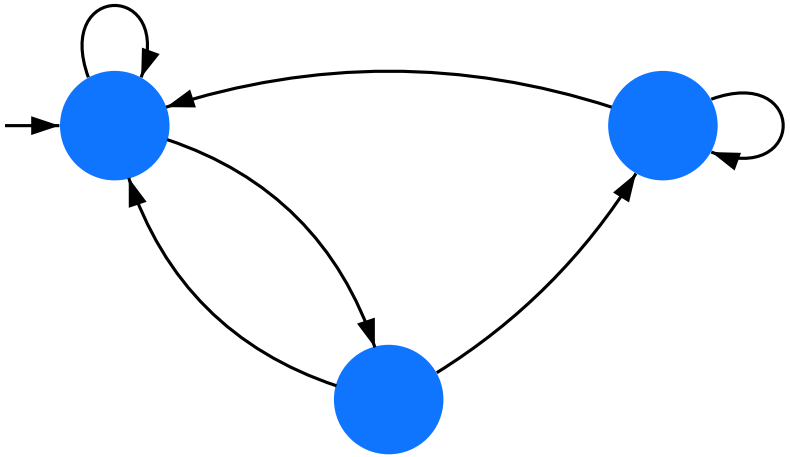
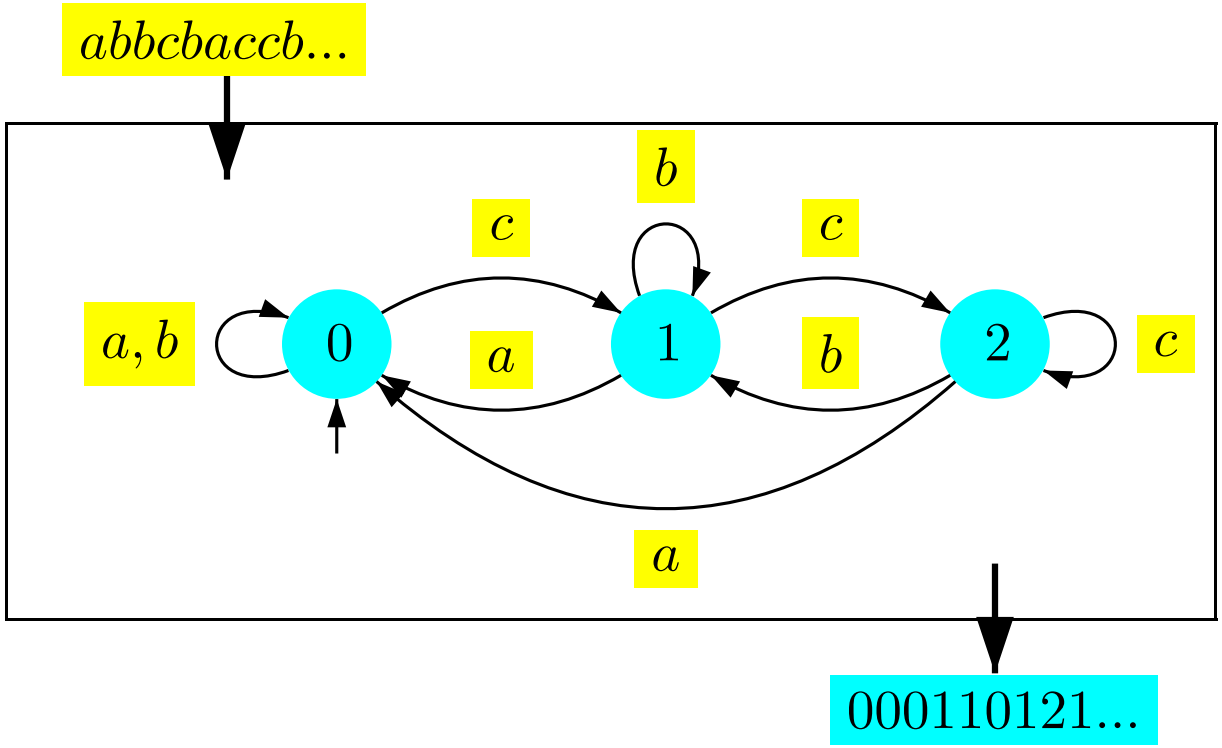
Vector Algorithms for Approximate String Matching [a]

Anne Bergeron et Sylvie Hamel



[a] International Journal of Foundations of Computer Science, Vol.13, No.1, pp.53-65, 2002.

Ancienne Technologie: utilisation **séquentielle** d'un automate pour convertir une chaîne d'entrée en une chaîne de sortie.



Analyse en temps: $\mathcal{O}(m)$ où m est la longueur de la chaîne d'entrée

Vecteurs caractéristiques:

Étant donné la séquence d'entrée

$$\boldsymbol{e} = abbcbaccb...,$$

on a les vecteurs caractéristiques suivants:

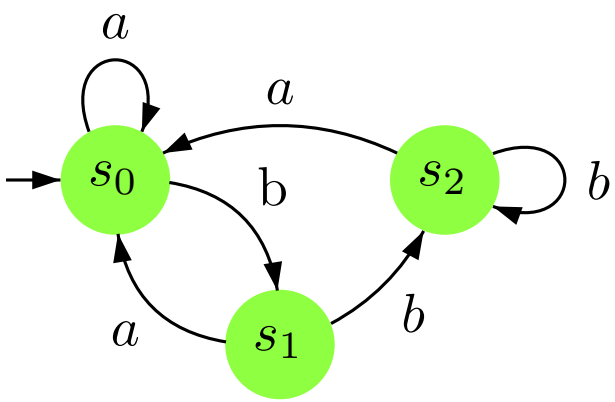
$$\begin{aligned} \boldsymbol{a} &= 100001000... \\ \boldsymbol{b} &= 011010001... \\ \boldsymbol{c} &= 000100110... \end{aligned}$$

Opérations sur les vecteurs de bits

Soit $\boldsymbol{x}$ et $\boldsymbol{y}$ des vecteurs booléens - ou vecteurs de bits. On permet les opérations suivantes sur ces vecteurs:

1. disjonction $\boldsymbol{x} \vee \boldsymbol{y}$
2. conjonction $\boldsymbol{x} \wedge \boldsymbol{y}$
3. négation $\neg \boldsymbol{x}$
4. addition binaire avec retenue $\boldsymbol{x} +_{\boldsymbol{b}} \boldsymbol{y}$
5. déplacement vers la droite $\uparrow_a \boldsymbol{x}$

Nouvelle Technologie: trouver la chaîne de sortie de façon **parallèle** en utilisant des **opérations vectorielles** pour calculer le vecteur caractéristique de chaque état.

