Notes on Gödel Numberings
September 29, 2005
Recursion Theory

Institute for Logic, Language and Computation
Universiteit van Amsterdam

These notes are based on [CT] and *Computability and Unsolvability* by Martin Davis.

1 Coding Example

Before studying Gödel coding of Turing machines, we show how to effectively code tuples of numbers. That is we define a function $f : \mathbb{N}^2 \to \mathbb{N}$ that is 1-1 and both f and f^{-1} are effectively computable. The function is

$$f(x, y) = \frac{1}{2}[(x + y)^2 + 3x + y]$$

Question 1.1: Prove for each x, y, $f(x, y) \in \mathbb{N}$. (**Hint:** first show that $(x + y)^2 + 3x + y = (x + y + 1)(x + y) + 2x)$

Question 1.2: Prove that f is primitive recursive.

We now show that f is 1-1. Make sure you understand each step of the proof below. Suppose that z, x and y are natural numbers such that

$$2z = (x + y)^2 + 3x + y \qquad (1)$$

Then

$$8z + 1 = (2x + 2y + 1)^2 + 8x$$

Question 1.3: Let $\lfloor x \rfloor$ be the largest integer $\leq x$, so $\lfloor \sqrt{8z + 1} \rfloor$ is the largest integer $\leq \sqrt{8z + 1}$. Prove that $\lfloor \sqrt{8z + 1} \rfloor$ is either $2x + 2y + 1$ or $2x + 2y + 2$.

Thus,

$$x + y = \lfloor (\lfloor \sqrt{8z + 1} + 1 \rfloor)/2 \rfloor - 1$$

And using (1),

$$3x + y = 2z - (\lfloor (\lfloor \sqrt{8z + 1} + 1 \rfloor)/2 \rfloor - 1)^2$$

Question 1.4: Using the above two equations, prove that for each z there is at most one pair (x, y) such that $f(x, y) = z$.

Quesiton 1.5: Given z describe a procedure for finding x and y. (Hint: let r be the largest integer such that $1+2+\cdots r \leq z$, then let $x = z-(1+2+\cdots+r)$ and $y = r - x$. Prove that $z = f(x, y)$.

Define two functions $g_1 : \mathbb{N} \to \mathbb{N}$ and $g_2\mathbb{N} \to \mathbb{N}$ as follows:

$$g_1(z) = \lfloor(\lfloor\sqrt{8z+1}\rfloor + 1)/2\rfloor \dot{-} 1$$

and

$$g_2(z) = 2z \dot{-} (r_l(z))^2$$

Question 1.6: Prove that g_1 and g_2 are recursive functions.

Finally define

$$l(z) = \lfloor(g_2(z) \dot{-} q_1(z)/2\rfloor$$

and

$$r(z) = g_1(z) \dot{-} \lfloor(g_2(z) \dot{-} g_1(z))/2\rfloor$$

It is easy to see that (given question 1.6), l and r are recursive functions. With a little more work (fill in the details) we have found recursive functions $f(x, y)$, $l(z)$ and $r(z)$ such that $f(l(z), r(z)) = z$, $l(f(x, y)) = x$ and $r(f(x, y)) = y$.

2 Coding Turing Machines

Let $\mathcal{T}$ be the set of all possible turing programs. A Gödel numbering is a function $gn : \mathcal{T} \to \mathbb{N}$ such that the following three properties hold

1. gn is effectively computable
2. gn^{-1} is effectively computable
3. If $gn(P_1) = gn(P_2)$ then $P_1 = P_2$

There are a number of functions that satisfy the above conditions. The function used in the book (pg. 62) can be described as follows:

- Define a function $gn_0 : \{L, R, q_0, S_0, q_1, S_1, q_2, S_2, \ldots\} \to \mathsf{Primes}$ as follows:

  - $gn_0(L) = 2$
  - $gn_0(R) = 3$
  - $gn_0(q_i) = p_{2+2i}$ (the $(2+2i)$th prime)
  - $gn_0(S_i) = p_{2+2i+1}$

