Transducteurs

Denise Maurice

13 février 2007

Introduction *Un transducteur est une notion assez intuitive quand on manipule des automates : au fur et à mesure qu'il lit un mot sur l'entrée, le transducteur écrit un mot sur une sortie. Les transductions sont alors les fonctions mathématiques ou les relations correspondant à ces transducteurs. En fait, il en existe une définition indépendante, et on peut montrer ensuite que ce sont les fonctions ou relations associées à ces fameux transducteurs. Ici, pour simplifier, je prends la définiton directement à partir des transducteurs.*

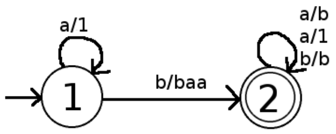
Table des matières

1 Transducteurs généraux

1.1 Définitions

Définition Un *transducteur* est la donnée d'un sextuplet (A, B, Q, i, F, E), où A est un alphabet, dit *d'entrée*, B un alphabet dit *de sortie*, Q est l'ensemble (fini) des états, $i \in Q$ est l'état initial, $F \subset Q$ est l'ensemble des états finaux, et E est l'ensemble des transitions :
$E \subset Q \times A^* \times B^* \times Q$

Comme les automates "normaux", on peut les représenter par un schéma : des cercles pour les états, une flèche entrante pour l'état initial, un double cercle pour les états finaux, et sur chaque transition, on écrit "a/bc" pour dire qu'on lit a et qu'on écrit bc.
Exemple :



Définitions Si $T = (A, B, Q, i, F, E)$ est un transducteur, alors :
- Si $e \in E^*, e = (p_1, u_1, v_1, q_1)...(p_n, u_n, v_n, q_n)$, on notera $|e| = (u, v)$ définis par $u = u_1...u_n$, $v = v_1...v_n$, avec la convention $|\varepsilon| = (\varepsilon, \varepsilon)$.
- e est appelé un *chemin* dans T de p_1 à q_n ssi $\forall i \in \{1, ..., n-1\}, q_i = p_{i+1}$.
- $\Lambda(p, q)$ est l'ensemble des chemins allant de p à q, avec la convention $\varepsilon \in \Lambda(p, p)$ pour tout état p.
  Par extension : $\Lambda(p, Q') = \bigcup \Lambda(p, q)$, pour $q \in Q' \subset Q$.
- $T(p, q) = \{|e|, e \in \Lambda(p, q)\}$ et $T(p, Q') = \{|e|, e \in \Lambda(p, Q')\}$ pour $Q' \subset Q$: ensemble des "entrées" et "sorties" pour aller d'un état à un autre.
- Un chemin e de p à q est dit *réussi* ssi $p = i, q \in F$. L'ensemble des entrées et sorties des chemins réussis est $\Lambda(i, F)$.

Définition Une *transduction* τ de A^* vers B^*, où A et B sont deux alphabets est une fonction de A^* vers l'ensemble des parties de B^*. On écrira $\tau : A^* \longrightarrow B^*$ pour simplifier.

C'est ni plus ni moins qu'une relation, sauf qu'au lieu de la voir du point de vue des couples de mots, on la voit du point de vue "fonctionnel" : à un mot on associe tous ceux qui lui sont reliés.

Définition Si T est un transducteur, la transduction réalisée par T, notée $|T|$, est définie par :
$|T|(u) = \{v \in B^*|(u,v) \in T(i,F)\}$

Intuitivement, $|T|(u) = \{$ *mots pouvant être écrits par T en lisant u* $\}$*. Ça correspond bien à l'idée naturelle qu'on peut avoir de la relation associée à un transducteur : un mot u est relié à un mot v ssi il existe un chemin allant de l'état initial à un état final, en lisant u et en écrivant v.*

Définition Une transduction est dite *rationnelle* ssi elle est réalisée par un transducteur.