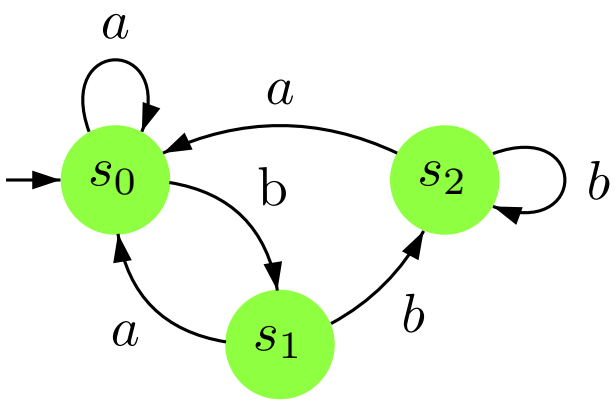


		b	a	b	b	a
$\boldsymbol{a}$	=	0	1	0	0	1
$\boldsymbol{b}$	=	1	0	1	1	0

$\boldsymbol{s_0} =$
$\boldsymbol{s_1} =$
$\boldsymbol{s_2} =$

$\boldsymbol{s_0} =$
$\boldsymbol{s_1} =$
$\boldsymbol{s_2} =$

Nouvelle Technologie: trouver la chaîne de sortie de façon **parallèle** en utilisant des **opérations vectorielles** pour calculer le vecteur caractéristique de chaque état.



		b	a	b	b	a
$\boldsymbol{a}$	=	0	1	0	0	1
$\boldsymbol{b}$	=	1	0	1	1	0

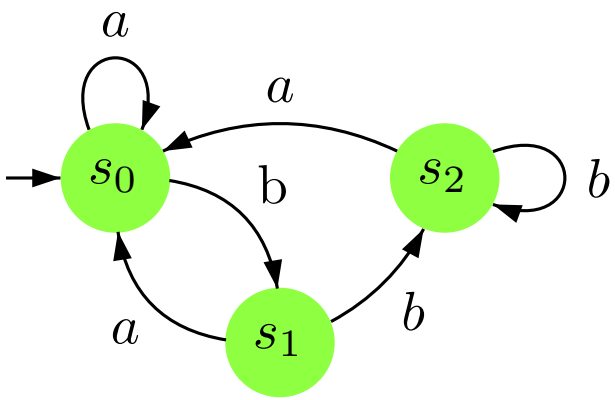
$\boldsymbol{s_0} = \boldsymbol{a}$

$\boldsymbol{s_1} =$

$\boldsymbol{s_2} =$

$\boldsymbol{s_0}$	=	0	1	0	0	1
$\boldsymbol{s_1}$	=					
$\boldsymbol{s_2}$	=					

Nouvelle Technologie: trouver la chaîne de sortie de façon **parallèle** en utilisant des **opérations vectorielles** pour calculer le vecteur caractéristique de chaque état.



		b	a	b	b	a
$\boldsymbol{a}$	$=$	0	1	0	0	1
$\boldsymbol{b}$	$=$	1	0	1	1	0

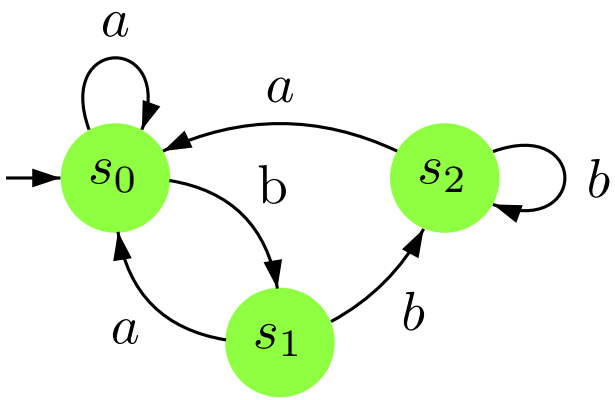
$\boldsymbol{s_0} = \boldsymbol{a}$

$\boldsymbol{s_1} = (\uparrow_1 \boldsymbol{s_0}) \wedge \boldsymbol{b}$

$\boldsymbol{s_2} =$

$\boldsymbol{s_0}$	$=$	0	1	0	0	1
$\boldsymbol{s_1}$	$=$	1	0	1	0	0
$\boldsymbol{s_2}$	$=$					

Nouvelle Technologie: trouver la chaîne de sortie de façon **parallèle** en utilisant des **opérations vectorielles** pour calculer le vecteur caractéristique de chaque état.



		b	a	b	b	a
$\boldsymbol{a}$	=	0	1	0	0	1
$\boldsymbol{b}$	=	1	0	1	1	0

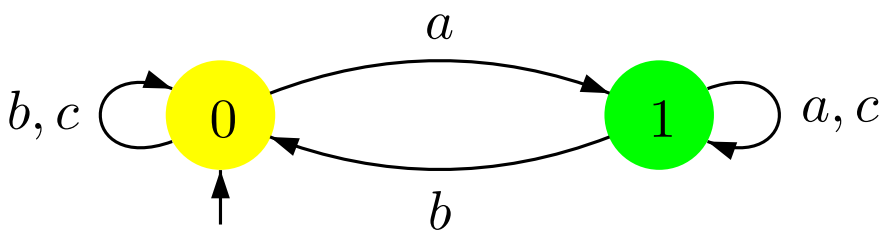
$\boldsymbol{s_0} = \boldsymbol{a}$

$\boldsymbol{s_1} = (\uparrow_1 \boldsymbol{s_0}) \wedge \boldsymbol{b}$

$\boldsymbol{s_2} = \neg\, (\boldsymbol{s_0} \vee \boldsymbol{s_1})$

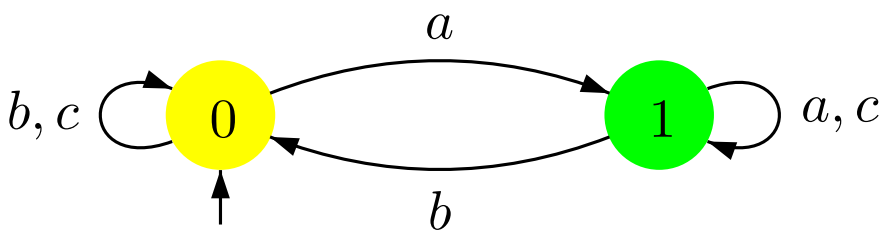
$\boldsymbol{s_0}$	=	0	1	0	0	1
$\boldsymbol{s_1}$	=	1	0	1	0	0
$\boldsymbol{s_2}$	=	0	0	0	1	0

Se souvenir d'événements passés: une **sortie** peut dépendre d'un **événement** arbitrairement **"éloigné"**...



