MMIM Modèles mathématiques en informatique musicale

Marc Chemillier

Master M2 Atiam (Ircam), 2010-2011

Oracle des facteurs (logiciel d'improvisation OMax)

- Construction de l'oracle
  - Liens suffixiels
  - Caractérisation des flèches addtionnelles
- Langage reconnu par l'oracle

1. Construction de l'oracle

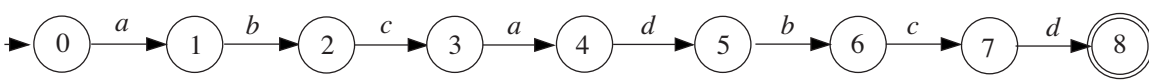
1.1 Liens suffixiels

L'ensemble des facteurs d'un mot x est fini, donc reconnaissable par automate. On obtient facilement un AFN reconnaissant tous les facteurs en prenant l'écorché de x et en rendant tous les états initiaux et terminaux.

Problème : **il n'existe pas de construction simple de l'AFD correspondant**. L'« oracle des facteurs » a été introduit pour pallier cette difficulté.

L'*oracle des facteurs* est un automate qui reconnaît tous les facteurs d'un mot x, mais avec quelques mots en plus. On peut donner directement l'AFD correspondant (sans déterminisation) par une technique analogue à l'algorithme de Morris & Pratt.

Exemple : écorché de la séquence $x = abcadbcd$



Construction de l'oracle : on part de l'écorché de x, et on ajoute des transitions à l'aide

d'une fonction *f* dite de « lien suffixiel » entre états, en construisant simultanément *f* et les transitions.

Au départ, on place un lien suffixiel de 1 vers 0. Puis on suppose l'oracle construit avec ses liens suffixiels jusqu'à l'état *p* compris. La lettre suivante *a* donne un nouvel état $\delta(p, a) = p+1$.

Pour ajouter les transitions, on suit les liens suffixiels déjà existants $p' = f(p)$, puis $p' = f(f(p))$, etc.

• si $\delta(p', a)$ est non défini, on ajoute une transition $\delta(p', a) = p+1$

- si $p' \neq 0$, on continue à suivre les liens,
- si $p' = 0$, on stoppe et on crée un lien suffixiel en posant $f(p+1) = 0$

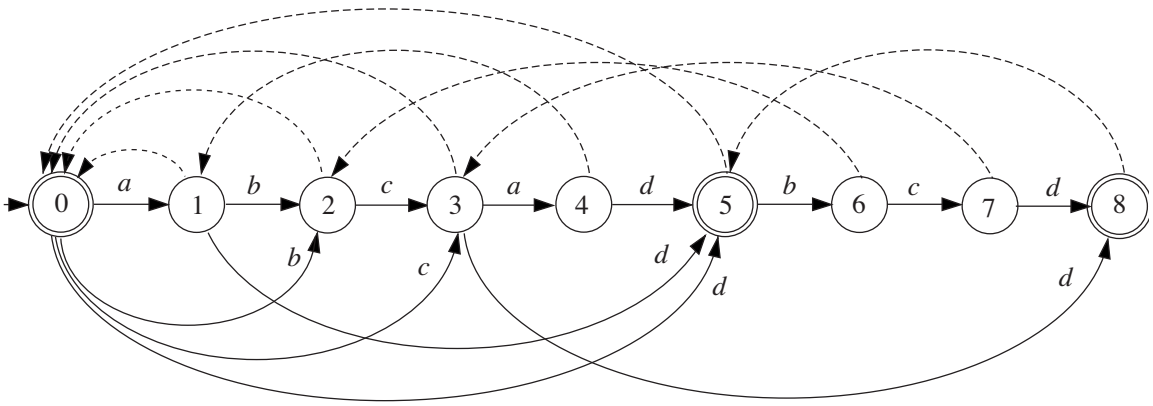
• si $\delta(p', a)$ est défini, on stoppe (pas de nouvelle transition), et on crée un lien suffixiel en posant $f(p+1) = \delta(p', a)$

Remarque : toutes les transitions arrivant dans un même état ont la même lettre.

