

# Pairing Functions and Gödel Numbers

## Coding Pairs of Numbers by a Single Number

consider the primitive recursive function

$$\langle x, y \rangle = 2^x(2y+1) - 1$$

Every different pair of numbers,  $x$  and  $y$ , maps to a unique value of

$$z = \langle x, y \rangle$$

$x$	0 0 0 0 ... 1 1 1 1 0 0 ... 2 2 2 2 ...
$y$	0 1 2 3 ... 0 1 2 3 ... 0 1 2 3 ...
$z$	0 2 4 6 ... 1 5 9 13 ... 3 11 19 ...

There is a unique solution  $(x, y)$  to the equation

$$\langle x, y \rangle = z$$

$x$ : the largest number such that  $2^x | (z+1)$

$y$ : subsequently,  $y$  is the solution to

$$2y+1 = \frac{z+1}{2^x}$$

If we do not choose the largest  $x$  such that  $2^x | (z+1)$ , then  $\frac{z+1}{2^x}$  will remain even and  $2y+1 = (z+1)/2^x$  will NOT have a solution.

It is always the case that the solution

$$x \leq z \quad y \leq z$$

This follows from

$$2^x(2y+1) \leq z+1$$

So we must have

$$\begin{array}{l} 2^x \leq z+1 \\ \text{implying } x \leq z \end{array} \quad \begin{array}{l} 2y+1 \leq z+1 \\ \text{implying } 2y \leq z \\ \text{implying } y \leq z \end{array}$$

In both cases, we cannot say that  $x < z$ ,  $y < z$  because of the values  $x=0, y=0, z=0$ .

Designate the solutions  $x$  and  $y$  by

$$x = l(z) \quad y = r(z)$$

$$l(z) = \min_{x \leq z} [(\exists y)_{\leq z} \ z = \langle x, y \rangle]$$

$$r(z) = \min_{y \leq z} [(\exists x)_{\leq z} \ z = \langle x, y \rangle]$$

$l(z)$  &  $r(z)$  are primitive recursive functions

## Coding Lists of Numbers by Single Numbers

The Gödel number representation of a finite sequence  $a_1, a_2, \dots, a_n$  is displayed as

$$[a_1, a_2, \dots, a_n]$$

$$[a_1, a_2, \dots, a_n] = \prod_{i=1}^n p_i^{a_i}$$

$\leftarrow$  a primitive recursive function

$$\begin{array}{l} p_1 = 2, p_2 = 3 \\ p_3 = 5, p_4 = 7, \dots \end{array}$$

Uniqueness Property of Gödel Numbers:

$$\text{If } [a_1, a_2, \dots, a_n] = [b_1, b_2, \dots, b_n]$$

$$\text{then } a_i = b_i \quad i = 1, \dots, n$$

follows from the uniqueness of the factorization of integers into primes. (Fundamental Th. of Arithmetic)

But note:

$$[a_1, \dots, a_n] = [a_1, \dots, a_n, 0, 0, 0 \dots]$$

$\uparrow$  arbitrary number of zeros

By declaration, we set the Gödel number of an empty sequence to be 1

$$[] = 1$$

This jibes with  $[0] = [0, 0] = [0, 0, 0 \dots] = 1$  so the Gödel number of an empty sequence does not change if we insert zero into the empty sequence just as ~~is~~ is the case with a non-empty sequence. (Smallest Gödel number is 1)

Given the form

$$x = [a_1, \dots, a_n]$$

we define the primitive recursive function

$$(x)_i = a_i$$

We have

$$(x)_i = \min_{t \leq x} ({}^n p_i^{t+1} | x)$$

Note that  $(x)_0 = 0$  always since

$$(x)_0 = \min_{t \leq x} ({}^n p_0^{t+1} | x)$$

$Lt(x) \equiv$  length  $n$  of the sequence  $a_1, a_2, \dots, a_n$  with Gödel number  $x$

$$Lt(x) = \min_{i \leq x} ((x)_i \neq 0 \wedge (\forall j)_{\leq x} (j \leq i \vee (x)_j = 0))$$

$$x = 20 = 2^2 \cdot 5^1 = [2, 0, 1], \quad Lt(x) = 3$$

$$Lt(0) = Lt(1) = 0$$

# Coding Programs By Numbers :

We can represent each program  $P$  of language  $S$  by a single integer shown as  $\#(P)$ . The program can be retrieved from this integer.

→ We first arrange the variables of  $P$  as follows :

$y, x_1, z_1, x_2, z_2, \dots$

note the interleaving  
of the input and  
the local vars

→ We arrange the labels as follows

$A_1, B_1, C_1, D_1, E_1, A_2, B_2, C_2, D_2, E_2, \dots$

→ We consider the unsubscripted  $X$  to be the same  $x_1$ , unsubscripted  $Z$  to be the same as  $z_1$ , unsubscripted  $E$  to be the same as  $E_1$ , and so on.

→ We write  $\#(V)$  for the position of a variable in the ordering shown above.  $\#(y) = 1$

→ We write  $\#(L)$  for the position of a label in the same way.  $\#(A_1) = 1$

→ Let  $I$  be an instruction of  $S$ . We associate a number  $\#(I)$  with  $I$ :

$$\#(I) = \langle a, \langle b, c \rangle \rangle$$

$a = 0$  if  $I$  is unlabeled  
 $a = \#(L)$  if  $I$  is labeled  $L$ .

If  $I$  mentions  $V$   
 $c = \#(V) - 1$

- $b = 0$  if  $I$  is  $V \leftarrow V$
- $b = 1$  if  $I$  is  $V \leftarrow V + 1$
- $b = 2$  if  $I$  is  $V \leftarrow V - 1$
- $b = \#(L') + 2$  if  $I$  is  $IF V \neq 0 GOTO L'$

Suppose  $I$  is  $x \leftarrow x + 1$  (unlabeled), what is  $\#(I)$  ?

$$\#(I) = \langle 0, \langle 1, 1 \rangle \rangle = \langle 0, 5 \rangle = 40$$

Suppose  $I$  is the dummy statement  $y \leftarrow y$ , what is  $\#(I)$  ?

$$\#(I) = \langle 0, \langle 0, 0 \rangle \rangle = 0$$

this is the reason for subtracting  
1 from  $\#(V)$  for setting  $c$

Let a program  $P$  consist of the statements  $I_1, I_2, \dots, I_n$ . We set

$$\#(P) = [\#(I_1), \#(I_2), \dots, \#(I_n)] - 1$$

NOTES ① For a long time, the number 1 was also considered to be a prime.

But now the first prime is 2. If we allow 1 to be a prime, the Fundamental Theorem of Arithmetic ceases to be valid. This theorem says that every natural number greater than 1 is either a prime or possesses a unique factorization into primes. The uniqueness is both in terms of the primes involved and their powers.

Since the Gödel number of an empty sequence is 1, subtracting 1 ensures that  $\#(P)$  for empty  $P$  is zero

② To prove that  $2^x \leq 3+1$  implies  $x \leq 3$  : In a proof by contradiction let's say that  $2^x \leq 3+1$  implies  $x > 3$ . That leads to  $2^x > 2^3$  which we write  $2^x > (1+1)^3$ . Now recall the binomial expansion  $(1+x)^3 = \sum_{k=0}^3 \binom{3}{k} x^k$ . So  $(1+1)^3 = \sum_{k=0}^3 \binom{3}{k}$ . Recall the formula for binomial coefficient  $\binom{3}{k} = \frac{3!}{(3-k)! k!}$ , we have  $(1+1)^3 = 1+3+\dots$ . That leads to  $2^x > 1+3$  which is a contradiction.