LINGI1123 : Calculabilité, logique et complexité

Semaine 8 : Ch04 Modèles de calculabilité

Sections 4.4 : Machine de Turing

Situations - Problèmes

Yves Deville

2020-2021

Les situations – problèmes décrits ci-après doivent être lus avant de regarder les vidéos 5 à 6 du chapitre 4. Ce sont de questions à vous poser en visionnant ces vidéos. Les liens vers ces vidéos sont reprises sur le site Moodle du cours.

- Vidéo 5 : Machines de Turing
- Vidéo 6 : Machines de Turing extensions

Situations problèmes

1. Pourquoi en 1936, Alan Turing a-t-il inventé ces machines de Turing ?
2. Les MT sont un modèle complet de la calculabilité. Toute fonction calculable peut donc être calculée par une MT. Trier un tableau est un problème standard en algorithmique. Seriez-vous capable de construire une MT qui trie un tableau d'entiers ? Que peut-on en conclure ?
3. Que se passe-t-il si, à partir d'un input donné, l'exécution d'une MT s'arrête parce qu'il n'y a pas de transition prévue dans la MT pour le couple symbole-état courant ?
4. T-calculable est- ce équivalent à Java-calculable ?
5. A quoi sert le modèle des MT ?
6. A quoi servent toutes ces extensions des MT ?
7. Que faire si pour un input donné une MT non déterministe donne des résultats différents dans ses différentes exécutions ?

1. réponse à un problème mathématique

1. Pourquoi en 1936, Alan Turing a-t-il inventé ces machines de Turing ?

- On Computable Numbers, With An Application To The Entscheidungsproblem
- Objectif : réponse au problème de décidabilité de David Hilbert (1928) : "Entscheidungsproblem" existe-t-il un algorithme qui décide si une proposition énoncée dans un système logique est valide ou non ?
- Réponse négative
  - Définition des Machines de Turing
  - Démonstration que Problème de l'arrêt non calculable par une MT
  - Réduction du Entscheidungsproblem au problème de l'arrêt

2. théoriquement possible mais très difficile

On peut en conclure que les MT sont des modèles théoriques
=> pas fait pour programmer

3. input x → $f(x) = \perp$

4. Java : modèle complet ⇒ équivalent
↑ MT : " "

5. Machine de Turing → modèle simple (calculabilité)

- facilite démonstrat° calculabilité

=> permet d'illustrer ce qu'est un algorithme élém- essentiels

- a) complexité : simplifie démonstrat° (nbre instruct°)

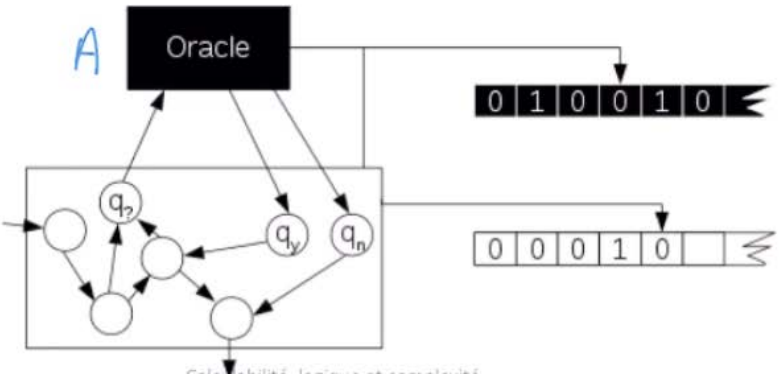
b. permet de simplifier les preuves :

exemple : construire MT universelle => quasi impossible avec MT simple

=> MT à plusieurs rubans, têtes, ...

Machine de Turing avec Oracle

- Que se passe-t-il si l'oracle est un ensemble récursif ? → même puissance qu'une MT
- Que se passe-t-il si l'oracle n'est pas un ensemble récursif ? → plus puissant qu'une MT



7. permet de déterminer si l'ensemble est NP-récursif