Pairing Functions and Gödel Numbers

This way the equation $\langle x, y\rangle = z$ defines functions $x = l(z)$ and $y = r(z)$.

$\langle x, y\rangle = z$ also implies that $x, y < z + 1$, and therefore $l(z) \le z$, $r(z) \le z$.

Then we can write:

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

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

showing that l(z) and r(z) are primitive recursive functions.

It is also true that $\langle x, y\rangle = z \Leftrightarrow x = l(z) \;\&\; y = r(z)$.

Pairing Functions and Gödel Numbers

Theorem 8.1 (Pairing Function Theorem):

The functions $\langle x, y\rangle$, l(z) and r(z) have the following properties:

1. they are primitive recursive;
2. $l(\langle x, y\rangle) = x$; $r(\langle x, y\rangle) = y$;
3. $\langle l(z), r(z)\rangle = z$;
4. $l(z), r(z) \le z$.

Pairing Functions and Gödel Numbers

We now want to develop primitive recursive functions that encode and decode arbitrary finite sequences of numbers.

Our method (actually invented by Gödel) will be based on the prime power decomposition of integers.

We define the Gödel number of the sequence $(a_1, \ldots, a_n)$ to be the number

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

For example, the Gödel number of the sequence (7, 6, 4, 4, 3) is

$2^7 \cdot 3^6 \cdot 5^4 \cdot 7^4 \cdot 11^3$.

Pairing Functions and Gödel Numbers

For each n, the function $[a_1, \ldots, a_n]$ is clearly primitive recursive.

Gödel numbering satisfies the following uniqueness property:

Theorem 8.2:

If $[a_1, \ldots, a_n] = [b_1, \ldots, b_n]$ then $a_i = b_i$ for $i = 1, \ldots, n$.

This follows immediately from the **fundamental theorem of arithmetic**, i.e., the uniqueness of the factorization of integers into primes.

Pairing Functions and Gödel Numbers

However, it is important to note that

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

because for any n+1, $p^0_{n+1} = 1$.

Actually, we could add any number of 0s to the right end of a sequence without changing its Gödel number.

Since we have $1 = 2^0 = 2^0 3^0 = 2^0 3^0 5^0 = \ldots$,

it is useful to define 1 as the Gödel number of the empty sequence of length 0.

Pairing Functions and Gödel Numbers

Obviously, adding a zero to the left of the sequence will lead to a Gödel number different from the initial one.

Examples:

$[1, 4] = 2^1 \cdot 3^4 = 162$

$[1, 4, 0] = 2^1 \cdot 3^4 \cdot 5^0 = 162$

$[0, 1, 4] = 2^0 \cdot 3^1 \cdot 5^4 = 1875$

Pairing Functions and Gödel Numbers

We will now define a primitive recursive function $(x)_i$ so that if $x = [a_1, \ldots, a_n]$, then $(x)_i = a_i$.

We set

$(x)_i = \min_{t \le x}(\sim p_i^{t+1} \mid x)$.

Then we define the length Lt(x) of the sequence for the Gödel number x:

$Lt(x) = \min_{i \le x}((x)_i \ne 0 \;\&\; (\forall j)_{\le x}(j \le i \vee (x)_j = 0))$.

Example: If $x = 20 = 2^2 \cdot 5^1 = [2, 0, 1]$,

then $(x)_1 = 2$, $(x)_2 = 0$, $(x)_3 = 1$, $(x)_4 = (x)_5 = \ldots 0$,

$Lt(x) = 3$.

Pairing Functions and Gödel Numbers

If $x > 1$ and $Lt(x) = n$, then p_n divides x but no prime greater than p_n divides x.

Note that $Lt([a_1, \ldots, a_n]) = n$ if and only if $a_n \ne 0$.

Theorem 8.3 (Sequence Number Theorem):

a. $([a_1, \ldots, a_n])_i = a_i$ if $1 \le i \le n$
b. $[(x)_1, \ldots, (x)_n] = x$ if $n \ge Lt(x)$

Coding Programs by Numbers

After having developed appropriate coding techniques, it will be our goal to **enumerate** all programs of the language $\mathcal{L}$.

In other words, each program $\mathcal{P}$ of $\mathcal{L}$ will receive a **number** $\#(\mathcal{P})$ so that the program can be retrieved from its number.

Let us first arrange the variables in the following order:

$Y\ X_1\ Z_1\ X_2\ Z_2\ X_3\ Z_3 \ldots$

And also the labels:

$A_1\ B_1\ C_1\ D_1\ E_1\ A_2\ B_2\ C_2\ D_2\ E_2\ A_3 \ldots$

Coding Programs by Numbers

We write #(V), #(L) for the position of a given variable or label in the appropriate ordering.

For example, $\#(X_2) = 4$, $\#(Z) = 3$, $\#(C_2) = 8$.

Now let I be an instruction (labeled or unlabeled) of the language $\mathcal{L}$.

Coding Programs by Numbers

Then we write

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

where

1. if I is unlabeled, then a = 0; if I is labeled L, then a = #(L);
2. if the variable V is mentioned in I, then c = #(V) – 1;
3. if the statement in I is $V \leftarrow V$, $V \leftarrow V+1$, or $V \leftarrow V-1$, then b = 0, 1, or 2, respectively;
4. if the statement in I is IF $V \ne 0$ GOTO L' then b = #(L') + 2.

Coding Programs by Numbers

Examples:

The number of the unlabeled instruction $X \leftarrow X-1$ is $\langle 0, \langle 2, 1\rangle\rangle = \langle 0, 11\rangle = 22$.

The number of the instruction [A] $X \leftarrow X-1$ is $\langle 1, \langle 2, 1\rangle\rangle = \langle 1, 11\rangle = 45$.

Coding Programs by Numbers

Note that for any given number q there is a unique instruction I with #(I) = q.

We first calculate l(q).

If l(q) = 0, I is unlabeled; otherwise I has the l(q)-th label in our list.

To find the variable mentioned in I, we compute i = r(r(q)) + 1 and locate the i-th variable V in our list.

Then the statement will be $V \leftarrow V$, $V \leftarrow V+1$, or $V \leftarrow V-1$, if l(r(q)) = 0, 1, or 2, respectively; otherwise, it will be the statement IF $V \neq 0$ GOTO L, where L is the j-th label in our list and j = l(r(q)) –2.

Coding Programs by Numbers

Finally, for a program $\mathcal{P}$ that consists of the instructions $I_1, I_2, \ldots, I_k$, we set

$\#(\mathcal{P}) = [\#(I_1), \#(I_2), \ldots, \#(I_k)] - 1$.

This way we associated every possible program in $\mathcal{L}$ with a **unique number**.

Coding Programs by Numbers

Gödel numbers are usually very large, even for small programs.

Let us look at the following example:

```
[A]   X ← X+1
      IF X≠0 GOTO A
```

$\#(I_1) = \langle 1, \langle 1, 1\rangle\rangle = \langle 1, 5\rangle = 21$

$\#(I_2) = \langle 0, \langle 3, 1\rangle\rangle = \langle 0, 23\rangle = 46$

So the number of our small program is

$2^{21} \cdot 3^{46} - 1$.