

MPI: Calculabilité

Cours

- Problème décidable, indécidable
- Le problème de l'arrêt est indécidable
- Réduction calculable
- Problèmes semi-décidable, co-semi-décidable (HP).
- Fonction calculable (HP)

Question de Cours

- Rapeller le problème de l'arrêt. Montrer qu'il est indécidable.
- Donner un exemple de problème semi-décidable et montrer qu'il est semi-décidable. (HP)
- Donner un exemple de problème qui n'est pas semi-décidable et le prouver. (HP)

Indécidabilité

Montrer que les problèmes sont indécidables:

- **ε -Arret: Entrée:** Un programme P , **Sortie:** Est-ce que $P(\varepsilon)$ termine ?
- **42-Arret: Entrée:** Un programme P , **Sortie:** Est-ce que $P(42)$ termine ?
- **\neg -Arret: Entrée:** Un programme P et une entrée w , **Sortie:** Est-ce que $P(w)$ ne termine pas ?
- **Equivalence: Entrée:** Deux programmes P, Q prenant en entrée des naturels, **Sortie:** Est-ce que pour tout $n \in \mathbb{N}$, $P(n)$ termine ssi $Q(n)$ termine ?
- **Arret- \exists : Entrée:** Un programme P prenant en entrée un naturel, **Sortie:** Est-ce qu'il existe un n tel que $P(n)$ termine ?
- **Arret- \forall : Entrée:** Un programme P prenant en entrée un naturel, **Sortie:** Est-ce que pour tout $n \in \mathbb{N}$ on a $P(n)$ qui termine ?

ARRET Bool \rightarrow Bool¹

On considère la grammaire G suivante (en gras les symboles non terminaux) :

$$\begin{aligned} S &\rightarrow \text{bool f (bool x[])} \{ \mathbf{I} \text{ return false ;} \} \\ E &\rightarrow \text{true} \mid \text{false} \mid \mathbf{V} \mid \mathbf{E} == \mathbf{E} \\ V &\rightarrow \text{x[i]} \qquad \qquad \qquad \forall i \in \mathbb{N} \\ I &\rightarrow \text{return } \mathbf{E} \text{ ;} \mid \mathbf{B} \mid \mathbf{C} \\ B &\rightarrow \text{while (} \mathbf{E} \text{) } \{ \mathbf{I} \} \\ C &\rightarrow \text{if (} \mathbf{E} \text{) } \{ \mathbf{I} \} \text{ else } \{ \mathbf{I} \} \end{aligned}$$

On considère $L_{\text{bool}} = L(G)$ et $L_{\text{bool bool}}$ le langage engendré par G sans la dérivation B .

1. Est-ce que le problème suivant est décidable ?
 - **Entrée:** $\kappa \in L_{\text{bool bool}}$,
 - **Sortie:** Est-ce que $\kappa(x)$ termine (avec ou sans erreur) pour tout x ?
2. Soit η obtenue à partir de \mathbf{E} . Montrer qu'il existe une formule ψ_η tel que pour tout ν une valuation, $\psi_\eta(\nu)$ correspond à la valeur de sortie de η pour $x[i] = \nu(X_i)$
3. Donner une formule équivalente à ce code

```
if (x[0]) {
  if (x[0] == x[1]) {return false;}
  else {return true;}
} else {
  return {x[1] == x[2];}
}
```

4. Est-ce que le problème suivant est décidable ?
 - **Entrée:** $\kappa \in L_{\text{bool}}$,
 - **Sortie:** Est-ce que $\kappa(x)$ termine pour tout x ?
5. Est-ce que le problème suivant est décidable ?
 - **Entrée:** $\kappa \in L_{\text{bool}}$,
 - **Sortie:** Est-ce qu'il existe x tel que $\kappa(x)$ termine ?

¹Un mines-télécom bof de 2024

Entrée bornée

Montrer que le problème suivant est indécidable:

- **Entrée:** Une fonction calculable $P : \mathbb{N} \rightarrow \mathbb{N}$ qui termine toujours.
- **Sortie:** Est-ce qu'il existe un n tel que $P(n) = 0$?

L'indécidable est partout

Soit S un ensemble de string, on considère le problème P_S suivant:

- **Entrée:** s une chaîne de caractères
 - **Sortie:** Est-ce que $n \in S$?
1. Montrer qu'il existe des S tel que P_S est indécidable.
 2. Montrer que si C est infini, alors il existe $K \subseteq C$ tel que P_K est indécidable. Est-ce vrai si C est fini ?

