



Posit Tutorial

LEONG Siew Hoon (Cerlane)

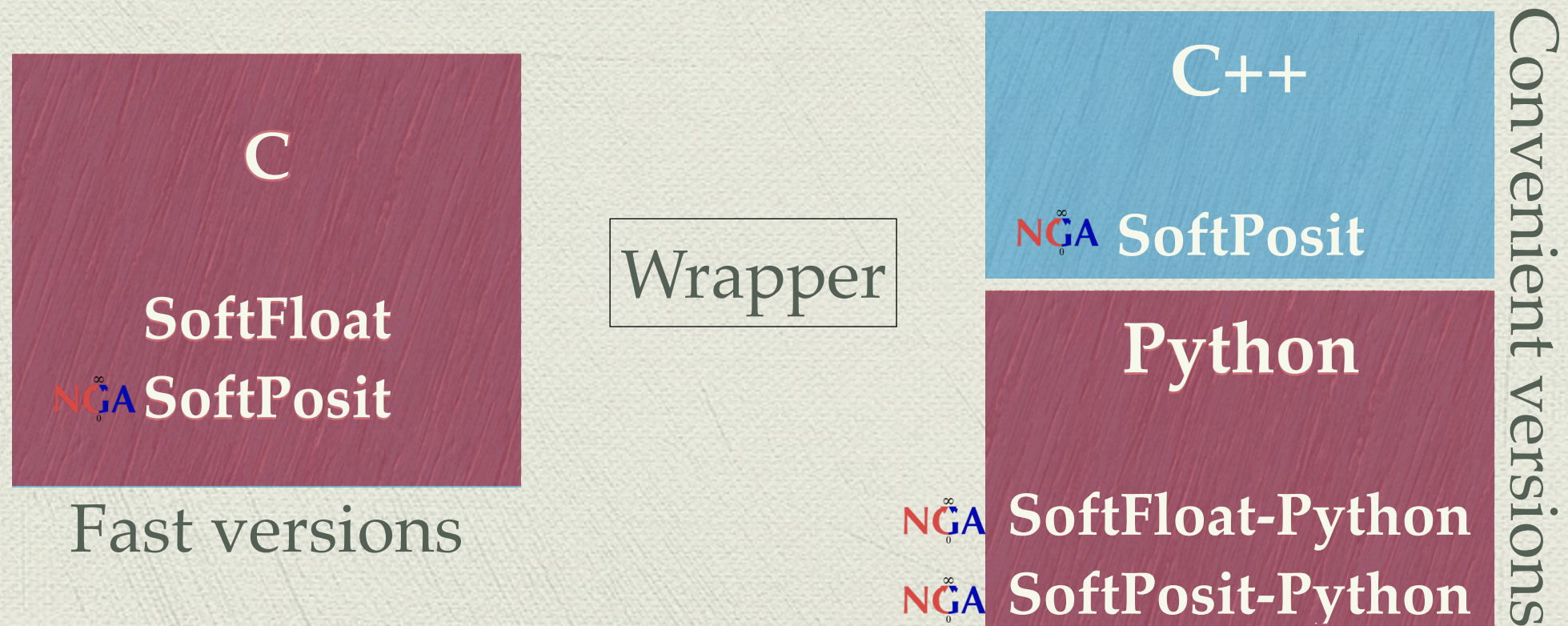
13 March 2019

Sponsors



Background Info

- ◆ Two main software used:
 - ◆ Berkeley SoftFloat 3d by John Hauser (22 yrs old code)
 - ◆ NGA SoftPosit (latest from GitLab)



Background Info

Data types:

