loop body had to be executed at least once.

Zero

Measuring must start at 0. Counting is measuring.

An octave is an interval of 8. What interval is 2 octaves? It's 15.

How many years from July 1 in year X to July 1 in year Y?   Y–X years?

Fortran 1955 loop body had to be executed at least once.

> *count*:= 0.
> **while** there's another one
> **do**
>   *count*:= *count* + 1
> **od**

Zero

Measuring must start at 0. Counting is measuring.

An octave is an interval of 8. What interval is 2 octaves? It's 15.

How many years from July 1 in year X to July 1 in year Y?   Y–X years?

Fortran 1955 loop body had to be executed at least once.

> *count*:= 0.
> **while** there's another one
> **do**
>   *count*:= *count* + 1
> **od**

Algol 1958, PL/I, Pascal: array must have at least 1 element.

Zero

Zero

first: preceding all others in time, order, or importance

Zero

first: preceding all others in time, order, or importance

last: following all others in time, order, or importance

Zero

first: preceding all others in time, order, or importance

last: following all others in time, order, or importance

second: following the first

Zero

first:	preceding all others in time, order, or importance	1st
last:	following all others in time, order, or importance	
second:	following the first	2nd

Zero

first:	preceding all others in time, order, or importance	~~1st~~	0st
last:	following all others in time, order, or importance		
second:	following the first	~~2nd~~	1nd

Zero

first: preceding all others in time, order, or importance	~~1st~~	~~0st~~
last: following all others in time, order, or importance		
second: following the first	~~2nd~~	~~1nd~~

Zero

first: preceding all others in time, order, or importance ~~1st~~ ~~0st~~

last: following all others in time, order, or importance

second: following the first ~~2nd~~ ~~1nd~~

third year of life = what age?

Zero

first: preceding all others in time, order, or importance ~~1st~~ ~~0st~~

last: following all others in time, order, or importance

second: following the first ~~2nd~~ ~~1nd~~

third year of life = what age? 2

Zero

first: preceding all others in time, order, or importance ~~1st~~ ~~0st~~

last: following all others in time, order, or importance

second: following the first ~~2nd~~ ~~1nd~~

third year of life = what age? 2

tenth annual picnic = how many years?

Zero

first: preceding all others in time, order, or importance ~~1st~~ ~~0st~~

last: following all others in time, order, or importance

second: following the first ~~2nd~~ ~~1nd~~

third year of life = what age? 2

tenth annual picnic = how many years? 9

Zero

first: preceding all others in time, order, or importance ~~1st~~ ~~0st~~

last: following all others in time, order, or importance

second: following the first ~~2nd~~ ~~1nd~~

third year of life = what age? 2

tenth annual picnic = how many years? 9

the eleventh hour: the latest possible time

Zero

first: preceding all others in time, order, or importance ~~1st~~ ~~0st~~

last: following all others in time, order, or importance

second: following the first ~~2nd~~ ~~1nd~~

third year of life = what age? 2

tenth annual picnic = how many years? 9

the eleventh hour: the latest possible time 10 to 11 o'clock?

Zero

first: preceding all others in time, order, or importance ~~1st~~ ~~0st~~

last: following all others in time, order, or importance

second: following the first ~~2nd~~ ~~1nd~~

~~third~~ year of life = what age? 2

~~tenth~~ annual picnic = how many years? 9

the ~~eleventh~~ hour: the latest possible time 10 to 11 o'clock?

Zero

first: preceding all others in time, order, or importance ~~1st~~ ~~0st~~

last: following all others in time, order, or importance

second: following the first ~~2nd~~ ~~1nd~~

~~third~~ year of life = what age? 2

~~tenth~~ annual picnic = how many years? 9

the ~~eleventh~~ hour: the latest possible time 10 to 11 o'clock?

the ~~fifteenth~~ item = item 15 ? item 14 ?

Zero

first: preceding all others in time, order, or importance ~~1st~~ ~~0st~~

last: following all others in time, order, or importance

second: following the first ~~2nd~~ ~~1nd~~

~~third~~ year of life = what age? 2

~~tenth~~ annual picnic = how many years? 9

the ~~eleventh~~ hour: the latest possible time 10 to 11 o'clock?

the ~~fifteenth~~ item = item 15 ? item 14 ?

zeroth item = item 0 ?

