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

\[
(x)_i = \min_{t \leq x} (\sim t\mid x).
\]

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

\[
L_t(x) = \min_{i \leq x} ((x)_i \neq 0 \& (\forall j \leq x \cdot (x)_j = 0)).
\]

Example: If \(x = 20 = 2^2 \cdot 5^1 = [2, 0, 1]\), then \((x)_1 = 2, (x)_2 = 0, (x)_3 = 1, (x)_4 = (x)_5 = \ldots = 0, L_t(x) = 3\).

Pairing Functions and Gödel Numbers

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

Note that \(L_t([a_1, \ldots, a_n]) = n\) if and only if \(a_n \neq 0\).

Theorem 8.3 (Sequence Number Theorem):

a. \(([a_1, \ldots, a_n])_i = a_i\) if \(1 \leq i \leq n\)

b. \(([(x)_1, \ldots, (x)_n]) = x\) if \(n \geq L_t(x)\)

Coding Programs by Numbers

After having developed appropriate coding techniques, it will be our goal to enumerate all programs of the language \(L\).

In other words, each program \(\varphi\) of \(L\) will receive a number \#(\varphi) 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:

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

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 \(L\).

Then we write

\[ \#(I) = (a, (b, c)) \]

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 \(V \leftarrow V + 1\), or \(V \leftarrow V - 1\), then \(b = 0, 1, or 2, respectively; 4. if the statement in \(I\) is IF \(V=0\) GOTO \(L’\), then \(b = \#(L) + 2\).
Coding Programs by Numbers

Examples:
The number of the unlabeled instruction $X \leftarrow X-1$ is 
$(0, \langle 2, 1 \rangle) = (0, 11) = 22$.
The number of the instruction $[A] \ X \leftarrow X-1$ is 
$(1, \langle 2, 1 \rangle) = (1, 11) = 45$.

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 $j = l(r(q)) – 2$.

Finally, for a program $P$ that consists of the
instructions $I_1, I_2, \ldots, I_k$, we set
$(P) = [l(I_1), l(I_2), \ldots, l(I_k)] – 1$.
This way we associated every possible program in $L$
with a unique number.

Gödel numbers are usually very large, even for small
programs.
Let us look at the following example:
$[A] \ X \leftarrow X+1$
IF $X=0$ GOTO $A$

Thus, the number of our small program is
$2^{21} \cdot 3^{46} – 1$.

Note that the number of the unlabeled instruction
$Y \leftarrow Y$ is $(0, \langle 0, 0 \rangle) = (0, 0) = 0$.
Thus, the number of a program will be unchanged if
an unlabeled instruction $Y \leftarrow Y$ is appended to it.
Although this ambiguity is harmless, we avoid it by
adding a sentence to our definition of programs of $L$:
The final instruction in a program is not permitted to
be the unlabeled statement $Y \leftarrow Y$.