Pour la séquence *abcadbcd*, l'oracle des suffixes donne l'AFD suivant (les liens suffixiels sont indiqués par des flèches en pointillé au-dessus).

• Oracle des facteurs : tous les états sont terminaux

• Oracle des suffixes : les états terminaux sont ceux du chemin suffixiel partant du dernier état, soit 8 (suffixe *cd*), 5 (suffixe *d*), 0 (suffixe vide)



Dans le logiciel d'improvisation OMax, on parcourt l'oracle en suivant les transitions, mais aussi en empruntant éventuellement les liens suffixiels :

d	*b*	*c*
5	6	7

motif prélevé = *a b c a* ***d b c*** *d*

On emprunte le lien suffixiel de 7 à 3 pour reprendre la lecture vers 4 :

d b c | *a d b* motif prélevé = *a b c* ***a d b*** *c d*

(3) 4 5 6

Mais le motif précédent *dbc* se terminait par *bc*. Or il se trouve que *bc* est aussi le facteur précédent *adb* dans la séquence. Donc en réalité, on a prélevé (*bc*)*adb* :

d | *b c a d b* motif réellement prélevé = *a*(***b c***)***a d b*** *c d*

(2 3) 4 5 6

C'est une propriété de l'oracle : les dernières lettres lues avant de quitter un état par un lien suffixiel sont identiques à celles qui précèdent l'état d'arrivée du lien (ex : *bc* dans le parcours ci-dessus). Ces lettres constituent une partie commune entre les motifs lus avant et après le lien suffixiel. Ainsi les liens suffixiels permettent d'enchaîner des motifs qui se chevauchent avec une partie commune (ex : *d*(*bc*) et (*bc*)*adb* dans le parcours ci-dessus). La propriété s'énonce ainsi :

Propriété fondamentale. *Si p est l'état d'arrivée d'un préfixe w de x, le lien suffixiel f(p) correspond à l'état d'arrivée du plus long suffixe de w qui est répété à gauche, c'est-à-dire qui est facteur non suffixe de w.*

Attention : Il ne faut pas confondre

• fonction de lien suffixiel :

f(*p*) = état d'arrivée du plus long suffixe propre de *w* qui est aussi **facteur** de *w* (donc de *x*)

• fonction de saut de Morris & Pratt :

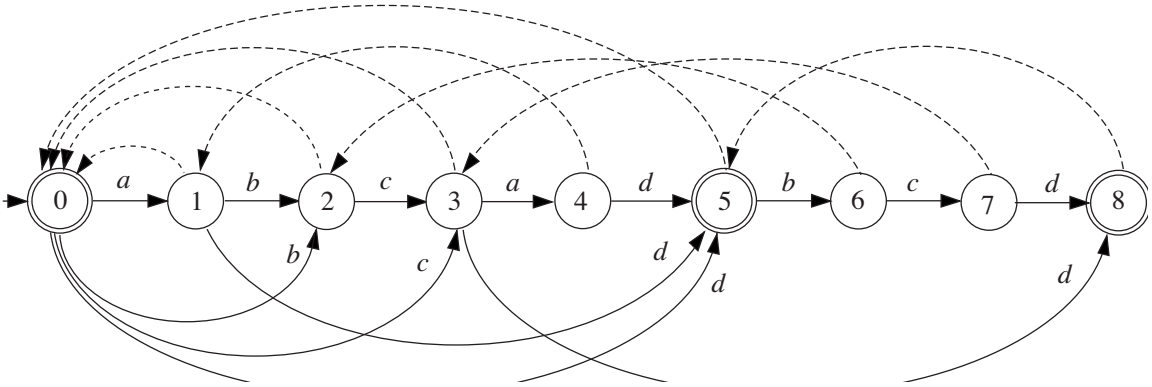
f(*p*) = état d'arrivée du plus long suffixe propre de *w* qui est aussi **préfixe** de *w* (donc de *x*)

1.2 Caractérisation des flèches additionnelles

Les flèches additionnelles de l'oracle (non présentes dans l'écorché) vérifient la caractérisation suivante. On note :

min(*p*) = ensemble des mots de **longueur minimale** reconnus de 0 à *p*

Propriété caractéristique. *Pour tout mot u de min(p) et pour toute lettre* $b \neq x_{p+1}$*, il existe une flèche additionnelle* $\delta(p, b) = q$ *si et seulement si q = état d'arrivée de la première occurrence de ub dans x telle que* $q > p + 1$.



