

Exercice 5.19

October 20, 2023

Problème considéré:

- Instance: deux mots w_1, w_2 de tailles respectives n_1 et n_2
- Question: trouver le plus long sous-mots commun entre w_1 et w_2

Ce problème de s'écrit pas facilement sous la forme d'un système dynamique discret contrairement aux autres exemples vu en cours et dans les exercices. On peut tout de même la résoudre via programmation dynamique en définissant la fonction valeur $V(i, j)$ définie par la longueur du plus long sous-mot commun entre $w_1[1 : i]$ et $w_2[1 : j]$, i.e. la restriction de des deux mots à leurs i et j premières lettres respectivement. Selon cette définition, Nous souhaitons donc calculer $V(n_1, n_2)$, ce qui peut se faire via les équations de récurrence suivantes:

- Initialisation : $\forall i, V(i, 0) = 0$, et $\forall j, V(0, j) = 0$
- $$V(i, j) = \begin{cases} 1 + V(i - 1, j - 1), & \text{si } w_1[i] = w_2[j] \\ \max(V(i, j - 1), V(i - 1, j)), & \text{si } w_1[i] \neq w_2[j] \end{cases}$$

Remarque : dans le cas des systèmes dynamiques discrets, en général on remplit un tableau indexé par l'état s et le temps t ligne par ligne pour chaque pas de temps, avec une ligne qui dépend uniquement des informations de la ligne précédente. Ici une case se remplit en utilisant les informations de la ligne précédente, mais aussi de la colonne précédente.

Voici un exemple de la programmation dynamique appliquée à l'instance $(ABCD, AEBBAC)$:

	0	1 (A)	2 (E)	3 (B)	4 (B)	5 (A)	C
0	0	0	0	0	0	0	0
1 (A)	0	1	1	1	1	1	1
2 (B)	0	1	1	2	2	2	2
3 (C)	0	1	1	2	2	2	3
4 (D)	0	1	1	2	2	2	3

Le chemin en rouge sont les "argmin" gardés en mémoire lors de l'exécution de l'algo, ce qui permet de reconstruire la solution optimale après coup: ABC .