Calculabilité et Complexité: CM3

Florian Bridoux

Université de Caen

2021-2022

Table des matières

Table of Contents

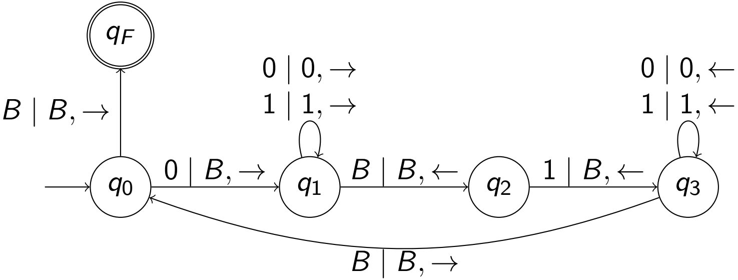
Propriétés de clôture

Theorem

Les propriétés suivantes sont vraies :

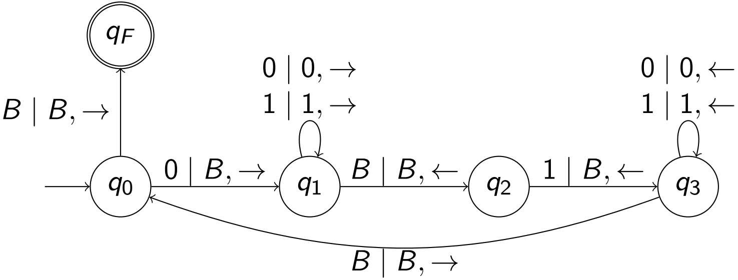
1. *la famille des langages décidables est close par complémentation ;*
2. *Les familles des langages décidables et semi-décidables sont closes par union et intersection ;*
3. *Un langage $L \subseteq \Sigma^*$ est décidable si et seulement si L est décidable et semi-décidable ($= \Sigma^* \setminus L$ est semi-décidables).*

Propriétés de clôture



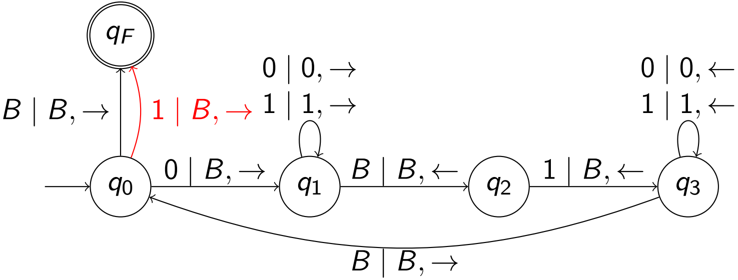
- $L(M) = \{0^n 1^n \mid n \in \mathbb{N}\}$.
- $L(M)$ est décidable: en un temps fini, on se retrouve dans l'état q_F et on accepte le mot ou on trouve une transition qui n'existe pas et on rejette.
- Mais comment décider le complémentaire de $L(M)$: $\{0, 1\}^* \setminus L(M)$?

Propriétés de clôture



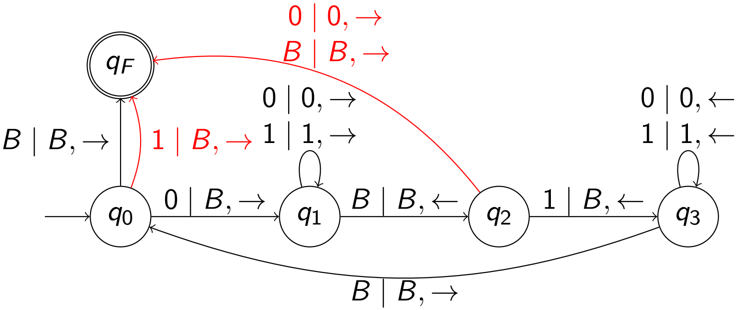
- Étape 1: On remplace les transitions qui n'existent pas par de nouvelles transition vers q_F: Si un mot était rejeté, il sera maintenant accepté (en un temps fini).

Propriétés de clôture



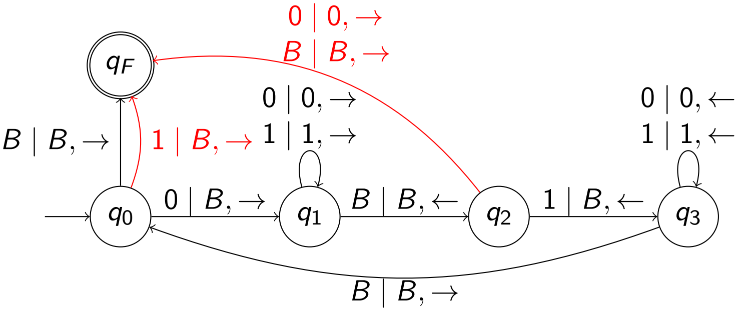
- Étape 1: On remplace les transitions qui n'existent pas par de nouvelles transition vers q_F: Si un mot était rejeté, il sera maintenant accepté (en un temps fini).

Propriétés de clôture



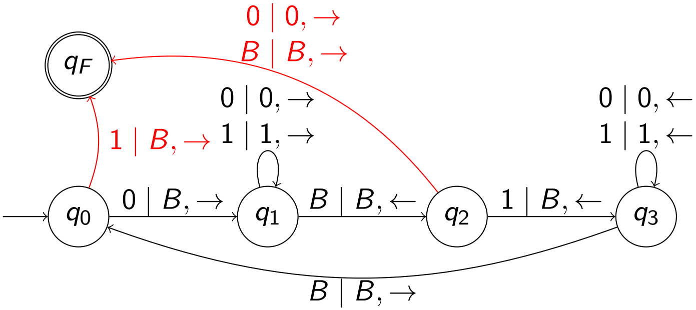
- Étape 1: On remplace les transitions qui n'existent pas par de nouvelles transition vers q_F: Si un mot était rejeté, il sera maintenant accepté (en un temps fini).

Propriétés de clôture



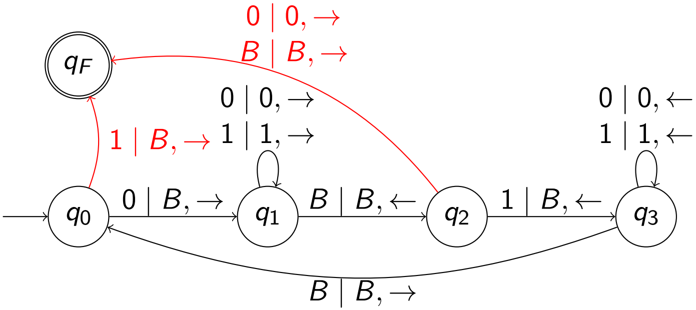
- Étape 1: On remplace les transitions qui n'existent pas par de nouvelles transition vers q_F: Si un mot était rejeté, il sera maintenant accepté (en un temps fini).
- Étape 2: On supprime les anciennes transition vers q_F: si un mot était accepté, il sera maintenant rejeté (en un temps fini).

Propriétés de clôture



- Étape 1: On remplace les transitions qui n'existent pas par de nouvelles transition vers q_F: Si un mot était rejeté, il sera maintenant accepté (en un temps fini).
- Étape 2: On supprime les anciennes transition vers q_F: si un mot était accepté, il sera maintenant rejeté (en un temps fini).

Propriétés de clôture



- Étape 1: On remplace les transitions qui n'existent pas par de nouvelles transition vers q_F: Si un mot était rejeté, il sera maintenant accepté (en un temps fini).
- Étape 2: On supprime les anciennes transition vers q_F: si un mot était accepté, il sera maintenant rejeté (en un temps fini).
- La nouvelle machine décide maintenant le complémentaire de $L(M)$: $\{0,1\}^* \setminus L(M)$: c'est un langage décidable.

Propriétés de clôture

Preuve

Les familles des langages décidables et semi-décidables sont closes par union et intersection.

Idée de preuve:

- On suppose qu'on a deux langages décidable/semi-décidables $L(M_1)$ et $L(M_2)$.
- On veut trouver une machine M_3 qui décide/semi-décide $L(M_1) \cup L(M_2)$ (*resp.* $L(M_1) \cap L(M_2)$).
- M_3 va simuler en parallèle M_1 et M_2 et va accepter ssi M_1 ou M_2 (*resp.* M_1 et M_2) acceptent.

Propriétés de clôture

Preuve

Un langage $L \subseteq \Sigma^*$ est décidable si et seulement si L est décidable et semi-décidable ($= \Sigma^* \setminus L$ est semi-décidables).

Le sens de gauche à droite est trivial.
Idée de preuve pour l'implication de droite à gauche:

- On a M_1 et M_2 qui semi-décident respectivement $L \subseteq \Sigma^*$ et $\Sigma^* \setminus L$.
- On veut trouver une machine M_3 qui décide L.
- M_3 va simuler en parallèle M_1 et M_2. Si M_1 dans un état acceptant alors L appartient au langage et M_3 accepte. Si M_2 s'arrête dans un état acceptant alors L appartient au langage $L \subseteq \Sigma^*$ M_3 rejette.
- Dans les deux cas, la machine M_3 s'arrête en un temps fini et L est décidable.

Table of Contents

Réduction

Theorem

Le langage $L_d = \{\langle M \rangle \mid M$ n'accepte pas le mot $\langle M \rangle\}$ n'est pas décidable.