Zero

first: preceding all others in time, order, or importance ~~1st~~ ~~0st~~

last: following all others in time, order, or importance

second: following the first ~~2nd~~ ~~1nd~~

~~third~~ year of life = what age? 2

~~tenth~~ annual picnic = how many years? 9

the ~~eleventh~~ hour: the latest possible time 10 to 11 o'clock?

the ~~fifteenth~~ item = item 15 ? item 14 ?

~~zeroth~~ item = item 0 ? **first item**

String Theory

Strings are indexed sequences.

String Theory

Strings are indexed sequences.

$3; 6; 4; 1 < 3; 7; 2$ order

String Theory

Strings are indexed sequences.

$3; 6; 4; 1 < 3; 7; 2$	order
$3; 6; 4 < 3; 6; 4; 1$	order

String Theory

Strings are indexed sequences.

$3;\ 6;\ 4;\ 1\ <\ 3;\ 7;\ 2$	order
$3;\ 6;\ 4\ <\ 3;\ 6;\ 4;\ 1$	order
$x;..y$	" x to y " for $x \le y$

String Theory

Strings are indexed sequences.

$3;\ 6;\ 4;\ 1\ <\ 3;\ 7;\ 2$	order
$3;\ 6;\ 4\ <\ 3;\ 6;\ 4;\ 1$	order
$x;..y$	" x to y " for $x \le y$
$\leftrightarrow(x;..y)\ =\ y-x$	length

String Theory

Strings are indexed sequences.

$3;\ 6;\ 4;\ 1\ <\ 3;\ 7;\ 2$	order
$3;\ 6;\ 4\ <\ 3;\ 6;\ 4;\ 1$	order
$x;..y$	"x to y" for $x{\le}y$
$\leftrightarrow(x;..y)\ =\ y{-}x$	length
$(x;..y)\ ;\ (y;..z)\ =\ x;..z$	join

String Theory

Strings are indexed sequences.

$3;\ 6;\ 4;\ 1\ <\ 3;\ 7;\ 2$	order
$3;\ 6;\ 4\ <\ 3;\ 6;\ 4;\ 1$	order
$x;..y$	" x to y " for $x{\le}y$
$\leftrightarrow(x;..y)\ =\ y{-}x$	length
$(x;..y)\ ;\ (y;..z)\ =\ x;..z$	join
"Don't say """no""". "	text

String Theory

Strings are indexed sequences.

3; 6; 4; 1 < 3; 7; 2	order
3; 6; 4 < 3; 6; 4; 1	order
x;..y	“ x to y ” for $x \le y$
$\leftrightarrow(x;..y) = y-x$	length
$(x;..y)$; $(y;..z) = x;..z$	join
“Don't say “““no”””.”	text

= “D”; “o”; “n”; “'”; “t”; “ ”; “s”; “a”; “y”; “ ”; ““““””; “n”; “o”; “””””””; “.”

String Theory

Strings are indexed sequences.

	$3;\ 6;\ 4;\ 1\ <\ 3;\ 7;\ 2$	order
	$3;\ 6;\ 4\ <\ 3;\ 6;\ 4;\ 1$	order
	$x;..y$	“ x to y ” for $x{\le}y$
	$\leftrightarrow(x;..y)\ =\ y{-}x$	length
	$(x;..y)\ ;\ (y;..z)\ =\ x;..z$	join
	“Don't say ““‘no’””.”	text
=	“D”; “o”; “n”; “'”; “t”; “ ”; “s”; “a”; “y”; “ ”; “““”; “n”; “o”; “”””; “.”	
	“abcdefghij”$_{3;..6}$ = “def”	subtext

String Theory

Strings are indexed sequences.

$3;\ 6;\ 4;\ 1\ <\ 3;\ 7;\ 2$	order
$3;\ 6;\ 4\ <\ 3;\ 6;\ 4;\ 1$	order
$x;..y$	" x to y " for $x \le y$
$\leftrightarrow(x;..y)\ =\ y-x$	length
$(x;..y)\ ;\ (y;..z)\ =\ x;..z$	join
"Don't say ""no""."	text
= "D"; "o"; "n"; "'"; "t"; " "; "s"; "a"; "y"; " "; """"; "n"; "o"; """"; "."	
"abcdefghij"$_{3;..6}$ = "def"	subtext
$nat;\ 1;\ (0,..10)$	distribution

String Theory

Strings are indexed sequences.