SoftFloat-Python

float16

float32

float64

SoftPosit-Python

posit8

quire8

posit16

quire16

posit32

quire32

posit_2

quire_2

posit_1

quire_1

posit_0

quire_0

Setup

- ◆ Visit

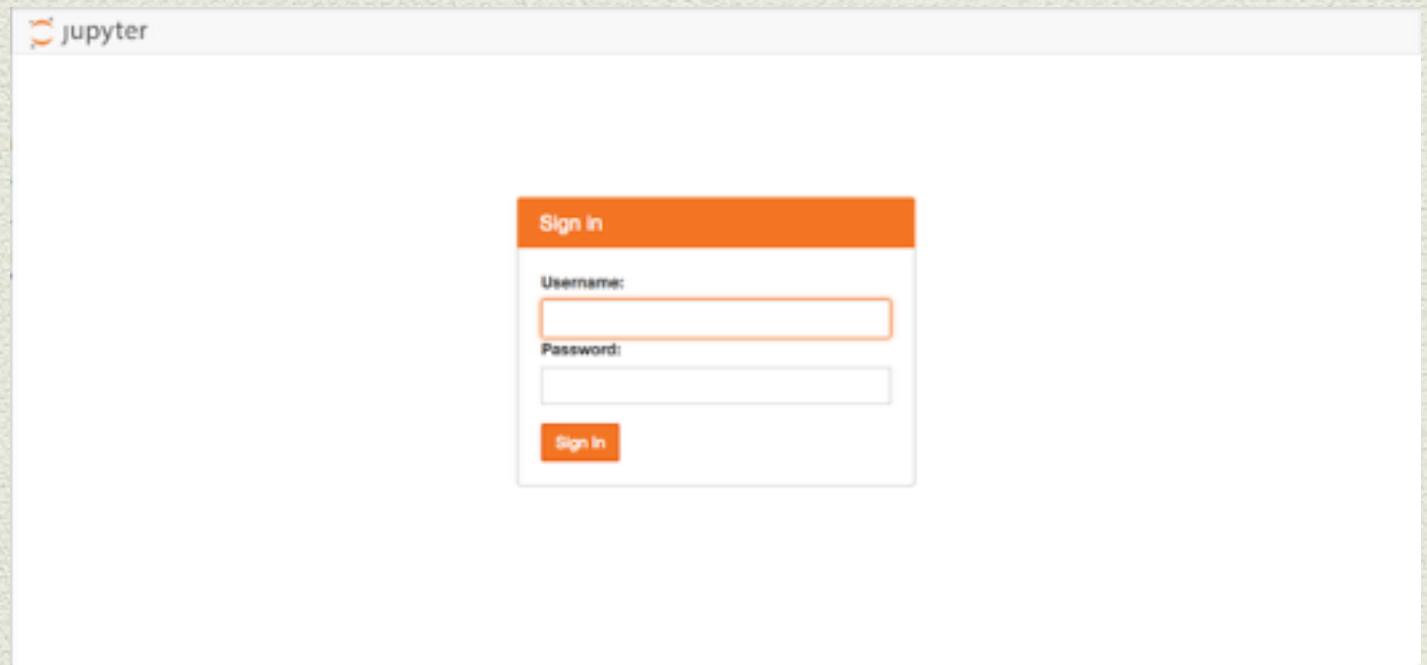
- ◆ <https://posit.comp.nus.edu.sg/>

- ◆ Username

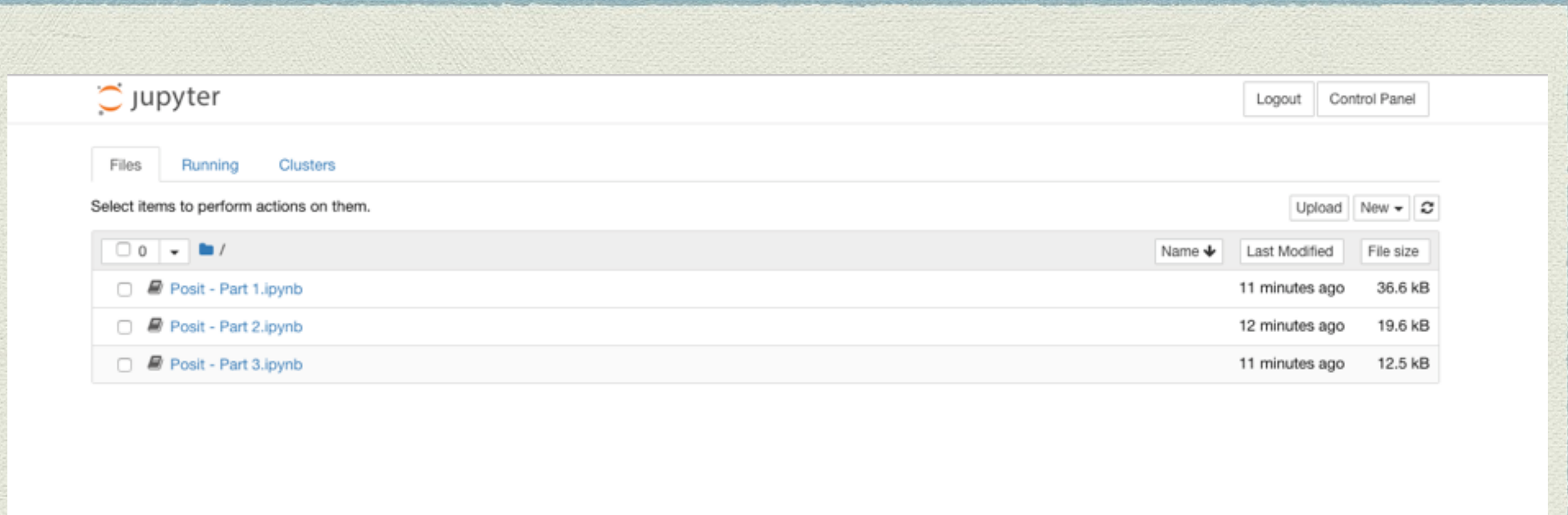
- ◆ [Redacted]