$acccccccccccccc\,c$ $bccccccccccccc\,c$

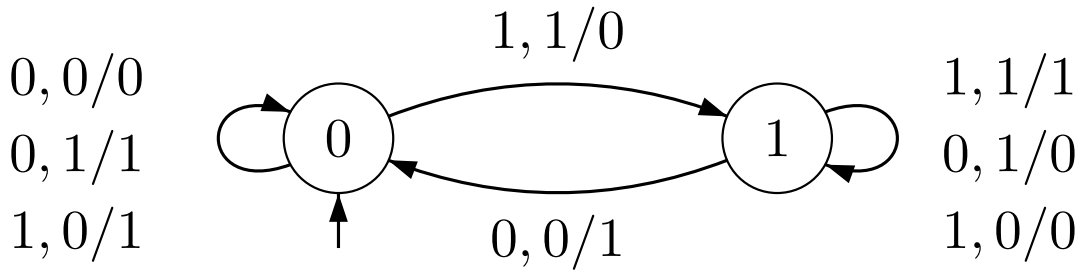
Se souvenir d'événements passés: une **sortie** peut dépendre d'un **événement** arbitrairement **"éloigné"**...



$acccccccccccccc\,c \rightarrow 1$

$bccccccccccccc\,c \rightarrow 0$

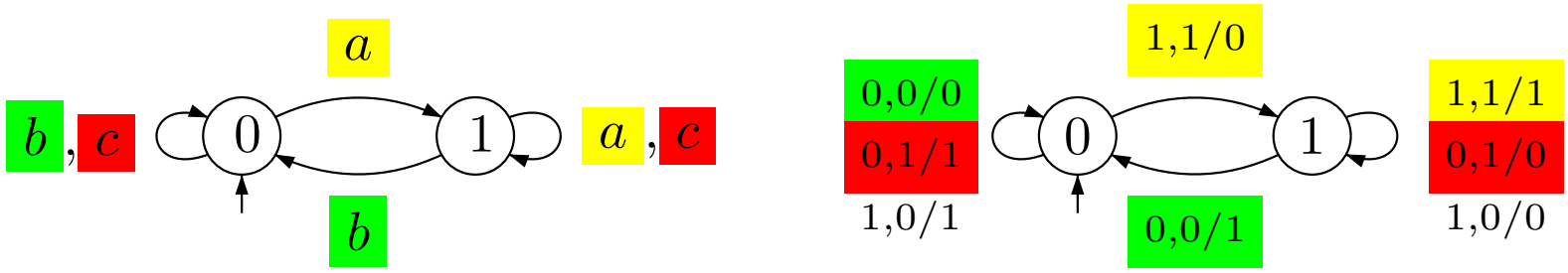
...mais il est encore possible de calculer, en parallèle, le vecteur caractéristique de chaque état en utilisant **l'automate d'addition binaire**:



Si deux vecteurs de bits $\boldsymbol{x_1}$ et $\boldsymbol{x_2}$ sont additionnés, la sortie sera **1** si

$$(\boldsymbol{x_1} \wedge \boldsymbol{x_2}) \vee [((\neg \boldsymbol{x_1} \wedge \boldsymbol{x_2}) \vee (\boldsymbol{x_1} \wedge \neg \boldsymbol{x_2})) \wedge \neg(\boldsymbol{x_1} +_b \boldsymbol{x_2})]$$

Maintenant, définissons $\boldsymbol{x_1 = a}$ et $\boldsymbol{x_2 = a \vee c}$:



Alors:

$a = \boldsymbol{x_1 \wedge x_2}$ = les deux bits sont 1

$b = \boldsymbol{\neg x_1 \wedge \neg x_2}$ = les deux bits sont 0

$c = \boldsymbol{\neg x_1 \wedge x_2}$ = le premier bit est 0 et le second est 1

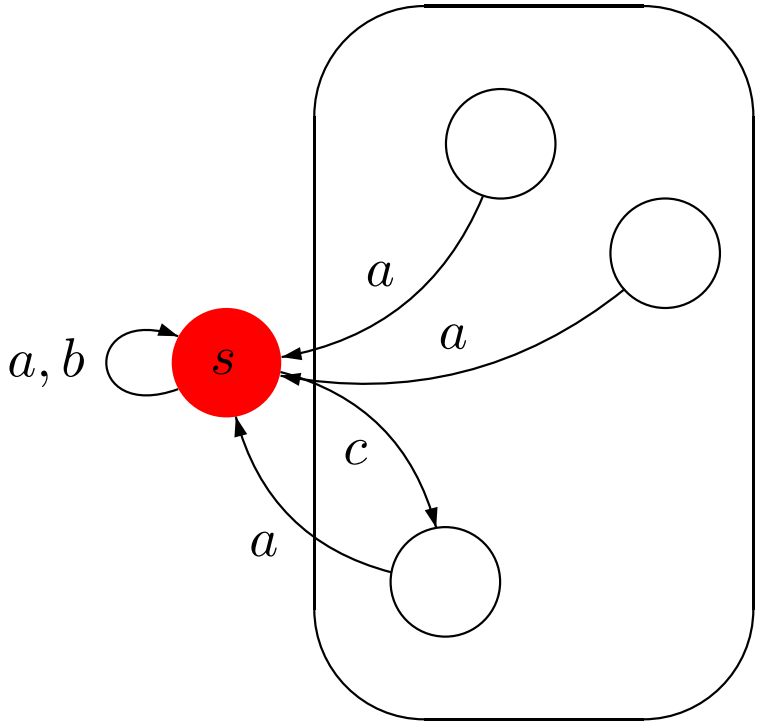
On obtient:

$$\boldsymbol{(r = 1) = a \vee [c \wedge \neg(a +_b (a \vee c))]}$$

De façon équivalente, on trouve

$$\boldsymbol{(r = 0) = b \vee [c \wedge (\neg b +_b \neg(b \vee c))]}$$

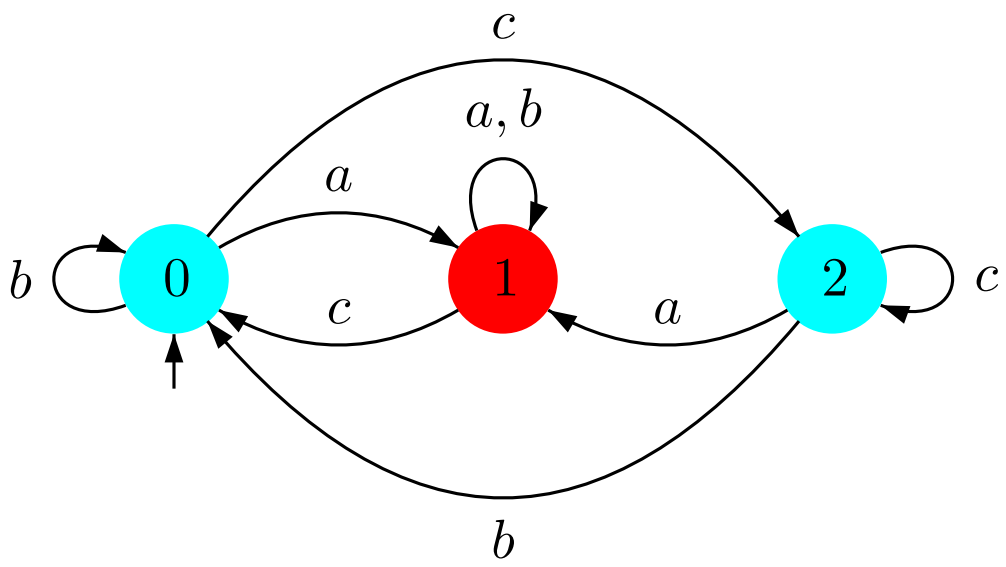
Résoudre des automates plus complexes: quels **automates** pouvons-nous **résoudre** en utilisant **nos techniques**?



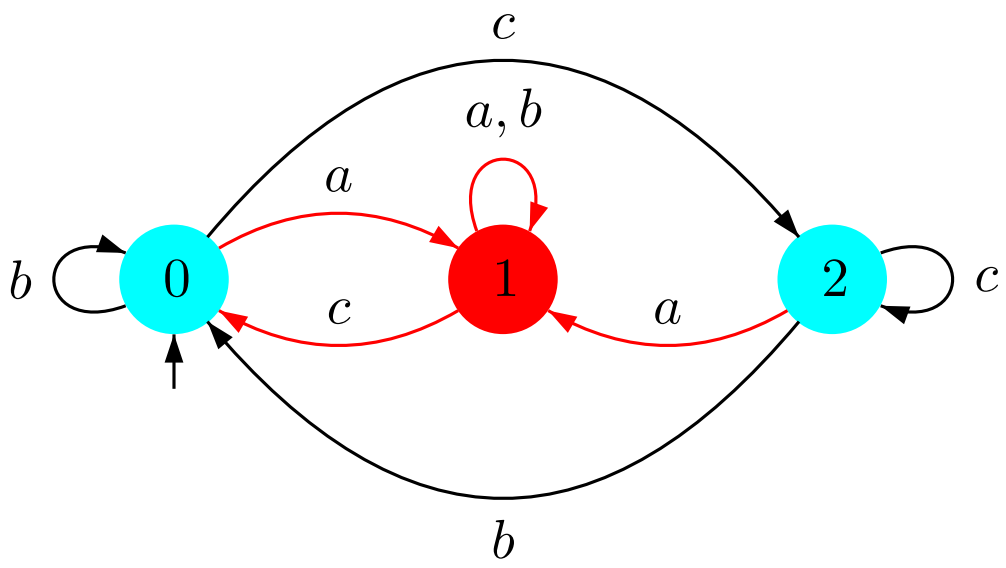
Quelques Définitions: soit $\mathcal{A}$ un automate complet déterministe sur l'alphabet Σ avec ensemble d'états Q et fonction de transition $Q \times \Sigma \xrightarrow{\delta} Q$.

Définition 1. **Un état s est effeuillable si, pour tout événement $x \in \Sigma$, on a**

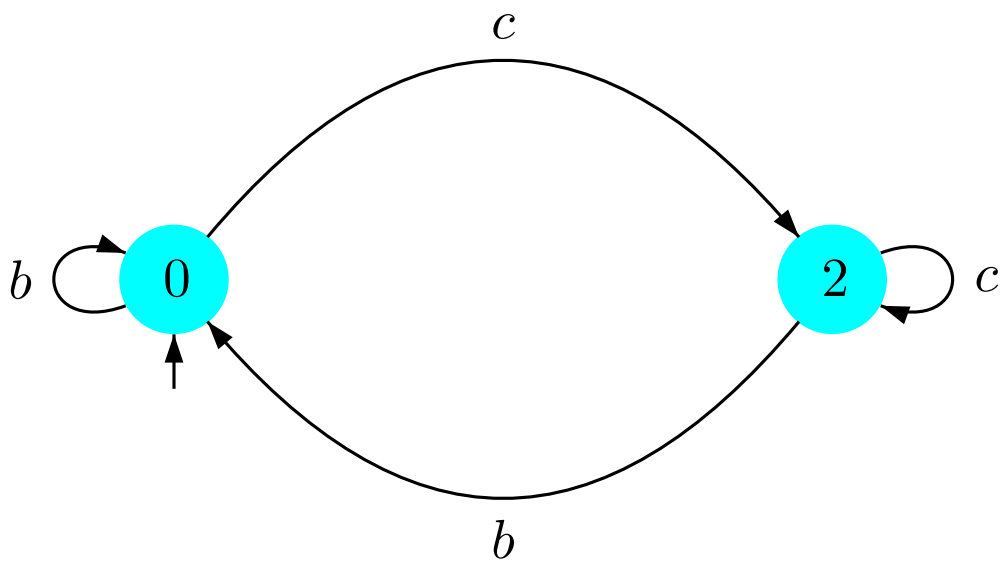
$$[\exists p \neq s \quad \text{tel que} \quad \delta(p, x) = s] \Rightarrow [\forall q \in Q, \delta(q, x) = s].$$



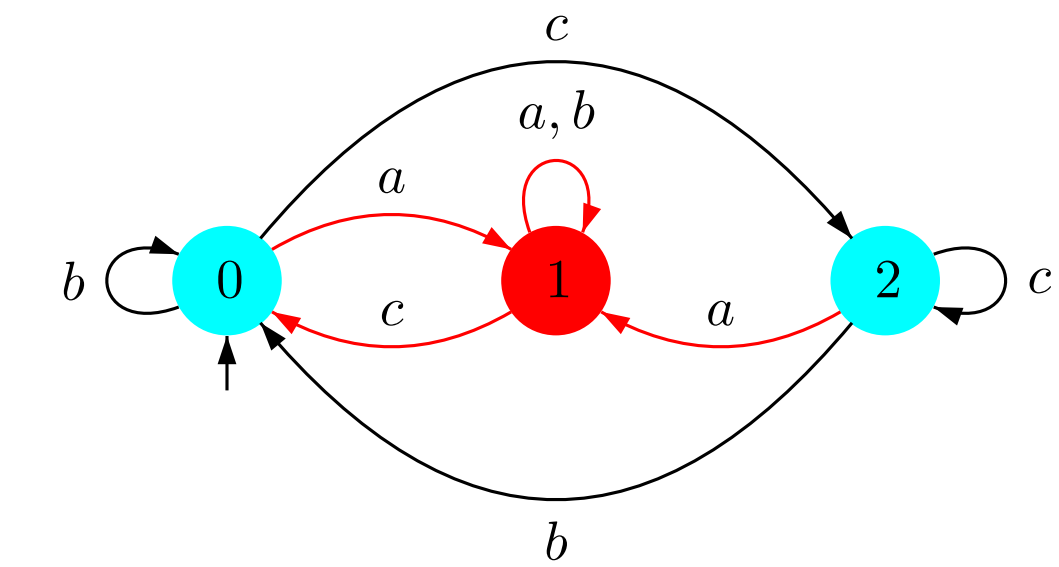
Definition 2. **Un automate $\mathcal{A}$ est *effeuillable* s'il a un seul état, ou s'il a un état effeuillable s et que $\mathcal{A} \setminus \{s\}$ est effeuillable.**

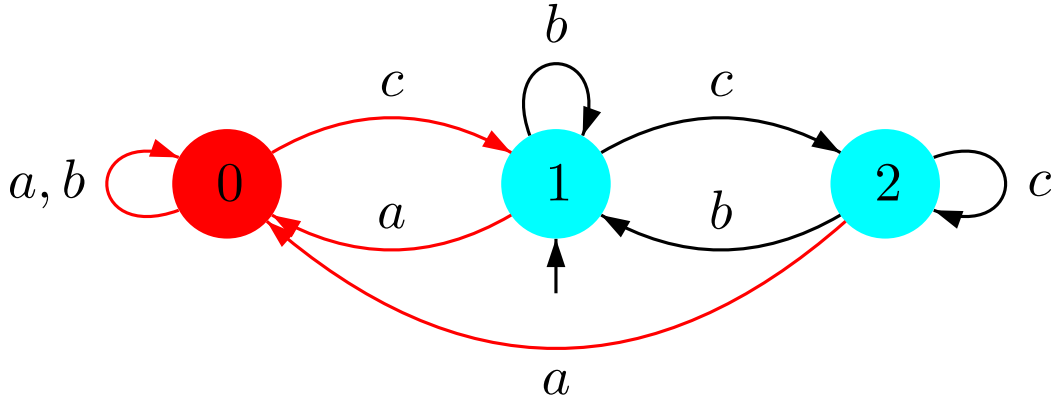


Definition 2. **Un automate $\mathcal{A}$ est *effeuillable* s'il a un seul état, ou s'il a un état effeuillable s et que $\mathcal{A} \setminus \{s\}$ est effeuillable.**



Ordre d'effeuillement:





Théorème 1. **Si un automate $\mathcal{A}$ est effeuillable, alors on peut trouver un algorithme vectoriel pour $\mathcal{A}$.**

Idée de la preuve:

1.

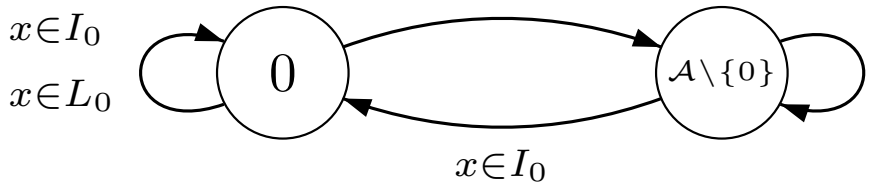
Idée de la preuve:

1. Ordonner les d états de l'automate $\mathcal{A}$ par ordre d'effeuillement et supposer que l'ensemble d'états est $Q = \{0, \ldots, d-1\}$.
2.

Idée de la preuve:

1. Ordonner les d états de l'automate $\mathcal{A}$ par ordre d'effeuillement et supposer que l'ensemble d'états est $Q = \{0, \ldots, d-1\}$.

2. On a alors la situation suivante:



   et le lemme de l'addition implique que

$$(\boldsymbol{r = 0}) = \begin{cases} \boldsymbol{I_0} \vee [\boldsymbol{L_0} \wedge (\neg \boldsymbol{I_0} + \neg(\boldsymbol{I_0} \vee \boldsymbol{L_0}))] & \text{ou} \\ \boldsymbol{I_0} \vee [\boldsymbol{L_0} \wedge \neg(\boldsymbol{I_0} + (\boldsymbol{I_0} \vee \boldsymbol{L_0}))] \end{cases}$$

   selon le fait que 0 soit l'état initial ou non.

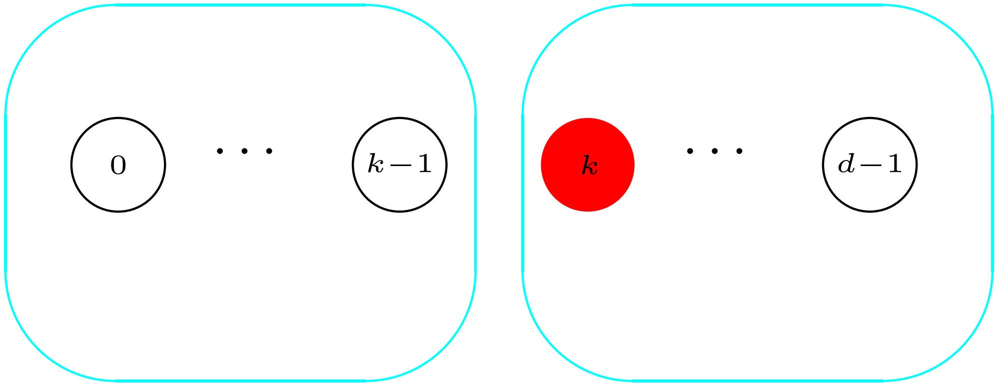
3.

Idée de la preuve (suite):

3. Calcul de $(\boldsymbol{r = k})$ pour $k \in \{1, \ldots d-2\}$:

Idée de la preuve (suite):

3. Calcul de $(\boldsymbol{r = k})$ pour $k \in \{1, \ldots d-2\}$:
   On connaît $\boldsymbol{K = (r < k)}$ et on veut trouver $(\boldsymbol{r = k})$:

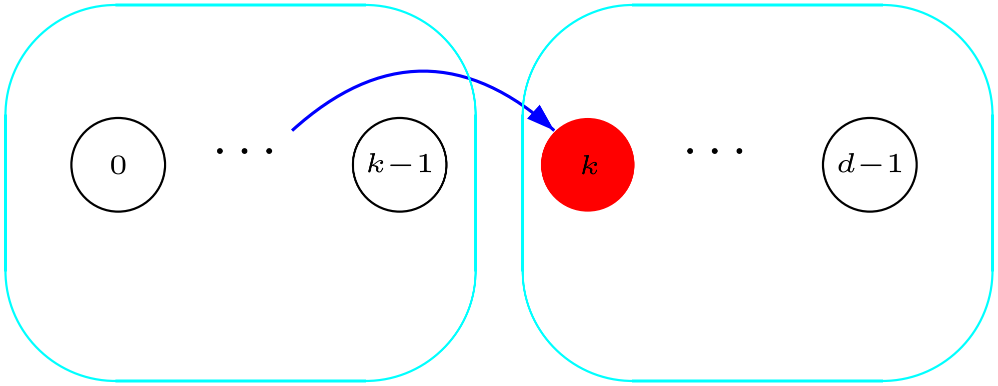


$\boldsymbol{N} \leftarrow [\qquad\qquad\qquad] \vee [\qquad\qquad\qquad]$

Idée de la preuve (suite):

