

第 3 章

Introduction to computer science

3.2 The analysis of computational problems

3.2.3 Decision problems and the complexity classes P and NP

多くの計算問題はたいてい「はい」か「いいえ」で答えられる決定問題としてはっきり定式化されている。計算複雑性の中心的なアイデアは簡単にそして頻繁に決定問題の言葉へ言い換えられる。これには 2 つの理由がある。1 つはそこでの定理がとても簡単かつシンプルな形になること、もう 1 は歴史的には計算複雑性というのは決定問題についての研究から生じた概念であるからである。

決定問題は「形式言語」というところから話が始まる。

形式言語

アルファベット Σ 上の言語 L というのは Σ の元からなるすべての有限列 (これを語という) を含む集合 Σ^* の部分集合のことである。

言語の例として、アルファベットを $\Sigma = \{0, 1\}$ をとると、語全体の集合は

$$\Sigma^* = \{\emptyset, 0, 1, 00, 01, 10, 11, 000, \dots\}$$

というようになる。ここで重要なのは長さが 0 の空単語ということも含むことである。これは非決定性オートマトンといったものを考えるときに必要になる。この中の言語 L というのは

$$L = \{0, 10, 100, 110, \dots\}$$

というようになる。ちょうど自然言語もすべてのアルファベットを組み合わせた語を取り入れてないのと同じである。

もともと、決定問題というのはチューリングマシンにある入力をしてそれが「はい」か「いいえ」で回答させるものである。ただ、現代的には入力を語としてそれが言語に含まれているかどうかというのを考えるものになる。入力を語として受け取ったオートマトンがちゃんとした終状態をとるか (受理するか) どうかで考えられる。

ある決定問題が与えられたとき、この判別にかかる時間を知りたいという問題を考える。それは最速で判別するチューリングマシンはどのようなものかというように置き換えれる。

決定問題の分類

長さ n の入力された語 x が言語として受理するのに判別する時間が $O(f(n))$ かかるようなチューリングマシンが存在するとき、そのような問題を $TIME(f(n))$ に属するという。また、チューリングマシンが存在するということは判別するための言語というのがあるということを意味する。つまり、問題の集合 $TIME(f(n))$ の元は言語である。

正直この本文だけだと、判別するのに必要な時間というのがどのように決まるかというのは与えられていない。おそらく、オートマトンのステップ数で決まるような気はする。

この定義を使って決定問題を分類していこう

P 問題

$TIME(n^k)$ のように入力の話の長さに対し、多項式時間で決まる問題の集まり (複雑性クラス) を P 問題という。

P 問題に入ることと言うことは実際にそういったオートマトン等を作ればよいが、P 問題に入らないことを言うのは結構難しい。というのも、予想自体はオートマトンを試しにいろいろ作ってみて計算量のオーダー自体はつかめるが、思いつかないようなものがあるかもしれないということだ。存在命題なので否定するには背理法使って証明するしかないというのが厳しいのだろう。そのような問題で単純かつ面白い決定問題が、素数であるかどうかを判別する問題である。具体的に因数を求めるのは大変であるがいったん求めると、すぐに確認 (witness) できる。このように答えがあっているかすぐ確認できるかどうかという基準は複雑性クラスを分類するのに使える。

NP 問題

与えられた入力を表す語 x が言語 L にとって受理される場合、確認に使える語が存在する。も与えたときに多項式時間で判別できる問題を NP 問題という。もっと専門用語を使うと非決定性チューリングマシンを用いて多項式時間で判別できる問題を NP 問題という。

この定義を見ると、P 問題は NP 問題に含まれているのがわかる。しかしその逆というのは証明するのは大変難しい。
($P \neq NP$ 問題)

2 つ目の定義は非決定性チューリングマシンは本文では扱っていないので 1 つ目の定義のみになっている。ここでは確認という単語は入っていないが、オートマトンの構造として非決定性をいれるというのが確認する語に相当するものになっている。

本文では入力された語 x が受理されるかされないかで条件を分けてもある。これは非決定オートマトンにおける終状態とそうでない状態の非対称性に来ているとも言え換えられる。

つまり、NP 問題における入力された語 x を言語が受理するかどうかというのを入れ替えた問題というのも作ることができる。

coNP 問題

ある決定問題 S の補問題 がクラス NP に属する場合、 S はクラス co-NP に属するという。

これは先ほど挙げた素数の判別の例でいうと、ある数が合成数であるかどうかというのは coNP 問題となる。この例だけ見ると NP も coNP も同じ複雑性クラスになっているように見えるが、これを証明するのは $P \neq NP$ を証明するのと同等以上に難しい。

Excercise 3.18

もし $coNP \neq NP$ なら $P \neq NP$ を示せ。