(A13)

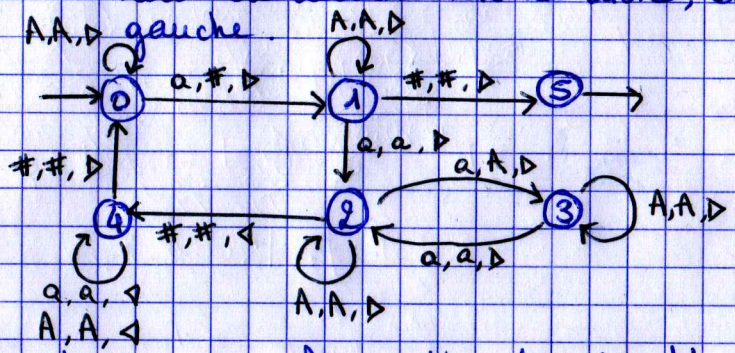
Leçon : Décidabilité et indécidabilité. Exemples

I - Préliminaires

Def: Une machine de Turing est un septuplet $(Q, \Sigma, \Gamma, E, q_0, F, \#)$ où :
- $Q = \{q_0, \dots, q_n\}$ ensemble fini d'états de contrôle
- Σ alphabet d'entrée, qui ne contient pas #
- Γ alphabet de bande, qui contient Σ et #
- E ensemble de transitions de la forme (p, a, q, b, x) avec $p, q \in Q$; $a, b \in \Gamma$, $x \in \{\triangleright, \triangleleft\}$; la transition pourra être notée $p, a \to q, b, x$
- $q_0 \in Q$ état initial
- $F \subseteq Q$ ensemble d'états finaux ou acceptants
- # symbole blanc.

Remarque :
- Les alphabets Σ et Γ sont finis
- Le symbole $\triangleright$ représente un déplacement de la tête de lecture vers la droite, et $\triangleleft$ vers la gauche.

Ex :



Représentation sous forme d'automate d'une machine de Turing, avec $\Sigma = \{a\}$, $\Gamma = \{a, A, \#\}$.
Une transition $p, a \to q, b, x$ étant représenté par un arc de p vers q étiqueté par (a, b, x).

Def: Une configuration d'une machine de Turing (MT) est un élément de $\Gamma'^* \times Q \times \Gamma'^*$

Cet élément représente le mot à gauche de la tête de lecture sur la bande, l'état de contrôle actuel et le mot à droite de la tête de lecture, en omettant l'infinité de symbole blanc #.

Ex :

tête de lecture
Etat q

Représentation de la bande pour une configuration $abA\#, q, aAA$

Def: Une configuration est initiale si elle est de la forme (ε, q_0, w) avec $w \in \Sigma^*$
Une configuration est finale si elle est de la forme (u, q, v) avec $q \in F$.

Def: Une étape de calcul d'une MT est une paire (C, C') de configurations, notée $C \to C'$ telle que :
- soit $C = (uc, pa, v)$ et $C' = (u, q, cbv)$ et $p, a \to q, b, \triangleleft \in E$
- soit $C = (u, p, av)$ et $C' = (ub, q, v)$ et $p, a \to q, b, \triangleright \in E$

Def: Un calcul est une suite de configurations successives $C_0 \to C_1 \to \dots \to C_k$.
Un calcul est acceptant si C_0 est initiale et C_k finale.

Ex : On reprend la machine de Turing de l'exemple précédent
- (ε, q_0, a) est une configuration initiale
- $(\varepsilon, q_0, a) \to (\#, q_1, \#)$ est une étape de calcul
- $(\varepsilon, q_0, a) \to (\#, q_1, \#) \to (\#\#, q_5, \#)$ est un calcul acceptant.

Def: Un mot $w \in \Sigma^*$ est accepté par une machine de Turing $\mathcal{M}$ s'il existe un calcul acceptant de configuration initiale (ε, q_0, w).

L'ensemble des mots acceptés par $\mathcal{M}$ est le langage accepté, noté $\mathcal{L}(\mathcal{M})$.

Ex : Dans la MT décrite précédemment, le langage reconnu est $\{a^{2^n} \mid n \in \mathbb{N}\}$

Def: Un langage $L \subseteq \Sigma^*$ est semi-décidable s'il existe une MT $\mathcal{M}$ telle que $L = \mathcal{L}(\mathcal{M})$

Un langage $L \subseteq \Sigma^*$ est décidable s'il existe une MT $\mathcal{M}$ sans calcul infini telle que $L = \mathcal{L}(\mathcal{M})$

Def: Un problème de décision est la donnée d'un ensemble E d'instances et d'un sous-ensemble $P \subseteq E$ des instances positives.

Ex: Problème de l'arrêt :
E : ensemble des couples machines, mots d'entrée
P : $\{(c\mathcal{M}, w) \mid \mathcal{M}$ s'arrête sur l'entrée $w\}$.

Def: Un codage est une fonction naturelle injective de E dans Σ^*, le codage de $x \in E$ est noté $\langle x \rangle$

Le langage d'un problème P est $L_P = \{\langle x \rangle \mid x \in P\}$

Def: On dit qu'un problème est décidable si le langage associé l'est.

Proposition: Le langage $L_\in = \{\langle c\mathcal{M}, w \rangle \mid w \in \mathcal{L}(\mathcal{M})\}$ est semi-décidable :
On peut simuler l'exécution d'une machine sur une entrée.