$3;\ 6;\ 4;\ 1\ <\ 3;\ 7;\ 2$	order
$3;\ 6;\ 4\ <\ 3;\ 6;\ 4;\ 1$	order
$x;..y$	“ x to y ” for $x{\le}y$
$\leftrightarrow(x;..y)\ =\ y–x$	length
$(x;..y)\ ;\ (y;..z)\ =\ x;..z$	join
“Don't say “““no”””.”	text
= “D”; “o”; “n”; “'”; “t”; “ ”; “s”; “a”; “y”; “ ”; “““”; “n”; “o”; “”””; “.”	
“abcdefghij”$_{3;..6}$ = “def”	subtext
$0;\ 1;\ 2:\quad nat;\ 1;\ (0,..10)$	distribution

String Theory

Strings are indexed sequences.

$3; 6; 4; 1 \ < \ 3; 7; 2$	order
$3; 6; 4 \ < \ 3; 6; 4; 1$	order
$x;..y$	" x to y " for $x{\le}y$
$\leftrightarrow(x;..y) \ = \ y{-}x$	length
$(x;..y) \ ; \ (y;..z) \ = \ x;..z$	join
"Don't say ""no""."	text

= "D"; "o"; "n"; "'"; "t"; " "; "s"; "a"; "y"; " "; """"; "n"; "o"; """"; "."

"abcdefghij"$_{3;..6}$ = "def"	subtext
$0; 1; 2\!: \ \ nat; 1; (0,..10)$	distribution
$3*(4; 5) \ = \ 4; 5; 4; 5; 4; 5$	repetition

String Theory

Strings are indexed sequences.

$3;\ 6;\ 4;\ 1\ <\ 3;\ 7;\ 2$	order
$3;\ 6;\ 4\ <\ 3;\ 6;\ 4;\ 1$	order
$x;..y$	“ x to y ” for $x{\le}y$
$\leftrightarrow(x;..y)\ =\ y\text{–}x$	length
$(x;..y)\ ;\ (y;..z)\ =\ x;..z$	join
“Don't say ““no””.”	text

= “D”; “o”; “n”; “'”; “t”; “ ”; “s”; “a”; “y”; “ ”; “““”; “n”; “o”; “”””; “.”

“abcdefghij”$_{3;..6}$ = “def”	subtext
$0;\ 1;\ 2{:}\quad nat;\ 1;\ (0,..10)$	distribution
$3*(4;\ 5)\ =\ 4;\ 5;\ 4;\ 5;\ 4;\ 5$	repetition
$*3\ =\ nil\ ,\ 3\ ,\ 3;3\ ,\ 3;3;3\ ,\ ...$	repetition

String Theory

Strings are indexed sequences.

	$3;\ 6;\ 4;\ 1\ <\ 3;\ 7;\ 2$	order
	$3;\ 6;\ 4\ <\ 3;\ 6;\ 4;\ 1$	order
	$x;..y$	“ x to y ” for $x{\le}y$
	$\leftrightarrow(x;..y)\ =\ y{-}x$	length
	$(x;..y)\ ;\ (y;..z)\ =\ x;..z$	join
	“Don't say “““no”””.”	text
=	“D”; “o”; “n”; “'”; “t”; “ ”; “s”; “a”; “y”; “ ”; “““”; “n”; “o”; “”””; “.”	
	“abcdefghij”$_{3;..6}$ = “def”	subtext
	$0;\ 1;\ 2\!:\ \ nat;\ 1;\ (0,..10)$	distribution
	$3*(4;\ 5)\ =\ 4;\ 5;\ 4;\ 5;\ 4;\ 5$	repetition
	$*3\ =\ nil\ ,\ 3\ ,\ 3;3\ ,\ 3;3;3\ ,\ ...$	repetition
	$(3;\ 5;\ 9) \triangleleft 2 \triangleright 8\ =\ 3;\ 5;\ 8$	modification