Exemple Si on reprend l'exemple plus haut, alors
$|T|(a^n) = \emptyset$ pour $n \in \mathbb{N}$
$|T|(a^n bu) = \{baab^k, |u|_b \leq k \leq |u|\}$

1.2 Premières propriétés

On a un équivalent des automates normalisés pour les transducteurs :

Propriété Toute transduction rationnelle peut être réalisée par un transducteur $T = (A, B, Q, i, F, E)$ tel que $E \subset Q \times (A \cup \{\varepsilon\}) \times (B \cup \{\varepsilon\}) \times Q$ (les transitions sont étiquetées, en lecture comme en écriture, seulement par une lettre ou le mot vide), $F = \{q_f\}$ (un seul état final), et enfin $(p, u, v, q) \in E \Rightarrow p \neq q_f$ et $q \neq i$ (aucune transition ne part de q_f, aucune n'arrive sur i).

Preuve Si $T = (A, B, Q, i, F, E)$ est un transducteur réalisant τ, alors :
On remplace chaque transition $(p, u_1...u_n, v_1...v_m, q)$ par les transitions de la forme $(p, u_1, v_1, p_1), (p_1, u_2, v_2, p_2)$...*etc.* Au besoin, on complète avec des $(p_{k-1}, u_k, \varepsilon, p_k)$ ou des $(p_{k-1}, \varepsilon, v_k, p_k)$.
Ensuite, on ajoute deux états : q_0 (nouvel état initial) et q_1 (nouvel état final), et les transitions :
- (q_0, u, v, q') pour $(i, u, v, q') \in E$
- (q, u, v, q_1) pour $(q, u, v, q') \in E$ et $q' \in F$
- $(q_0, \varepsilon, \varepsilon, q_1)$ si $i \in F$

Remarque *Si en plus on sait que* $\tau(\varepsilon) = \emptyset$*, alors on peut prendre uniquement des transitions du type* (p, u, ε, q) *ou* (p, ε, v, q)*, avec u et v des lettres.*

2 Transductions séquentielles

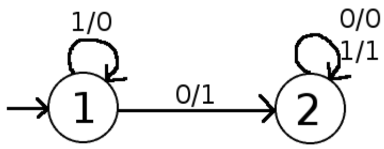
2.1 Définitions

Les automates décrits plus haut ne sont pas faciles à utiliser en pratique : non seulement ils peut y avoir plusieurs chemins possibles pour un mot d'entrée donné, mais en plus il peut y avoir plusieurs chemins acceptants, et donc plusieurs mots en retour. On va voir ici une version déterministe des transducteurs.

Définition Un *transducteur séquentiel à gauche* (ou transducteur séquentiel tout court) L est la donnée d'alphabets d'entrée et de sortie A et B, d'un ensemble fini d'états Q, d'un état initial $i \in Q$ et de deux fonctions partielles :
$\delta : Q \times A \longrightarrow A$, donnant l'état suivant, et
$\lambda : Q \times A \longrightarrow B$, donnant le mot à écrire pour cette transition.

Ces fonctions sont étendues aux mots, c'est à dire :
$\delta(q, \varepsilon) = q$ et $\lambda(q, \varepsilon) = \varepsilon$
si $u \in A^*$ et $x \in A$,
$\delta(q, ux) = \delta(\delta(q, u), x)$ et $\lambda(q, ux) = \lambda(u, x).\lambda(\delta(q, u), x)$

Exemple Ce transducteur effectue l'opération "+1" sur un nombre u écrit en binaire, modulo $2^{|u|}$, les bits de poids faible étant à gauche (en premier).



L'état 1 correspond à la "retenue" : il faut encore faire +1 sur le nombre restant. Donc tant qu'on voit des 1, on écrit 0 et on retient 1 (ie on reste dans l'état 1). Dès qu'on voit un 0, on peut écrire 1 et on a fini l'opération : on passe en état 2 (où on patiente).

Définition Si α est une fonction partielle, α est une transduction séquentielle (à gauche) si $\alpha = |L|$ pour L un transducteur séquentiel.

