

Autòmat Diccionari

Com s'ha vist en pràctiques anteriors, una aproximació no determinista al problema del pattern matching permet abordar el problema amb un comportament temporal millor. En aquesta pràctica es mostra com, a partir del conjunt de patrons a buscar, és possible construir un autòmat determinista que permet obtenir una solució al problema amb un cost temporal encara menor.

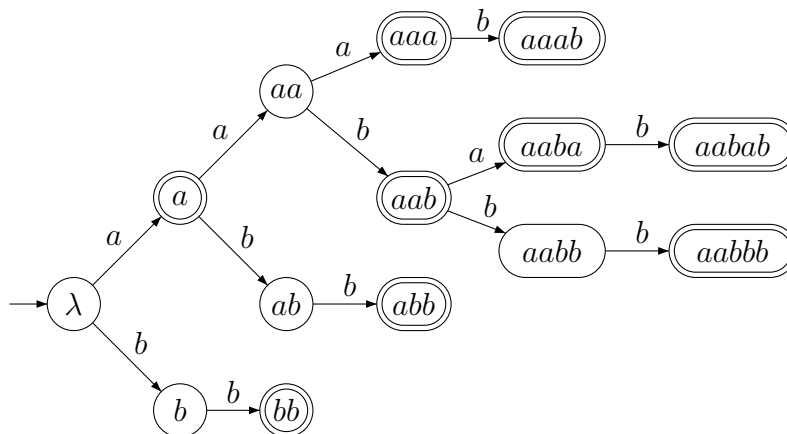
El nou mètode es basa en la construcció de l'*autòmat diccionari* del conjunt de patrons. A partir d'un conjunt M de cadenes sobre un determinat alfabet Σ , es defineix l'autòmat diccionari $AD_M = (Q, \Sigma, \delta, q_0, F)$ com segueix:

- $Q = \{x \in \Sigma^* : x \in Pref(M)\}$
- $q_0 = \lambda$
- $F = Pref(M) \cap \Sigma^* M$
- $\delta(x, a) = h(xa)$ on, per a qualsevol cadena u , $h(u)$ és el sufix més llarg d' u que pertany a $Pref(M)$.

És important notar les similituds entre l'autòmat diccionari del conjunt M (AD_M) i l'arbre acceptor de prefixos del mateix conjunt ($AAP(M)$). Les diferències són, d'una banda el conjunt de finals, i d'altra, la definició de la funció de transicions. De fet, tant els finals com totes les transicions d' $AAP(M)$ són estats finals i transicions del AD_M , per la qual cosa en l'exemple següent considerem el que vam utilitzar en la pràctica anterior:

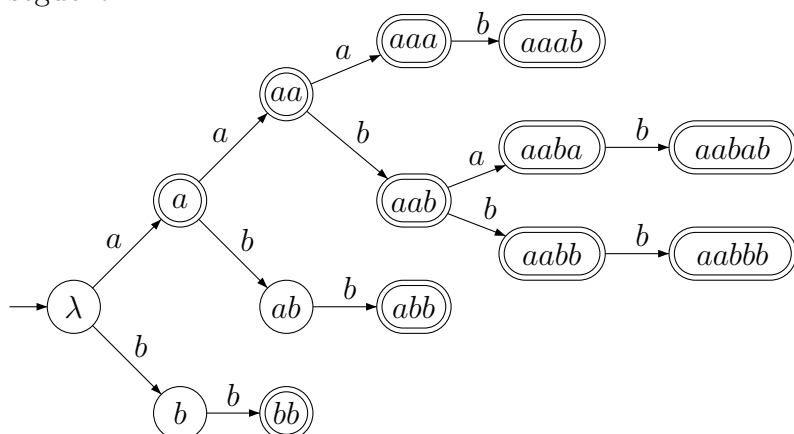
$$M = \left\{ \begin{array}{l} p_1 = a, p_2 = bb, p_3 = aaa, p_4 = aab, p_5 = abb, \\ p_6 = aaab, p_7 = aaba, p_8 = aabab, p_9 = aabbb \end{array} \right\}$$

Posteriorment necessitarem referir-nos a aquests patrons individualment, per la qual cosa hem associat a cada patró un identificador. L'arbre acceptor de prefixos corresponent a aquest conjunt (on identifiquem cada estat amb el prefix d' M amb el qual està associat) és el següent:



