Cours

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

Question de Cours

- Rapeller le problème de l'arret. 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 → Bool[1]

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

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

On considère $L_{\text{bool}} = L(G)$ et $L_{\text{bool bool}}$ le langage engendrée 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 $\boldsymbol{E}$. Montrer qu'il existe une formule ψ_η tel que pour tout ν une valuation, $\psi_\eta(\nu)$ correspond à la valuer 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 ?

[1] Un mines-télécom bof de 2024

Entrée bornée

Montrer que le problème suivant est indécidable:

- **Entrée:** Une fonction calculable $P : \mathbb{N} \to \mathbb{N}$ 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 chaine 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 décidabilité[2]

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

---

[2] tiré 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 $w \in \mathrm{ASCII}^*$ tel que l'exécution de ce code affiche le `string` w.

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 set` donné indique s'il est vide ou non est calculable.