II - Techniques de preuves pour la décidabilité et l'indécidabilité

A - Décidabilité : Décrire une MT convenant.

Proposition: Si deux langages $L, L' \subseteq \Sigma^*$ sont décidables, alors les langages $L \cup L'$, $L \cap L'$ et $\Sigma^* \setminus L$ le sont aussi.

Idée de preuve : Construire des machines $\mathcal{M}_\cup$, $\mathcal{M}_\cap$ et $\mathcal{M}_{comp}$ à partir des machines $\mathcal{M}$ et $\mathcal{M}'$ décidant les langages L et L'.

Def: On dit qu'une théorie est décidable si le problème de savoir si une formule close est vraie est décidable.

Thm : L'arithmétique de Presburger (théorie au premier ordre des entiers munis de l'addition) est décidable

Preuve : Développement 1

Remarque : Si on ajoute la multiplication à la théorie précédente, elle devient indécidable.

B - Argument diagonal

Principe : Construire une machine D à partir d'une machine $\mathcal{M}$ supposée existante dont l'exécution sur un mot "diagonal" mène à une contradiction

Exemple : · Les MT comme les mots sur Σ^* sont dénombrables donc numérotables.
Proposition: Le langage $L = \{w_i \mid w_i \notin L(c\mathcal{M}_i)\}$ est indécidable, car non semi-décidable

Preuve : Supposons qu'il existe $\mathcal{M}$ reconnaissant L.
Il existe $j \in \mathbb{N}$ tel que $\mathcal{M} = \mathcal{M}_j$
$w_j \in L(c\mathcal{M}_j)$ ssi $w_j \notin L = L(c\mathcal{M}_j)$ absurde

Ex: Problème de l'arrêt indécidable

Def: Halt = $\{(M, w) \mid M$ s'arrête sur $w\}$

Preuve indécidabilité:
- Supposons M_H décidant Halt sur $\langle M, w\rangle$
- D sur $\langle M\rangle$: simule M_H sur $\langle M, M\rangle$:
  si accepte : boucle
  si rejette : accepte
- Execution de D sur $\langle D\rangle$ s'arrête-elle ?

Autre exemple: Le langage L_ϵ est indécidable.

Preuve: - Supposons M_ϵ décidant L_ϵ
- D sur $\langle M\rangle$ tel que : simule M_ϵ sur $\langle M, M\rangle$
  si accepte : rejette
  sinon : accepte
- Execution de D sur $\langle D\rangle$ absurde.

C- Réduction

Def: Une fonction $f: \Sigma^* \to \Gamma^*$ est calculable s'il existe une machine de Turing qui pour toute entrée w, s'arrête avec $f(w)$ sur la bande

Remarque: On omet l'infinité de symbole blanc de la bande.

Def: Soient A, B deux problèmes sur les alphabets Σ_A et Σ_B et de langages L_A et L_B.
Une réduction de A à B est une fonction f $f: \Sigma_A^* \to \Sigma_B^*$ calculable telle que $w \in L_A$ ssi $f(w) \in L_B$.

Si il existe une réduction de A à B, on note $A \leq_m B$.

Propriété:
- Si $A \leq_m B$ et si B décidable, alors A décidable
- Si $A \leq_m B$ et si A indécidable, alors B indécidable.

- Exemples d'application:

1. Proposition: Les langages $L_\emptyset = \{\langle M\rangle \mid L(M) \neq \emptyset\}$ et $L_{\neq} = \{\langle M, M'\rangle / L(M) \neq L(M')\}$ sont indécidables

Preuve: On a $L_\emptyset \leq_m L_{\neq}$ par la réduction $\langle M\rangle \to \langle M, M_\emptyset\rangle$ où $M_\emptyset$ est telle que $L(M_\emptyset) = \emptyset$.
On a $L_\epsilon \leq_m L_\emptyset$ par la réduction :
$\langle M, w\rangle \to$ (u → si $u = w$ alors simuler M sur u sinon rejeter)

2- Théorème de Rice: Pour toute propriété non triviale P sur les langages semi-décidables, le problème de savoir si une langage $L(M)$ vérifie P est indécidable.

Idée de preuve: On pose L vérifiant P (ou $\bar{P}$) et M_L une MT telle que $L = L(M_L) \neq \emptyset$. On fait une réduction de L_ϵ au problème considéré

Pour $\langle M, w\rangle$ on construit Q s'exécutant sur x tel que
Q: simule M sur w
si accepte : simule M_L sur x
sinon : rejette
alors L(Q) vérifie P ssi M s'accepte w.

3- Indécidabilité du problème de correspondance de Post (PCP)

Développement 2

Def:
- Une instance de PCP est une suite de mots $(u_1, v_1), \dots, (u_m, v_m)$ sur alphabet Σ
- Une solution est une suite d'indices $i_1, i_2, \dots, i_n$ de $[\![1; m]\!]$ telle que $u_{i_1} u_{i_2} \dots u_{i_n} = v_{i_1} v_{i_2} \dots v_{i_n}$
- Une instance est positive si elle admet au moins une solution

Application PCP: Indécidabilité de problèmes sur les grammaires par exemple.
- $L_G(S) \cap L_{G'}(S') = \emptyset$? pour deux grammaires HC
- Une grammaire hors contexte G est-elle ambiguë.