art.1 Coding Terms

A term is simply a certain kind of sequence of symbols: it is built up inductively from constants and variables according to the formation rules for terms. Since sequences of symbols can be coded as numbers—using a coding scheme for the symbols plus a way to code sequences of numbers—assigning Gödel numbers to terms is not difficult. The challenge is rather to show that the property a number has if it is the Gödel number of a correctly formed term is computable, or in fact primitive recursive.

**Proposition art.1.** The relations $\text{Term}(x)$ and $\text{CI}Term(x)$ which hold iff $x$ is the Gödel number of a term or a closed term, respectively, are primitive recursive.

**Proof.** A sequence of symbols $s$ is a term iff there is a sequence $s_0, \ldots, s_{k-1} = s$ of terms which records how the term $s$ was formed from constant symbols and variables according to the formation rules for terms. To express that such a putative formation sequence follows the formation rules it has to be the case that, for each $i < k$, either

1. $s_i$ is a variable $v_j$, or
2. $s_i$ is a constant symbol $c_j$, or
3. $s_i$ is built from $n$ terms $t_1, \ldots, t_n$ occurring prior to place $i$ using an $n$-place function symbol $f^n_j$.

To show that the corresponding relation on Gödel numbers is primitive recursive, we have to express this condition primitive recursively, i.e., using primitive recursive functions, relations, and bounded quantification.

Suppose $y$ is the number that codes the sequence $s_0, \ldots, s_{k-1}$, i.e., $y = \langle \text{^s}_0, \ldots, \text{^s}_{k-1} \rangle$. It codes a formation sequence for the term with Gödel number $x$ iff for all $i < k$:

1. there is a $j$ such that $(y)_i = \text{^v}_j$, or
2. there is a $j$ such that $(y)_i = \text{^c}_j$, or
3. there is an $n$ and a number $z = \langle z_1, \ldots, z_n \rangle$ such that each $z_i$ is equal to some $(y)_{i'}$ for $i' < i$ and
\[
(y)_i = \text{^f}^n_j(\text{^flatten}(z) \text{^z})^a,
\]
and moreover $(y)_{k-1} = x$. The function $\text{flatten}(z)$ turns the sequence $\langle \text{^t}_1, \ldots, \text{^t}_n \rangle$ into $\text{^t}_1, \ldots, \text{^t}_n$ and is primitive recursive.

The indices $j, n$, the Gödel numbers $z_i$ of the terms $t_i$, and the code $z$ of the sequence $\langle z_1, \ldots, z_n \rangle$, in (3) are all less than $y$. We can replace $k$ above with $\text{len}(y)$. Hence we can express “$y$ is the code of a formation sequence of the
term with Gödel number $x^\#$ in a way that shows that this relation is primitive recursive.

We now just have to convince ourselves that there is a primitive recursive bound on $y$. But if $x$ is the Gödel number of a term, it must have a formation sequence with at most $\text{len}(x)$ terms (since every term in the formation sequence of $s$ must start at some place in $s$, and no two subterms can start at the same place). The Gödel number of each subterm of $s$ is of course $\leq x$. Hence, there always is a formation sequence with code $\leq x^{\text{len}(x)}$.

For ClTerm, simply leave out the clause for variables.

**Alternative proof of Proposition art.1.** The inductive definition says that constant symbols and variables are terms, and if $t_1, \ldots, t_n$ are terms, then so is $f^j_n(t_1, \ldots, t_n)$, for any $n$ and $j$. So terms are formed in stages: constant symbols and variables at stage 0, terms involving one function symbol at stage 1, those involving at least two nested function symbols at stage 2, etc. Let’s say that a sequence of symbols $s$ is a term of level $l$ iff $s$ can be formed by applying the inductive definition of terms $l$ (or fewer) times, i.e., it “becomes” a term by stage $l$ or before. So $s$ is a term of level $l + 1$ iff

1. $s$ is a variable $v_j$, or

2. $s$ is a constant symbol $c_j$, or

3. $s$ is built from $n$ terms $t_1, \ldots, t_n$ of level $l$ and an $n$-place function symbol $f^j_n$.

To show that the corresponding relation on Gödel numbers is primitive recursive, we have to express this condition primitive recursively, i.e., using primitive recursive functions, relations, and bounded quantification.

The number $x$ is the Gödel number of a term $s$ of level $l + 1$ iff

1. there is a $j$ such that $x = v^\#_j$, or

2. there is a $j$ such that $x = c^\#_j$, or

3. there is an $n$, a $j$, and a number $z = (z_1, \ldots, z_n)$ such that each $z_i$ is the Gödel number of a term of level $l$ and

$$x = f^\#_j((z^\# \triangleright \text{flatten}(z) \triangleright ^\#)^\#),$$

and moreover $(y)_{k-1} = x$.

The indices $j$, $n$, the Gödel numbers $z_i$ of the terms $t_i$, and the code $z$ of the sequence $(z_1, \ldots, z_n)$, in (3) are all less than $x$. So we get a primitive recursive definition by:

$$\text{lTerm}(x, 0) = \text{Var}(x) \lor \text{Const}(x)$$

$$\text{lTerm}(x, l + 1) = \text{Var}(x) \lor \text{Const}(x) \lor$$

$$(\exists z < x) (((\forall i < \text{len}(z)) \text{lTerm}(z, i)) \land$$

$$(\exists j < x) x = (f^\#_j(z^\# \triangleright \text{flatten}(z) \triangleright ^\#)^\#))$$
We can now define $\text{Term}(x)$ by $\text{ITerm}(x,x)$, since the level of a term is always less than the Gödel number of the term.

**Problem art.1.** Show that the function flatten$(z)$, which turns the sequence $\langle t_1^#, \ldots, t_n^# \rangle$ into $t_1^#, \ldots, t_n^#$, is primitive recursive.

**Proposition art.2.** The function $\text{num}(n) = n^#$ is primitive recursive.

*Proof.* We define $\text{num}(n)$ by primitive recursion:

\[
\begin{align*}
\text{num}(0) &= 0^# \\
\text{num}(n + 1) &= \text{num}(n) \bowtie \text{num}(n + 1)^#.
\end{align*}
\]

Photo Credits

Bibliography