Calculabilité - Décidabilité (ICC)

Cours n°7

Stef Graillat

Sorbonne Université

Résumé du cours précédent (1/2)

- Forme normale de Chomsky : étant donnée une grammaire, on peut construire une autre grammaire générant le même langage (excepté ε) sous forme normale de Chomsky : il n'y a pas de symbole inutile, et le corps de chaque règle de production consiste soit en deux variables soit en un terminal
- Lemme de pompage : pour tout langage hors-contexte et pour tout mot suffisamment long, on peut trouver un sous-mot tel que les deux fins de ce mot peuvent être "itérées" en tandem
- Opérations préservant les langages hors-contexte : les langages hors-contexte sont clos par substitution, union, concaténation, cloture et renversement. Les langages hors-contexte ne sont pas clos par intersection mais l'intersection d'un langage hors-contexte et d'un langage régulier est encore hors-contexte

Résumé du cours précédent (2/2)

- Tester la vacuité d'un langage hors-contexte : étant donné une grammaire hors-contexte, il existe un algorithme pour tester si la grammaire génère au moins un mot non vide
- Tester l'appartenance à un langage hors-contexte : l'algorithme CYK permet de tester si un mot est généré par une grammaire hors-contexte. Le coût de l'algorithme est en $\mathcal{O}(n^3)$ si le mot à tester est de longueur n

Partie III :
Machines de Turing

Machines de Turing

Alan Mathison Turing (23 juin 1912 - 7 juin 1954) était un mathématicien britannique auteur de l'article fondateur de la science informatique « On Computable Numbers with an Application to the Entscheidungsproblem » qui allait donner le coup d'envoi à la création de l'ordinateur programmable. Il y présente sa machine de Turing, le premier calculateur universel programmable, et invente les concepts de programmation et de programme.

Quels problèmes peut-on résoudre avec un ordinateur ?

Modélisation d'un ordinateur $\rightsquigarrow$ Machine de Turing

Problèmes non résolubles par des ordinateurs

Écrire un programme testant si ce programme affiche `hello world`

```
int main(){
    printf("hello world \n");
}
```

Problèmes non résolubles par des ordinateurs

Écrire un programme testant si ce programme affiche `hello world`

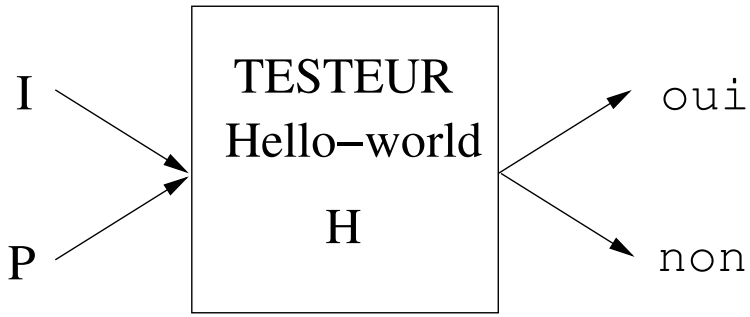
```
int exp(int i, int n)
  {
  int ans, j;
  ans=1;
  for(j=1; j<=n; j++) ans*=i;
  return(ans);
  }
```

```
int main(){
  int n, total=3,x,y,z;
  scanf("%d", &n);
  while(1){
    for(x=1; x<=total-2; x++){
      for(y=1; x<=total-x-1; y++){
       z=total-x-y;
       if (exp(x,n)+exp(y,n)==exp(z,n))
            printf("hello world");
        }
      }
     total++;
    }
  }
```

Étant donné n en entrée, le programme affiche `hello world` si l'équation $x^n + y^n = z^n$ a une solution avec x, y, z entiers.

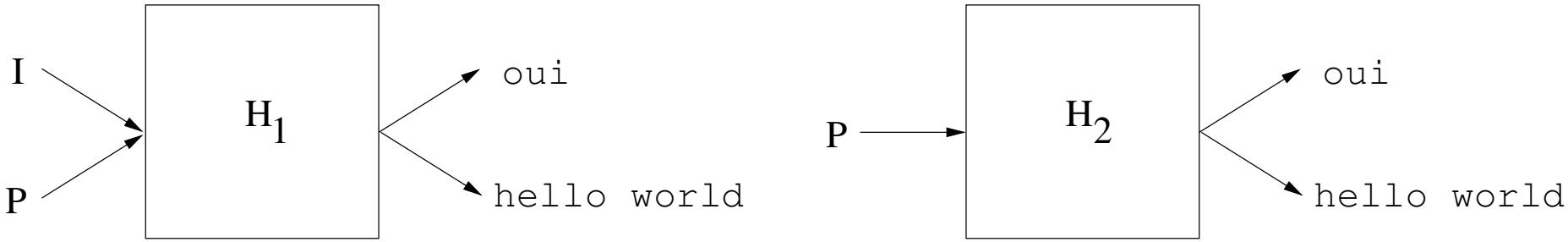
Un problème indécidable

Tester l'affichage de `hello world`



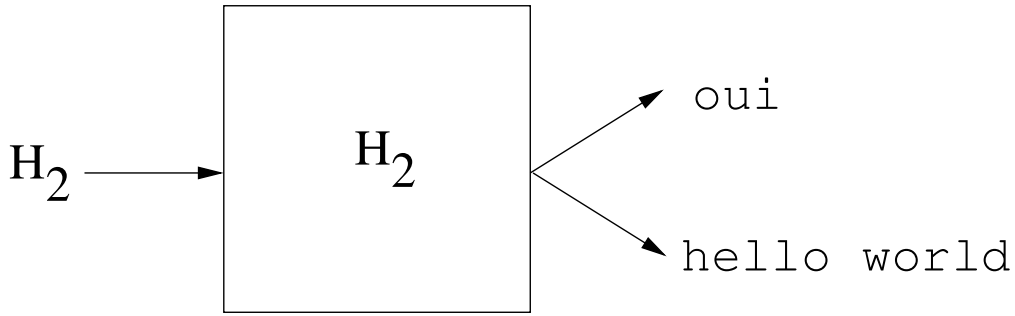
- Soit H un programme qui prend entrée un programme P et une entrée I et qui teste si P affiche `hello world` quand il prend en entrée I.
- Si oui, H affichera `oui` si non il affichera `non`.

Un problème indécidable (suite)



- On modifie H pour obtenir H_1 qui fait la même chose que H sauf qu'il affiche `hello world` à la place de `non`
- On remplace ensuite H_1 par un programme H_2 qui simule le comportement de H_1 avec comme entrée P et une copie de P.

Un problème indécidable (suite)



Que fait H_2 quand il a en entrée une copie de lui-même ?

- Si H_2 affiche `hello world` alors il affiche `oui` $\Rightarrow$ contradiction
- Si H_2 affiche `oui` alors il affiche `hello world` $\Rightarrow$ contradiction

Problèmes indécidables

Un problème est indécidable s'il n'existe pas de programme pour le résoudre

Ici, problème = décider si un mot appartient à un langage

Les langages sur un alphabet (fini) sont indénombrables

Les programmes (suite finie de symboles de l'alphabet) sont dénombrables

Par conséquent, il y a infiniment plus de langages que de programmes

Il doit exister des problèmes indécidables

Réduction de problème

Soit P_1 un problème indécidable et P_2 un nouveau problème.

Peut-on utiliser le fait que P_1 soit indécidable pour savoir si P_2 est aussi indécidable ?

Réduction de P_1 à P_2 : trouver un algorithme f tel que étant donnée une instance w de P_1, on peut la transformer algorithmiquement en une instance $f(w)$ de P_2 tel que $w \in P_1 \Leftrightarrow f(w) \in P_2$

Théorème 1

Si on peut réduire P_1 à P_2 alors P_1 indécidable implique que P_2 indécidable

Un peu d'histoire...

Hilbert (début XX-ième siècle) : existe-t-il un algorithme décidant si une formule logique du premier ordre avec quantificateurs sur les entiers est vraie ?

David Hilbert (23 janvier 1862 à Königsberg, Prusse-Orientale - 14 février 1943 à Göttingen, Allemagne) est un mathématicien allemand. Il est souvent considéré comme un des plus grands mathématiciens du XXe siècle, au même titre que Henri Poincaré. Il a créé ou développé un large éventail d'idées fondamentales, que ce soit la théorie des invariants, l'axiomatisation de la géométrie ou les fondements de l'analyse fonctionnelle (avec les espaces de Hilbert).

Un peu d'histoire...

1931 : théorème d'incomplétude de Gödel (une formule qui ne peut ni être prouvée ni être contre-dite une fois donnée une axiomatique) $\longrightarrow$ procédé de preuve similaire à celui de la contruction de H_2

Kurt Gödel (28 avril 1906 - 14 janvier 1978) est un mathématicien et logicien austro-américain. Son résultat le plus connu, le théorème d'incomplétude de Gödel, affirme que n'importe quel système logique suffisamment puissant pour décrire l'arithmétique des entiers admet des propositions sur les nombres entiers ne pouvant être ni infirmées ni confirmées à partir des axiomes de la théorie. Gödel a également démontré la complétude du calcul des prédicats du premier ordre.

Un peu d'histoire...

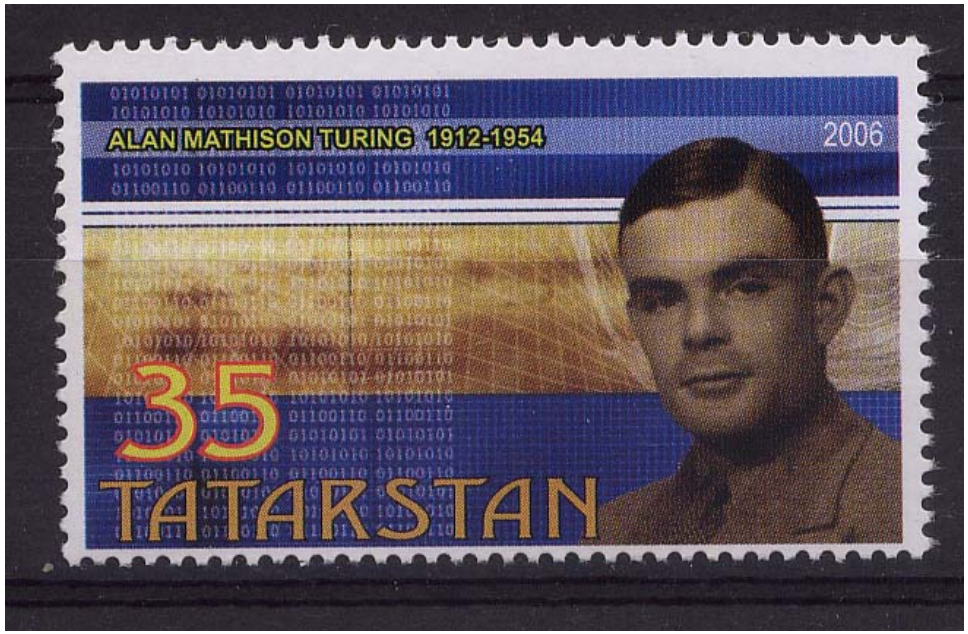
- 1936 : Machines de Turing, modèle calculatoire plutôt que déclaratif
- Thèse de Church : tous les modèles de calcul sont équivalents

Alonzo Church (14 juin 1903 - 11 août 1995) fut un mathématicien (logicien) américain à qui l'on doit certains des fondements de l'informatique théorique. Il est connu principalement pour le développement du lambda-calcul, son application à la notion de fonction récursive, pour la première démonstration de l'existence d'un problème indécidable.

Problèmes riches et fins : si on remplace dans les problèmes de Hilbert les entiers par des réels, alors les problèmes sont décidables !

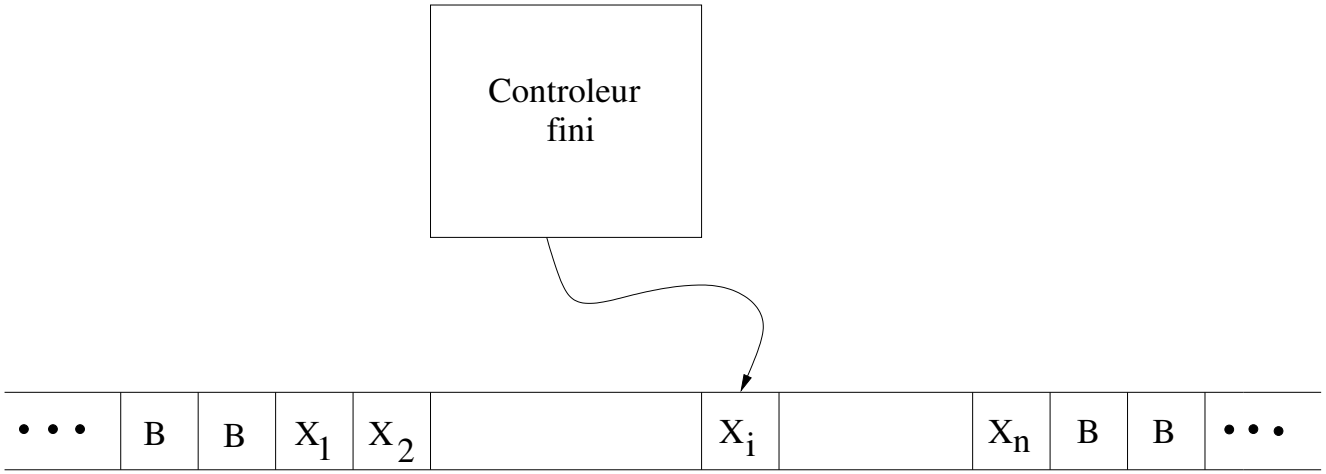
La machine de Turing

Objectif : pas seulement l'établissement d'énoncés d'indécidabilité !