3. Calcul de $(\boldsymbol{r = k})$ pour $k \in \{1, \ldots d-2\}$:

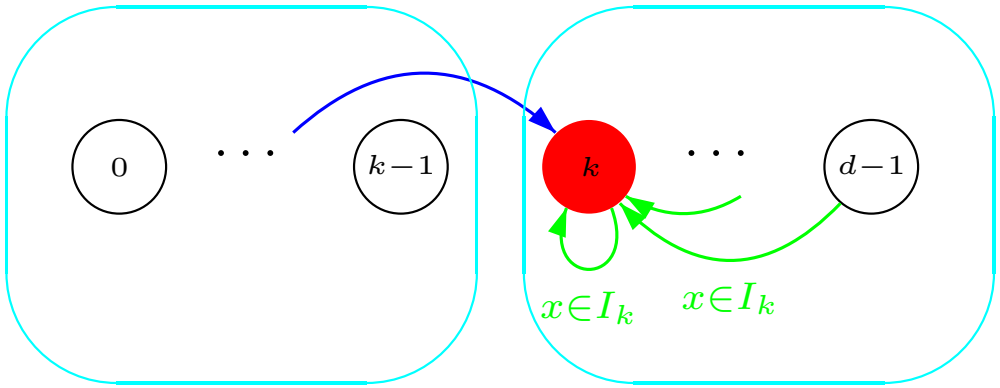
   On connaît $\boldsymbol{K = (r < k)}$ et on veut trouver $(\boldsymbol{r = k})$:



$\boldsymbol{N} \leftarrow [(\uparrow_i \boldsymbol{K}) \wedge (\boldsymbol{\sigma}(\uparrow_i \boldsymbol{r}, \boldsymbol{e}) = \boldsymbol{k})] \vee [\qquad\qquad\qquad\qquad]$

Idée de la preuve (suite):

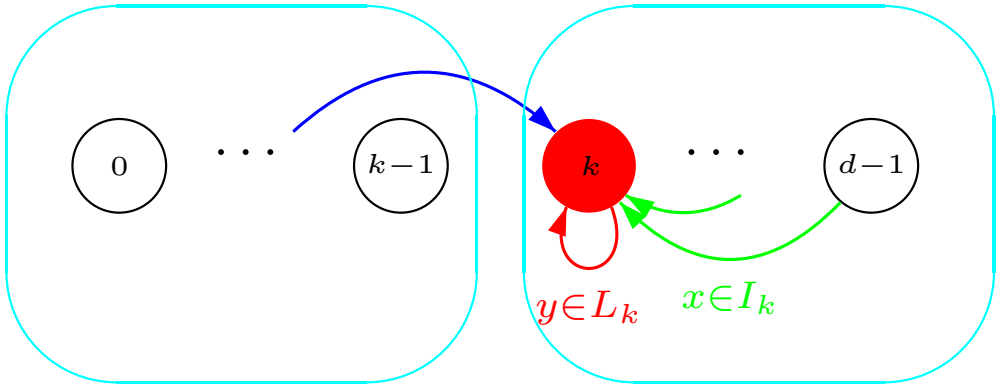
3. Calcul de $(\boldsymbol{r = k})$ pour $k \in \{1, \ldots d-2\}$:
   On connaît $\boldsymbol{K = (r < k)}$ et on veut trouver $(\boldsymbol{r = k})$:



$$\boldsymbol{N} \leftarrow [(\uparrow_i \boldsymbol{K}) \wedge (\boldsymbol{\sigma}(\uparrow_i \boldsymbol{r}, \boldsymbol{e}) = \boldsymbol{k})] \vee [\neg \boldsymbol{K} \wedge (\boldsymbol{e} \in \boldsymbol{I_k})]$$

Idée de la preuve (suite):

3. Calcul de $(\boldsymbol{r = k})$ pour $k \in \{1, \ldots d-2\}$:
   On connaît $\boldsymbol{K = (r < k)}$ et on veut trouver $\boldsymbol{(r = k)}$:



$$\boldsymbol{N} \leftarrow [(\uparrow_i \boldsymbol{K}) \wedge (\boldsymbol{\sigma}(\uparrow_i \boldsymbol{r}, \boldsymbol{e}) = \boldsymbol{k})] \vee [\neg \boldsymbol{K} \wedge (\boldsymbol{e} \in \boldsymbol{I_k})]$$

$$(\boldsymbol{r = k}) \leftarrow \begin{cases} \boldsymbol{N} \vee [\boldsymbol{L_k} \wedge (\neg \boldsymbol{N} + \neg(\boldsymbol{N} \vee \boldsymbol{L_k}))] & \text{ou} \\ \boldsymbol{N} \vee [\boldsymbol{L_k} \wedge \neg(\boldsymbol{N} + (\boldsymbol{N} \vee \boldsymbol{L_k}))] \end{cases}$$

$$\boldsymbol{K} \leftarrow \boldsymbol{K} \wedge (\boldsymbol{r = k})$$

4.

Idée de la preuve (suite):

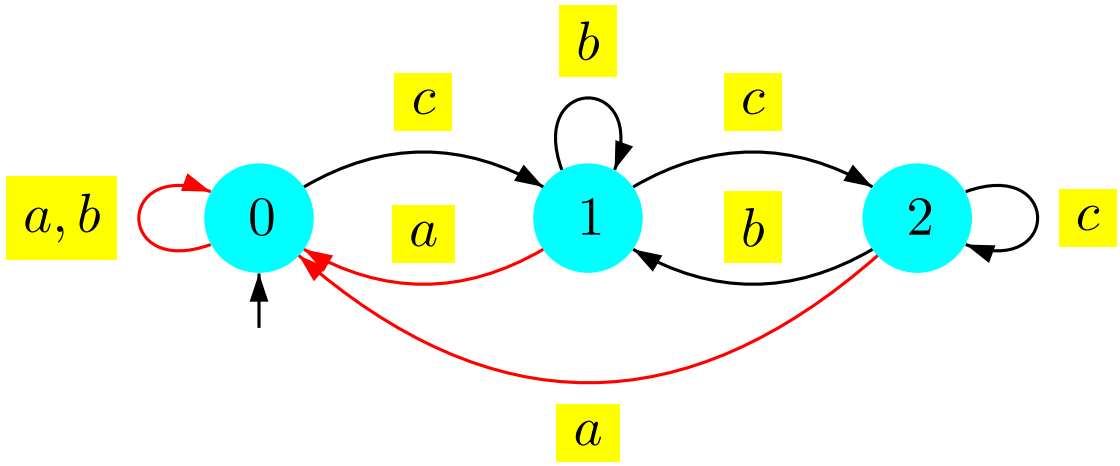
4. On calcule le vecteur caractéristique du dernier état:

$$(\boldsymbol{r = d - 1}) = \neg \boldsymbol{K}$$

Complexité: $\mathcal{O}(d|\Sigma|)$, où d est le nombre d'états de l'automate.