état 0 : min(0) = {ε}, $x_1 = a$, flèches additionnelles : δ(0, *b*) = 2, δ(0, *c*) = 3, δ(0, *d*) = 5

état 1 : min(1) = {*a*}, $x_2 = b$, flèche additionnelle : δ(1, *d*) = 5 (1ère occurrence de *ad*)

état 3 : min(3) = {*c*}, $x_4 = a$, flèche additionnelle : δ(3, *d*) = 8 (1ère occurrence de *cd*)

Remarque : Dans l'article de Allauzen, Crochemore & Raffinot 1999, l'oracle est défini par cette propriété caractéristique. La construction par liens suffixiels n'arrive qu'après une série de lemmes. Mais comme elle est plus efficace et donne le même automate, c'est elle qui est utilisée.

Lemme 1. *Si $u \in min(p)$, alors u est suffixe de $x_1 \ldots x_p$ et p = état d'arrivée de la première occurrence de u dans x.*

Dem. C'est vrai pour l'état $p = 1$, car min(1) est réduit à la première lettre du mot x_1 et l'état 1 = arrivée de la première occurrence de x_1. On procède par récurrence en montrant que si la propriété vraie pour tous les états $\leq p - 1$, elle se conserve par construction quand on passe à l'état p. L'idée est de considérer la dernière flèche du chemin de 0 à p permettant de lire u : il s'agit soit d'une flèche de l'écorché de x, soit d'une flèche additionnelle. Notons que cette flèche a toujours pour étiquette x_p, car les flèches arrivant à un même état ont la même étiquette. Soit j l'état de départ de cette flèche. On pose $u = zx_p$. On a nécessairement $z \in \min(j)$ car $u = zx_p \in \min(p)$.

• 1er cas : $j = p - 1$ (la flèche fait partie de l'écorché de x). Par récurrence comme $z \in \min(p - 1)$, z est suffixe de $x_1 \ldots x_{p-1}$ n'apparaissant pas à gauche. La dernière flèche d'étiquette x_p fait partie de l'écorché de x. Donc $u = zx_p$ est suffixe de $x_1 \ldots x_p$ et il n'apparaît pas à gauche car sinon, z apparaîtrait à gauche.

• 2ème cas : $j < p - 1$ (il s'agit d'une flèche additionnelle). Par récurrence comme $z \in \min(j)$, z est suffixe de $x_1 \ldots x_j$ n'apparaissant pas à gauche. Comme la flèche d'étiquette x_p est additionnelle, elle conduit vers l'état d'arrivée de la première occurrence de $zx_p = u$ qui est donc suffixe de $x_1 \ldots x_p$ et n'apparaît pas à gauche.

Remarques :

1) On voit que tous les mots de min(*p*) sont

- suffixes du même mot $x_1 \ldots x_p$,

- de même longueur minimale,

donc ils sont égaux. Il en résulte que min(p) est un mot unique.

2) La propriété caractéristique des flèches additionnelles s'étend à toutes les flèches de l'oracle, c'est-à-dire aux flèches de l'écorché :

Pour $u = \min(p)$, le Lemme 1 montre que p = état d'arrivée de la première occurrence de u, donc pour une flèche de l'écorché $\delta(p, x_{p+1}) = p + 1$, l'état $p + 1$ = arrivée de la première occurrence de ux_{p+1}.

Lemme 2. *Si u = min(p), alors u est suffixe de tout mot w reconnu de 0 à p.*

Dem. C'est vrai pour l'état $p = 1$, car $\min(1) = x_1$ et c'est le seul mot reconnu de 0 à 1. On procède par récurrence en montrant que si la propriété vraie pour tous les états $\leq p - 1$, elle se conserve par construction quand on passe à l'état p. L'idée est de considérer la dernière flèche du chemin de 0 à p permettant de lire un mot w. On la note $\delta(j, a) = p$ avec un état $j < p$ et $w = w'a$. On pose $v = \min(j)$ et d'après la remarque ci-dessus sur la caractérisation des flèches, p = état d'arrivée de la première occurrence de va, c'est-à-dire que va est suffixe de $x_1 \ldots x_p$. Comme $u = \min(p)$, il est suffixe de $x_1 \ldots x_p$ et sa longueur est inférieure à celle de va, donc u est suffixe de va. Par hypothèse de récurrence, v est suffixe de w', donc va suffixe de w. D'où u suffixe de w.