List Theory

List Theory

[0; 1; 2] a string in a package

List Theory

[0; 1; 2] a string in a package

[nat; 1; (0,..10)] distribution

List Theory

[0; 1; 2]	a string in a package
[0; 1; 2]: [*nat*; 1; (0,..10)]	distribution

List Theory

[0; 1; 2]	a string in a package
[0; 1; 2]: [*nat*; 1; (0,..10)]: [3**nat*]	distribution

List Theory

[0; 1; 2]	a string in a package
[0; 1; 2]: [*nat*; 1; (0,..10)]: [3**nat*]: [**nat*]	distribution

List Theory

[0; 1; 2]   a string in a package

[0; 1; 2]:   [*nat*; 1; (0,..10)]:   [3**nat*]:   [**nat*]   distribution

$-(a, b) = -a, -b$

negation of bunch = bunch of negations

List Theory

[0; 1; 2] a string in a package

[0; 1; 2]: [*nat*; 1; (0,..10)]: [3*nat]: [*nat] distribution

$-(a, b) = -a, -b$

negation of bunch = bunch of negations

$(a+b)\times(c+d) = a\times c + a\times d + b\times c + b\times d$

product of sums = sum of products

List Theory

$[0; 1; 2]$ a string in a package

$[0; 1; 2]$: $[nat; 1; (0,..10)]$: $[3*nat]$: $[*nat]$ distribution

$-(a, b) = -a, -b$

negation of bunch = bunch of negations

$(a+b)\times(c+d) = a\times c + a\times d + b\times c + b\times d$

product of sums = sum of products

$(a\vee b)\wedge(c\vee d) = a\wedge c \vee a\wedge d \vee b\wedge c \vee b\wedge d$

conjunction of disjunctions = disjunction of conjunctions

List Theory

$[0; 1; 2]$ a string in a package

$[0; 1; 2]$: $[nat; 1; (0,..10)]$: $[3{*}nat]$: $[{*}nat]$ distribution

$-(a, b) = -a, -b$

negation of bunch = bunch of negations

$(a+b)\times(c+d) = a\times c + a\times d + b\times c + b\times d$

product of sums = sum of products

$(a\vee b)\wedge(c\vee d) = a\wedge c \vee a\wedge d \vee b\wedge c \vee b\wedge d$

conjunction of disjunctions = disjunction of conjunctions

$[(2, 3); (4, 5)] = [2; 4], [2; 5], [3; 4], [3; 5]$

list of bunches = bunch of lists

List Theory

[0; 1; 2]	a string in a package
[0; 1; 2]: [*nat*; 1; (0,..10)]: [3**nat*]: [**nat*]	distribution

List Theory

[0; 1; 2]	a string in a package
[0; 1; 2]: [*nat*; 1; (0,..10)]: [3**nat*]: [**nat*]	distribution
~[3; [5; 7]; 4] = 3; [5; 7]; 4	content

List Theory

[0; 1; 2]	a string in a package
[0; 1; 2]: [*nat*; 1; (0,..10)]: [3**nat*]: [**nat*]	distribution
~[3; [5; 7]; 4] = 3; [5; 7]; 4	content
#[3; [5; 7]; 4] = 3	length

List Theory

$[0; 1; 2]$	a string in a package
$[0; 1; 2]$: $[nat; 1; (0,..10)]$: $[3*nat]$: $[*nat]$	distribution
$\sim[3; [5; 7]; 4] = 3; [5; 7]; 4$	content
$\#[3; [5; 7]; 4] = 3$	length
$[3; [5; 7]; 4]\ 2 = 4$	index

List Theory

[0; 1; 2]	a string in a package
[0; 1; 2]: [*nat*; 1; (0,..10)]: [3**nat*]: [**nat*]	distribution
~[3; [5; 7]; 4] = 3; [5; 7]; 4	content
#[3; [5; 7]; 4] = 3	length
[3; [5; 7]; 4] 2 = 4	index
□[3; [5; 7]; 4] = 0,..3	domain

List Theory

[0; 1; 2]	a string in a package
[0; 1; 2]: [*nat*; 1; (0,..10)]: [3**nat*]: [**nat*]	distribution
~[3; [5; 7]; 4] = 3; [5; 7]; 4	content
#[3; [5; 7]; 4] = 3	length
[3; [5; 7]; 4] 2 = 4	index
□[3; [5; 7]; 4] = 0,..3	domain
[3; 5; 7; 4] [2; 1; 2] = [7; 5; 7]	composition

