In computability theory, we often need to work with functions that take multiple arguments or return multiple values. However, the formal models of computation, such as Turing Machines or the URM (Unlimited Register Machine), operate on natural numbers. To handle multiple arguments or return values, we use a technique called pairing.

Pairing Function

A pairing function is a bijective function that encodes a pair of natural numbers into a single natural number. The most common pairing function is the Cantor pairing function, denoted as $\langle x, y \rangle$ or $\pi(x, y)$.

The Cantor pairing function is defined as follows: $\pi(x, y) = \langle x, y \rangle = (1/2)(x + y)(x + y + 1) + y$

This function has the following properties:

- It is a bijection from $N \times N$ to N.
- It is primitive recursive.

Projection Functions

To extract the first and second components of a pair encoded by the Cantor pairing function, we use the projection functions w_1 and w_2.

The projection functions are defined as follows:

- $w_1(\langle x, y \rangle) = x$
- $w_2(\langle x, y \rangle) = y$

These functions are also primitive recursive.

Example: Encoding and Decoding a Function with Multiple Arguments

$(w)_1, (w)_2$ are meant to be encoding in pairs and represent basically tuples – they are used to correctly replace w, z, y and variables like that *inside* minimalization operator. They exist basically because there is not a pair-minimizing operator. Nested minimalization doesn't work either, because I would scroll the table first only on the columns and then only on the rows.

Basically, they are used to map x, y as projection elements to transform a predicate into a mathematical expression (coding a couple as an integer). Consider this example which extends what was written before; basically, we use this encoding to replace x, y, t (example taken from exercise 8.26 – one of the very few to make us understand because the process is clearly written – would love it if was always like that):

$$\begin{aligned} sc_A(x) &= \mathbf{1}(\mu(y,z,t).H(x,y,t) \wedge S(x,z,y,t)) \\ &= \mathbf{1}(\mu w.H(x,(w)_1,(w)_3) \wedge S(x,(w)_2,(w)_1,(w)_3) \end{aligned}$$

Usually, in other cases (but the encoding depends on the specific problem, remember):

- $(w)_1$: y
- $(w)_2$: t (number of steps)

The introduction of their variables is explained at the end of the Universal Function lesson; basically, when we talk about the inverse function to determine if it terminates over input x in a defined number of steps, we might use the encoding in pairs π, but instead we use the exponent of the first prime number 1 and the exponent of the second prime number 2. So, we have: