INSA Centre Val de Loire
Département STI

Année 2018-2019

Calculabilité et Complexité

TD 1 – Premières notions de calculabilité & Machines de Turing

Exercice 1 : Relations indécidables

a : Soit $(f_n)_{n\in\mathbb{N}}$ une suite de fonctions de $\mathbb{N}$ vers $\mathbb{N}$ telle que $(n, x) \mapsto f_n(x)$ est calculable par un algorithme.

Montrer que la suite (f_n) n'épuise pas toutes les fonctions calculables par un algorithme.

b : On considère n'importe quel langage de programmation (possédant des boucles). Soit Φ la fonction de $\mathbb{N}$ vers $\mathbb{N}$ telle que $\Phi(n)$ donne la longueur du plus petit programme permettant d'écrire n en base 10. Montrer que Φ n'est pas calculable par un algorithme.

On pourra faire un raisonnement par l'absurde.

Exercice 2 : Opérations arithmétiques

a : Donner une machine de Turing qui effectue l'addition de deux nombres codés en binaire avec 3 rubans. On choisira la forme du codage la plus adaptée.

b : Comment réaliser cette opération avec 2 rubans ?

c : Peut-on réaliser la même opération avec un seul ruban ?

d : En utilisant l'une des machines précédentes comme base, décrire fonctionnement d'une machine de Turing qui réalise la multiplication de 2 nombres codés en binaire.

Exercice 3 : Reconnaissance de langages

Soit $\Sigma = \{a, b, c\}$

a : Donner une machine de Turing qui reconnaît les mots sur Σ de la forme $a^n b^n c^n$.

b : Donner une machine de Turing qui reconnaît les mots sur Σ qui contiennent autant de a que de b et que de c :

$$L = \{x \in \Sigma^* |\ |x|_a = |x|_b = |x|_c\}$$