Lemme 3. *Si w est facteur de x, alors w est reconnu dans l'oracle de 0 à p ≤ état d'arrivée de sa première occurrence dans x.*

Dem. On procède par récurrence sur la longueur du facteur w. En posant $w = w'a$, w' est un facteur plus court, donc il est reconnu de 0 à $j \leq$ état d'arrivée de sa première occurrence dans x. Soit $v = \min(j)$, d'après le Lemme 2, v est suffixe de w'. Comme va apparaît dans x, il existe une transition $\delta(j, a) = p$ état d'arrivée de la première occurrence de va. Donc il existe un chemin tel que $w = w'a$ soit reconnu de 0 à p. De plus, $p \leq$ état d'arrivé de sa première occurrence, car va est suffixe de w et p = état d'arrivé de la première occurrence de va, donc il ne peut y avoir d'occurrence de w plus à gauche.

Le Lemme 3 montre que l'oracle reconnaît tous les facteurs de x.

En fait, l'oracle reconnaît **un peu plus** que les facteurs de x.

2. Langage reconnu par l'oracle

2.1 Facteurs canoniques et contractions

L'article Mancheron & Moan 2005 propose une caractérisation du langage reconnu par l'oracle en introduisant les notions suivantes (p. 142-143 de l'article).

Ensemble des facteurs canoniques d'un mot s par rapport à son oracle :

F_s = {min(p) / l'état p a au moins deux flèches entrantes, ou deux flèches sortantes}

Une contraction de s est un couple (p, q) où p et q sont les états d'arrivée de la dernière lettre de deux occurrences d'un facteur canonique

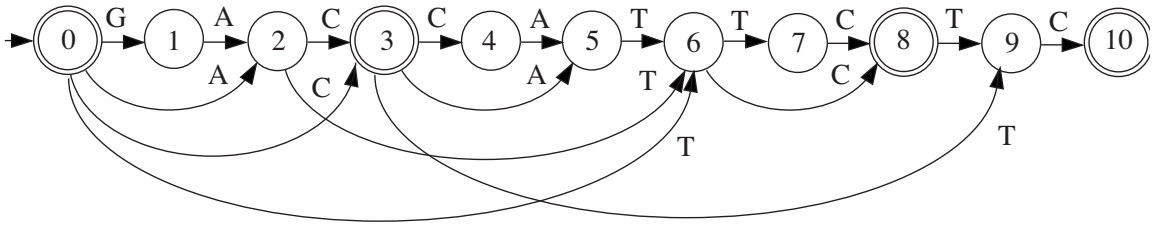
Appliquer une contraction (p, q) à s : on supprime les lettres de s comprises entre p inclus et q non inclus

Fermeture de s :

$\varepsilon(s)$ = ensemble des mots obtenus en appliquant à s toutes les combinaisons de contractions possibles

Théorème (3.1, p. 150). *Le langage reconnu par l'oracle d'un mot s est l'ensemble des suffixes de tous les mots de sa fermeture ε(s).*

Exemple : oracle du mot s = GACCATTCTC



Etats = {0,	2,	3,	5,	6,	8,	9}
F_s =	{A,	C,	CA,	T,	TC,	CT}
Contractions	(2, 5)	(3, 4)		(6, 7)	(7, 9)	
		(3, 8)		(6, 9)		
		(3, 10)				

Noter qu'on limite l'énumération en fixant le premier état p = celui de la première occurrence du facteur.

Exemple d'application de la combinaison de contractions (2, 5) et (7, 9) :

	1	2	3	4	5	6	7	8	9	10
	G	A	C	C	A	T	T	C	T	C
(2, 5)	G	-	-	-	A	T	T	C	T	C
(7, 9)	G	-	-	-	A	T	-	-	T	C

d'où le mot GATTC

Petite remarque : pourquoi les lettres G, C, T, A ?