- ◆ Password

- ◆ [Redacted]

A screenshot of a web browser window displaying the JupyterLab login page. The browser's title bar shows the Jupyter logo and the word "jupyter". The main content area is white and contains a "Sign in" form. The form has an orange header bar with the text "Sign in". Below this, there are two input fields: "Username:" and "Password:". The "Username:" field is highlighted with an orange border. At the bottom of the form is an orange "Sign in" button.

Successful Login



Jupyter keyboard shortcuts:

- **Ctrl+Enter** (to run code) or **Cmd+Enter** (OSX)
- **Tab** key when typing a variable or function name to get hints

Tutorial Part 1

Posit types

◆ Recommended

	posit8	posit16	posit32
posit size (<i>ps</i>)	8 bits	16 bits	32 bits
exp. size (<i>es</i>)	0 bit	1 bit	2 bits
significant digits	~3 decimal digits	~5 decimal digits	~10 decimal digits
range	-64 — 64	-2.68E+8 — 2.68E+8	-1.33E+36 — 1.336E+36

float32 range:
-3.4E+38 — 3.4E+38

float32:
7 digits

float32 *es*:
8 bits

Posit recommended types

SoftPosit:

Python:

$pA = sp.\text{posit8}(value)$

E.g.
import softposit as sp
 $pA = sp.\text{posit8}(0.3)$

C++:

$\text{posit8 } pA = value;$

C:

$\text{posit8_t } pA = \text{convertDoubleToP8}(value);$

Python:

Faster way:

$pA = sp.\text{posit16}(value)$

C++:

$\text{posit16_t } pA =$
 $\{ .v = 0x2F10 \};$

$\text{posit16 } pA = value;$

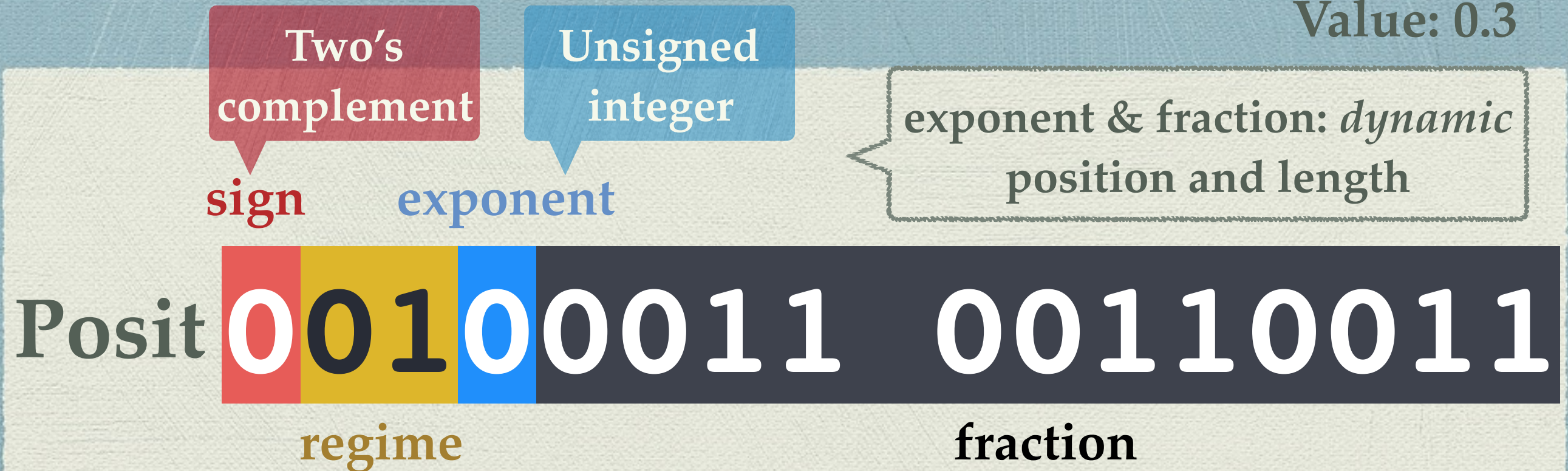
Recommended
 $\text{posit}<32,2>$ can be
declared similarly

