

Rapport : Transducteurs et machines de Moore et Mealy

30 novembre 2024

Table des matières

| | |
|---|----------|
| 1 Généralités sur les transducteurs finis | 1 |
| 1.1 Définition des transducteurs | 1 |
| 1.2 Relations rationnelles | 2 |
| 1.3 Equivalence entre transducteurs finis et relations rationnelles | 2 |
| 2 Machines de Moore | 2 |
| 2.1 Définition et exemple | 3 |

1 Généralités sur les transducteurs finis

Un transducteur fini est une généralisation des automates finis, implémentant non pas des langages mais plutôt des relations entre des ensembles de mots. Cela se fait à l'aide d'étiquettes supplémentaires sur chaque transition entre deux états.

1.1 Définition des transducteurs

Définition 1. Un transducteur fini est un 6-uplet $T = (\Sigma, \Gamma, Q, I, F, \delta)$ où

- Σ est l'alphabet d'entrée (fini)
- Γ l'alphabet de sortie (fini)
- Q l'ensemble des états (fini)
- $I \subset Q$ l'ensemble des états initiaux
- $F \subset Q$ l'ensemble des états finaux
- $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \longrightarrow \mathcal{P}(Q)$ est la fonction de transition non déterministe

On peut comme pour les automates définir $\delta^* : Q \times \Sigma^* \times \Gamma^* \longrightarrow \mathcal{P}(Q)$ par induction :

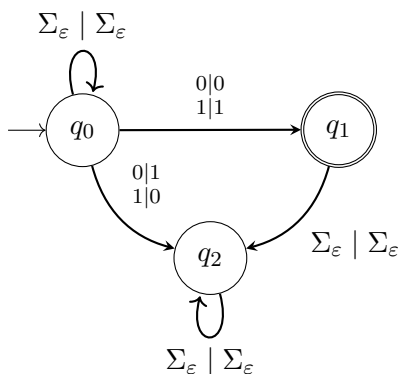
$$\delta^*_{|Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon} = \delta$$
$$\forall (a, b) \in \Sigma_\varepsilon \times \Gamma_\varepsilon, \forall (w, v) \in \Sigma^* \times \Gamma^*, \forall q \in Q, \delta^*(q, wa, vb) = \bigcup_{p \in \delta^*(q, w, v)} \delta(p, a, b)$$

On définit ici la relation reconnue par un transducteur fini, de manière similaire que les langages reconnus par les automates.

Définition 2. On définit $[T]$ la relation reconnue par un transducteur fini T , définie comme suit :

$$\forall (u, v) \in \Sigma^* \times \Gamma^*, u[T]v \Leftrightarrow \exists q \in I, \exists r \in F, r \in \delta^*(q, u, v)$$

FIGURE 1 – Un exemple de transducteur fini reconnaissant la relation de congruence modulo 2
Ici $\Sigma = \Gamma = \{0, 1\}$ et les transitions sont étiquetées par lettre d'entrée, lettre de sortie



1.2 Relations rationnelles

Soient Σ et Γ des alphabets finis. Nous allons ici étudier une classe particulière de relations entre Σ^* et Γ^* , que l'on peut voir comme des parties de $\Sigma^* \times \Gamma^*$, les relations rationnelles. Généralisons pour cela les opérations usuelles sur les langages

Définition 3. Soient $R, R' \subset \Sigma^* \times \Gamma^*$ des relations. On pose

- $R \cdot R' = \{(xx', yy') \mid (x, y) \in R, (x', y') \in R'\}$, la concaténation ou produit
- $R^* = \bigcup_{n \geq 0} R^n$ où R^n est la concaténation répétée, et $R^0 = \{(\varepsilon, \varepsilon)\}$

Définition 4. L'ensemble des relations rationnelles est défini comme le plus petit ensemble de langages stable par étoile, concaténation et union, et contenant \emptyset et les singletons.

Pour être plus général, une relation rationnelle est une partie rationnelle du monoïde produit $\Sigma^* \times \Gamma^*$. Comme pour les automates on a bien équivalence entre les relations reconnues par des transducteurs finis et les relations rationnelles, ce qui est le sujet de la prochaine section

1.3 Equivalence entre transducteurs finis et relations rationnelles

Théorème 1. Une relation R est rationnelle si et seulement si elle est reconnue par un transducteur fini T

Démonstration. La preuve est exactement du même acabi que celle pour les automates finis et langages rationnels □

Cette équivalence permet de voir plus simplement certaines propriétés des transducteurs finis ou des relations rationnelles. On obtient par exemple toutes les propriétés de clôture des automates finis en considérant un transducteur comme un automate sur l'alphabet $\Sigma \times \Gamma$. Il est plus simple d'obtenir la stabilité par composition et inverse par la vision en tant que relation.

Ajouter la preuve de composition qd mm, à trouver dans le livre de Sakarovitch + propriété des projections et transformation des réguliers en régulier

2 Machines de Moore

Une machine de Moore est un cas particulier d'un transducteur fini : d'abord une telle machine est déterministe et implémente donc non pas une relation rationnelle mais une fonction, qui sera appelée rationnelle. La contrainte supplémentaire sur ces machines est que la lettre de sortie lue ne dépend que de l'état actuel de l'automate.

2.1 Définition et exemple

On peut évidemment définir une machine de Moore en partant de la définition de transducteur et en posant des contraintes sur la fonction de transition. Il est toutefois plus commode de la définir ainsi

Définition 5. Une machine de Moore \mathcal{M} est un 7-uplet $(\Sigma, \Gamma, Q, q_0, F, \delta, \lambda)$ où

- Σ est l'alphabet d'entrée
- Γ l'alphabet de sortie
- Q l'ensemble fini des états
- q_0 l'état initial
- $F \subset Q$ les états finaux
- $\delta : Q \times \Sigma \rightarrow Q$ la fonction de transition déterministe
- $\lambda : Q \rightarrow \Gamma$ la fonction de sortie

Un calcul de \mathcal{M} sur un mot w est défini de la même manière que pour un automate déterministe. On ajoute toutefois la notion de mot produit par ce calcul, qui est le mot obtenu en concaténant les lettres obtenues par application de λ sur chaque état pris lors du calcul de \mathcal{M} sur w . Plus formellement

Définition 6. On définit pour $q \in Q$ et $w \in \Sigma^*$ $q \star w$ récursivement :

$$\forall a \in \Sigma, q \star a = \lambda(\delta(q, a)) \text{ et } \forall u \in \Sigma^*, q \star ua = (q \star u)(\delta^*(q, u) \star a)$$

Le mot produit par le calcul de \mathcal{M} sur w est défini comme $q_0 \star w$. Notons $L \subset \Sigma^*$ le langage reconnu par l'automate déterministe contenu dans la définition de \mathcal{M} . La fonction rationnelle réalisée par \mathcal{M} est alors définie par

$$f : \begin{array}{ll} L & \longrightarrow \Gamma^* \\ w & \longmapsto q_0 \star w \end{array}$$

On peut généraliser la fonction réalisée par \mathcal{M} en la définissant non pas sur L mais sur tout Σ^* . Cela revient à rendre tout les états acceptants (ça reste une fonction rationnelle ???)

FIGURE 2 – Un exemple de machine de Moore implémentant le xor chiffre à chiffre entre deux entiers en binaires

Si l'on souhaite faire le xor de 101 et 100, on donne à notre machine l'entrée 110010 et cette dernière renvoie 000001 (un 0 puis le résultat du xor pour chaque chiffre). On ne précise $\lambda(q_0)$ car cette valeur n'est pas utilisée pour définir la fonction de cette machine