List Theory

[0; 1; 2]	a string in a package
[0; 1; 2]: [*nat*; 1; (0,..10)]: [3**nat*]: [**nat*]	distribution
~[3; [5; 7]; 4] = 3; [5; 7]; 4	content
#[3; [5; 7]; 4] = 3	length
[3; [5; 7]; 4] 2 = 4	index
□[3; [5; 7]; 4] = 0,..3	domain
[3; 5; 7; 4] [2; 1; 2] = [7; 5; 7]	composition
[3; 5; 7; 4];;[2; 1; 2] = [3; 5; 7; 4; 2; 1; 2]	join

List Theory

[0; 1; 2]	a string in a package
[0; 1; 2]: [*nat*; 1; (0,..10)]: [3**nat*]: [**nat*]	distribution
~[3; [5; 7]; 4] = 3; [5; 7]; 4	content
#[3; [5; 7]; 4] = 3	length
[3; [5; 7]; 4] 2 = 4	index
□[3; [5; 7]; 4] = 0,..3	domain
[3; 5; 7; 4] [2; 1; 2] = [7; 5; 7]	composition
[3; 5; 7; 4];;[2; 1; 2] = [3; 5; 7; 4; 2; 1; 2]	join
[3; 6; 4; 1] < [3; 7; 2]	order
[3; 6; 4] < [3; 6; 4; 1]	order

List Theory

[0; 1; 2]	a string in a package
[0; 1; 2]: [*nat*; 1; (0,..10)]: [3**nat*]: [**nat*]	distribution
~[3; [5; 7]; 4] = 3; [5; 7]; 4	content
#[3; [5; 7]; 4] = 3	length
[3; [5; 7]; 4] 2 = 4	index
□[3; [5; 7]; 4] = 0,..3	domain
[3; 5; 7; 4] [2; 1; 2] = [7; 5; 7]	composition
[3; 5; 7; 4];;[2; 1; 2] = [3; 5; 7; 4; 2; 1; 2]	join
[3; 6; 4; 1] < [3; 7; 2]	order
[3; 6; 4] < [3; 6; 4; 1]	order
2→22 | [10;..15] = [10; 11; 22; 13; 14]	modification

List Theory

modification

Let $L = [10;..15]$

$$2 \rightarrow L\, 3 \mid 3 \rightarrow L\, 2 \mid L \quad =$$

List Theory

modification

Let L = [10;..15] = [10; 11; 12; 13; 14]

$$2 \rightarrow L\,3 \mid 3 \rightarrow L\,2 \mid L \quad =$$

List Theory

modification

Let L = $[10;..15]$ = $[10;\ 11;\ 12;\ 13;\ 14]$

$$2 \to L\,3 \mid 3 \to L\,2 \mid \underline{L} \ =$$

List Theory

modification

Let L = [10;..15] = [10; 11; 12; 13; 14]

$$2 \rightarrow L\,3 \mid 3 \rightarrow L\,2 \mid L \quad = \qquad [10;\ 11;\ 12;\ 13;\ 14]$$

List Theory

modification

Let $L = [10;..15] = [10;\ 11;\ 12;\ 13;\ 14]$

$$2 \rightarrow L\,3 \mid \underline{3 \rightarrow L\,2} \mid L \quad = \quad [10;\ 11;\ 12;\ 13;\ 14]$$

List Theory

modification

Let L = [10;..15] = [10; 11; 12; 13; 14]

$$2 \rightarrow L\,3 \mid 3 \rightarrow L\,2 \mid L \quad = \quad [10;\ 11;\ 12;\ 12;\ 14]$$

List Theory

modification

Let L = [10;..15] = [10; 11; 12; 13; 14]

$$\underline{2 \rightarrow L\, 3} \mid 3 \rightarrow L\, 2 \mid L \quad = \quad [10;\ 11;\ 12;\ 12;\ 14]$$

List Theory

modification

Let L = [10;..15] = [10; 11; 12; 13; 14]

$$2 \rightarrow L\,3 \mid 3 \rightarrow L\,2 \mid L \quad = \quad [10;\ 11;\ 12;\ 12;\ 14] \qquad ✗$$