C:

$\text{posit16_t } pA = \text{convertDoubleToP16}(value);$

Posit vs Float Format

Value: 0.3



Regime

Positive value <1 :

0001

Repeating **Zeros**
terminated by a **One**

Positive value ≥ 1 :

11110

Repeating **Ones**
terminated by a **Zero**

Regime size = $ps-1$

runlength = 14

runlength = 15

0 00000000 00000000**1**

0 1111111 11111111

0001

sign of regime bit (r): 0
runlength (m): 3

$k = -3$

11110

sign of regime bit (r): 1
runlength (m): 4

$k = 3$

To calculate the magnitude of regime:

$$k = \begin{cases} -m & \text{if } r = 0 \\ m - 1 & \text{if } r = 1 \end{cases}$$

$$useed = 2^{2^{es}}$$

exponent
size : es

Special Posits

Only one bit pattern
for zero

Zero

000000000 0000000000

Not a Real

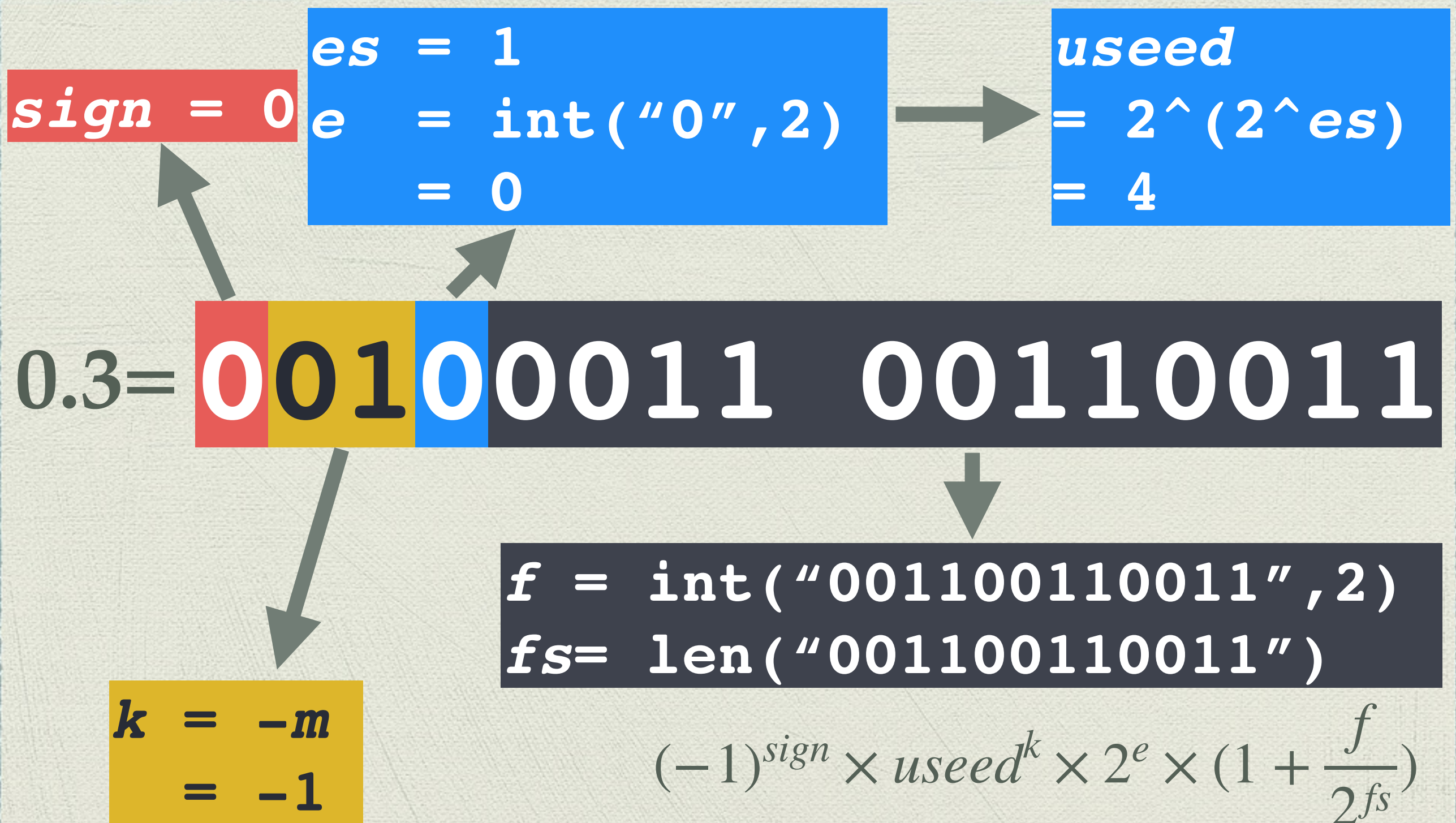
Always a 1 follows by 0s

NaR

100000000 0000000000

