encoding words

Canonical name	EncodingWords
Date of creation	2013-03-22 19:06:09
Last modified on	2013-03-22 19:06:09
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	20
Author	CWoo (3771)
Entry type	Feature
Classification	msc 94A60
Classification	msc 68Q05
Classification	msc 68Q45
Classification	msc 03D20

Let Σ be an alphabet, and Σ^* the set of all words over Σ. An encoding of words over Σ is, loosely speaking, an assignment of words to numbers such that the numbers uniquely identify the words.

Definition. An encoding for a language $L \subseteq \Sigma^*$ is a one-to-one function $E : L \to \mathbb{N}$.

If L is finite, then it is easy to find an encoding for L. We are interested mainly in encoding infinite sets. By the definition above, L can not be encoded if it is uncountable. We can therefore limit the discussion to Σ that is at most countably infinite by listing some common methods of encoding L.

Among the methods liststed, Σ is an enumerated set $\{a_1, a_2, \ldots\}$. In the first method, Σ is assumed to be finite, and countably infinite in the last three. In addition, L is assumed to be Σ^* in the first three methods.

Method 1. First, set $E(a_i) := i$. In addition, for the empty word λ, we set $E(\lambda) := 0$. Next, inductively define $E(w)$ on the length of w. Suppose now that $E(w)$ has been defiend. Set

$$E(wa) := nE(w) + E(a), \text{ where } a \in \Sigma.$$

It is easy to see that if any non-empty word $w \in A$, with $w = b_1 \cdots b_m$, where $b_i \in \Sigma$. Then

$$E(w) = E(b_1)n^{m-1} + \cdots + E(b_m).$$

Then E is an encoding of L. This is really just the base-n digital representation of integers, where each a_i can be thought of as digits used by the representation. The only difference here is that 0 is not used as a "digit" (every letter gets mapped to a positive integer), except when the word is empty.

For example, let $\Sigma = \{0, 1, \ldots, 9\}$. Then the words $01, 001$, and 10 have code numbers $12, 112$, and 21.

It is easy to see that the encoding is one-to-one (see proof `http://planetmath.org/Uniqueness`

Method 2. Pick three positive number p, q, r such that p, q are coprime, with $p > 1$. Set $E(a_i) := p^i$ and $E(\lambda) := 1$. Inductively define $E(w)$ on length of w. Suppose now that $E(w)$ has been defined. Then

$$E(wa) := (rE(w) + q)E(a), \text{ where } a \in \Sigma.$$

For example, $E(a_2a_5a_3) = (r(rp^2 + q)p^5 + q)p^3 = r^2p^{10} + rqp^8 + qp^3$.

To see that E is injective, we make the following series of observations:

1. E is injective on Σ. In addition, either $E(a)|E(b)$ or $E(b)|E(a)$ for any $a, b \in \Sigma$.

2. $p|E(w)$ iff $w \neq \lambda$.

3. If $E(w) = E(a)$ for some $a \in \Sigma$, then $w = a$. First, note that $w \neq 1$, and if $w \in \Sigma$, then $w = a$. So suppose $w = vb$, with $b \in \Sigma$ and $v \neq \lambda$. Then $(rE(v) + q)E(b) = E(a)$. If $E(b)|E(a)$, then $rE(v) + q = p^i$. Since $E(v) > 1$, $i \neq 0$. But if $i > 0$, p and q would not be coprime as $p|E(v)$. On the other hand, if $E(a)|E(b)$, then $(E(v) + q)p^j = 1$, again impossible. So w must be a letter, and therefore is a.

4. Now, suppose $E(w) = E(v)$, and $E(a)|E(b)$, where a, b are the rightmost letters of w, v respectively. By the same argument as previously, $a = b$, so we may cancel the letters, leaving us with the equation $E(w') = E(v')$, where $w = w'a$ and $v = v'b$. Continue the process of canceling the last letters in pairs, we end up with $E(u) = E(c)$ for some letter $c \in \Sigma$. So $u = c$. This shows that $w = v$.

A variation of this method is to set $E(aw) := (rE(w) + q)E(a)$.

If we set $p = 2$ and $r = q = 1$, then the range of E is the set of all positive integers.