On veut aussi mesurer la difficulté de résolution d'un problème. De plus, on veut se doter d'un formalisme un peu plus puissant pour prouver des propriétés d'indécidabilité (correspondance de Post).

La machine de Turing



L'entrée est placée sur le ruban et toutes les autres cellules du ruban contiennent le symbole blanc *B*. Initialement la tête de lecture est sur la cellule la plus à gauche de l'entrée

Un mouvement d'une machine de Turing (MT) est une fonction de l'état du contrôleur et du symbole lu sur le ruban

Dans un mouvement, la machine de Turing va

1. changer d'état
2. écrire un symbole sur le ruban
3. bouger la tête de lecture à gauche ou à droite

Définition formelle d'une machine de Turing

Définition 1

Une machine de Turing (MT) est un septuplet $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

- Q *un ensemble fini d'états du contrôleur*
- Σ *l'alphabet des symboles d'entrées*
- Γ *l'alphabet des symboles contenus dans le ruban, on a toujours* $\Sigma \subset \Gamma$
- δ *la fonction de transition qui prend en entrée* $q \in \Sigma$ *et* $X \in \Gamma$ *et renvoie* (p, Y, D) *où*
  - $p \in Q$ *est le nouvel état*
  - $Y \in \Gamma$ *qui remplace* X *dans le ruban*
  - D *est une direction (L ou R) indiquant si on lit ensuite le ruban à gauche ou à droite.*
- q_0 *est l'état de départ*
- B *le symbole blanc*
- F *l'ensemble des états finaux (ou acceptants).*

Descriptions instantannées

Représenter les cases visitées sur le ruban : chaine de caractère finie pour les cases visitées

Décrire l'état du contrôleur : l'état est indiqué à gauche de la case visitée

$$X_1 X_2 \cdots X_{i-1} q X_i X_{i+1} \cdots X_n$$

- q est l'état de la machine de Turing
- La case visitée est la i-ième et contient X_i
- $X_1 X_2 \cdots X_n$ est la portion du ruban située entre le symbole blanc le plus à gauche et le symbole blanc le plus à droite.

Descriptions instantannées (suite)

Si $\delta(q, X_i) = (p, Y, L)$ alors

$$X_1 X_2 \cdots X_{i-1} q X_i X_{i+1} \cdots X_n \vdash X_1 X_2 \cdots X_{i-2} p X_{i-1} Y X_{i+1} \cdots X_n$$

Exceptions si

- $i = 1$ alors $q X_1 X_2 \cdots X_n \vdash p B Y X_2 \cdots X_n$
- $i = n$ et $Y = B$ alors $X_1 X_2 \cdots X_{n-1} q X_n \vdash X_1 X_2 \cdots X_{n-2} p X_{n-1}$

Si $\delta(q, X_i) = (p, Y, R)$ alors

$$X_1 X_2 \cdots X_{i-1} q X_i X_{i+1} \cdots X_n \vdash X_1 X_2 \cdots X_{i-1} Y p X_{i+1} \cdots X_n$$

Exceptions si

- $i = n$ alors $q X_1 X_2 \cdots X_n \vdash X_1 X_2 \cdots X_n Y p B$
- $i = 1$ et $Y = B$ alors $q X_1 X_2 \cdots X_{n-1} X_n \vdash p X_2 \cdots X_{n-2} X_{n-1}$

Table de transition

Table de transition : table représentant l'action de δ ; les rangées correspondent aux états, les colonnes aux symboles du ruban.

L'état de départ est reconnu grâce à une flèche, les états acceptants grâce à une étoile.

Soit la MT $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, X, Y, B\}, \delta, q_0, B, \{q_4\})$ où δ est donné par la table de transition

	0	1	X	Y	B
$\rightarrow q_0$	(q_1, X, R)			(q_3, Y, R)	
q_1	$(q_1, 0, R)$	(q_2, Y, L)		(q_1, Y, R)	
q_2	$(q_2, 0, L)$		(q_0, X, R)	(q_2, Y, L)	
q_3				(q_3, Y, R)	(q_4, B, R)
$\star\ q_4$					

Diagramme de transition

Diagramme de transition

- Ensemble de nœuds correspondant aux états de la machine de Turing.
- Une arête entre l'état p et l'état q labellisée par X/YD signifie $\delta(q, X) = (p, Y, D)$. La direction D est noté $\leftarrow$ pour L et $\rightarrow$ pour R
- Une flèche vers l'état de départ.
- Les nœuds correspondant à des états acceptants sont entourés par un double cercle

Soit la MT $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, X, Y, B\}, \delta, q_0, B, \{q_4\})$