- Par l'absurde, supposons qu'une machine M_d reconnaisse L_d. Considérons alors l'entrée $\langle M_d \rangle$ donnée à la machine M_d. Deux cas sont possibles.
  - Si M_d n'accepte pas $\langle M_d \rangle$ alors, par définition du langage L_d, le mot $\langle M_d \rangle$ est dans L_d. Or M_d reconnait L_d, donc M_d accepte $\langle M_d \rangle$, une contradiction.
  - Si M_d accepte $\langle M_d \rangle$ alors, par définition du langage L_d, le mot $\langle M_d \rangle$ n'est pas dans L_d. Or M_d reconnait L_d, donc M_d n'accepte pas $\langle M_d \rangle$, une contradiction.
- Dans les deux cas nous arrivons à une contradiction.
- Donc le langage L_d n'est pas décidable.

Réduction

Définition: Réduction Turing

Une **réduction (many-one) Turing** du langage A au langage B est une fonction calculable $f : \Sigma_A^* \to \Sigma_B^*$ telle que $w \in A \iff f(w) \in B$. On note $A \leq_m^T B$.

- Intuitivement, un problème A se réduit à un problème B si connaissant un algorithme pour décider/calculer B, on peut obtenir un algorithme pour décider/calculer A.
- A est alors plus facile (ou aussi facile) que B!

Remarque

many-one qualifie la fonction f (de plusieurs vers un, il faut comprendre par là que ni l'injectivité ni la surjectivité ne sont imposées). On écrit *one-to-one* quand on impose la bijectivité.

Réduction

Définition

On appelle *instance* de L un mot $w \in \Sigma_L^*$ dont on se demande s'il appartient au langage L.

Théorème

Si le langage A se réduit au langage B ($A \leq_m^T B$), alors:

- si A n'est pas décidable alors B non plus;
- si A n'est pas semi-décidable alors B non plus.
- si A n'est pas co-semi-décidable alors B non plus.

Preuve au TD3.

Remarque

Pour montrer qu'un langage B est indécidable, on choisit un langage A bien connu pour être indécidable, et l'on réduit A à B. En effet, 1) B décidable $\implies$ A décidable et 2) A indécidable implique que B est indécidable.

Réduction

Théorème

Le langage $L_u = \{\langle M\rangle\#w \mid M \text{ accepte le mot } w\}$ n'est pas décidable: son complément $L_{\bar{u}}$ n'est pas semi-décidable.

Preuve:

- Nous allons réduire L_d à $L_{\bar{u}}$ en décrivant une procédure algorithmique pour transformer les instances de L_d en des instances de $L_{\bar{u}}$. Soit w' une instance de L_d.
  1. la machine vérifie $w' \in L_{enc}$, si w' n'est pas un encodage valide alors on retourne l'instance $\langle M_{palindrome}\rangle\#abba$ (on a bien $w' \notin L_d$ et $\langle M_{palindrome}\rangle\#abba \notin L_{\bar{u}}$) ;
  2. si $w' = \langle M\rangle$ est un encodage valide, alors on retourne l'instance $w'\#w' = \langle M\rangle\#\langle M\rangle$ (on a bien $w' \in L_d$ si et seulement si $\langle M\rangle\#\langle M\rangle \in L_{\bar{u}}$).

  Cette réduction montre que si $L_{\bar{u}}$ est décidable alors L_d l'est également, or nous savons que L_d n'est pas décidable, donc $L_{\bar{u}}$ non plus, et par conséquent L_u n'est pas décidable.

Réduction

Théorème

Le langage $L_{halt\epsilon} = \{\langle M \rangle \mid M$ s'arrête quand on la lance sur l'entrée vide $\}$ n'est pas décidable.

- Prouvons que $L_u \leq_m^T L_{halt\epsilon}$. Etant donnée $\langle M \rangle \# w$ une instance de L_u, construisons l'instance $\langle M' \rangle$ suivante de $L_{halt\epsilon}$:
  - Si $\langle M \rangle$ n'est pas un encodage valide alors on retourne $\langle M \rangle$;
  - sinon on construit $\langle M' \rangle$ avec M' la machine qui commence par écrire w sur le ruban (cela est possible en utilisant $|w|$ états), puis entre dans l'état initial de M (M' va alors se comporter comme M), et nous rajoutons également à M' des transitions, depuis tous les états non finaux où M s'arrête (transition indéfinie), vers un état qui boucle à l'infini.
- Dans tous les cas, nous avons bien $\langle M \rangle \# w \in L_u \iff \langle M' \rangle \in L_{halt\epsilon}$, donc si $L_{halt\epsilon}$ est décidable alors L_u est décidable. Or, L_u n'est pas décidable, donc $L_{halt\epsilon}$ n'est pas décidable.