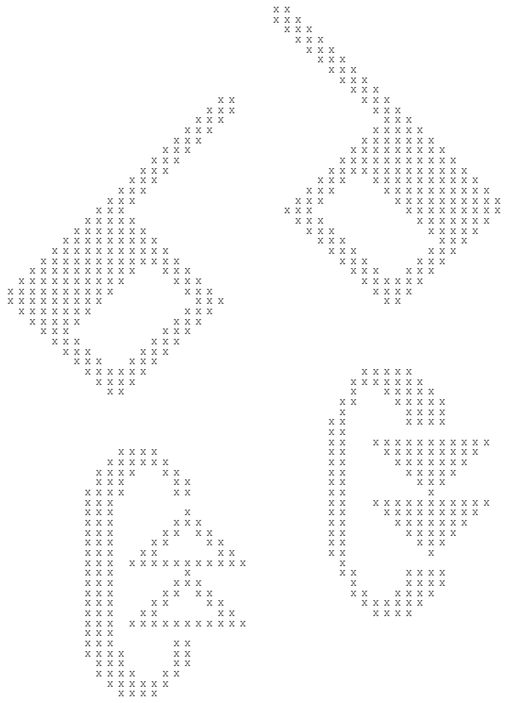
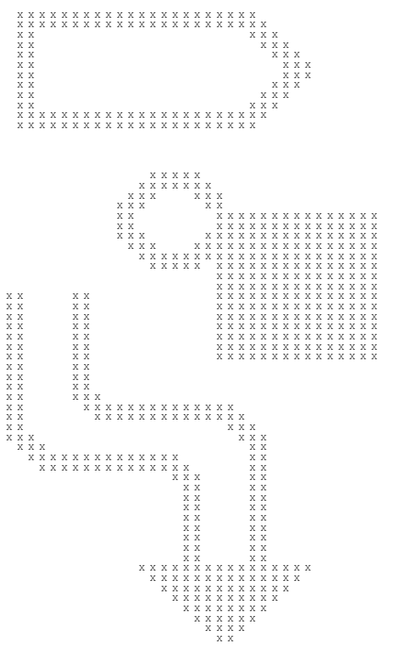
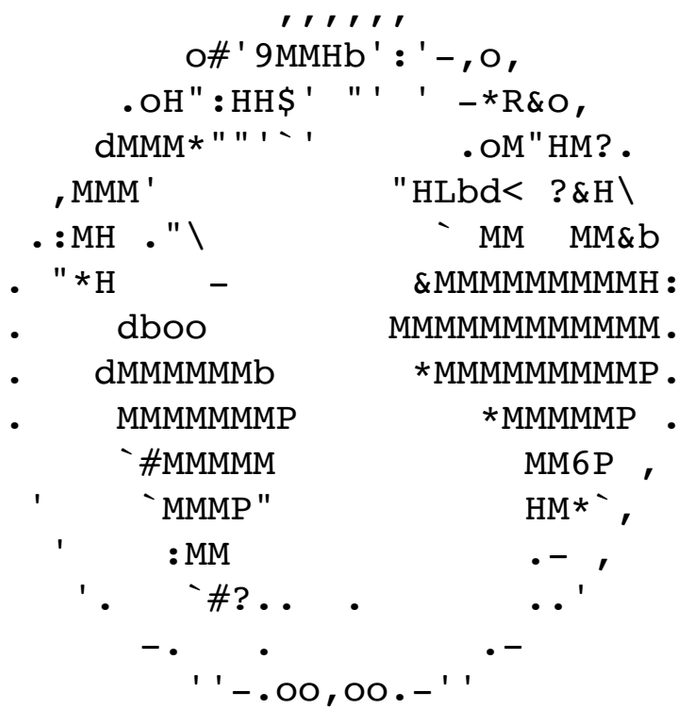




SignWriting in an ASCII World!



by Stephen E Slevinski Jr

It's an ASCII World!

A. Starting Point

   1. ASCII
   2. Regular Expressions
   3. Token Patterns
   4. Hexadecimal

B. Definition

   1. Symbol Keys
   2. Numbers
   3. Formal SignWriting
   4. Query Strings

Starting Point

American Standard Code for Information Interchange.

ASCII is the basis for text processing and standard definition.

Unicode is ASCII plus additional characters.

ASCII is UTF-8, character for character.

ASCII will outlive Unicode.

ASCII

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	(space)
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29	)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,
010 1101	055	45	2D	-
010 1110	056	46	2E	.
010 1111	057	47	2F	/
011 0000	060	48	30	0
011 0001	061	49	31	1
011 0010	062	50	32	2
011 0011	063	51	33	3
011 0100	064	52	34	4
011 0101	065	53	35	5
011 0110	066	54	36	6
011 0111	067	55	37	7
011 1000	070	56	38	8
011 1001	071	57	39	9
011 1010	072	58	3A	:
011 1011	073	59	3B	;
011 1100	074	60	3C	<
011 1101	075	61	3D	=
011 1110	076	62	3E	>
011 1111	077	63	3F	?

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z
101 1011	133	91	5B	[
101 1100	134	92	5C	\
101 1101	135	93	5D	]
101 1110	136	94	5E	^
101 1111	137	95	5F	_

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l
110 1101	155	109	6D	m
110 1110	156	110	6E	n
110 1111	157	111	6F	o
111 0000	160	112	70	p
111 0001	161	113	71	q
111 0010	162	114	72	r
111 0011	163	115	73	s
111 0100	164	116	74	t
111 0101	165	117	75	u
111 0110	166	118	76	v
111 0111	167	119	77	w
111 1000	170	120	78	x
111 1001	171	121	79	y
111 1010	172	122	7A	z
111 1011	173	123	7B	{
111 1100	174	124	7C	|
111 1101	175	125	7D	}
111 1110	176	126	7E	~

https://en.wikipedia.org/wiki/ASCII/

Starting Point

Regex

A regular expression is used to examine text and identify strings that match a stated pattern.

It is faster to recognize a string with regular expressions then to analyze a string with a routine.

https://xkcd.com/208/

Starting Point Regex

Character	Description	Example
*	Match a character 0 or more times	**ABC*** matches *AB, ABC, ABCC,* ...
+	Match a character 1 or more times	**ABC+** matches *ABC, ABCC, ABCCC,* …
?	Match a character 0 or 1 times	**ABC?** matches *AB or ABC*
{#}	Match a character explicit number times	**AB{2}** matches *ABB*

Test Regex Online https://regex101.com/

Starting Point Regex

Character	Description	Example
[ ]	Match any character from a list	**[ABC]** matches *A, B, or C*
[-]	Match any character in a range	**[A-C]** matches *A, B, or C*
( )	Create a group for matching	**A(BC)+** matches *ABC, ABCBC, ABCBCBC, …*
(|)	Match one of several alternatives	**(ABIBCICD)** matches *AB, BC, or CD*

Test Regex Online https://regex101.com/

Starting Point

Find a number between 122 and 455

Step 1) 10's don't match and the min 1's are not zero (last number to 9)

Step 2) Bring up the 10's if hundreds are different

Step 3) Bring up the 100's if different

Step 4) Bring up the 10's

Step 5) Bring up the 1's

Regex

Regex	Matches
12[2-9]	122 to 129
1[3-9][0-9]	130 to 199
[2-3][0-9][0-9]	200 to 399
4[0-4][0-9]	400 to 449
45[0-5]	450 to 455

Final Regex (12[1-9]|1[3-9][0-9]|[2-3][0-9][0-9]|4[0-4][0-9]|45[0-5])

Test Regex Online https://regex101.com/

Starting Point

Token Patterns

A B L M R → Structural Markers

W S P → Symbol Bases

i o → Symbol Modifiers

n → Numbers

Starting Point

Token Patterns

A	Sequence Marker
B	SignBox Marker
L	Left Lane Marker
M	Middle Lane Marker
R	Right Lane Marker
w	Writing Base Symbol
s	Sequence Base Symbol
P	Punctuation Base Symbol
i	Fill Modifier
o	Rotation Modifier
n	Number

Writing Symbol

W	i	O

Cartesian Coordinate

n	n

Starting Point

Hexadecimal

Character	Value
x0	0
x1	1
x2	2
x3	3
x4	4
x5	5
x6	6
x7	7
x8	8
x9	9
xA	10
xB	11
xC	12
xD	13
xE	14
xF	15

Decimal Value

Standard numbers that we use are base 10, using only the 0 thru 9.

