Computability and Complexity

Problem Set 2

Computable functions and recursive sets

Y. Deville

C. Bertrand Van Ouytsel & V. Coppé & A. Gerniers & N. Golenvaux & M. Parmentier

February, 2021

1. Let $X \subseteq \mathbb{N}$ and $f(n)$ a total function defined by

$$f(n) = \#\{x \in X | x < n\}$$

where $\#A$ is the cardinal of A. Show that X is recursive iff f is computable.

2. True or false ?
   (a) If the domain of a function is finite, then the function is computable.
   (b) If a function is computable, then its domain is recursive.
   (c) A function whose table can be defined in a finite way is necessarily computable.

3. The Matiyasevich theorem is a famous theorem in number theory which gives an answer to Hilbert's tenth problem. The formulation of the theorem is quite simple :

   A subset of $\mathbb{N}$ is diophantine iff it is recursively enumerable.

   (Any subset of $\mathbb{N}$ is called diophantine iff it is of the form $E_D = \{a \in \mathbb{N} \mid \exists x_1, ..., x_k \in \mathbb{N} : D(a, x_1, ..., x_k) = 0\}$ where $D(a, x_1, ..., x_k)$ is a fixed polynomial with one parameter a and with k variables $x_1, ..., x_k$.)

   Prove half of the theorem : show that if a set X is diophantine, then it is recursively enumerable.
   Hint : write a program which proves that the set of perfect squares, which is a diophantine set ($\{a \in \mathbb{N} \mid \exists x \in \mathbb{N} : x^2 - a = 0\}$), is recursively enumerable but does not prove that it is recursive (even though it is). Then do the same for the set of sums of two perfect squares.

4. Show that all monotonically decreasing total functions $f : \mathbb{N} \to \mathbb{N}$ are computable.

   Challenge : Construct a monotonically increasing total function that is not computable. (Hint : use HALT)

1. • Supposons X récursif. Alors il existe P_x qui calcule la fonct° f.

$$\Rightarrow P_f(n) \equiv$$

```
result = 0
count = 0
while count < n:
    if Px(h) == 1:
        result = result + 1
count += 1
return result
```

• Supposons f calculable. Alors $\exists$ programme qui calcule f. Programme décide X ;

$$P_X(n) \equiv \text{return } P_f(n+1) - P_f(n)$$

2.

1) Vrai $\rightarrow$ voir slides TP2

$\rightarrow$ ne marche que pour des fcts où le domaine est fini

2) Faux, on peut montrer qu'une fct f est calculable avec K comme domaine (qu'on sait non récursif)

- Si n dans K, pgm retourne 1
- " n pas " K, programme ne se termine pas

$$P_f(n) \equiv \left[\begin{array}{l} P_n(n) \\ \text{return } 1 \end{array} \right.$$

3) Faux, contre-exemple : halt

3. Prouvons que l'ensemble des carrés parfaits est un ensemble récursivement énumérable.

⇒ Il existe un programme qui calcule les carrés parfaits :

$P_n \equiv$

```
x = 0
while True :
    if (x**2 - a) == 0 :
        print(1)
    else :
        x += 0
```

Faisons pareil avec la somme de 2 carrés parfaits ⟶

```
x = 0
while True :
    for y in range (x+1) :
        if (x**2 + y**2) - a == 0 :
            return (1)
    else :
        x += 1
```

⇒ La solut° sur les slides est une généralisat° de ce cas

4. Définit° fct décroissante :

$f : \mathbb{N} \to \mathbb{N}$ décroissante ssi :

$\forall x, y \in \mathbb{N} : x \leq y \Rightarrow f(x) \geq f(y)$

On peut dire que im f est finie