Décidabilité - EISTI - ING 2

Yannick Le Nir

Ecole Internationale des Sciences du Traitement de l'Information

yannick.lenir@eisti.fr

Origine du problème

Machines de Turing Universelles

- L'existence de MTU engendre l'apparition de problèmes indécidables :
- i.e. problèmes n'ayant pas d'algorithmes
- i.e. langages non-récursifs

Indécidabilité

- Apparition dès le début de l'informatique
- Analogie problèmes/langages et algorithmes/machines de Turing
- Précurseur de la théorie de la complexité

Halting Problème

Définition

Soit une machine de Turing M d'entrée x. Est-ce que M va s'arrêter sur x ?

Définition par les langages

$H = \{M; x : M(x) \neq \emptyset\}$

Diagonalisation

Preuve de l'indécidabilité de H

```
int zero()
{
  return 0;
}
```

```
int  loop()
{
   while true;
   return 0;
}
```

```
int weird()
{
  if (halt(weird())
    return loop();
  else return zero();
```

- Si `halt(weird())` retourne `true` alors weird() retourne `loop()` (contradiction)
- Si `halt(weird())` retourne `false` alors `weird()` retourne `zero()` (contradiction).
- Donc `halt` n'existe pas!!!

Propriétés

Proposition

H est récursivement énumérable

Théorème

H n'est pas récursif

Corrolaire

H est un problème indécidable.

Autres problèmes indécidables

Exemples

1. $\{M : M$ s'arrête sur toute entrée$\}$
2. $\{M; x :$ il existe un y tel que $M(x) = y\}$
3. $\{M; x :$ l'exécution de M sur x passe par tous les états de $M\}$
4. $\{M; x; y : \ M(x) = y\}$

La technique de preuve de ces propriétés consiste à toujours réduire le problème à celui de l'arrêt : si le problème est décidable, alors cela implique que H l'est aussi, d'où contradiction.

Conséquences effectives

Exemples de programmes

Comme conséquence de l'existence de problèmes indécidables, il s'avère dans le cas général impossible de déterminer si

- Si un programme boucle indéfiniement
- Si un test dans un programme sera vrai ou faux
- Si une suite d'instruction dans un programme sera toujours exécutée
- Si une variable sera toujours utilisée
- Si un programme est un virus

Heureusement,dans des cas particuliers, pour des algorithmes particuliers, ces problèmes peuvent devenir décidables.

Langages récursifs

Proposition

Si L est récursif alors $\bar{L}$ l'est aussi.

Preuve

- L est récursif

Langages récursifs

Proposition

Si L est récursif alors $\bar{L}$ l'est aussi.

Preuve

- L est récursif
- Donc il existe une machine M qui décide L

Langages récursifs

Proposition

Si L est récursif alors $\bar{L}$ l'est aussi.

Preuve

- L est récursif
- Donc il existe une machine M qui décide L
- Nous devons construire une machine $\bar{M}$ qui décide $\bar{L}$?

Langages récursifs

Proposition

Si L est récursif alors $\bar{L}$ l'est aussi.

Preuve

- L est récursif
- Donc il existe une machine M qui décide L
- Nous devons construire une machine $\bar{M}$ qui décide $\bar{L}$?
- Il suffit d'inverser les rôles de *yes* et *no* dans M.

Langages récursivements énumérables

Théorème

L est récursif ssi L et $\bar{L}$ sont récursivements énumérables.

Preuve

- L est récursif

Langages récursivements énumérables

Théorème

L est récursif ssi L et $\bar{L}$ sont récursivements énumérables.

Preuve

- L est récursif donc $\bar{L}$ aussi,

Langages récursivements énumérables

Théorème

L est récursif ssi L et $\bar{L}$ sont récursivements énumérables.

Preuve

- L est récursif donc $\bar{L}$ aussi,donc tous les deux récursivement énumérables.

Langages récursivements énumérables

Théorème

L est récursif ssi L et $\bar{L}$ sont récursivements énumérables.

Preuve

- L est récursif donc $\bar{L}$ aussi,donc tous les deux récursivement énumérables.
- Si L et $\bar{L}$ sont récursivement énumérables, ils sont acceptés par deux machines M et $\bar{M}$.

Langages récursivements énumérables

Théorème

L est récursif ssi L et $\bar{L}$ sont récursivements énumérables.

Preuve

- L est récursif donc $\bar{L}$ aussi,donc tous les deux récursivement énumérables.
- Si L et $\bar{L}$ sont récursivement énumérables, ils sont acceptés par deux machines M et $\bar{M}$.
- L est-il décidé par la machine M' ?

Langages récursivements énumérables

Théorème

L est récursif ssi L et $\bar{L}$ sont récursivements énumérables.

Preuve

- L est récursif donc $\bar{L}$ aussi,donc tous les deux récursivement énumérables.
- Si L et $\bar{L}$ sont récursivement énumérables, ils sont acceptés par deux machines M et $\bar{M}$.
- L est-il décidé par la machine M' ?
  - Pour une entrée x, M' simule sur deux rubans M et $\bar{M}$
  - Une étape de M puis une étape de $\bar{M}$ etc ...
  - Comme M accepte L, $\bar{M}$ accepte son complément et x est forcément accepté par l'une des deux.
  - L'une des deux machine va s'arrêter et accepter x.
  - Si M accepte, alors M' s'arrête sur *yes*, sinon M' s'arrête sur *no*.