

## art.1 Coding Symbols

inc:art:cod:  
sec

The basic language  $\mathcal{L}$  of first order logic makes use of the symbols

$$\perp \quad \neg \quad \vee \quad \wedge \quad \rightarrow \quad \forall \quad \exists \quad = \quad ( \quad ) \quad ,$$

together with **enumerable** sets of variables and **constant symbols**, and **enumerable** sets of **function symbols** and **predicate symbols** of arbitrary arity. We can assign *codes* to each of these symbols in such a way that every symbol is assigned a unique number as its code, and no two different symbols are assigned the same number. We know that this is possible since the set of all symbols is **enumerable** and so there is a **bijection** between it and the set of natural numbers. But we want to make sure that we can recover the symbol (as well as some information about it, e.g., the arity of a **function symbol**) from its code in a computable way. There are many possible ways of doing this, of course. Here is one such way, which uses primitive recursive functions. (Recall that  $\langle n_0, \dots, n_k \rangle$  is the number coding the sequence of numbers  $n_0, \dots, n_k$ .)

**Definition art.1.** If  $s$  is a symbol of  $\mathcal{L}$ , let the *symbol code*  $c_s$  be defined as follows:

1. If  $s$  is among the logical symbols,  $c_s$  is given by the following table:

$\perp$	$\neg$	$\vee$	$\wedge$	$\rightarrow$	$\forall$
$\langle 0, 0 \rangle$	$\langle 0, 1 \rangle$	$\langle 0, 2 \rangle$	$\langle 0, 3 \rangle$	$\langle 0, 4 \rangle$	$\langle 0, 5 \rangle$
$\exists$	$=$	$($	$)$	$,$	
$\langle 0, 6 \rangle$	$\langle 0, 7 \rangle$	$\langle 0, 8 \rangle$	$\langle 0, 9 \rangle$	$\langle 0, 10 \rangle$	

2. If  $s$  is the  $i$ -th variable  $v_i$ , then  $c_s = \langle 1, i \rangle$ .
3. If  $s$  is the  $i$ -th **constant symbol**  $c_i^n$ , then  $c_s = \langle 2, i \rangle$ .
4. If  $s$  is the  $i$ -th  $n$ -ary **function symbol**  $f_i^n$ , then  $c_s = \langle 3, n, i \rangle$ .
5. If  $s$  is the  $i$ -th  $n$ -ary **predicate symbol**  $P_i^n$ , then  $c_s = \langle 4, n, i \rangle$ .

**Proposition art.2.** *The following relations are primitive recursive:*

1.  $\text{Fn}(x, n)$  iff  $x$  is the code of  $f_i^n$  for some  $i$ , i.e.,  $x$  is the code of an  $n$ -ary **function symbol**.
2.  $\text{Pred}(x, n)$  iff  $x$  is the code of  $P_i^n$  for some  $i$  or  $x$  is the code of  $=$  and  $n = 2$ , i.e.,  $x$  is the code of an  $n$ -ary **predicate symbol**.

**Definition art.3.** If  $s_0, \dots, s_{n-1}$  is a sequence of symbols, its *Gödel number* is  $\langle c_{s_0}, \dots, c_{s_{n-1}} \rangle$ .

Note that *codes* and *Gödel numbers* are different things. For instance, the variable  $v_5$  has a code  $c_{v_5} = \langle 1, 5 \rangle = 2^2 \cdot 3^6$ . But the variable  $v_5$  considered as a term is also a sequence of symbols (of length 1). The *Gödel number*  $\#v_5\#$  of the term  $v_5$  is  $\langle c_{v_5} \rangle = 2^{c_{v_5}+1} = 2^{2^2 \cdot 3^6 + 1}$ .

explanation

**Example art.4.** Recall that if  $k_0, \dots, k_{n-1}$  is a sequence of numbers, then the code of the sequence  $\langle k_0, \dots, k_{n-1} \rangle$  in the power-of-primes coding is

$$2^{k_0+1} \cdot 3^{k_1+1} \cdot \dots \cdot p_{n-1}^{k_{n-1}},$$

where  $p_i$  is the  $i$ -th prime (starting with  $p_0 = 2$ ). So for instance, the formula  $v_0 = 0$ , or, more explicitly,  $=(v_0, c_0)$ , has the Gödel number

$$\langle c_=, c_{(}, c_{v_0}, c_{,}, c_{c_0}, c_{)} \rangle.$$

Here,  $c_=$  is  $\langle 0, 7 \rangle = 2^{0+1} \cdot 3^{7+1}$ ,  $c_{v_0}$  is  $\langle 1, 0 \rangle = 2^{1+1} \cdot 3^{0+1}$ , etc. So  $\#=(v_0, c_0)\#$  is

$$\begin{aligned} 2^{c_=+1} \cdot 3^{c_{(}+1} \cdot 5^{c_{v_0}+1} \cdot 7^{c_{,}+1} \cdot 11^{c_{c_0}+1} \cdot 13^{c_{)}+1} = \\ 2^{2^1 \cdot 3^8 + 1} \cdot 3^{2^1 \cdot 3^9 + 1} \cdot 5^{2^2 \cdot 3^1 + 1} \cdot 7^{2^1 \cdot 3^{11} + 1} \cdot 11^{2^3 \cdot 3^1 + 1} \cdot 13^{2^1 \cdot 3^{10} + 1} = \\ 2^{13\ 123} \cdot 3^{39\ 367} \cdot 5^{13} \cdot 7^{354\ 295} \cdot 11^{25} \cdot 13^{118\ 099}. \end{aligned}$$

## Photo Credits

## Bibliography