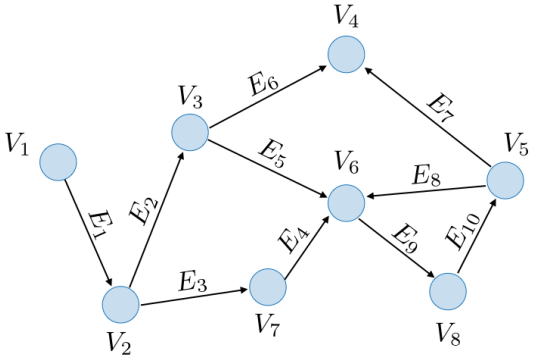
2.1.2 Parcours de graphe

Cette catégorie d'algorithmes basés sur les états vise à trouver des chemins optimaux avec une complexité moins grande qu'exponentielle. Dans cette section, nous allons nous concentrer sur la programmation dynamique et la recherche à coût uniforme.

❒ **Graphe** – Un graphe se compose d'un ensemble de sommets V (aussi appelés nœuds) et d'arêtes E (appelés arcs lorsque le graphe est orienté).



Remarque : un graphe est dit être acyclique lorsqu'il ne contient pas de cycle.

❒ **État** – Un état contient le résumé des actions passées suffisant pour choisir les actions futures de manière optimale.

❒ **Programmation dynamique** – La programmation dynamique (en anglais *dynamic programming* ou *DP*) est un algorithme de recherche de type retour sur trace qui utilise le principe de mémoïsation (i.e. les résultats intermédiaires sont enregistrés) et ayant pour but de trouver le chemin à coût minimal allant de l'état s à l'état final s_{end}. Cette procédure peut potentiellement engendrer des économies exponentielles si on la compare aux algorithmes de parcours de graphe traditionnels, et a la propriété de ne marcher que dans le cas de graphes acycliques. Pour un état s donné, le coût futur est calculé de la manière suivante :

$$\text{FutureCost}(s) = \begin{cases} 0 & \text{if IsEnd}(s) \\ \min\limits_{a \in \text{Actions}(s)} \big[\text{Cost}(s,a) + \text{FutureCost}(\text{Succ}(s,a))\big] & \text{otherwise} \end{cases}$$