Hexadecimal

Prefixed with an ‘x’, hexadecimal characters and strings are base 16, using 0 thru 9 and A thru F.

String	Value
x00	0
x10	16
x20	32
x30	48
x40	64
x50	80
x60	96
x70	112
x80	128
x90	144
xA0	160
xB0	176
xC0	192
xD0	208
xE0	244
xFF	255

Definition

Symbol keys are used to identify each of the 37,811 symbols of the ISWA 2010.

Symbols keys are built using three tokens.

1. Symbol Base of “w”, “s”, or “P”
2. Fill Modifier of “i”
3. Rotation Modifier of “o”

Symbol Key

Writing symbol

w	i	o

Sequence symbol

s	i	o

Punctuation symbol

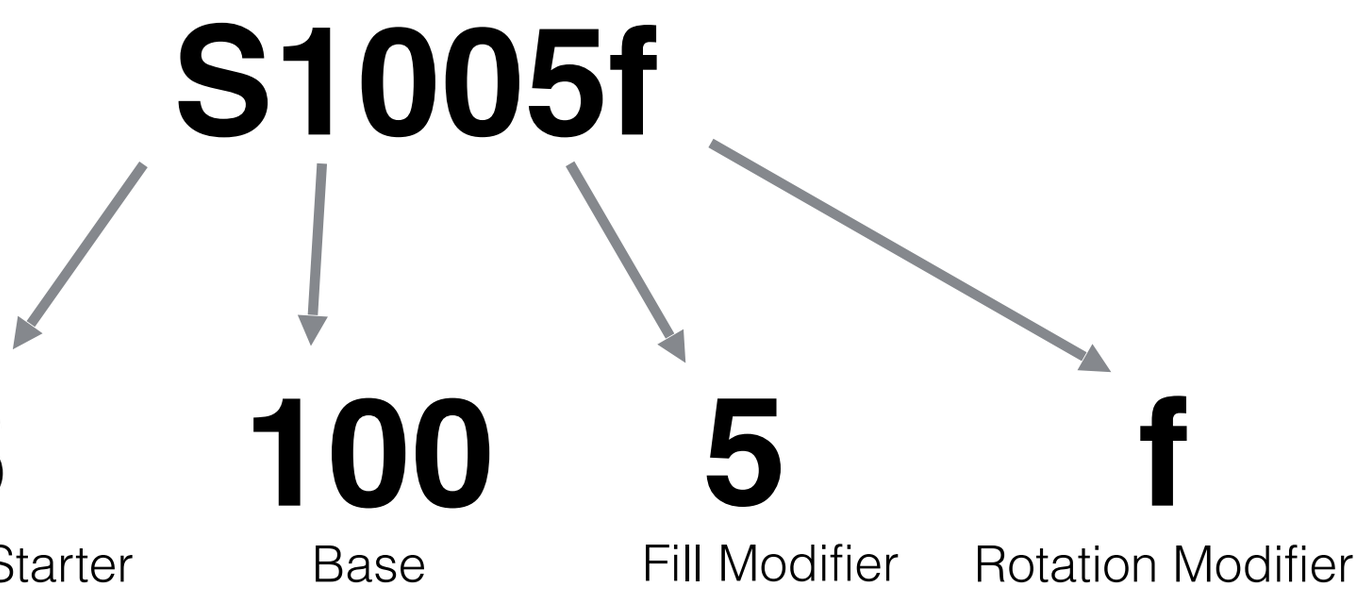
P	i	o

https://tools.ietf.org/html/draft-slevinski-signwriting-text-05#section-2.1

Definition

Symbol Key

Symbol keys are 6 characters long.



Regex

S[123][0-9a-f]{2}[0-5][0-9a-f]

http://signbank.org/SignWriting_Character_Viewer.html#?set=key

Definition

The number characters encode the ruler principle for 2-dimensions.

Numbers range from 250 to 749, with 500 being the center.

Cartesian Coordinates are built using two tokens.