Le génome est fait d'ADN. Les gènes contenus dans le génome sont codés sous forme chimique le long des molécules d'ADN. Celles-ci sont constituées par l'enchaînement de "maillons" élémentaires nommés nucléotides. Les nucléotides ont une partie variable - une base, du point de vue chimique - qui peut exister sous 4 formes différentes ; ces formes sont symbolisées par les lettres A, T, G et C. Les instructions sont donc écrites dans un alphabet chimique à 4 lettres seulement.

Remarque : Dans les applications bioinformatiques, l'oracle est utilisé de façon négative. Si un mot n'est pas reconnu par l'oracle, on est sûr qu'il n'est pas suffixe.

2.2 Exemple de langage reconnu par l'oracle

Exemple : oracle du mot s = GCTCA

On peut vérifier que les suffixes de GCTCA sont reconnus (outre le mot vide) :

GCTCA

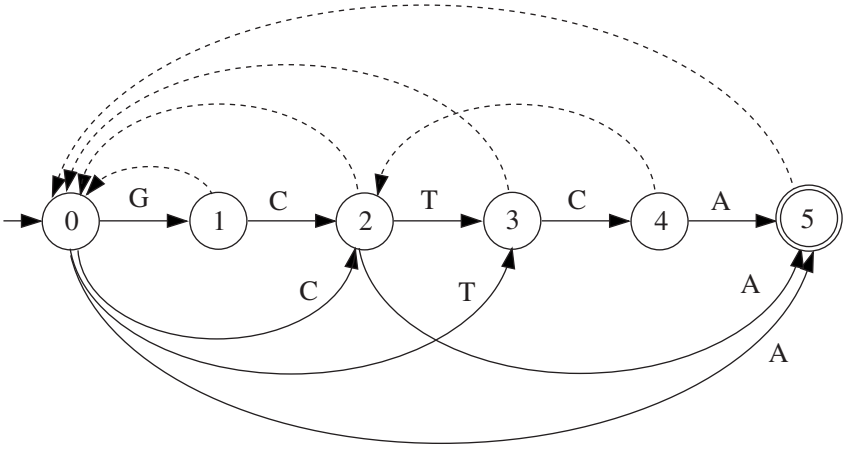
CTCA

TCA

CA

A

Existe-t-il d'autres mots reconnus par l'oracle qui ne sont pas suffixes de GCTCA ?



Oui, il y en a un : GCA

Etats = {0, 2, 3, 5}

F_s = {C, T, A}

Contractions (2, 4)

Si l'on applique la contraction (2, 4), on obtient précisément ce mot GCA.

La fermeture ε(*s*) contient les suffixes de *s* = GCTCA et de GCA. Comme les suffixes de GCA sont aussi suffixes de *s* excepté GCA, cela donne le langage reconnu par l'oracle :

ε(*s*) = {GCTCA, CTCA, TCA, CA, A, ε, GCA}

Références

• oracle des suffixes (simulation stylistique)

Allauzen, Cyril & Maxime Crochemore, Mathieu Raffinot, Factor oracle : A new structure for pattern matching, *SOFSEM '99*: *Proceedings of the 26th Conference on Current Trends in Theory and Practice of Informatics*, Lecture Notes in Computer Science, Springer-Verlag, 1999, p. 291-306.

Mancheron Alban , Moan Christophe, Combinatorial Characterization of the Language Recognized by Factor and Suffix Oracles, *International Journal of Foundations of Computer Science*, 16 (6) (2005) 1179-1191.

Assayag, Gérard, Shlomo Dubnov, Olivier Delerue, Guessing the Composer's Mind: Applying Universal Prediction to Musical Style, *Proceedings of the ICMC (Int. Computer Music Conf.)*, 1999, p. 496-499.

Assayag, Gérard, Shlomo Dubnov, Using factor oracles for machine improvisation, *Soft Computing*, special issue on Formal Systems and Music, G. Assayag, V. Cafagna, M. Chemillier (eds.), 8 (9) (2004) 604–610.