Introducció

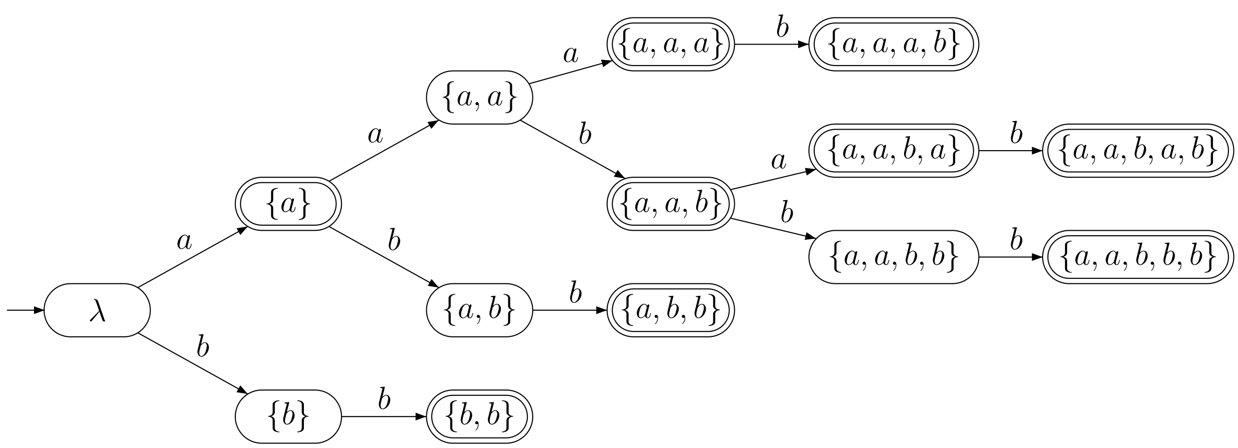
El problema de detectar en quines posicions apareix una o un conjunt de distintes cadenes (usualment denominades patrons) en una cadena més llarga x (text) es coneix com *String Matching* o *Pattern Matching*. Aquest problema és de gran interés algorísmic i té utilitat pràctica en camps com, per exemple, la Biologia Molecular o la Genètica, ja que permet el processament ràpid de seqüències biològiques.

Una primera aproximació (*naive*) al problema comporta buscar cada patró en la seqüència cosa que suposa un cost de $\mathcal{O}(n \cdot |p| \cdot |x|)$, on n és el nombre de patrons a localitzar, $|p|$ és la longitud del patró més llarg i $|x|$ és la longitud del text.

Donat un conjunt M de cadenes sobre determinat alfabet Σ, l'*arbre acceptor de prefixos* per a M ($AAP(M)$) és un autòmat determinista que accepta exclusivament M. Per exemple, donat el conjunto:

$$M = \{\{a\}, \{b, b\}, \{a, a, a\}, \{a, a, b\}, \{a, b, b\}, \{a, a, a, b\}, \{a, a, b, a\}, \{a, a, b, a, b\}, \{a, a, b, b, b\}\}$$

l'$AAP(M)$ seria el següent:

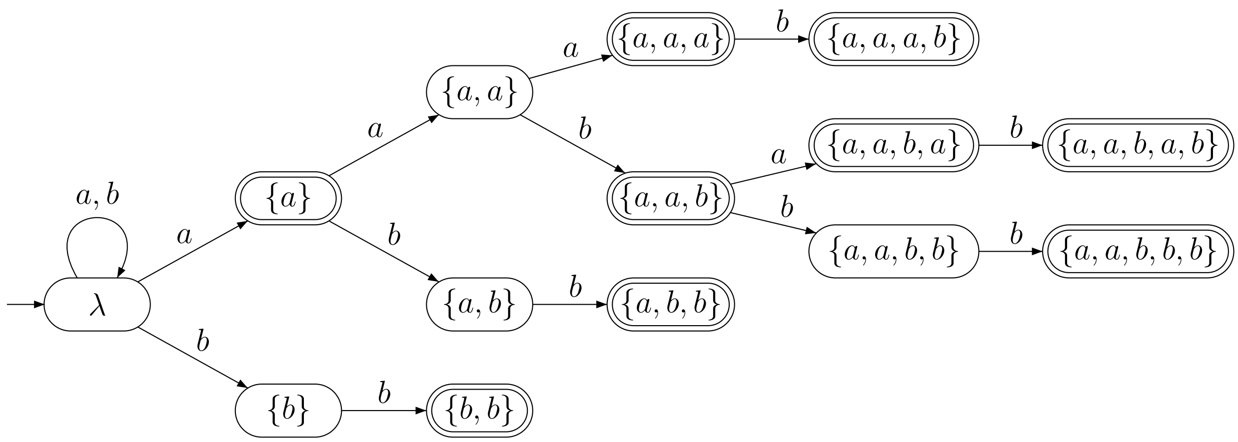


És fàcil construir l'AAP d'un conjunt M de cadenes si considerem els prefixos de totes las cadenes en M. Intuïtivament, per a acceptar una cadena qualsevol, cal processar cada símbol d'aquesta en l'ordre correcte.

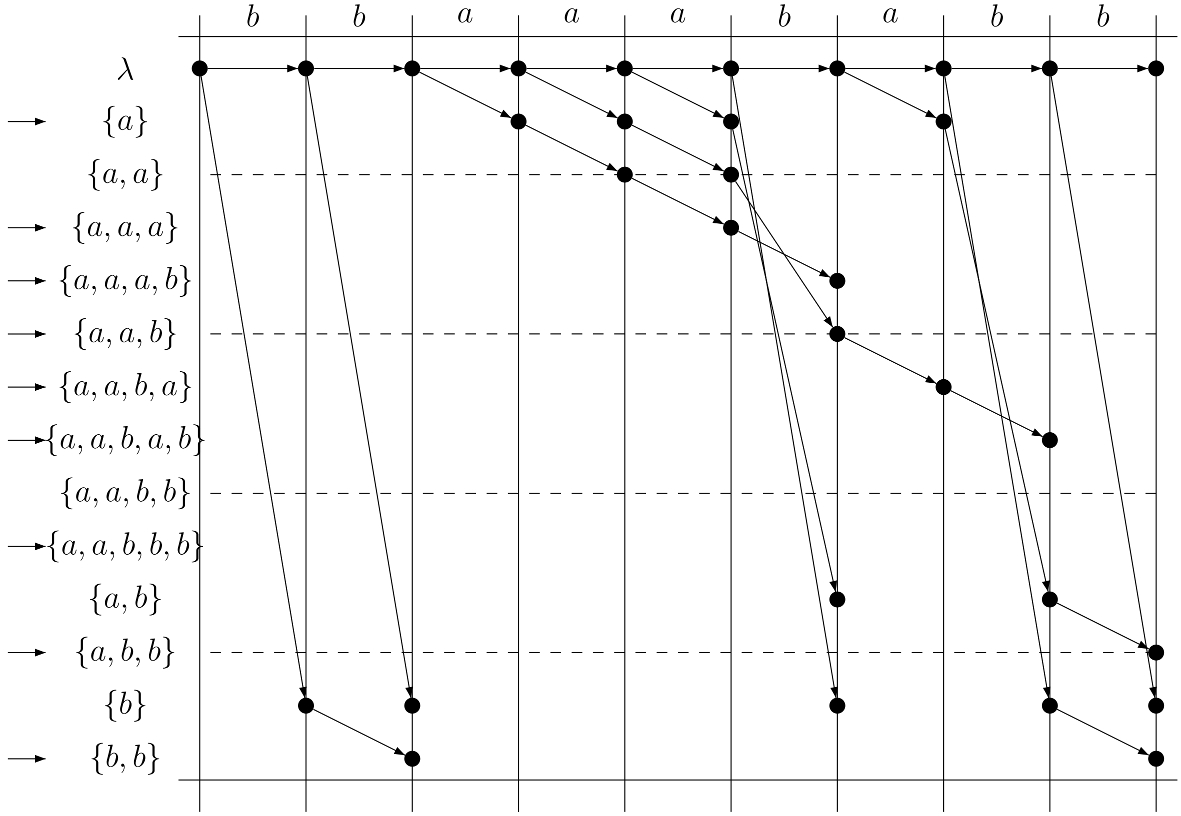
Formalment, l'$AAP(M)$ es defineix com l'autòmat $A = (Q, \Sigma, \delta, q_0, F)$ on:

- $Q = \{x \in \Sigma^* \ : \ x \in Pref(M)\}$
- $q_0 = \lambda$
- $F = M$
- $\delta(x, a) = xa$ si $xa \in Q$, estant indefinida en cas contrari.

Aquest autòmat pot modificar-se fàcilment per a obtenir un AFN que accepte $\Sigma^* M$. Note's que per a açò prou afegir un bucle sobre l'estat inicial amb tots els símbols de l'alfabet. A continuació es mostra l'AFN obtingut a partir de l'exemple anterior.



A partir d'aquest autòmat, és possible detectar totes les posicions on apareix una cadena de M en un text x. Per a açò prou realitzar una anàlisi no determinista modificada lleugerament. Aquesta modificació consisteix a detectar cada vegada que s'aconsegueix un estat final en el conjunt d'estats actius. Per exemple, considerant el text $x = \{b, b, a, a, a, b, a, b, b\}$, l'anàlisi no determinista pot representar-se com segueix:



En aquest diagrama es pot vore que després d'analitzar el segon símbol s'arriba als estats $\{b\}$ y $\{b, b\}$. El fet que l'estat l'estat $\{b, b\}$ siga final indica que s'ha detectat un patró del conjunto M (el patró bb). De la mateixa manera, per exemple: després d'analitzar bba i $bbaa$ s'arriba a l'estat $\{a\}$ la qual cosa indica que s'ha detectat el patró a; quan s'analitza $bbaaa$ s'arriba als estats finals $\{a\}$ y $\{a, a, a\}$ la qual cosa indica que s'han detectat els patrons a i aaa, y així successivament.

Exercicis

Exercici 1

Es demana implementar un mòdul Mathematica que, prenent un conjunt de cadenes M com entrada, torne l'arbre acceptor de prefixos del conjunt.

Exercici 2

Es demana implementar un mòdul Mathemática que, prenent un conjunt de cadenes M com entrada, torne un AFN que accepte el llenguatge Σ^*M.

Exercici 3

Es demana implementar un mòdul Mathematica que, donats un conjunt de patrons M i un text x, construisca un AFN que accepte el llenguatge Σ^*M i l'utilitze per a, fent una anàlisi eficient del text x, torne les posicions de x en les quals apareix un patró de M i de quin patró es tracta.

Exemple: Donats:

$$x = \{b, a, b, a, a, b, b, a, b, b, b, a, b, b, a, b, a, a, a, a, a, b, b, a, a, b, b, a, b, a\}$$
$$M = \{\{b, b\}, \{a, b, b, b\}, \{b, b, a, b\}, \{a, a, a, a\}\}$$

el mòdulo hauria de tornar:

$$\{\{6, \{b, b\}\}, \{6, \{b, b, a, b\}\}, \{9, \{b, b\}\}, \{10, \{b, b\}\}, \{10, \{b, b, a, b\}\}, \{8, \{a, b, b, b\}\},$$
$$\{13, \{b, b\}\}, \{13, \{b, b, a, b\}\}, \{17, \{a, a, a, a\}\}, \{18, \{a, a, a, a\}\}, \{22, \{b, b\}\},$$
$$\{26, \{b, b\}\}, \{26, \{b, b, a, b\}\}\}$$

Nota: Per a resoldre l'exercici es recomana modificar l'exercici de la pràctica 2 que aborda el processament d'una cadena en un autòmat no determinista.

Bibliografia

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