2.2 Propriétés

Définition On dit d'une fonction partielle α qu'elle conserve les préfixes si $\alpha(\varepsilon) = \varepsilon$ et si $\alpha(uv)$ est défini, alors $\alpha(uv) \in \alpha(u)B^*$

Propriétés des transducteurs séquentiels Si α est une transduction séquentielle, alors
$\alpha(\varepsilon) = \varepsilon$
$\alpha(uv) = \alpha(u).\lambda(\delta(i,u),v)$, ce qui implique entre autres qu'elle conserve les préfixes.

Preuve

On va montrer par récurrence sur la taille de v que $\forall u, v \in A^*, \forall p, q \in Q$
$\delta(q, uv) = \delta(\delta(q, u), v)$
$\lambda(q, uv) = \lambda(q, u).\lambda(\delta(q, u), v)$

Pour $v = \varepsilon$, rien à démontrer. Si $v = xw, x \in A$. Soit $u \in A^*, p, q \in Q$
$\delta(q, uv) = \delta(q, (ux).w)$
$= \delta(\delta(q, ux), w)$ par hypothèse de récurrence ($|w| < |v|$)
$= \delta(\delta(\delta(q, u), x), w)$ par définition de l'extension de δ aux mots
$= \delta(\delta(q, u), xw)$ par hypothèse de récurrence.

$\lambda(q, uxw) = \lambda(q, ux).\lambda(\delta(q, ux), w)$ par hypothèse de récurrence
$= \lambda(q, u).\lambda(\delta(q, u), x).\lambda(\delta(\delta(q, u), x), w)$
On pose $\delta(q, u) = p$ pour simplifier :
$= \lambda(q, u).\lambda(p, x).\lambda(\delta(p, x), w)$
$= \lambda(q, u).\lambda(p, xw)$ Car $|w| < |v|$
$= \lambda(q, u).\lambda(\delta(q, u), xw)$

Exemple 1 Dans l'exemple cité plus haut, α conserve bien les préfixes.
Par contre, si on pose β la fonction "+1" sur les chiffres en binaire en partant des bits de poids fort, cette propriété nous montre que β n'est pas une transduction séquentielle.
En effet $\beta(00) = 01$, et $\beta(0) = 1$. On voit que $\beta(00)$ est défini, mais ne commence pas par 1.

Exemple 2 Tous les morphismes sont des transductions séquentielles : il suffit de prendre un seul état, et des transitions étiquetées par "$a/\mu(a)$"

Exemple 3 Sur les alphabets $A = \{x\}$ et $B = \{a, b\}$, on définit τ par :
$\tau(x^n) = a^n$ si n est pair, b^n sinon.
De la même façon, vu que $\tau(x^n+1)$ est composé de lettres différentes de $\tau(x^n)$, cette fonction ne conserve pas les préfixes, et n'est donc pas une transduction séquentielle.

De manière générale, ça correspond à une idée intuitive qu'on peut avoir des transducteurs si on les met en parallèle avec les automates déterministes ou pas : il faudrait prévoir la fin (la parité du nombre de lettres dans le 3, ou si il va y avoir ou non retenue dans le cas du 1) avant de pouvoir commencer à écrire. Alors que dans les cas qui "marchent", on peut écrire au fur et à mesure.

Remarque *On voit que dans le cas de la fonction "+1", on n'a pu le faire que modulo quelque chose, parce qu'il était difficile de prévoir qu'on arrivait au bout du mot. Et c'est le cas assez souvent : on va donc se donner droit à un outil supplémentaire, les transductions subséquentielles.*

2.3 Les transductions subséquentielles : gérer les "effets de bord"

Définition Une transduction subséquentielle (à gauche) S est la donnée d'un transducteur séquentiel et d'une fonction partielle $\rho : Q \longrightarrow B^*$.
La fonction subséquentielle réalisée $|S|$ est alors définie par :
$|S|(u) = \lambda(i, u).\rho(\delta(i, u))$
De façon intuitive, on "fait marcher" le transducteur, puis à la fin selon l'état dans lequel on se trouve on rajoute un supplément au mot écrit.

Exemple 1 Une transduction séquentielle est bien évidemment aussi une fonction subséquentielle, il suffit de poser $\rho(q) = \varepsilon, \forall q$.