Method 3. The third method utilizes the uniqueness of prime decomposition of integers. First, define $f : \Sigma \to \mathbb{N}$ by $f(a_i) = i$. Then, for any $w = b_1 \cdots b_m$, with $b_i \in \Sigma$, define

$$E(w) := p_1^{f(b_1)} \cdots p_m^{f(b_m)} = \prod_{i=1}^{m} p_i^{f(b_i)},$$

where p_i is the i-th prime number (for example, $p_1 = 2$). We again set $E(\lambda) := 1$. By the fundamental theorem of arithmetic, and the fact that f is a bijection, E is injective (and maps onto the set of positive integers). This method is known as the multiplicative encoding of Gödel.

Method 4. Once an encoding E is found for Σ^*, an encoding for $L \subseteq \Sigma^*$ can be obtained by restricting the domain of E to L. Depending on how L is defined, other methods of encoding L via E are possible. We illustrate one example.

Let $L = L_1 \cup L_2\Sigma^*$, where L_1, L_2 are disjoint non-empty finite sets not containing the empty word. Encode L as follows: suppose $L_1 = \{v_1, \ldots, v_m\}$ and $L_2 = \{w_1, \ldots, w_n\}$. Define $E' : L \to \mathbb{N}$ such that:

1. $E'(v_i) := 10^{i-1}$ and $E'(w_j) := 10^{m+j-1}$, where $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$;

2. $E'(w) := E'(w_j)E(u)10^{m+n-1}$, where $w = w_j u$, and $\lambda \neq u \in \Sigma^*$.

Essentially, the first $m + n$ digits are reserved for encoding words v_i and w_j. E' is easily seen to be injective.

Method 5. Let L_2 be the language consisting of all words of length 2. Define $E_2 : L_2 \to \mathbb{N}$ by $E_2(a_i a_j) := J(i, j)$, where J is a pairing function that codes pairs of positive integers. Since J is an injection (actually maps onto the set of positive integers), so is E_2. Using J, one can encode the language L_3 of all length 3 words. Define $E_3 : L_2 \to \mathbb{N}$ by $E_3(a_i a_j a_k) := J(i, J(j, k))$. Again, E_3 is an injection. By induction, one can encode the language L_n of all words of length n, for any positive integer n.

Method 6. Let $L(n)$ be the language consisting of all words of length at most n. We can utilize Method 5 to code L. First, let $\Sigma_1 := \Sigma \cup \{a_0\}$, where a_0 is a letter not in Σ. Define $\phi : L(n) \to \Sigma_1^*$ by $\phi(w) := a_0^{n-|w|}w$, where $|w|$ is the length of w. Then $\phi(L) \subseteq L_n$, the language of all length n words over Σ_1^*. It is easy to see that ϕ is one-to-one. Then $E(n) := E_n \circ \phi$ is an encoding for L, where E_n is defined in Method 5 that encodes L_n, via the modified version of the pairing function $J'(i, j) := J(i + 1, j + 1)$, where $i, j \geq 0$.

Method 7. Can Method 5 be used to encode Σ^*? The answer is yes. However, a direct extension of E_n does not work. By this we mean that $E : \Sigma^* \to \mathbb{N}$, given by $E(w) = E_n(w)$ where $|w| = n$, though a function, is not injective. For any positive integer m, there is a word w_n of length n for every $n > 0$, such that $E_n(w_n) = m$. Instead, define E so that $E(w) := E_2(|w|, E_{|w|}(w))$ if $w \neq \lambda$, and $E(\lambda) := 0$. It is easy to see that E is injective, since both E_2 and E_n are (in fact, E is a bijection).

Remark. An encoding E for L can be thought of as a partial function from Σ^* to $\mathbb{N}$, whose domain is $L \subseteq \Sigma^*$. E is said to be *effective* if $E(L)$ is a recursive set. Equivalently, the partial function E^* on Σ^* given by

$$E^*(w) = \begin{cases} a_1^{E(w)} & \text{if } w \in L, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

is Turing-computable. An enumeration of Σ can be thought of as an encoding for Σ. If Σ is finite, any enumeration of Σ is effective. Assume that Σ is effectively enumerated, whether or not Σ is finite (so that a_1 in the definition of E^* can be effectively chosen). Then it is not hard to see that all of the encodings described above are effective. In fact, all of the sets $E(L)$ described are primitive recursive.