

1 Review of Gödel Numbering

In Lecture 21, we very briefly talked about the Gödel Numbering presented in the assigned readings. Some students found it somewhat unintuitive. There are many ways to design such a numbering scheme. Formally, a gödel numbering is function f from the type of formulae (say, in \mathcal{Q}) to the type natural numbers which satisfies the following three properties

- (1) f is computable
- (2) f is injective(one-to-one), i.e., different gödel numbers are assigned to different expressions
- (3) there exists a corresponding computable "inverse" function g , which takes a number as input and decides whether it is the gödel number of an expression and if so, also computes that expression.

We will present here an easy to understand gödel numbering given by Gödel. To begin with, assign a positive number to each symbol of the language. This can be done in many ways. For example, just use the unicode number for the character(add 1 if necessary to make sure that all assignments are positive). Using this, we can represent every formula by a list of numbers, such that i^{th} element of the list is the number assigned to i^{th} symbol of the formula. E.g., if symbols $\forall, x, y, +$ and $=$ were mapped to 1,2,3,4,5, respectively, the list representing $\forall x.\forall y.x + y = y + x$ is [1, 2, 1, 3, 2, 4, 3, 5, 3, 4, 2] So far, we have a function h which maps a formula to a member of the type N^+List , whose members are lists of positive numbers. Now, we will provide a injective function g from N^+List to N^+ .

$$g([n_1, n_2, \dots, n_m]) = \prod_{i=1}^m p_i^{n_i} \quad (1)$$

p_i denotes the i^{th} prime number(e.g $p_1 = 2, p_2 = 3$ and so on...). Note the following constructively provable facts.

- (1) There are infinitely many primes. It is possible to compute the i^{th} prime number. There are many proofs of this fact. We don't expect you to know any of these, but if you can't control your curiosity, you can find some at <http://primes.utm.edu/notes/proofs/infinite/>
- (2) **Every** number greater than 1 is either a prime, or can be **uniquely** factorized into a product of prime numbers. Since multiplication is symmetric, the order of factors is immaterial. This is also known as the fundamental theorem of arithmetic. Moreover, there is an algorithm to compute these factors
- (3) $g([]) = 1 \wedge \forall l : N^+List.(g(l) = 1) \Rightarrow (l = [])$

To obtain a gödel numbering, just compose the function g with h ($g \circ h$). It's easy to see that all the three properties are satisfied by this function.

Exercise Note that this gödel numbering works even if we had infinitely many symbols in the logic. Define a gödel numbering for the case when we have only finite number of symbols(say M). We will continue the proof of Undecidability of Q in the next lecture.

2 Natural Induction and Primitive Recursion

As mentioned in lecture 20,in Heyting Arithemetic(HA), we have the following Induction axiom schema:

$$(P(0) \wedge \forall x.(P(x) \implies P(x + 1))) \implies \forall xP(x) \quad (2)$$

Since HA is constructive, we need to provide evidence for this rule. So, we first add a new operator to the computation system, called *primrec*, with the following rules of evaluation.

$$\begin{aligned} \text{primrec}(0; \text{base}; \text{ind}) &\mapsto \text{base} \\ \text{primrec}(n + 1; \text{base}; \text{ind}) &\mapsto ((\text{ind})n)(\text{primrec}(n; \text{base}; \text{ind})) \end{aligned}$$

Note that the first argument to *primrec* must be a number. For example, the function *add* in type $N \rightarrow (N \rightarrow N)$ can be represented as $\lambda x.\lambda y.\text{primrec}(x; y; \lambda n.\lambda r.(r + 1))$ When applied to 1 and 2, we get the following execution.

$$\begin{aligned} &(\lambda x.\lambda y.\text{primrec}(x; y; \lambda n.\lambda r.(r + 1)))(1) (2) \\ &\mapsto (\lambda y.\text{primrec}(1; y; \lambda n.\lambda r.(r + 1)))2 \\ &\quad \mapsto \text{primrec}(1; 2; \lambda n.\lambda r.(r + 1)) \\ &\mapsto ((\lambda n.\lambda r.(r + 1))0)\text{primrec}(0; 2; \lambda n.\lambda r.(r + 1)) \\ &\quad \mapsto (\lambda r.(r + 1))\text{primrec}(0; 2; \lambda n.\lambda r.(r + 1)) \\ &\quad \quad \mapsto (\lambda r.(r + 1))2 \\ &\quad \quad \quad \mapsto (2 + 1) \\ &\quad \quad \quad \quad \mapsto 3 \end{aligned}$$

Now, we can express the evidence for natural induction (eq. 2) as:

$$\lambda p.\lambda n.\text{primrec}(n; p_1; p_2) \quad (3)$$