Exemple 2 Si on reprend α comme définie plus haut, et qu'on lui rajoute $\rho(1) = 1$ et $\rho(2) = \varepsilon$, on obtient une "vraie" fonction "+1".

Exemple 3 La fonction τ définie plus haut n'est pas subséquentielle par contre. En effet, si $\tau = |S|$, alors pour n pair on a :
$|S|(x^n) = \lambda(i, x^n).\rho(\delta(i, x^n)) = a^n$
$|S|(x^{n+1}) = \lambda(i, x^n).\lambda(\delta(i, x^n), x).\rho(\delta(i, x^{n+1}) = b^{n+1}$
Si on prend n assez grand, c'est à dire n plus grand que $max(|\rho(q)|)$, alors le "début" $\lambda(i, x^n)$ n'est pas le mot vide. Et il ne peut pas à la fois être composé uniquement de a et de b : contradiction.

2.4 composition des transducteurs subséquentiels

Proposition Si $\alpha : A^* \longrightarrow B^*$ et $\beta : B^* \longrightarrow Z^*$ sont deux fonctions subséquentielles (resp séquentielles), alors $\beta \circ \alpha$ est subséquentielle (resp séquentielle).

Preuve Soient $S = (A, B, Q, i_s, \rho)$ et $T = (B, Z, P, i_t, \sigma)$ les transducteurs réalisant α et β. Alors on pose :
$T \circ S = (A, Z, P \times Q, [i_s, i_t], \omega)$ où [p,q] désigne le coupe (p,q) (pour alléger la notation).
$\delta([p, q], x) = [\delta_t(p, \lambda_s(q, x)), \delta_s(q, x)]$
$\lambda([p, q], x) = \lambda_t(p, \lambda_s(q, x))$
$\omega([p, q]) = \lambda_t(p, \rho(q)).\sigma(\delta_t(p, \rho(q)))$.

Il faut vérifier que ce transducteur effectue bien ce qu'on veut :

D'abord, on va vérifier que ces définitions restent valables si on les étend aux mots. Par récurrence sur la longueur du mot : pour le mot vide c'est évident. Si un mot s'écrit $ux, u \in A*, x \in A$, alors, notons $u' = \lambda_s(q, u)x' = \lambda_s(\delta_s(q, u), x)$. On a :
$\delta([p, q], ux) = \delta(\delta([p, q], u), x)$ par définition de l'extension aux mots
$= \delta([\delta_t(p, u'), \delta_s(q, u)], x)$ par hypothèse de récurrence
$= [\delta_t(\delta_t(p, u'), x'), \delta_s(\delta_s(q, u), x)]$ par la définition de δ pour les lettres
$= [\delta_t(p, u'x'), \delta_s(q, ux)]$
$= [\delta_t(p, \lambda_s(q, ux)), \delta_s(q, ux)]$

Le raisonnement est le même à peu de choses près pour λ et ω.

On va maintenant montrer que $|T \circ S| = \beta \circ \alpha$:
On écrira $\alpha(u) = v_0.v_\rho$ pour le "séparer" en sa partie obtenue par le transducteur et par l'application ρ. Ensuite on notera $\beta(\alpha(u)) = w_0.w_1.w_\sigma$, qui "sépare" le mot en la partie obtenue par le transducteur à partir de v_0, celle obtenue par le transducteur à partir de v_ρ et l'ajout obtenu par σ
On a alors :
$|T \circ S|(u) = \lambda([i_t, i_s], u).\omega(\delta([i_t, i_s], u)$
La première partie donne : $\lambda([i_t, i_s], u) = \lambda_t(i_t, \lambda_s(i_s, u)) = \lambda_t(i_t, v_0) = w_0$
Ensuite, $\omega(\delta([i_t, i_s], u) = \omega([\delta_t(i_t, \lambda_s(i_s, u)), \delta_s(i_s, u)])$
$= \omega([\delta_t(i_t, v_0), \delta_s(i_s, u)])$
$= \lambda_t(\delta_t(i_t, v_0), \rho(\delta_s(i_s, u))).\sigma(\delta_t(\delta_t(i_t, v_0), \rho(\delta_s(i_s, u))))$
La première partie donne :$\lambda_t(\delta_t(i_t, v_0), v_\rho) = w_1$
La seconde : $\sigma(\delta_t(\delta_t(i_t, v_0), v_\rho)) = \sigma(\delta_t(i_t, v_0.v_\rho)) = w_\sigma$
On a bien au final $|T \circ S|(u) = w_0.w_1.w_\sigma = \beta \circ \alpha$