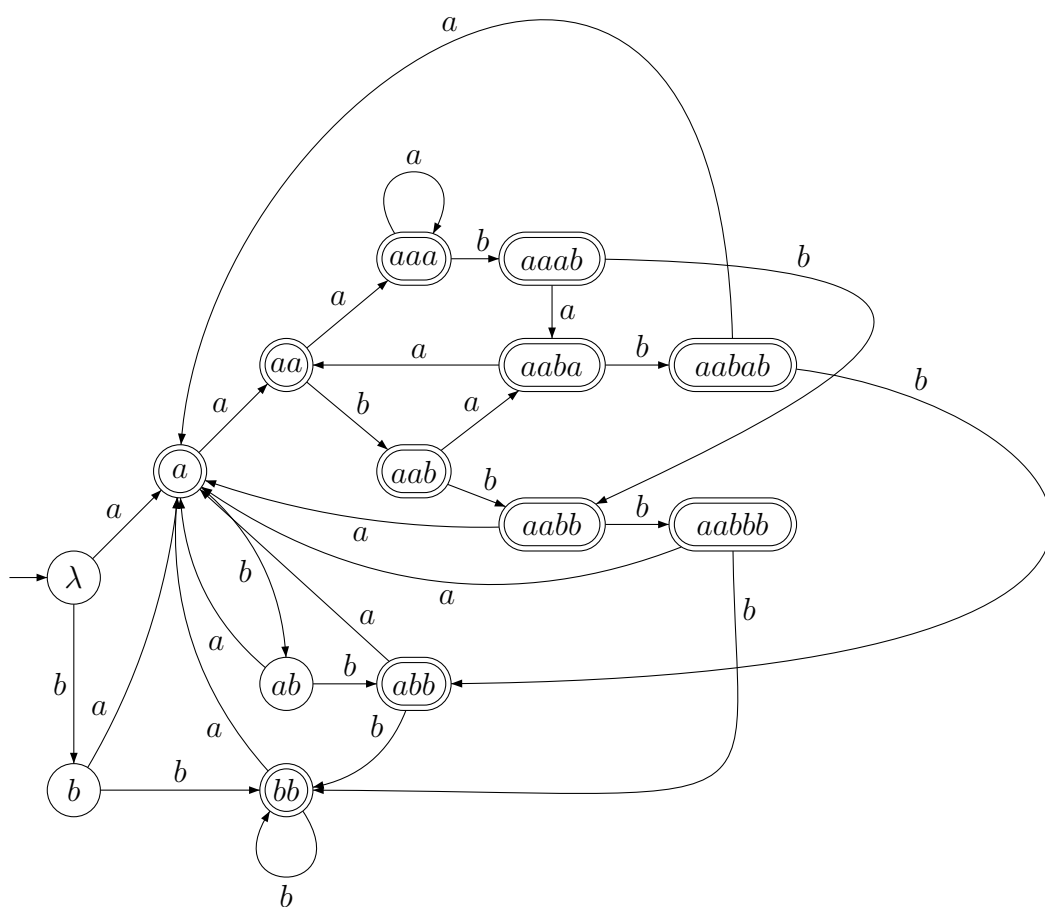
Com a primer pas modifiquem el conjunt de finals. En aquest sentit, noteu com els estats aa i $aabb$, que no són finals en el $AAP(M)$, sí que ho són en AD_M (els dos tenen un sufix que

està en M). No hi ha cap altre estat en aquest cas, per tant, aquesta primera modificació ens du a l'autòmat següent:



Ara considerem les transicions no definides en aquest autòmat, entre altres, les transicions de l'estat bb amb els símbols a i b o de l'estat ab amb el símbol a .

Per exemple, si apliquem la definició de la funció de transicions de l' AD_M , $\delta(ab, a) = h(aba)$. El sufix més llarg d' aba en $Pref(M)$ és a , per tant $\delta(ab, a) = a$. A continuació mostrem el AD_M després d'aplicar aquest procés a les transicions no definides.