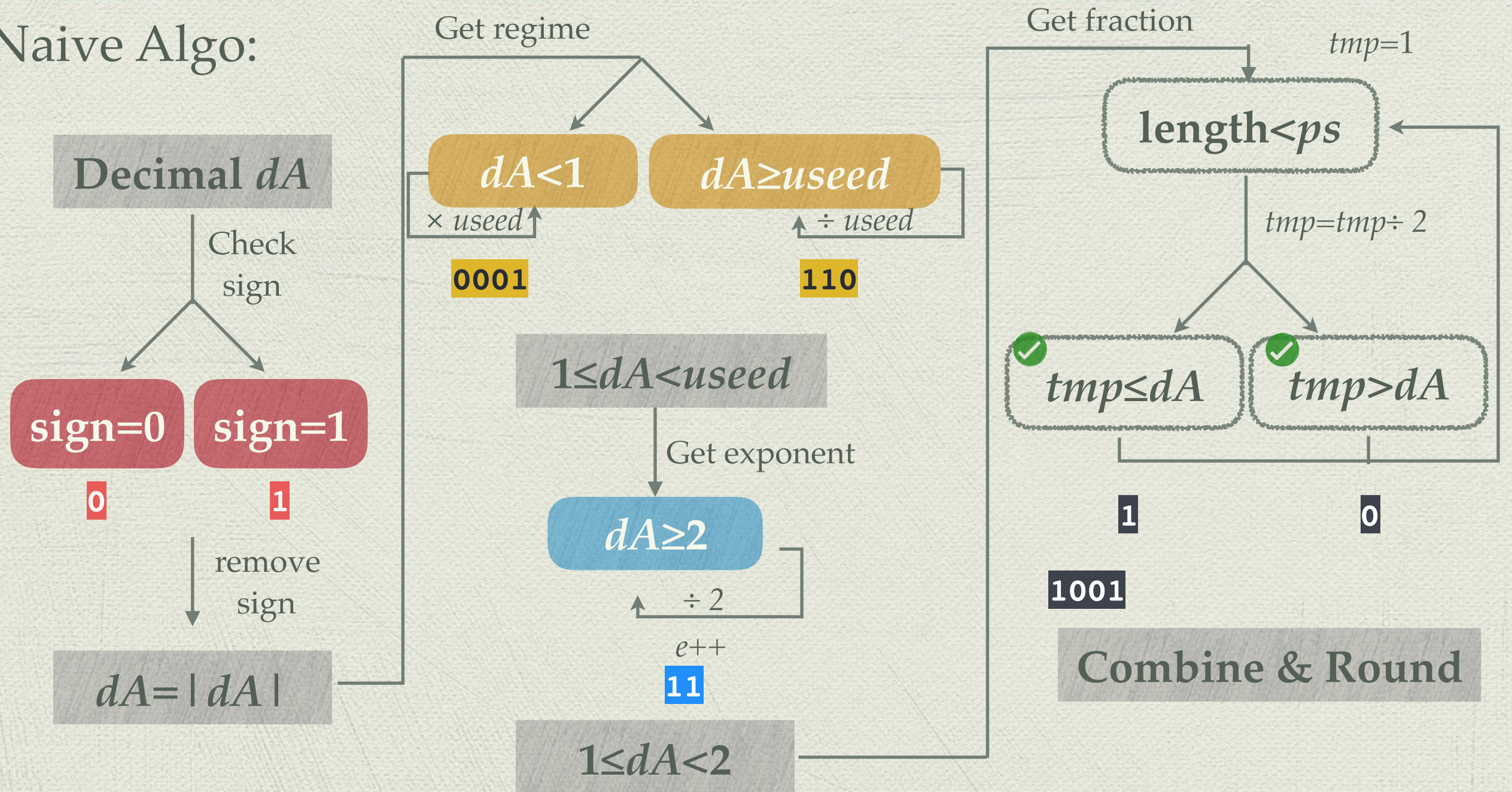
Represents NaN and $\pm\text{INF}$

Calculating the value of a posit

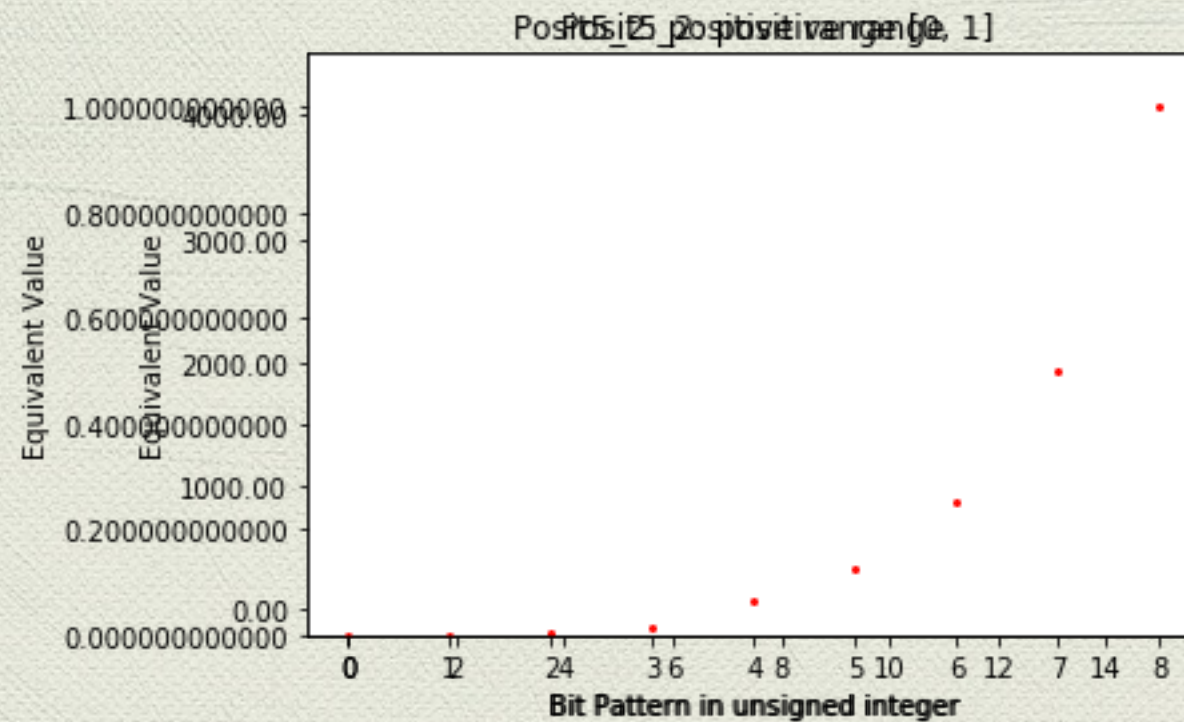


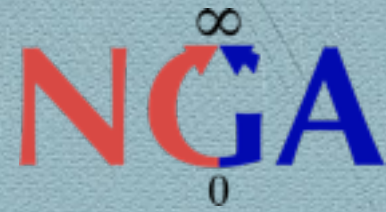
Converting decimal to posit

Naive Algo:



Posit Intervals





Special thanks to NUS for giving me
a VM to run the tutorial!



Questions?

Feedback form

<https://bit.ly/2tEN9bL>

Please kindly fill it!

*Thank
You*