List Theory

modification

Let L = [10;..15] = [10; 11; 12; 13; 14]

$$2 \to L\,3 \mid 3 \to L\,2 \mid L \quad =$$

List Theory

modification

Let L = [10;..15] = [10; 11; 12; 13; 14]

$$2 \rightarrow L\,3 \mid 3 \rightarrow L\,2 \mid \underline{L} \;\; =$$

List Theory

modification

Let $L = [10;..15] = [10; 11; 12; 13; 14]$

$$2 \rightarrow L\,3 \mid 3 \rightarrow L\,2 \mid L \quad = \quad [10; 11; 12; 13; 14]$$

List Theory

modification

Let L = [10;..15] = [10; 11; 12; 13; 14]

$$2 \rightarrow L\, 3 \mid \underline{3 \rightarrow L\, 2} \mid L \quad = \qquad [10;\ 11;\ 12;\ 13;\ 14]$$

List Theory

modification

Let $L = [10;..15] = [10;\ 11;\ 12;\ 13;\ 14]$

$$2 \rightarrow L\, 3 \mid 3 \rightarrow L\, 2 \mid L \quad = \quad [10;\ 11;\ 12;\ 12;\ 14]$$

List Theory

modification

Let L = [10;..15] = [10; 11; 12; 13; 14]

$$\underline{2 \rightarrow L\,3} \mid 3 \rightarrow L\,2 \mid L \quad = \qquad [10;\ 11;\ 12;\ 12;\ 14]$$

List Theory

modification

Let $L = [10;..15] = [10;\ 11;\ 12;\ 13;\ 14]$

$$2 \rightarrow L\ 3 \mid 3 \rightarrow L\ 2 \mid L \quad = \quad [10;\ 11;\ 13;\ 12;\ 14] \quad ✔$$

String and List Theory

$S_{n,m} = S_n, S_m$

$L\,(n, m) = L\,n, L\,m$

String and List Theory

$$S_{n,\,m} \;=\; S_n, S_m$$

$$L\,(n, m) \;=\; L\,n, L\,m$$

$$S_{\{n,\,m\}} \;=\; \{S_n, S_m\}$$

$$L\,\{n, m\} \;=\; \{L\,n, L\,m\}$$

$$S_{n;\,m} \;=\; S_n; S_m$$

$$L\,(n; m) \;=\; L\,n; L\,m$$

$$S_{[n;\,m]} \;=\; [S_n; S_m]$$

$$L\,[n; m] \;=\; [L\,n; L\,m]$$

String and List Theory

$$S_{n,\,m} \;=\; S_n, S_m \qquad\qquad L\,(n, m) \;=\; L\,n, L\,m$$

$$S_{\{n,\,m\}} \;=\; \{S_n, S_m\} \qquad\qquad L\,\{n, m\} \;=\; \{L\,n, L\,m\}$$

$$S_{n;\,m} \;=\; S_n; S_m \qquad\qquad L\,(n; m) \;=\; L\,n; L\,m$$

$$S_{[n;\,m]} \;=\; [S_n; S_m] \qquad\qquad L\,[n; m] \;=\; [L\,n; L\,m]$$

$$S_{0,\,\{1,\,[2;\,1];\,0\}}$$
$$=\; S_0, \{S_1, [S_2; S_1]; S_0\}$$

$$L\,(0, \{1, [2; 1]; 0\})$$
$$=\; L\,0, \{L\,1, [L\,2; L\,1]; L\,0\}$$

List Theory

multidimensional structures

$$
\begin{aligned}
A \;=\; [\quad & [6;\ 3;\ 7;\ 0]\ ; \\
& [4;\ 9;\ 2;\ 5]\ ; \\
& [1;\ 5;\ 8;\ 3]\ ]
\end{aligned}
$$

List Theory

multidimensional structures

$$
\begin{aligned}
A \;=\; [\quad & [6;\, 3;\, 7;\, 0]\ ; \\
& [4;\, 9;\, 2;\, 5]\ ; \\
& [1;\, 5;\, 8;\, 3]\ ]
\end{aligned}
$$

A: [3*[4*nat]]

List Theory

multidimensional structures

$$A = [\quad [6; 3; 7; 0] ;$$
$$[4; 9; 2; 5] ;$$
$$[1; 5; 8; 3] \,]$$

A: [3*[4*nat]]

