How do we describe general while-loops in terms of μ-revision?

\[ \mu n = \varnothing \text{ or } \emptyset (\varnothing, n) \text{ only we have } \]

\[ \text{int } n \leftarrow 1; \]

\[ \text{while } (n < a) \]

\[ n \leftarrow n \times 2; \]

\[ \}

For loop

\[ \text{int } n \leftarrow 1; \]

\[ \text{for } (\text{int } j = 1; j < m; j++) \]

\[ n \leftarrow 2 \times n; \]

\[ f (0) = g (1) \]

\[ f (m+1) = h (m, f (m)) \]

\[ g = 1 \circ 0 \circ \emptyset \]

\[ h = 2 \circ \emptyset^2 \]

\[ f (m) = P E (0 \circ \emptyset \circ \text{prod}(2, \emptyset^2)) \]

# of iterations \( m \) is the smallest natural number for which \( f (m) < a \)
\[ \mu_n(\lfloor \frac{m}{n} \rfloor < a) \]
\[
\mathcal{L}(\mu_n(\lfloor \frac{m}{n} \rfloor < a)) \]

where
\[
\mathcal{L}(m) = \mathcal{P} \mathcal{E}(\alpha, \emptyset, \text{mult}(\tau_1^3, \tau_2^3))
\]

\[ \text{for loop } \]
\[ \text{int } x = a; \]
\[ \text{while } (x < b) \Rightarrow \]
\[ x = x \times x; \]

\[ x(\alpha, \emptyset) = \alpha \]
\[ x(\alpha, m+1) \neq x(\alpha, m) \]:

\[ \mathcal{L}(n_1, \emptyset) = n_1 \]
\[ \mathcal{L}(n_1, m+1) = \mathcal{L}(n_1, m) \times \mathcal{L}(n_1, 1) \]
\[ \mathcal{L}(n_1, \emptyset) = \mathcal{L}(n_1) = n_1 \]
\[ \mathcal{L}(n_1, m+1) = \mathcal{L}(n_1, m, \mathcal{L}(n_1, m)) = \text{prod}(\tau_3^3, \tau_1^3) \]

\[ x(\alpha, m) = \mathcal{P} \mathcal{E}(\tau_1^3, \text{prod}(\tau_3^3, \tau_1^3)) \]

we stop at smallest \( m \) for which
\[ x(\alpha, m) < b \]

\[ \mu_n(\lfloor \frac{m}{n} \rfloor < b) \Rightarrow \#_{\text{of iterations}} \]
Exercise

```c
int a = 1; i = 1;
while (a < b*c) {
    a = a * i;
    i = i + 1;
}
```

```c
int a = 1;
for (int j = 1; j <= m; j++)
    a = a * j;
```

\( a(0) = 1 \)
\( a(m+1) = a(m) \times (m+1) \)

\( f(0) = 1 \)
\( f(m+1) = f(m) \times (m+1) \)

\( q = 1 \)
\( h(n, f(m)) = \text{mult}(\pi_2, 0, \pi_3) \)
\( a(m) = \text{prod}(\pi_1, \text{mult}(\pi_2, a, \pi_2)) \)

```c
int x = s;
while (x < 1 || x > 0)
    x = (a * x + b) % N;
```
Computable = μ-recursive

Intuition based on computing -
Math-wise it is complicated

But math also has basic & reasonably clear
concepts. Main concept of math is set.
There are basic operations: U, N, -
Union, Intersection, Complement

Sets of natural numbers:
Def. A set \( A \) is called decidable if there's an
algorithm that, given a natural number \( n \),
checks whether \( n \in A \)

\[
f(n) = \begin{cases} 
1 & \text{if } n \in A \\
0 & \text{if } n \notin A 
\end{cases}
\]

\( \chi_A(n) \) characteristic function

Theorem 1, \( \emptyset \) is decidable
Proof: return false

Theorem 2, \( \mathbb{N} \) is decidable
Proof: return true

Th 3, every finite set is decidable
Th 4, if \( A \) and \( B \) are decidable,
\( A \cup B \) is decidable? (\( n \) has to belong at least 1 set)
Th 5, if \( A \) and \( B \) are decidable,
A \& B is decidable (\(n\) has to belong to both sets)

Some sets are not decidable

Def. A set 'A' is recursive enumerable (r.e.)

if there exists an algorithm that

\[\text{eventually lists every element of this set.}\]

or

\[\text{if there exists an algorithm for which every element of this set will be eventually printed}\]

Th 1, \( \emptyset \) is r.e.
Th 2, \( N \) is r.e.
Th 3, Every finite set is r.e.
Th 4, Every decidable set is r.e.

\[
\text{int m=0;}
\text{while (true) }
\text{if (m \in A)}
\text{\{ s.o.p. (m); m; \}}
\text{\}}
\text{\}
\]

Th 5, If \( A, B \) are r.e. then \( A \cup B \) is r.e.