Semi Tally Numerical System

by Sven Nilsen, 2020

In this paper I present a numeral system that is semi tally with canonization symbols `01248g`.

Tally marks is a unary numeral system that groups numbers into five:

I II III IIII ~~IIII~~

The benefit of tally marks is that the order does not matter, making them ideal for counting on write-only mediums. However, suppose that one is not just counting one by one, but also building on communicated numbers. Also, the medium could be read-write for visual addition or subtraction.

It makes sense to reuse existing symbols from the decimal system:

n	**ceil(n/2)floor(n/2)**
1	10
2	11
3	21
4	22
5	32
6	33
7	43
8	44
9	54

The numbers 1-9 are compatible with the decimal system.

Each number has a canonical split form which divides the sum as close as possible into two equal parts, with the greater value being ordered first.

Taking inspiration from the hexadecimal system, one can generalize to include the entire alphabeth:

a	55	= 10	n	cb	= 23
b	65	= 11	o	cc	= 24
c	66	= 12	p	dc	= 25
d	76	= 13	q	dd	= 26
e	77	= 14	r	ed	= 27
f	87	= 15	s	ee	= 28
g	88	= 16	t	fe	= 29
h	98	= 17	u	ff	= 30
i	99	= 18	v	gf	= 31
j	a9	= 19	w	gg	= 32
k	aa	= 20	x	hg	= 33
l	ba	= 21	y	hh	= 34
m	bb	= 22	z	ih	= 35

One property of this numeral system, is that there is an easy algorithm for canonizing using 6 symbols:

1 2 4 8 g w

To canonize a number, one splits repeatedly into the 6 symbols:

sven
421421421421g842142142121212121212121212

Order the symbols from high to low:

g84444444222222222222222211111111111111

Merge to get the canonical form:

www

Therefore:

sven = www

The symbol `w` can be converted into `1` with space as separator, using `0` if there are none:

www = 111 0 = 21 0

Since any word or sentence can be converted into the canonical form, one can use this as a simple hash. One can also easily convert back and forth from split tally marks to a binary hash. This gives a grouping of 5 symbols padded with zeroes, which nicely continues the 5 alignment from tally numbers. As a consequence, this numerical system also has a natural associated one-hand sign language.

Some examples:

Word	**Canonical Semi Tally Marks**	**Binary Format**
merge	2 g841	00010 11101
independence	42 842	01001 01110
articulate	42 8	00110 01000
tradition	41 g8421	00101 11111
sentence	4 g841	00100 11101
million	4 g21	00100 10011
future	4 g1	00100 10000
incredible	41 821	00101 01011
minute	4 8	00100 01000
hour	21 2	00011 00011

Since canonicalization is strongly normalizing, one can improve effiency by memorizing the canonical form of letters, words and small sentences. Common expressions can improve communication.

Here is the English alphabeth, which is easy to construct by starting at `a = 82`:

a	82
b	821
c	84
d	841
e	842
f	8421
g	g
h	g1
i	g2
j	g21
k	g4
l	g41
m	g42
n	g421
o	g8
p	g81
q	g82
r	g821
s	g84
t	g841
u	g842
v	g8421
w	w
x	w1
y	w2
z	w21

When adding numbers, one can concatenate the groups and simplify:

g42 821 g421 + 42 0 81 = g4242 821 g42181 = g84 84 0

Similarly, when subtracting numbers, one can remove symbols or use minus sign before simplifying:

g42 821 g421 - 42 0 81 = g 821 g-842 = g 821 842

Summary:

- Semi-tally, compatible with unary number systems and tally numbers
- Compatible with decimals in range `0-9`
- Easy to translate to binary and hexadecimal number systems from canonical form
- Strongly normalizing to canonical form
- English letters, words and sentences have an associated numerical hash
- Addition and subtraction is very efficient