1. Number of “n”
2. Number of “n”

Numbers

Cartesian Coordinate

250x749

250 **749**

X Coordinate Y Coordinate

(250,749)

Top-Left Coordinate of symbol

https://tools.ietf.org/html/draft-slevinski-signwriting-text-05#section-2.2

Definition

Numbers

Description	Example	Regex
Numbers are 3 characters long.	**500**	[0-9]{3}
Coordinates are 7 characters long.	**500x500** X Coordinate Y Coordinate	[0-9]{3}x[0-9]{3}

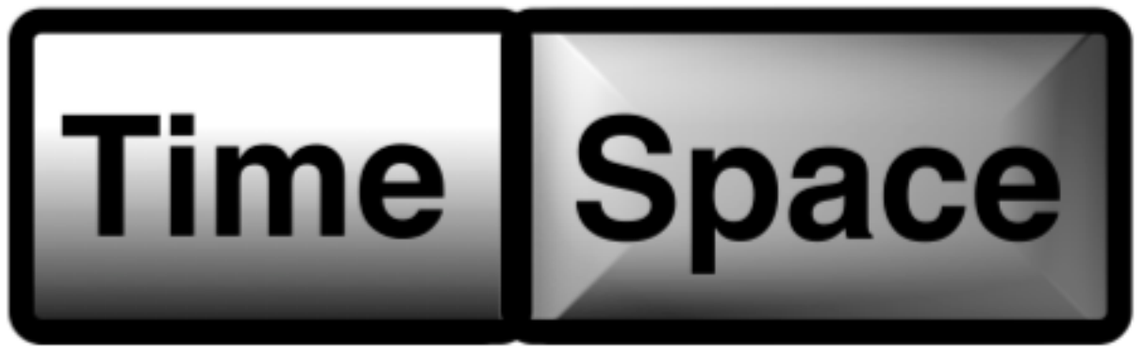
Regex for explicit number between 250 and 749

(2[5-9][0-9]|[3-6][0-9]{2}|7[0-4][0-9])

Definition Formal SignWriting

According to Wikipedia, "In mathematics, computer science, and linguistics, a formal language is a set of strings of symbols that may be constrained by rules that are specific to it."

Sign as Word



- Mathematical ASCII name
- Optional time for sorting
- Mandatory space for visual

https://tools.ietf.org/html/draft-slevinski-signwriting-text-05#section-2

Definition Formal SignWriting

Spatial symbol

Identifies a specific symbol with coordinate information.

Used in the
Spatial Signbox

w	i	o	n	n

Spatial symbols are built using five tokens.

1. Symbol Base of “w”
2. Fill Modifier of “i”
3. Rotation Modifier of “o”
4. Number of “n”
5. Number of “n”

Regex

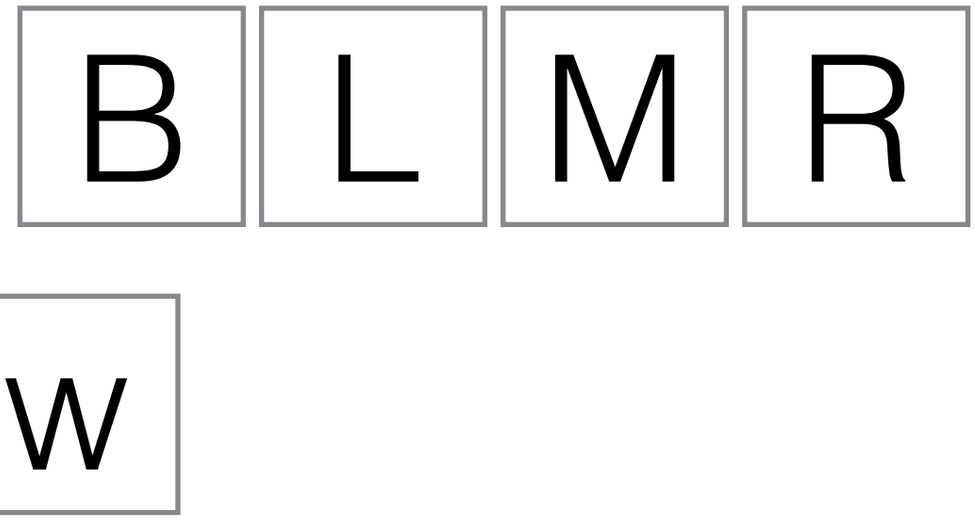
S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3}

