Quelques exercices sur la calculabilité Terminale NSI

Exercice 1 : Montrer qu'il n'existe pas de fonction Python prenant en entrée une fonction Python f, sans paramètre, et renvoyant True si et seulement si l'exécution f() de f s'arrête.

Exercice 2 : Définir une machine de Turing qui, partant de l'extrémité gauche de son entrée, parcourt cette entrée à la recherche du symbole 0.

Si le symbole est trouvé elle devra écrire 1 dans la case précédent l'entrée, et sinon elle devra y écrire 0.

On supposera que l'entrée est composée de symboles 0 et 1 et bordée de symboles blancs . .

Universalité de la calculabilité

Dès 1936, Turing établit que les fonctions pouvant être définies comme des fonctions $\lambda - calculables$ de Church et les fonctions pouvant être calculées par des machines de Turing sont exactement les mêmes. Autrement dit, les deux approches, si différentes soient-elles, décrivent la même notion de fonction calculable. Cette convergence entre les résultats obtenus par deux approches si radicalement différentes a vite étayé l'idée, baptisée thèse de Church-Turing, que l'on venait effectivement de réussir à capturer la vraie notion de fonction calculable.