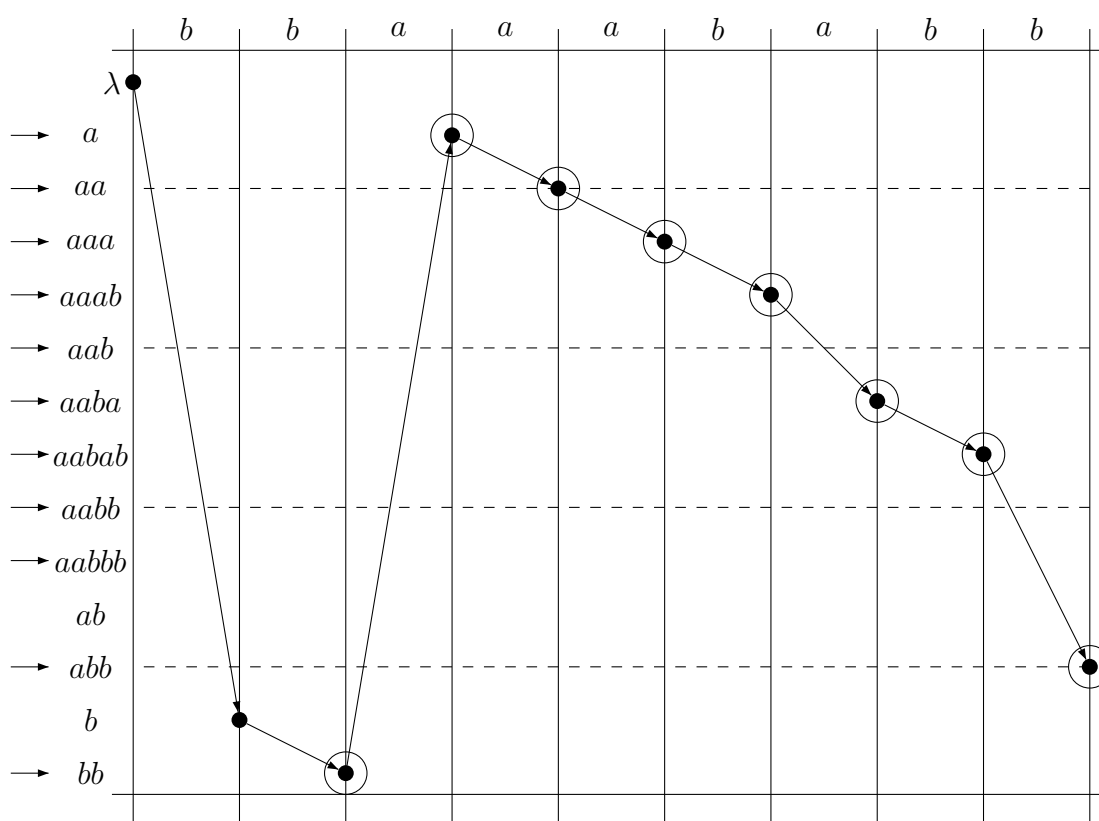


Com ja passava en la pràctica anterior, estem construint un autòmat (l'autòmat diccionari A_M) que reconeix el llenguatge Σ^*M , de manera que, mentre s'analitza un text qualsevol x , arribar a un estat final implica que s'ha trobat, almenys, un patró. De fet, s'han trobat tots els patrons que són sufixe de la cadena que denota l'estat final.

Per tant, modifiquem l'autòmat per a anotar en cada estat final u , quins patrons són sufix de la cadena u . Aquesta informació es resumeix en la taula següent:

estado	a	aa	bb	aaa	aab	abb	$aaab$	$aaba$	$aabb$	$aabab$	$aabbb$
patrones	p_1	p_1	p_2	p_1, p_3	p_4	p_2, p_5	p_4, p_6	p_1, p_7	p_2, p_5	p_8	p_2, p_9

Una vegada obtingut l'autòmat diccionari, és possible detectar totes les posicions on apareix una cadena d' M en un text x . Per a açò només cal realitzar una anàlisi determinista, i sempre que s'arribi a un estat final, indicar que s'han detectat els patrons associats al corresponent estat final. Per exemple, considerant el text $x = bbaaababb$, l'anàlisi determinista es pot representar com segueix:



En aquest diagrama hem marcat els estats finals visitats durant l'anàlisi. Es pot veure que després d'analitzar el segon símbol s'arriba a l'estat bb , que en ser final indica que s'ha detectat un patró p_2 del conjunt M (el patró bb). De la mateixa manera, per exemple: després d'analitzar bba s'arriba a l'estat a que indica que s'ha detectat el patró a ; quan s'ha

analitzat *bbaaa* s'arriba a l'estat *aaa* que indica que s'han detectat els patrons *a* i *aaa*, i així succesivament.

Exercicis

Exercici 1

Implementar un mòdul Mathematica que, prenent una cadena u i un conjunt de cadenes M com entrada, torne el sufix més llarg d' u que siga un element d' M .

Exercici 2

Implementar un mòdul Mathematica que, prenent un conjunt de cadenes M com entrada, torne l'autòmat diccionari d'aquest conjunt.

Exercici 3

Implementar un mòdul Mathematica per a, donats l'autòmat diccionari d'un conjunt de patrons M i un text x , torne el conjunt de posicions d' x en las quals apareix un element d' M .

Bibliografia

Maxime Crochemore, Christophe Hancart and Thierry Lecroq ALGORITHMS ON STRINGS. *Cambridge University Press*, 2007.