https://tools.ietf.org/html/draft-slevinski-signwriting-text-05#section-2.3

Definition Formal SignWriting

Spatial SignBox

A cluster of symbols
used in 2-dimensions



Regex for Token Description

[BLMR]nn(wionn)*

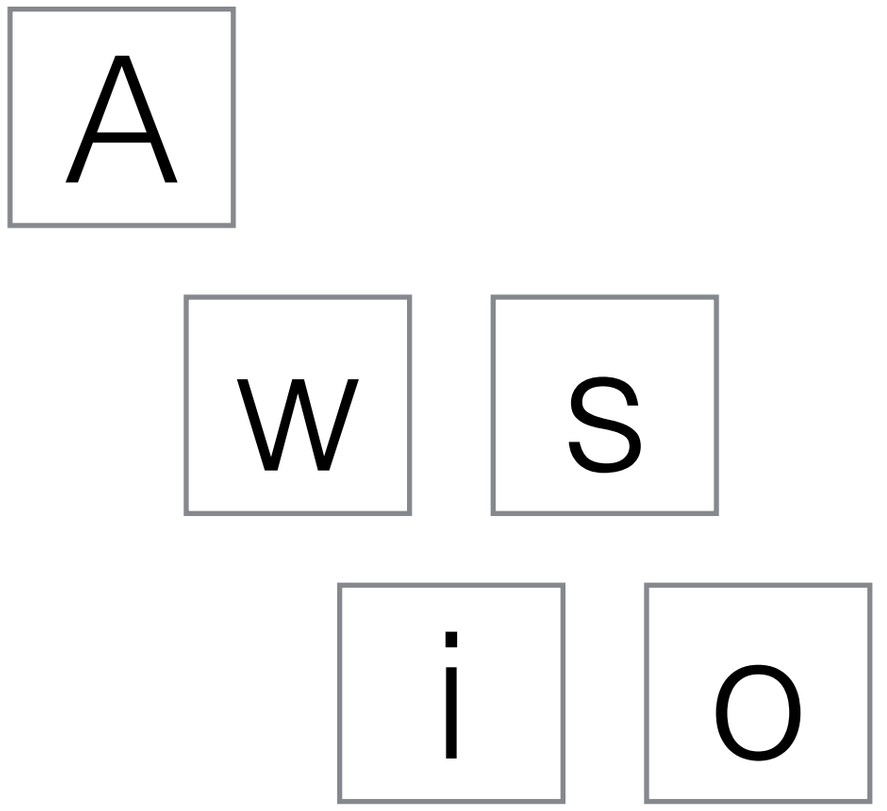
Regex for Formal SignWriting

[BLMR]([0-9]{3}x[0-9]{3})(S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*

Definition Formal SignWriting

Temporal Prefix

An optional ordered list of symbols used for sorting.



