PLUS COURT CHEMIN INTER
Entree: Depart-> Vecteur d'etats
Resultat: Modifier l'attribut Pere de chaque etat avec son etat pere dans l'automate
Variable: Etats-> Vecteur d'etats
Algorithme:
si l'attribut pere de l'etat final est null
pour i de 0 a taille de vecteur Depart
pour j de 0 a taille lettres alphabet
si existe transition(lettre j, Depart[i])
ajouter transition(lettre j, Depart[i]) au vecteur Etats
si l'attribut pere de transition(lettre j, Depart[i]) est null
marque cet attribut avec Depart[i]
appel recursif de la fonction avec comme parametre le vecteur Etats

PLUS COURT CHEMIN
Entree: rien
Resultat: Mot ->String qui contient les ettiquetes du plus court chemin
Variable: petitchemin -> String initial-> Vecteur initialise avec l'etat initial Temp-> etat
Algorithme:
si il existe un chemin
appel initial de la fonction Plus court chemin Inter avec le vecteur initial pour marquer les attributs pere des etats
Temp <- etat final
tant que Temp n'est pas l'etat initial
pour i de 0 a taille ensembleGlobal (il contient l'ensemble d'etats de l'automate)
pour j de 0 a taille lettres de l'alphabet
si transition(lettre j, ensembleGlobal[i])=Temp ET ensembleGlobal[i] est le pere de Temp
incrementer la variable petitchemin avec la lettre alphabet[j]
temp <- ensembleGlobal[i]
inverse petitchemin et la renvoyer