- Let Q be a quadruple, $Q = q_i S A q_j$, then let

$$gn_1(Q) = 2^{gn_0(q_i)} \times 3^{gn_0(S)} \times 5^{gn_0(A)} \times 7^{gn_0(q_j)}$$

- Finally, let $P = Q_1 Q_2 \cdots Q_k$ be a Turing program, the Gödel number of P is:

$$gn(P) = 2^{gn_1(Q_1)} \times 3^{gn_1(Q_2)} \times \cdots \times p_k^{gn_1(Q_k)}$$

Question 2.1: Find the Gödel number of the Turing program that computes the zero function ($\mathbf{0}(n) = 0$ for all n).

Using the unique factorization theorem it is easy to convince yourself that the above function satisfies the three properties described above. That is, to compute $gn^{-1}(n)$ for some (large!) n, factor n into a product of primes, then factor the exponents into a product of primes. Finally, we can simply read off the exponents to find the program. In particular, note that this last step requires that gn_0^{-1} be effectively computable.

Question 2.2: Convince yourself, that given the above definitions gn_1 and gn, then gn_0 is the most sensible function. Given n, how do you determine if n is the Gödel number of a quadruple or the Gödel number of a program? What if we define[1] gn_0 such that $gn_0(q_1) = 4$ and $gn_0(q_2) = 16$?

We now show that determining whether x is a Gödel number is primitive recursive. First of all, we need some primitive recursive functions and predicates.

Question 2.3: Prove that the following functions and predicates are primitive recursive.

- $Pr(n)$ is a function that gives the nth prime number.
- $x \mid y$ is true iff x divides y

[1] This is not to say that such a definition *cannot* work, but rather that it will create a number of unnecessary problems.

- Let R be a primitive recursive predicate, $\mu_{x=0}^{n}(R(x))$ gives the smallest $x \leq n$ such that $R(x)$ is true (and is 0 if no such x exists).

Question 2.4: (Answers on the next page)

1. Find a primitive recursive function $Gl(n, x)$ returns the number associated with the nth symbol encoded by x. That is, if M is an expression (either a quadruple or a program) consisting of the symbols $\gamma_1, \ldots, \gamma_p$, then if $0 < n \leq p$, $Gl(n, x)$ is the number associated with γ_n, whereas if $n = 0$ or $n > p$, then $Gl(n, x) = 0$

2. Find a primitive recursive function $l(x)$ that gives the number of symbols encoded by x. That is if $x = gn(M)$ for some expression (either a quadruple or a program) $M = \gamma_1\gamma_2 \cdots \gamma_n$, then $l(x) = n$.

3. Find a primitive recursive predicate $G(x)$ which is true if and only if x is the Gödel number of some expression.

4. Find a primitive recursive predicate $Term(x, z)$ which is true if and only if z is the Gödel number of some expression and x is the Gödel number of a term in that expression. That is if $M = \gamma_1 \cdots \gamma_k$ and $z = gn(M)$ and x is the Gödel number of γ_i, then $Term(x, z)$ holds.

This gives you an idea of how to formally prove that the theory of Turing machines can be coded using Gödel numbers. Consult[2] *Computability and Unsolvability*, Ch. 4 by Martin Davis for more details.

[2]A number of other text books on Recursion Theory will also provide details of Gödel numbering.

Answers to question 2.4:

1. $Gl(n,x) = \mu^{n}_{y=0}[(Pr(n)^{y} \mid x) \wedge \neg(Pr(n)^{y+1}|x)]$
2. $l(x) = \mu^{x}_{y=0}[(Gl(y,x)x > 0) \wedge \bigwedge^{x}_{i=0}((Gl(y+i+1,x) = 0)]$
3. $GN(x) \leftrightarrow \neg \bigvee^{l(x)}_{y=1}[Gl(y,x) = 0) \wedge (Gl(y+1,x) \neq 0)]$
4. $Term(x,z) \leftrightarrow GN(z) \wedge \bigvee^{l(x)}_{n=0}[(x = Gl(n,z) \wedge (n \neq 0)]$