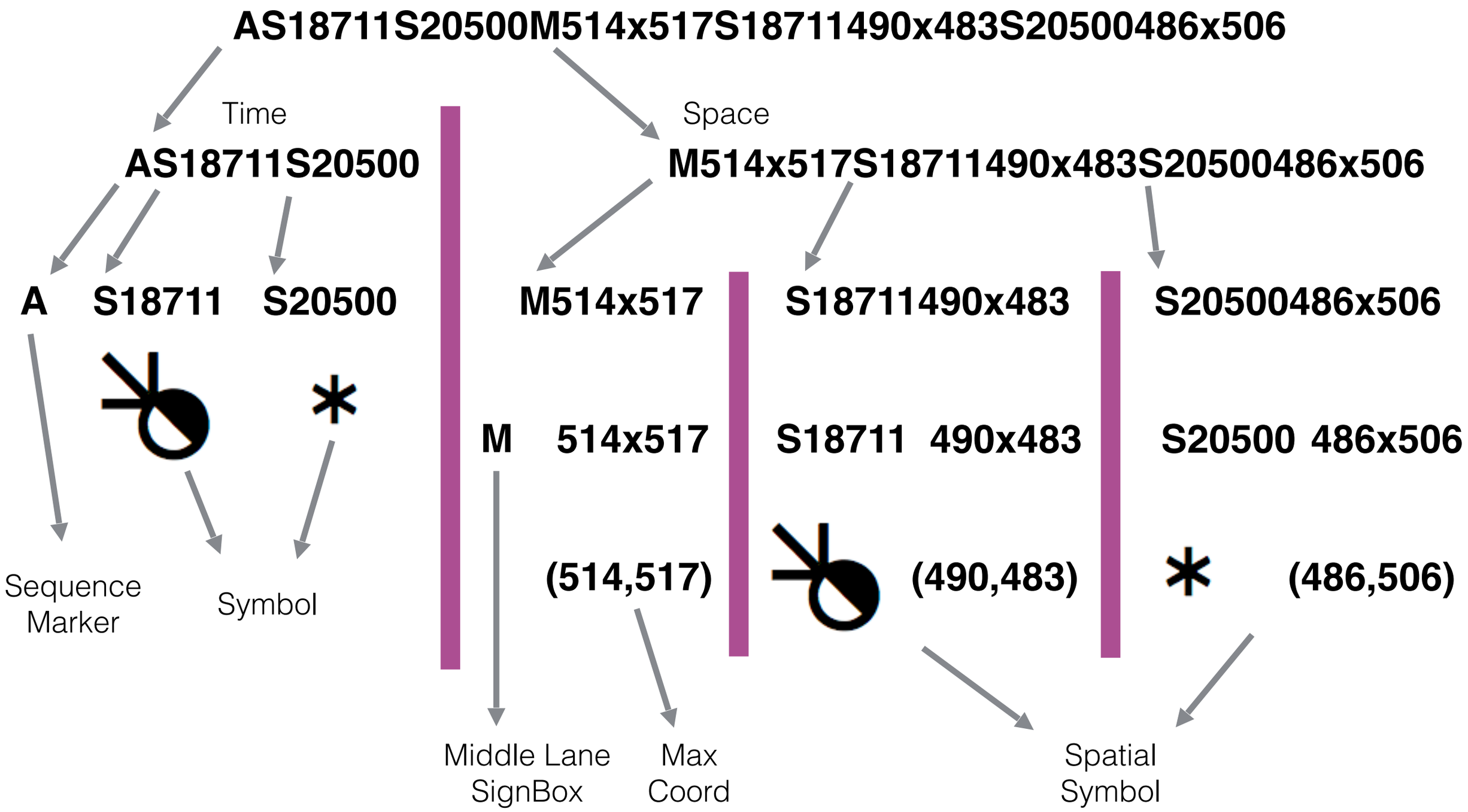
Regex for Token Description

(A([ws]io)+)?

Regex for Formal SignWriting

(A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?

Definition Formal SignWriting



Definition

Query String

The query string is a lite ASCII markup similar to Formal SignWriting.

Query strings are used to search Formal SignWriting.

Formal SignWriting can be converted into several different query string, depending on the search parameters.

Query String for all signs

Q

Query String for sortable signs

QT

https://tools.ietf.org/html/draft-slevinski-signwriting-text-05#section-3

Definition

Query String

Query Strings

Q

(A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?
[BLMR]([0-9]{3}x[0-9]{3})(S[123][0-9a-f]
{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*

15 to 50 times expansion

Regular Expressions

process million of characters per second

Formal SignWriting

search results

Definition Query String

The query string is a concise representation for a much larger and detailed set of regular expressions.

When a query string returns more than one regular expression, a filter and repeat step is required.

QS10000S20500

(A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?[BLMR]([0-9]{3}x[0-9]{3})(S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})***S10000**[0-9]{3}x[0-9]{3}(S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*

(A(S[123][0-9a-f]{2}[0-5][0-9a-f])+)?[BLMR]([0-9]{3}x[0-9]{3})(S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})***S20500**[0-9]{3}x[0-9]{3}(S[123][0-9a-f]{2}[0-5][0-9a-f][0-9]{3}x[0-9]{3})*

Definition Query String

Two main sections of a query string

Temporal Sequence Searching	**Spatial SignBox Searching**
1-Dimensional order	2-Dimensional order

Both sections use the same definition for a symbol or a range.

Search Symbol	S[123][0-9a-f]{2}[0-5u][0-9a-fu]
Search Range	R[123][0-9a-f]{2}t[123][0-9a-f]{2}

Definition Query String

Temporal Sequence Searching

It is possible to specify the start order of the temporal sequence by identifying a series of symbols and/or ranges.

