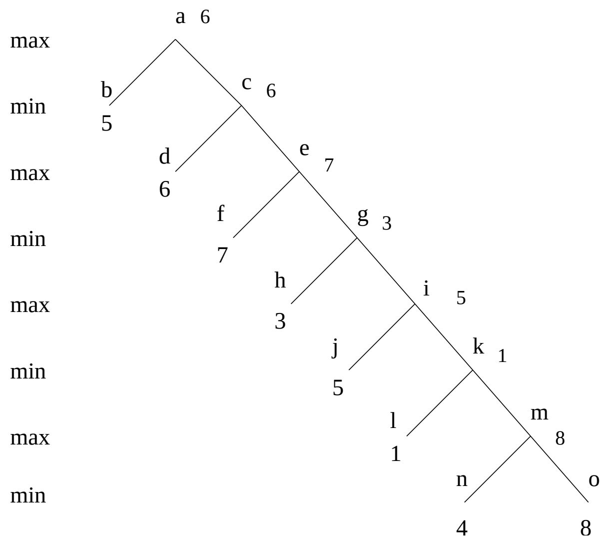
Corrigé

Exercice 1 *Le coût du chemin de D vers A est donné par $f = g + h$ pour le noeud A. On a $8 = g + 5$, donc le coût est 3. Pareil pour le chemin de D vers B. Ca donne 4. Ensuite on sait qu'à la première apparation de C on a $10 = g + 2$ donc, le chemin de D vers C en passant par B a coût 8, donc le chemin entre B et C a coût 4. Pour le chemin entre A et C on sait seulement que sa valeur est ≥ 5, puisque la valeur de f pour C ne change pas quand on dévéloppe A.*

De la même façon, on obtient le cout pour le chemin C vers E (4), le chemin de E vers F (2) et le chemin C vers F (7).

L'heuristique est adminissible. Elle ne sous-estime jamais.

Exercice 2

- 


- ```
  (a,v,$)
  (b,v,$)(c,v,$)
  (c,v,$)(b,f,5)
  (d,v,$)(b,f,5)
  (d,f,6)(b,f,5)
  (e,v,6)(b,f,5)
  (f,v,6)(g,v,6)(b,f,5)
  (f,f,6)(g,v,6)(b,f,5)
  (e,f,6)(b,f,5)
  (c,f,6)(b,f,6)
  (a,f,6)
  ```

- *Oui. On peut s'inspirer de l'arbre de l'exercice. On met pour b la valeur 5 et pour d la valeur 4. Et on peut prolonger l'arbre à droite autant qu'on veut. α-β va seulement regarder b et d.*

Exercice 3 *C1 est* 1 *ssi au moins 2 entrées sont activées. C2 est* 1 *ssi 3 entrées sont activées. C3 est* 1 *ssi au moins 1 entrées est activée. S1 est comme C1. S2 est* 1 *ssi exactement* 1 *ou* 3 *entrées sont activées. Donc S1 et S2 donne le nombre d'entrées activées en codage binaire.*

Exercice 4 *La sortie est donnée par*

$$o = w_1(x_1 + x_1^3) + w_2(x_2 + x_2^3) \cdots + w_n(x_n + x_n^3)$$

Il suffit de définir l'erreur sur un échantillon comme

$$E(\vec{w}) = \frac{1}{2} \sum_{(\vec{x}^s, c^s)} (c^s - o^s)^2$$
  ```

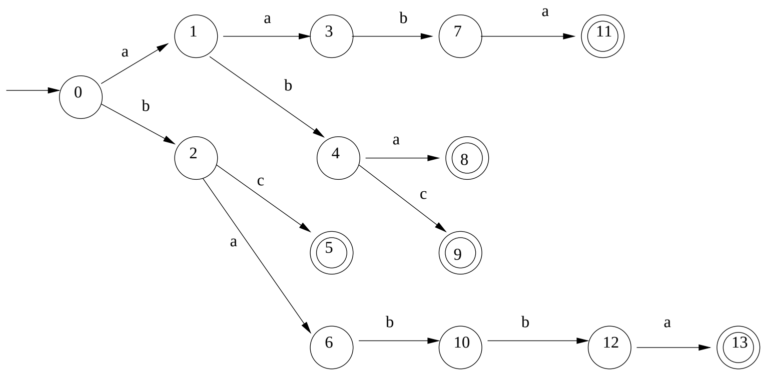
Ensuite on prend les dérivées partielles par rapport aux w_i. Ça donne:

$$\frac{\partial(\vec{w})}{\partial w_i}E = \sum_S (c^s - o^s)(-x_i^s - (x_i^s)^3)$$

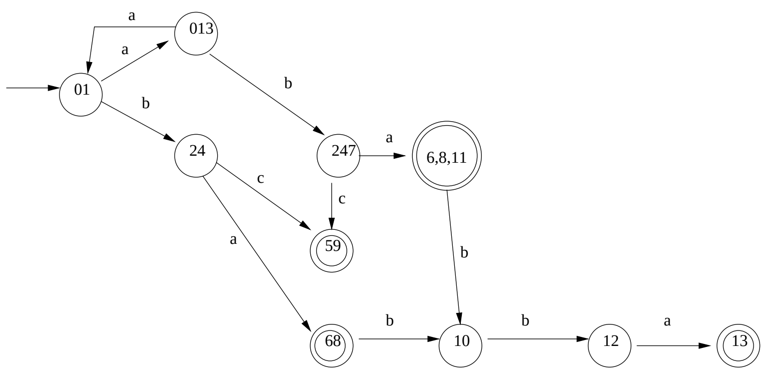
Et après on obtient facilement un algorithme d'apprentissage en remplaçant dans l'algorithme d'apprentissage par descente du gradient le calcul du Δw_i par $\Delta w_i \leftarrow \Delta w_i + \epsilon(c^s - o^s)(x_i^s + (x_i^s)^3)$.

Exercice 5

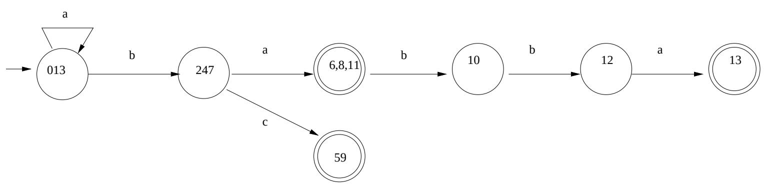
- *On commence avec:*



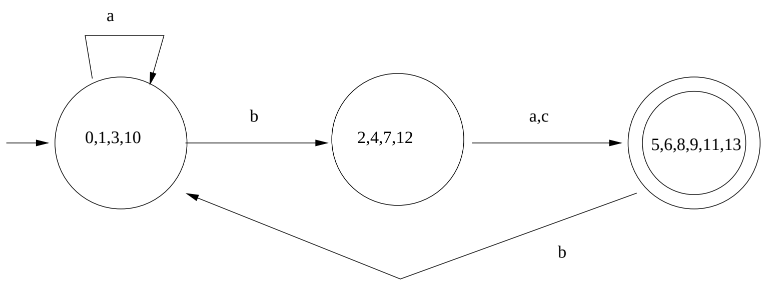
Ensuite on fusionne 0 *et* 1 *(Ils ont les mots aba et bc en commun) et on détérminise:*



Ensuite on fusionne 3 *avec* 01 *et on détérminise :*

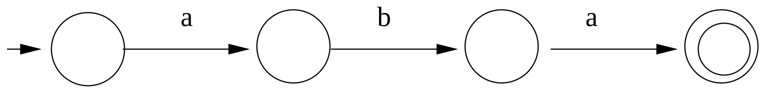


On fusionne 5 avec 6 (ils ont ϵ en commun) et on fusionne 10 avec 0 (ils ont ba en commun), on détérminise et ça donne le résultat final :

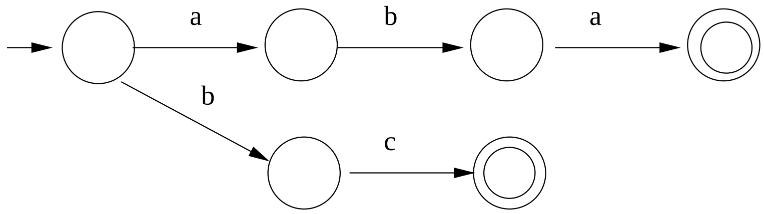


- *On ajoute un mot w : Si $w \in L(A)$ on ne fait rien. Sinon, on suit les états successivement obtenu en essayant d'accepter le mot w avec l'automate, jusqu'à ce qu'on est bloqué. Ensuite on ajoute à l'automate des nouveaux états permettant d'accepter w. On applique ensuite ZR sur l'automate obtenu.*

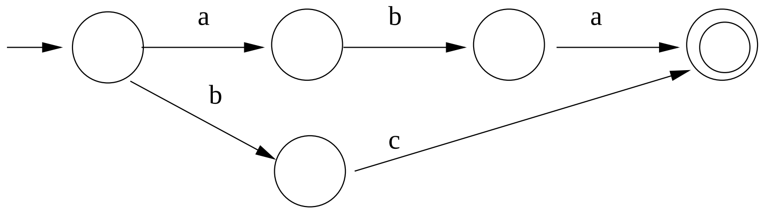
- *On commence avec aba:*



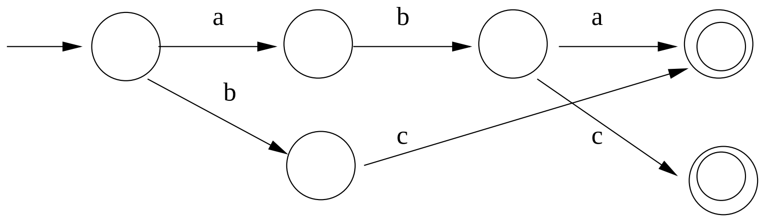
On ajoute bc:



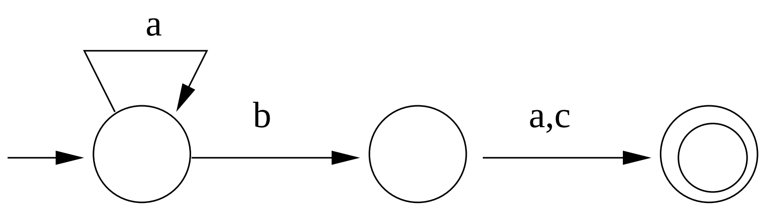
On applique ZR :



On ajoute abc :



Appliquer ZR donne :



Le mot aaba est déjà accepté. On ajoute babba et on obtient avec ZR ensuite le même automate que le résultat de l'exercice précédent.