Complexité et décidabilité²

Montrer que pour toute fonction calculable $g : \text{String} \rightarrow \mathbb{N}$, il existe une fonction calculable $f : \text{String} \rightarrow \text{char}$ si complexe que pour tout programme P qui calcule f , il existe une entrée $u \in \text{String}$ sur laquelle $P(u)$ prend plus de $g(u)$ étapes de calcul.

²tiré d'un oral d'Ulm 2021 https://a3nm.net/work/exams/ens/exercices_info_ulm_2021.pdf

Existence d'une quine

Donner un code OCaml non vide $w \in \text{ASCII}^*$ tel que l'exécution de ce code affiche le string w .

Réels calculables

1. Rappeler la représentation binaire des flottants en machine.

Les flottants machine ne peuvent pas représenter tout les nombres décimaux qui existe, et encore moins tout les réel. On introduit ici une autre manière de représenter plus de réels via des fonctions.

Un réel $x \in [0, 1[$ est dit *calculable au sens de Cauchy* s'il existe un programme $P : \mathbb{N} \rightarrow \mathbb{Q}$ tel que, pour tout $n \in \mathbb{N}$,

$$|P(n) - x| \leq 2^{-n}.$$

On dit alors que P est un *nom de Cauchy* de x . On note \mathbb{K} l'ensemble des réels calculables.

2. Montrer que tout rationnel $q \in [0, 1[$ est calculable.

3. Montrer que si $x, y \in \mathbb{K}$, alors $x + y$ (quand $x + y < 1$) et xy sont calculables.

Ind: On rappelle que $|ab - a'b'| \leq |ab - ab'| + |ab' - a'b'|$.

4. Montrer que le probleme de tester si deux nombres réels sont égaux est indécidable avec cette représentation. On pourra admettre que le problème suivant est indécidable:

- **Entrée:** Un programme $A : \mathbb{N} \rightarrow \text{bool}$ qui termine toujours
- **Sortie:** Est-ce qu'il existe un n tel que $A(n) = \text{true}$?

On dit qu'un problème de décision est *semi-décidable* s'il existe un algorithme A tel que :

- A répond VRAI sur toutes les instances positives
- A répond FAUX ou ne termine pas sur toutes les instances négatives.

5. Montrer que le problème suivant est semi-décidable :

- **Entrée:** $x, y \in [0, 1[$ représenté par leur nom de Cauchy.
- **Sortie** Est-ce que $x < y$?

6. Montrer qu'il existe des réels qui ne sont pas calculable.

On étudie maintenant les réels calculable sur tout \mathbb{R} et pas que $[0, 1[$. On garde la meme définition, mais pour $x \in \mathbb{R}$.

7. Montrer que \mathbb{K} est un \mathbb{Q} -espace vectoriel.

8. (*) Est-ce qu'il existe un algorithme $A : \mathbb{N} \rightarrow \mathbb{K}$ tel que $(A(n))_{n \in \mathbb{N}}$ forme une base de \mathbb{K} en tant que \mathbb{Q} -espace vectoriel?

Représentation d'ensembles infini

On se propose ici de créer une structure de données permettant de représenter un ensemble infini sous la forme d'une fonction de `'a -> bool` déterministe et qui **termine toujours**. On définit donc le type suivant en OCaml:

```
type 'a set = 'a -> bool;;
```

1. Donner un ensemble $A \subseteq \mathbb{N}$ que l'on ne pourra pas représenter avec notre structure. Le nombre d'ensembles non représentable est-t'il fini ? Dénombrable ? Indénombrable ?
2. Montrer que la fonction `val is_empty: int set -> bool` qui à un ensemble donné indique s'il est vide ou non n'est pas calculable.
3. Soit `val P: int set set`, montrer qu'il existe un $N \in \mathbb{N}$ tel que pour tout `val x: int set`, dans le calcul de `P x`, `P` n'évalue `x` que sur entrées inférieure à N . Est-ce vrai avec `val P: (int -> int) set` ?
4. (*) Montrer que la fonction `val is_empty: (int set set) -> int set` qui à un `int set` donné indique s'il est vide ou non est calculable.