Q((A(S[123][0-9a-f]{2}[0-5u][0-9a-fu]|R[123][0-9a-f]{2}t[123][0-9a-f]{2})+)?T)?

Symbol Search

Fill and Rotation values of ‘u’ represent unknown and allow all possible values.

W	İ	O	S	İ	O

Range Search

Finds all symbols between two specified symbol bases

W	W	W	S	S	S

Definition

Query String

Spatial SignBox Searching

Allowable distance from specified coordinates

It is possible to specify one or more symbols (or ranges of symbols) that must be included in the spatial SignBox with optional coordinates for each symbol or range.

Optional Variance

Q(S[123][0-9a-f]{2}[0-5u][0-9a-fu]([0-9]{3}x[0-9]{3})?IR[123][0-9a-f]{2}t[123][0-9a-f]{2}([0-9]{3}x[0-9]{3})?)*(V[0-9]+)?

Symbol Search

Fill and Rotation values of ‘u’ represent unknown and allow all possible values.

w	i	o

n	n

Optional Coordinates

Specified coordinates will limit the search results for the previous symbol.

Range Search

Finds all symbols between two specified symbol bases

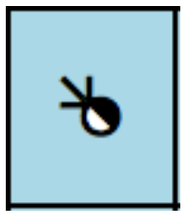
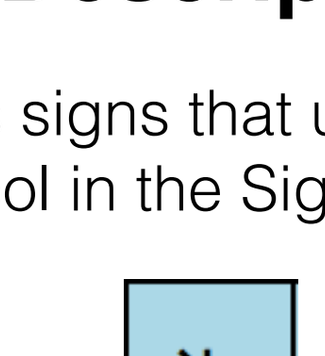
w	w

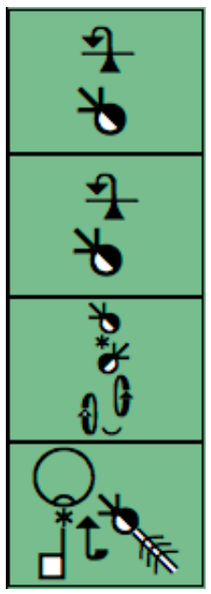
n	n

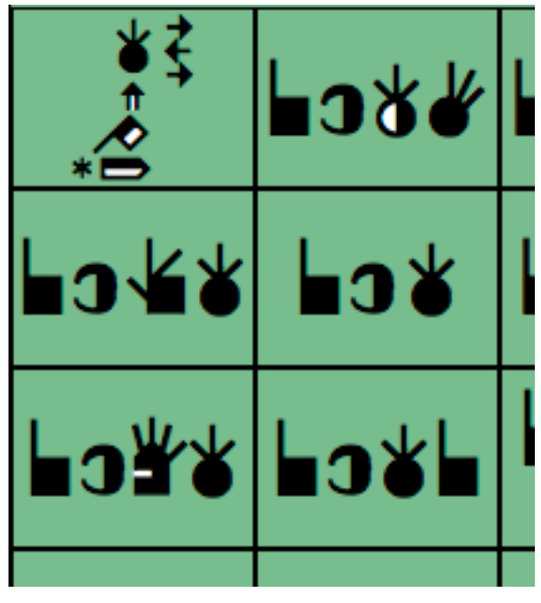
Optional Coordinates

Specified coordinates will limit the search results for the previous range.