Exemple:

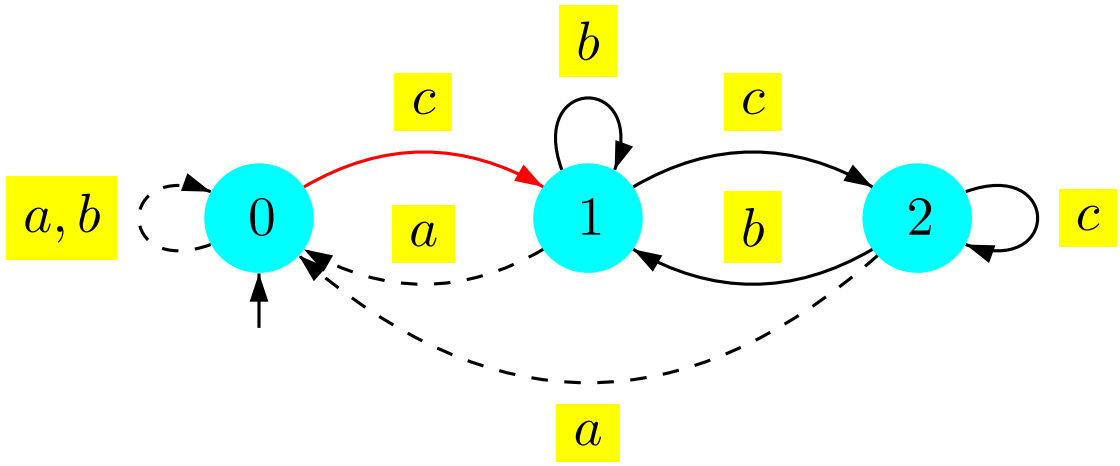
Étape 1. **Premier état effeuillable:** 0



a	b	b	c	b	a	c	c	b
0	0	0			0			

Exemple:

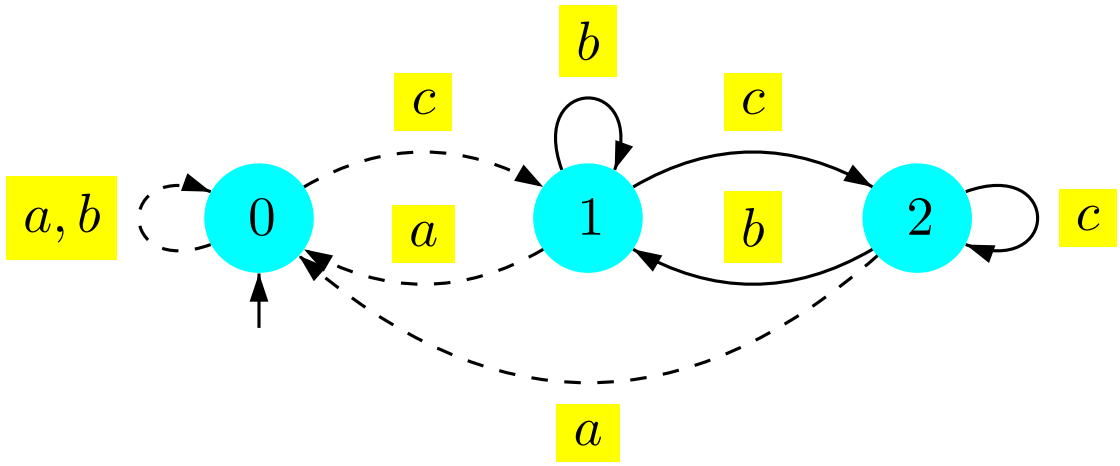
Étape 2. **"Voisins de 0"**



a	b	b	c	b	a	c	c	b
0	0	0	1		0	1		

Exemple:

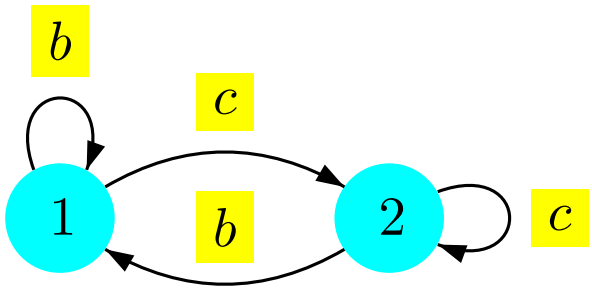
Étape 3. **"Étape récursive"**



a	b	b	c	b	a	c	c	b
0	0	0	1		0	1		

Exemple:

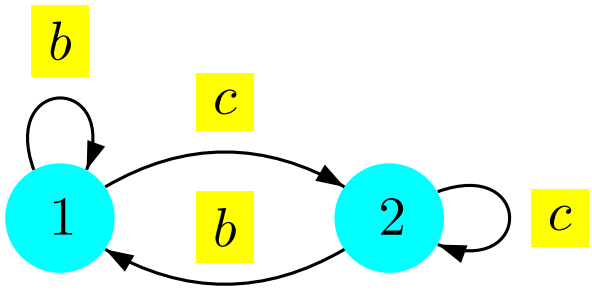
Étape 3. **"Étape récursive"**



a	b	b	c	b	a	c	c	b
0	0	0	1		0	1		

Exemple:

Étape 3. **"Étape récursive"**



a	b	b	c	b	a	c	c	b
0	0	0	1	1	0	1	2	1

Solution Classique: Calculer la matrice des distances $D[i,j]$

		A	C	G	T	A	A	T	A	G	C	...
	0	0	0	0	0	0	0	0	0	0	0	...
T	1	1	1	1	0	1	1	0	1	1	1	...
A	2	1	2	2	1	0	1	1	0	1	...	...
T	3	2	2	3	2	1	1	1	1	1	...	...
A	4	3	3	3	3	2	**1**	2	**1**	2	...	...
							↑		↑			

où

$$D[i,j] = \min \begin{cases} D[i-1,j-1] + \delta(p_i, t_j) \\ D[i,j-1] + 1 \\ D[i-1,j] + 1 \end{cases} \quad (1)$$

avec les conditions initiales $D[0,j] = 0$ et $D[i,0] = i$.

Nouvelle Solution: Calculer les distances avec un automate ...

Définition 3. **On définit les différences horizontales et verticales entre les éléments de la table des distances D de la façon suivante:**

$$\Delta v_{i,j} = D[i,j] - D[i-1,j],$$

$$\Delta h_{i,j} = D[i,j] - D[i,j-1].$$

Lemme 1. **Les différences horizontales et verticales sont bornées, c'est-à-dire**

$$\Delta v_{i,j} \in \{-1,0,1\},$$

$$\Delta h_{i,j} \in \{-1,0,1\}.$$

Connaître les différences horizontales nous permet de résoudre le problème...

		A	C	G	T	A	A	T	A	G	C	...
	0	0	0	0	0	0	0	0	0	0	0	...
		1										
T	1											...
		1										
A	2											...
		1										
T	3											...
		1										
A	4											...

car on sait que $D[m, 0] = m$ et alors

$$D[m, j] = D[m, j-1] + \Delta h_{m,j}.$$

Connaître les différences horizontales nous permet de résoudre le problème...

		A	C	G	T	A	A	T	A	G	C	...
	0	0	0	0	0	0	0	0	0	0	0	...
		1										
T	1	*										...
		1										
A	2	*										...
		1										
T	3	*										...
		1										
A	4	*										...

car on sait que $D[m, 0] = m$ et alors

$$D[m, j] = D[m, j-1] + \Delta h_{m,j}.$$

Notre algorithme:

Au départ, on a que $\mathbf{\Delta v_o} = 11\ldots1$.

Pour j de 1 à n faire

1. Calculer $\mathbf{\Delta h_j}$ avec un automate $\mathcal{B}$.
2. Calculer $\mathbf{\Delta v_j}$:

$$\mathbf{\Delta v_j} = \mathbf{\Delta h_j} + \mathbf{\Delta v_{j-1}} - \uparrow_0 \mathbf{\Delta h_j}$$

3. Calculer $D[m, j]$

$$D[m, j] = D[m, j-1] + \Delta h_{m,j}$$

Finalement, si $D[m, j] \leq t$ alors on a une occurrence de P en position j du texte.

$$D[i,j]=\min\begin{cases} D[i-1,j-1]+\delta(p_i,t_j) \\ D[i,j-1]+1 \\ D[i-1,j]+1 \end{cases} \Longrightarrow \Delta h_{i,j}=\min\begin{cases} \delta(p_i,t_j)-\Delta v_{i,j-1} \\ \Delta h_{i-1,j}-\Delta v_{i,j-1}+1 \\ 1 \end{cases}$$

On peut donc définir un automate $\mathcal{B}$ qui calculera la colonne de différences horizontales $\mathbf{\Delta h_j} = \Delta h_{1,j} \ldots \Delta h_{m,j}$ étant donné la séquence de couples suivante:

$$(\mathbf{\Delta v_{j-1}}, \delta(\mathbf{P}, \mathbf{t_j})) = ((\Delta v_{1,j-1}, \delta(p_1, t_j)), \ldots, (\Delta v_{m,j-1}, \delta(p_m, t_j)))$$

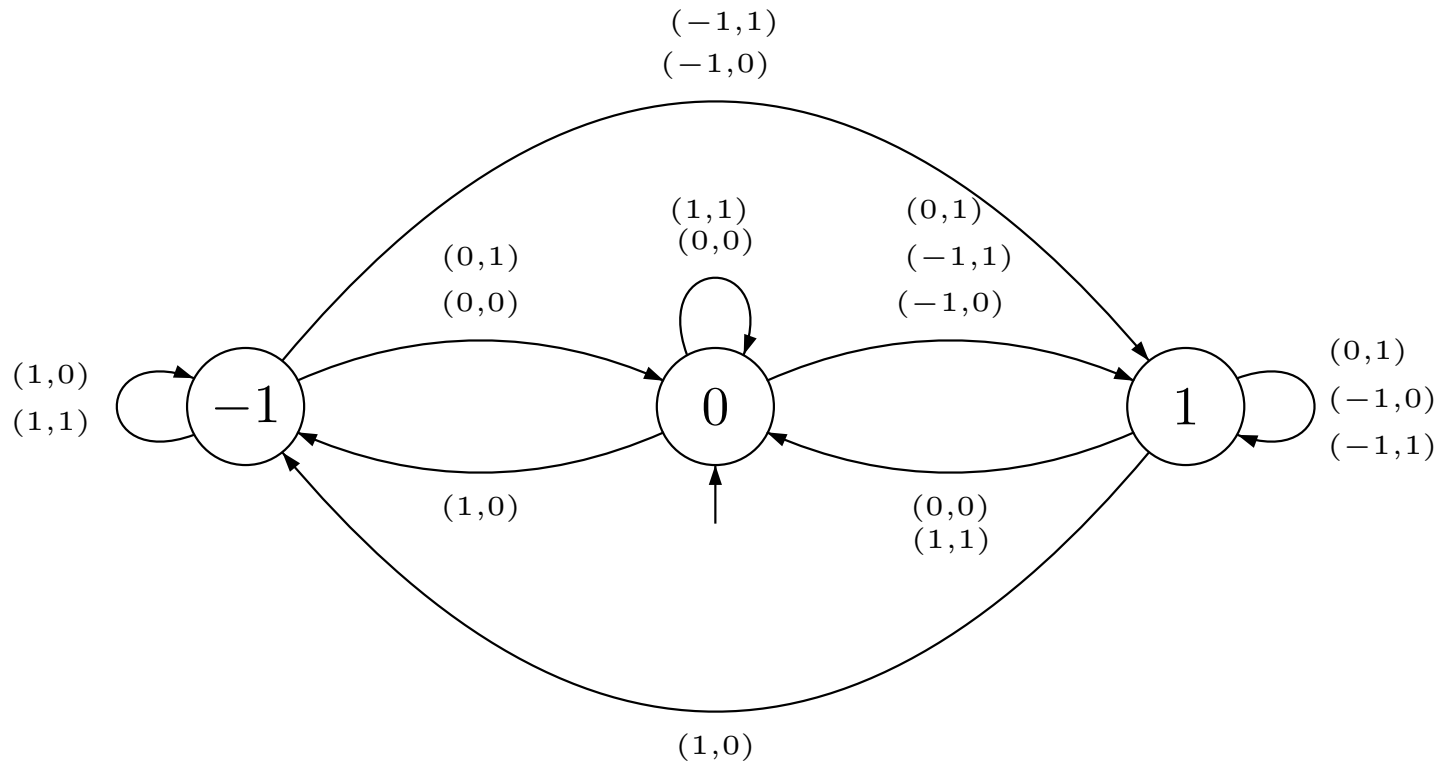
Les états de l'automate $\mathcal{B}$ sont $\{-1, 0, 1\}$, l'état initial est 0, et la fonction de transition σ de $\mathcal{B}$ est donné par:

$$\Delta h_{i,j} = \min \begin{cases} \delta(p_i, t_j) - \Delta v_{i,j-1} \\ \Delta h_{i-1,j} - \Delta v_{i,j-1} + 1 \\ 1 \end{cases} \Longrightarrow \quad \sigma(s, (\Delta v, \delta)) = \min \begin{cases} \delta - \Delta v \\ s - \Delta v + 1 \\ 1 \end{cases}$$

pour un événement $(\Delta v, \delta)$ dans le produit cartésien $\{-1, 0, 1\} \times \{0, 1\}$. On obtient la table de transitions suivante:

	(-1,0)	(-1,1)	(0,0)	(0,1)	(1,0)	(1,1)
-1	1	1	0	0	-1	-1
0	1	1	0	1	-1	0
1	1	1	0	1	-1	0

L'automate $\mathcal{B}$ permettant de calculer les colonnes de différences horizontales dans le cas où la distance utilisée est la distance d'édition est donc le suivant:



Théorème (Myers 99). **L'automate à 3 états permettant de résoudre le problème de la recherche approximative d'un mot dans un texte, avec l'emploi de la distance d'édition, est effeuillable.**

Theorem 2. **L'automate $\mathcal{B}$ à $2c + 2$ états permettant de résoudre le problème de la recherche approximative d'un mot dans un texte, avec l'emploi de la distance d'édition généralisée, est effeuillable.**