$A\ 1 = [4; 9; 2; 5]$

List Theory

multidimensional structures

$$A = [\ [6;\ 3;\ 7;\ 0]\ ;$$
$$[4;\ 9;\ 2;\ 5]\ ;$$
$$[1;\ 5;\ 8;\ 3]\ ]$$

A: $[3*[4*nat]]$

$A\ 1\ =\ [4;\ 9;\ 2;\ 5]$

$A\ 1\ 2\ =\ 2$

List Theory

multidimensional structures

$$
\begin{aligned}
A \ = \ [\ \ & [6;\ 3;\ 7;\ 0]\ ; \\
& [4;\ 9;\ 2;\ 5]\ ; \\
& [1;\ 5;\ 8;\ 3]\ ]
\end{aligned}
$$

A: $[3*[4*nat]]$

$A\ 1 \ = \ [4;\ 9;\ 2;\ 5]$

$A\ 1\ 2 \ = \ 2$

$A\ (1\,,\,2)$

$A\ [1\,,\,2]$

List Theory

multidimensional structures

$$A \;=\; [\quad [6;\, 3;\, 7;\, 0]\ ;$$
$$[4;\, 9;\, 2;\, 5]\ ;$$
$$[1;\, 5;\, 8;\, 3]\ ]$$

A: $[3{*}[4{*}nat]]$

$A\ 1 \;=\; [4;\, 9;\, 2;\, 5]$

$A\ 1\ 2 \;=\; 2$

$A\ (1\,,2) \;=\; A\ 1\,, A\ 2$

$A\ [1\,,2]$

List Theory

multidimensional structures

$$A = [\ [6; 3; 7; 0] ;$$
$$[4; 9; 2; 5] ;$$
$$[1; 5; 8; 3]\ ]$$

A: $[3*[4*nat]]$

$A\,1 = [4; 9; 2; 5]$

$A\,1\,2 = 2$

$A\,(1, 2) = A\,1, A\,2 = [4; 9; 2; 5], [1; 5; 8; 3]$

$A\,[1, 2]$

List Theory

multidimensional structures

$$A \;=\; [\quad [6;\, 3;\, 7;\, 0]\;;$$
$$[4;\, 9;\, 2;\, 5]\;;$$
$$[1;\, 5;\, 8;\, 3]\;]$$

A: [3*[4*nat]]

$A\ 1 \;=\; [4;\, 9;\, 2;\, 5]$

$A\ 1\ 2 \;=\; 2$

$A\ (1\,,2) \;=\; A\ 1\,, A\ 2 \;=\; [4;\, 9;\, 2;\, 5], [1;\, 5;\, 8;\, 3]$

$A\ [1\,,2] \;=\; [A\ 1\,, A\ 2]$

List Theory

multidimensional structures

$$
\begin{aligned}
A \;=\; [\;\; & [6;\,3;\,7;\,0]\;; \\
& [4;\,9;\,2;\,5]\;; \\
& [1;\,5;\,8;\,3]\;]
\end{aligned}
$$

A: [3*[4*nat]]

$A\;1 \;=\; [4;\,9;\,2;\,5]$

$A\;1\;2 \;=\; 2$

$A\,(1\,,2) \;=\; A\;1\,, A\;2 \;=\; [4;\,9;\,2;\,5],\,[1;\,5;\,8;\,3]$

$A\,[1\,,2] \;=\; [A\;1\,, A\;2] \;=\; [\;[4;\,9;\,2;\,5],\,[1;\,5;\,8;\,3]\;]$

List Theory

multidimensional structures

$$A \;=\; [\;\; [6;\, 3;\, 7;\, 0]\;;$$
$$[4;\, 9;\, 2;\, 5]\;;$$
$$[1;\, 5;\, 8;\, 3]\;]$$

A: [3*[4*nat]]

$A\;1 \;=\; [4;\, 9;\, 2;\, 5]$

$A\;1\;2 \;=\; 2 \quad \longleftarrow$

$A\,(1\,,2) \;=\; A\;1\,, A\;2 \;=\; [4;\, 9;\, 2;\, 5], [1;\, 5;\, 8;\, 3]$

$A\,[1\,,2] \;=\; [A\;1\,, A\;2] \;=\; [\;[4;\, 9;\, 2;\, 5], [1;\, 5;\, 8;\, 3]\;]$