Problems and Languages

Encoding of a Set

Let S be any set, and let Σ be any alphabet. An *encoding* of S over Σ is defined to be a one-to-one function $S \to \Sigma^*$. Given such an encoding, the *language* of S is defined to be the set of all strings which are encodings of members of S.

When we say that a set S is *regular*, *recursive*, *polynomial time*, *etc.*, we always have a specific encoding of S in mind, and we mean that the language of S is *regular*, *recursive*, *polynomial time*, *etc.*.

For most applications, $S \subseteq U$, where U is some universal set which has a given encoding, which S inherits.

For example, the set of natural numbers can be encoded over $\Sigma = \{1\}$ by using *unary notation*, *i.e.*, the number n is encoded as the string 1^n. Then, for example, the set of multiples of 3 is regular, because the language $\{1^{3i}\}$ is a regular language.

Encoding of a Problem

A *problem* consists of a set of *instances*, where for each instance, there is a *solution*. Fix an encoding of both instances and solutions of a given problem. Then, the problem is a function from strings to strings, *i.e.*, if w is the encoding of an instance of the problem, $f(w)$ is the encoding of of the solution of that instance.

Consider, for example, addition of natural numbers. A problem instance is encoded as a non-empty string of digits, followed by the symbol $+$, followed by another non-empty string of digits, such as "$43+517$". The solution to a problem instance is a string of digits, in this case, "560".

Encoding of a 0-1 Problem

A *0-1 problem* is defined to be a problem where, for each instance, the solution is either true (1) or false (0). In this special case, the *language* of the problem is defined to be the set of encodings of problem instances for which the solution is true. In this case, we usuall identify the problem with its language.

For example, we can define a 0-1 addition problem to be strings of symbols which encode addition facts, such as "$43 + 517 = 560$".

Encodings of Numbers

An encoding of numbers is called a *numeration system*, and the encoding of one number is called a *numeral*.

Standard numeration systems for $\mathcal{N}$, the set of natural numbers, include the unary, binary, and decimal numeration systems, over the alphabets $\{1\}$, $\{0,1\}$, and $\{0,1,2,3,4,5,6,7,8,9\}$, respectively. We can encode the set of integers by using the additional symbol "$-$" for negative numbers.

Exercise 1 The knapsack problem with integer input is well-known to be $\mathcal{NP}$-complete if binary or decimal encoding is used for the inputs. Prove that the problem is polynomial time if unary encoding is used for the inputs.

Rational Numbers

Let $\mathcal{Q}$ be the set of rational numbers. We can encode $\mathcal{Q}$ by using *fractions*, where each rational number is written as the encodings of two integers separated by the symbol "/", such as "23/57". To ensure uniqueness, we insist that the denominator never be negative, and that the fraction is reduced to the lowest terms.

Real Numbers

Let $\Re$ be the set of real numbers. There is no encoding of $\Re$, since it is an uncountable set. However, there is a way to express every real number as a 0-1 problem.

Given a real number x, let $\mathcal{Q}_x \subseteq \mathcal{Q}$ be the set of all rational numbers less than or equal to x. Intuitively, if we "know" $\mathcal{Q}_x$, then we "know" x, since two real numbers x and y can be distinguished by a rational number that lies between them. We then identify the real number x with the set $\mathcal{Q}_x$, which means, with the language of all encodings of members of $\mathcal{Q}_x$.

Thus, for instance, if we say x is *recursive*, we mean that the language of all encodings of members of $\mathcal{Q}_x$ is recursive.

For example, every rational number can be considered to be a real number, and as such, is trivially recursive, since it is easy to decide whether a given fraction is less than or equal to another given fraction.

As another example, $\sqrt{2}$ is recursive, since the truth of the statement $n/m \leq \sqrt{2}$ can be easily decided.

The well known transcendental number π, the ratio of the circumference of a circle to its diameter, is also recursive. Hint: $\frac{\pi}{4} = \sum_{i=0}^{\infty}(-1)^i \frac{1}{2i+1}$.

Exercise 2 Prove that a real number x is recursive if and only if there is a machine that runs forever, writing the decimal digits of x onto an output tape.

Exercise 3 Let S be a set of natural numbers. Let $x_S \in \Re$ be defined by

$$x_S = \sum_{i \in S} 2^{-i}$$

Prove that x_S is recursive if and only if S is recursive.

Note: since not all sets of natural numbers are recursive, not all real numbers are recursive.

Exercise 4 Suppose that x is a real number, and that there exists a machine M that runs forever, printing a sequence of rational numbers that converges to x. Does this prove that x is recursive? Prove or give a counter-example.