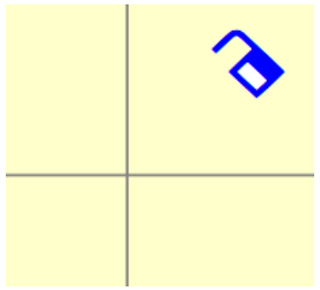
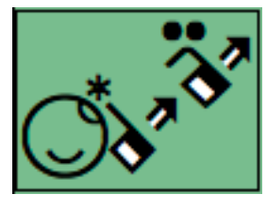
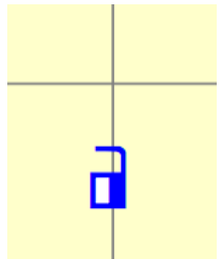
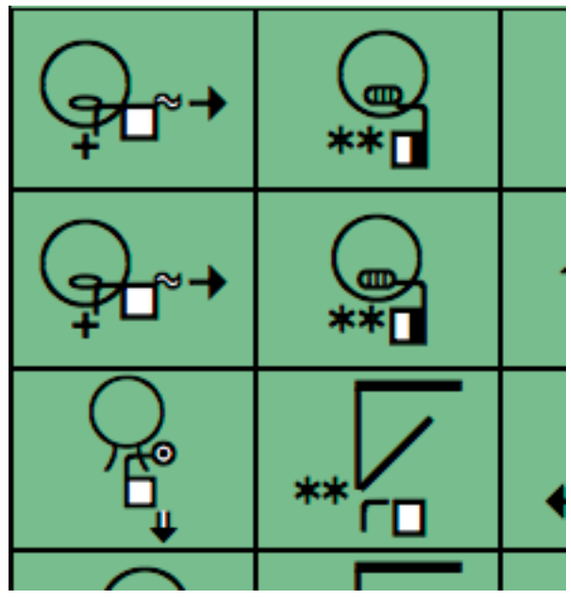
Examples Query String

Query	Description	Results
QS18711	Finds signs that use an exact symbol in the SignBox	
QS187uu	Finds signs that use a general base symbol in the SignBox with any fill or rotation.	



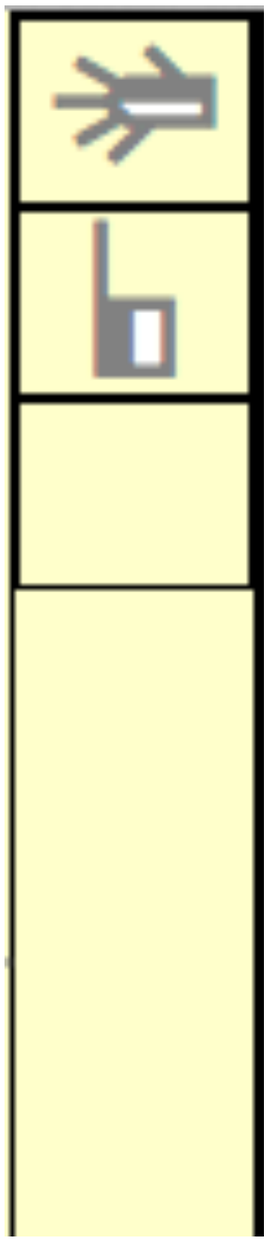


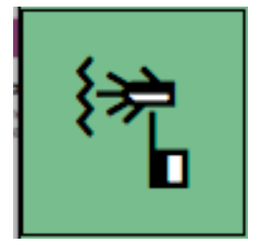
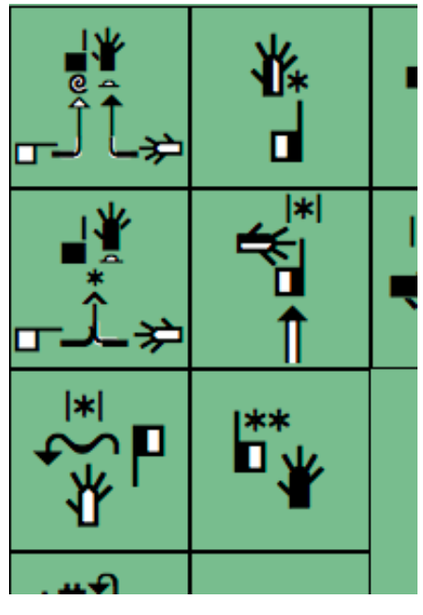
Examples Query String

Query	Description	Results
QS10a11532x445	Finds signs that use an exact symbol near a specific coordinate in the sign box 	
QS10auu491x526	Finds signs that use a general base symbol near a specific coordinate in the SignBox with any fill or rotation. 	

Examples

Query String



Query	Description	Results
QAS14c12S10018T	Finds signs with a temporal sequence that starts with specified symbols	
QAS14cuuS100uuT	Finds signs with a temporal sequence that starts with the ordered base symbols and any fills or rotations	

Examples Query String

For mixed searching, the Temporal Sequence searching is defined first.

QAS100uuS100uuTS20500470x470

Temporal Sequence Searching

Temporal Sequence starts with any two index hand shapes

Spatial SignBox Searching

Contact star is used near coordinate (470,470)