Enfin, si les transducteurs sont séquentiels, on remarque qu'on a $\omega([p,q]) = \lambda_t(p,\varepsilon).\sigma(\delta_t(p,\varepsilon)) = \varepsilon$, ce qui montre bien que $\beta\circ\alpha$ est séquentielle.

3 Fonctions rationnelles et transducteurs non ambigus

3.1 Définition

Intérêt *On a d'une part des transducteurs très généraux, peu manipulables, et d'autre part des transducteurs séquentiels, très faciles à utiliser mais qui en permettent pas de tout faire. On va voir ici une moyenne, les transducteurs non ambigus : ils sont non déterministes au sens des automates habituels, mais en revanche il ne peut y avoir (au plus) qu'une sortie.*

Définition Soit $T = (A, B, Q, i, F, E)$ un transducteur, vérifiant :
$E \subset Q \times A \times B^* \times Q$ (on lit seulement des lettres, et on n'a pas d'ε-transition)
$(p, x, u, q), (p, x, u', q) \in E \Rightarrow u = u'$ (une seule écriture possible pour un départ, une arrivée et une lettre lue).
T est dit *non-ambigu* si pour tout mot $u \in A^*$ il est l'étiquette *d'au plus* un chemin réussi.

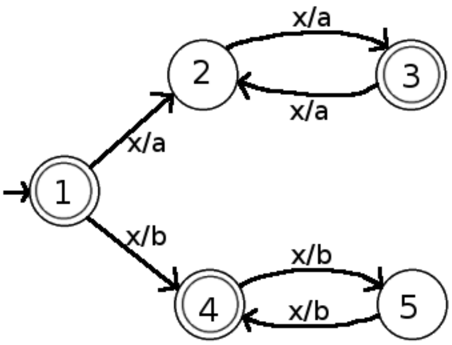
3.2 Théorème important

Il s'avère que les transducteurs non-ambigus, bien qu'à priori beaucoup plus restrictifs que les tranducteurs généraux, sont presque aussi puissants : ils permettront de réaliser toutes les transductions rationnelles ayant au plus un mot dans leur image : les fonctions rationnelles.

Définition Une fonction rationnelle est une transduction telle que $Card(\tau(u)) \leq 1$ pour tout u.

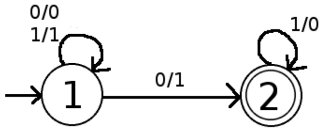
Théorème *Eilenberg, 1974* Toute fonction rationnelle peut-être réalisée par un transducteur non-ambigu.
(admis car démonstration longue et difficile)

Exemple 1 La fonction τ définie plus haut, qui n'était pas subséquentielle, peut quand même être réalisée par le transducteur non-ambigu suivant :



En effet, il est bien non-ambigu car si on "emprunte" le chemin du haut (et qu'on écrit des a), le seul moyen de s'arrêter sur un état final est d'avoir un nombre pair de lettres. Et pareil en bas.

Exemple 2 La fonction β définie plus haut, qui faisait le "+1" dans le cas où les bits de poids fort étaient à gauche, ne peut pas être réalisée par un transducteur séquentiel, par contre le transducteur non-ambigu suivant fonctionne :



On remarque que c'est le même que pour α, à l'envers.

4 Conclusion

Les transducteurs, à l'instar des automates, sont un outil simple (par rapport à une machine de Türing) et légers à implémenter en pratique sur un ordinateur (pas besoin de beaucoup de mémoire, tout est dans les états). On peut par exemple effectuer grâce à ça des opérations arithmétiques de base comme l'addition.