Math 104B Homework 1

Rad Mallari
8360828

1.) Consider a reduced system where floating point numbers are represented in binary as $\pm S \cdot 2^E$ where $S = 1.b_1b_2$ and the exponent can only be $-1, 0, 1$.

(a) How many numbers can this system represent?

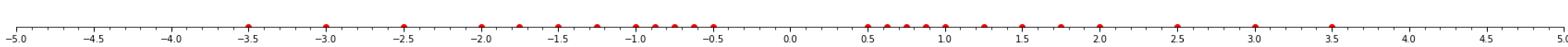
This system can represent 24 numbers.

(b) Display these numbers in the real line.

In [1]:
```
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
f_point_numbers = [1/2, 5/8, 3/4, 7/8, 1, 5/4, 3/2, 7/4, 2, 5/2, 3, 7/2]
f_point_numbers += [-num for num in f_point_numbers]
f_point_numbers.sort()
print(f_point_numbers)

fig = plt.figure(figsize=(30, 20))
n=8
ax = plt.subplot(n, 1, 2)
plt.plot(f_point_numbers, [0 for x in range(len(f_point_numbers))], 'ro')
ax.spines['right'].set_color('none')
ax.spines['left'].set_color('none')
ax.yaxis.set_major_locator(ticker.NullLocator())
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.tick_params(which='major', width=1.00)
ax.tick_params(which='major', length=5)
ax.tick_params(which='minor', width=0.75)
ax.tick_params(which='minor', length=2.5)
ax.set_xlim(-5, 5)
ax.set_ylim(0, 1)
ax.patch.set_alpha(0.0)
ax.xaxis.set_major_locator(ticker.MultipleLocator(0.5))
ax.xaxis.set_minor_locator(ticker.MultipleLocator(0.1))
```

```
[-3.5, -3, -2.5, -2, -1.75, -1.5, -1.25, -1, -0.875, -0.75, -0.625, -0.5, 0.5, 0.625, 0.75, 0.875, 1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5]
```



(c) What is the *eps* of this system?

$eps = \frac{2^{-2}}{2}$

2.) How many numbers are there in double precision?

Double precision have exponents $E_{min} = -1022$ and $E_{max} = 1023$.
So for we have $N_{min} = min_{x\in DP}|x| = 2^{-1022} \approx 2.2 \times 10^{308}$ and $N_{max} = max_{x\in DP}|x| \approx 1.8 \times 10^{308}$.
Therefore, in total $N = 2.2 \times 10^{308} + 1.8 \times 10^{308}$

3.) Suppose we do arithmetic with only two digits using rounding. For example $x = 3.47$ is represented as $x^* = 3.5$.
Let $x = 2.5$ and $y = 2.4$. Show that using this system, $(x-y)^2 = 0.01$, but $x^2 - 2xy + y^2 = 0.1$.

For this system, $(x-y)^2 = (2.5-2.4)^2 = (0.1)^2 = 0.01$. Meanwhile, $2.5^2 - 2 \cdot 2.5 \cdot 2.4 + 2.4^2 = 6.3 - 12 + 5.8 = 0.1$

4.) Suppose you need to compute $y = x - sinx$ for x small. There is going to be a significant cancelation of digits if the computation is performed directly.
How many digits are lost in double precision when $x = 0.05$? Propose an alternative way to compute y with nearly full machine precision.

Taylor expansion of $sinx$ about 0, is given by

$$sinx = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots$$

We can use this expansion to compute y with better precision.

5.) Let $y = \sqrt{1+x} - 1$, where x is very small

(a) Prove that y can be written as

$$y = \frac{x}{\sqrt{1+x}+1}$$

(b) Explain why (**??**) removes the digit cancellation problem that $y = \sqrt{1+x} - 1$ has.

Solution:

(a) Multiplying by $\frac{\sqrt{1+x}+1}{\sqrt{1+x}+1}$ gives us:

$$y = \frac{x}{\sqrt{1+x}+1} \cdot \left(\frac{\sqrt{1+x}+1}{\sqrt{1+x}+1}\right) = \frac{1+x-1}{\sqrt{1+x}+1}$$

Which simplifies to:

$$y = \frac{x}{\sqrt{1+x}+1}$$

(b) We know that subtracting $\sqrt{1+x}$ and -1 causes digit cancellation for small values of x.
Rewriting our y this way is ≈ 2 in the denominator for small x, thereby removing digit cancellation problem.

6.) Machine precision ($eps = 2^{-52}$) can be computer by the following program (attributed to Cleve Moler):

Machine precision

$a = \frac{4}{3}$

$b = a - 1$

$c = b + b + b$

$eps0 = |c - 1|$

Run the program and prove its validity.

In [3]:
```
# machine precision
print("Showing machine precision (eps=2**-(52))")
a = 4/3
print(f"a: {a}")
b = a-1
print(f"b: {b}")
c = b+b+b
print(f"c: {c}")
eps_0 = abs(c-1)
print(f"eps0: {eps_0}")
```

```
Showing machine precision (eps=2**-(52))
a: 1.3333333333333333
b: 0.33333333333333326
c: 0.9999999999999998
eps0: 2.220446049250313e-16
```

In [ ]: