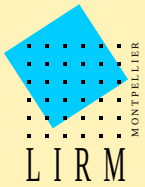
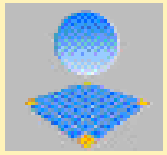
Alignement de séquences : application à la comparaison de chaînes d'ADN

Sèverine Bérard





Laboratoire d'Informatique, de Robotique
et de Micro-électronique de Montpellier
CNRS - Université Montpellier II
FRANCE

Pourquoi comparer des séquences ?

- Biologie : identification de nouvelles séquences, de leur fonction
- Traitement du langage naturel : reconnaissance vocale
- Correction d'erreurs
  String-to-string correction problem [Wagner & Fisher 74]
- ...

Alignement

- Pour comparer des séquences on construit un *alignement*

Ex :

A	G	G	T	C	A
|		|		|	|
A	-	G	C	C	A

- 3 opérations : *Insertion* (I), *Délétion* (D), *Mutation* (M)

- Coût d'un alignement = somme des coûts des opérations qui le composent

- Distance entre 2 séquences = coût minimum d'alignement

Notations

- Deux séquences S et R, de longueur respective n et m
- $S[i]$ = Caractère à la position i de S
- S_i = Préfixe de longueur i de S

Exemple : S = A B C D E

- $S[1]$=A ; $S[2]$=B ; . . . ; $S[5]$=E
- S_1=A ; S_3 = A B C ; S_5 = S

Programmation dynamique

*P*rincipe : construire progressivement une matrice dans laquelle chaque case (i, j) contient la distance entre S_i et R_j

- La case (n, m) contient la distance entre les deux séquences
- Pour obtenir un alignement entre S_i et R_j, il suffit de considérer les trois alignements suivants :
  - S_{i-1} et R_{j-1} et d'ajouter la mise en correspondance de $S[i]$ et $R[j]$
  - S_i et R_{j-1} et d'ajouter l'insertion de $R[j]$
  - S_{i-1} et R_j et d'ajouter le délétion de $S[i]$

Matrice de programmation dynamique

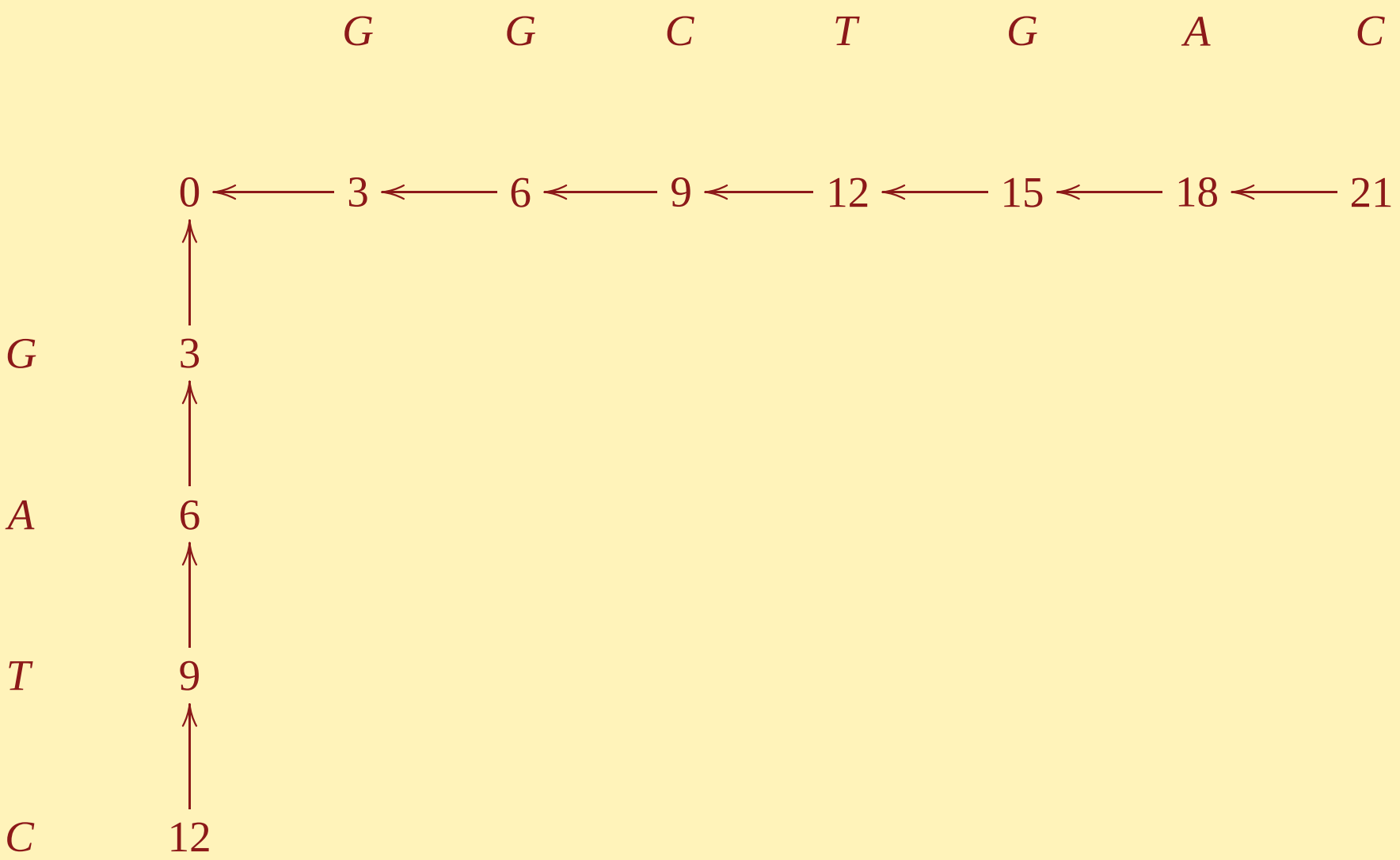
- Soit M la matrice
- Initialisation : $M(0,0) = 0$; $M(i,0) = i * D$; $M(0,j) = j * I$
- $$M(i,j) = min \begin{cases} M(i-1,j-1) + \mathrm{s}(S[i],R[j]) \\ M(i,j-1) + I \\ M(i-1,j) + D \end{cases}$$

$$\text{avec s}(S[i],R[j]) = \begin{cases} 0 \text{ si } S[i] = R[j] \\ M \text{ sinon} \end{cases}$$

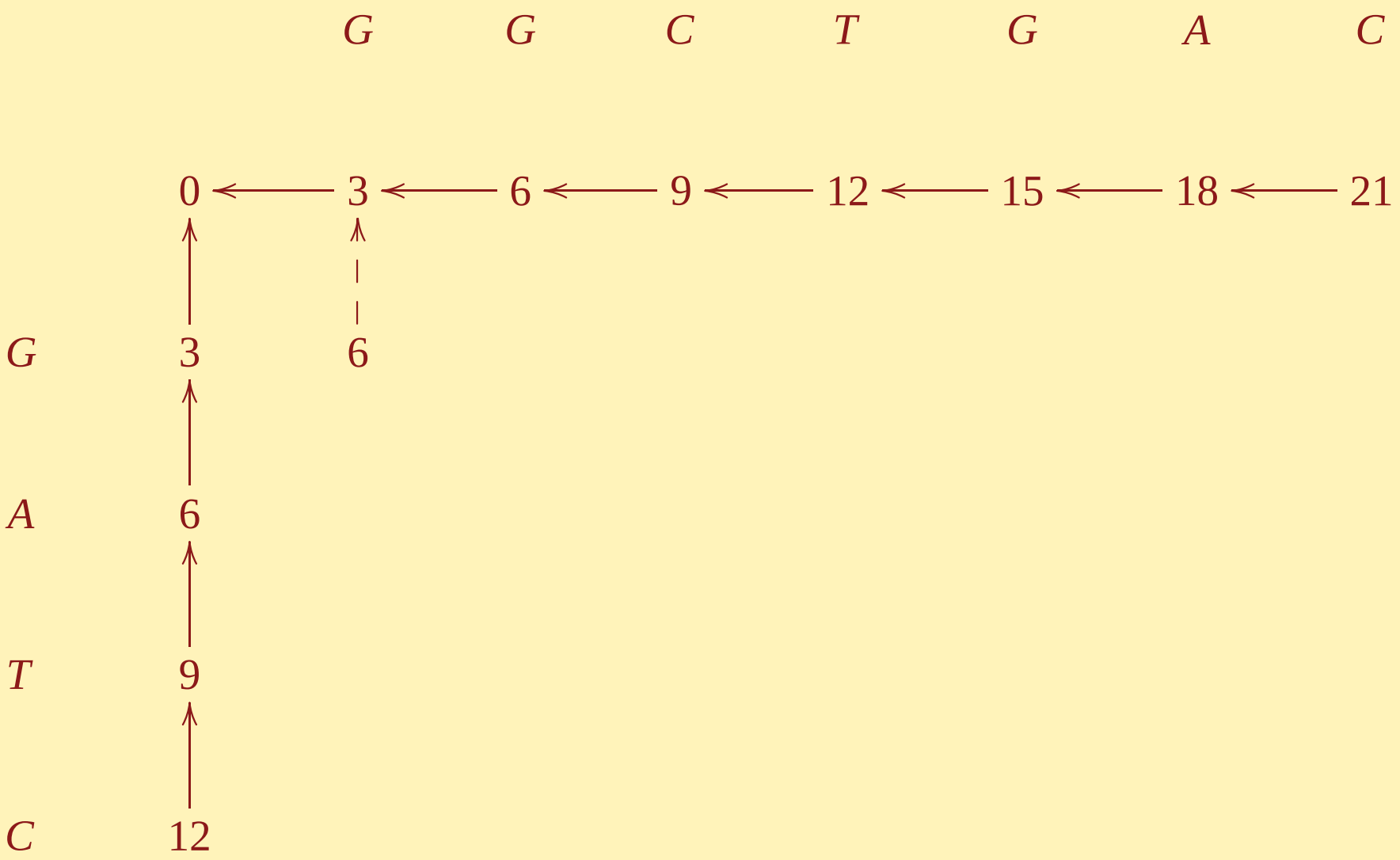
Un exemple :

- S = G A T C
- R = G G C T G A C
- $I = D = M = 3$

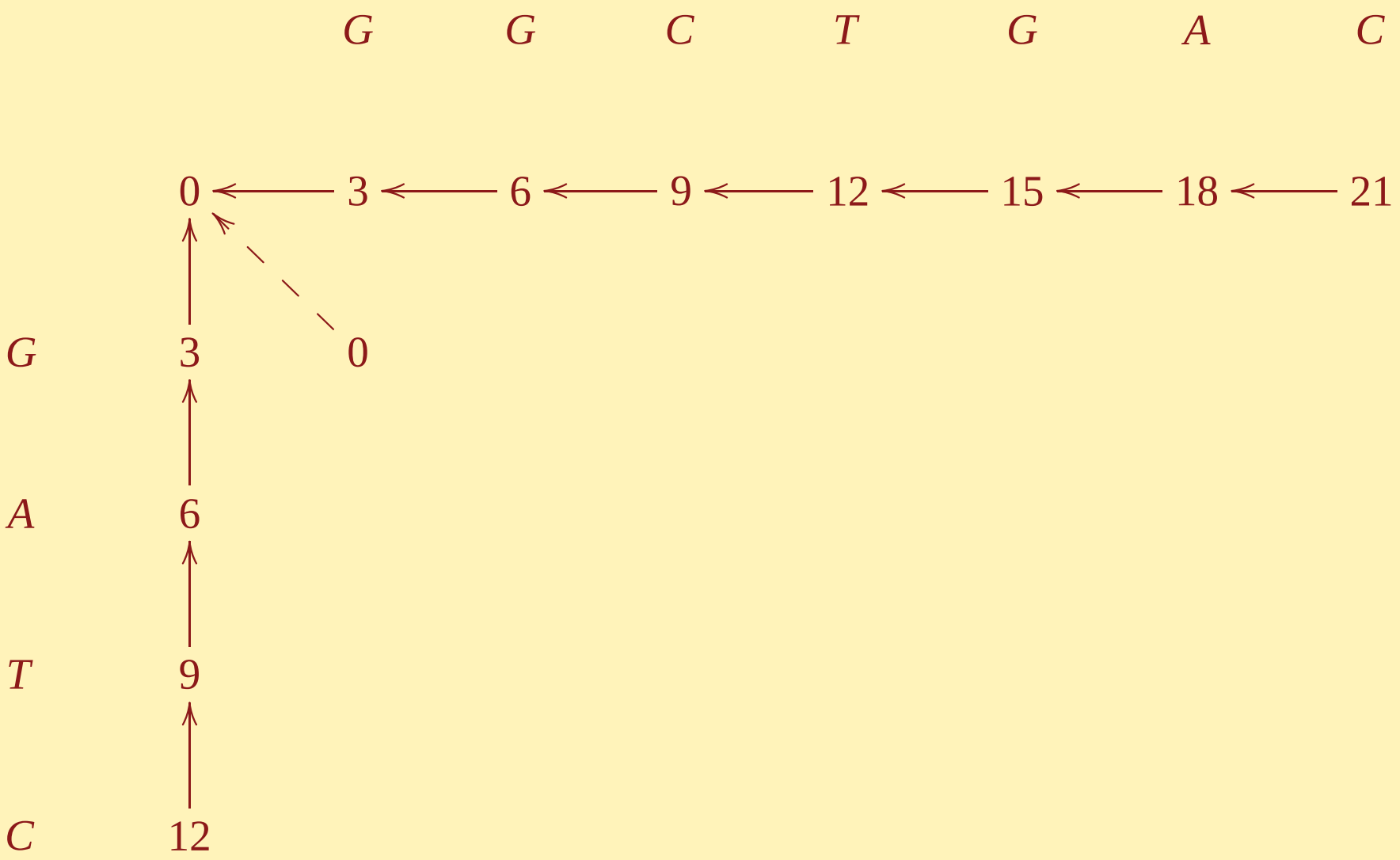
Matrice de programmation dynamique (Initialisation)



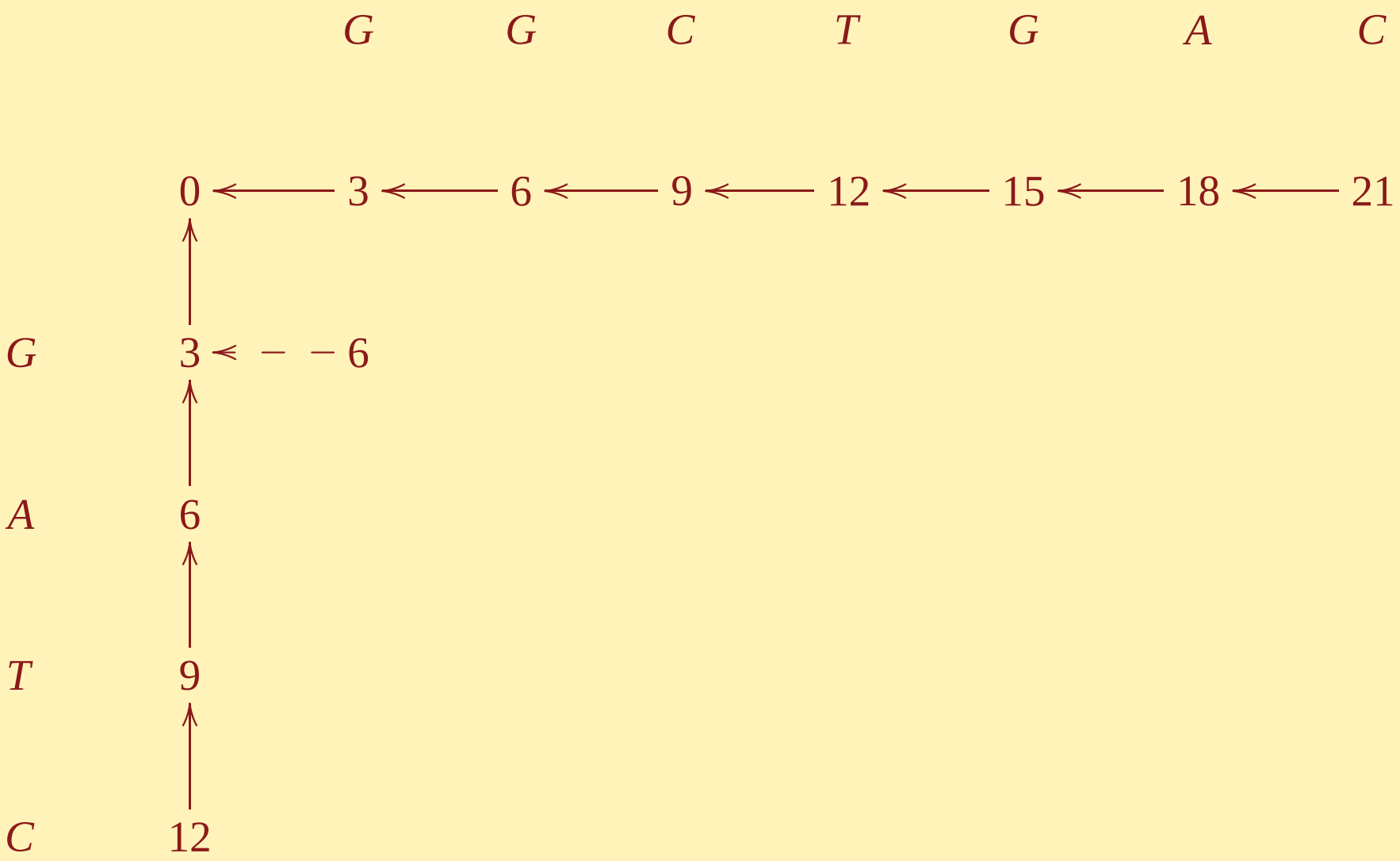
Matrice de programmation dynamique



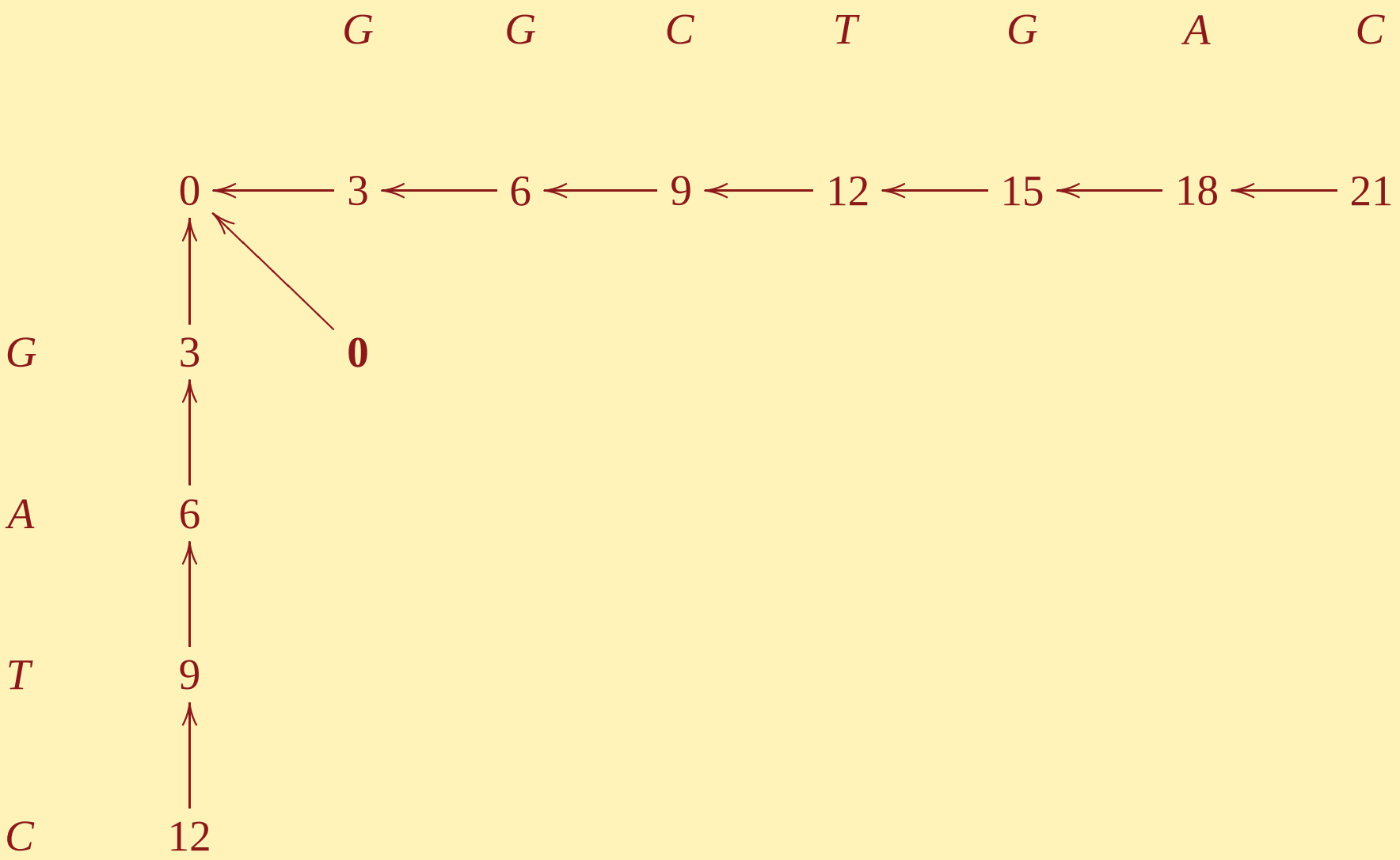
Matrice de programmation dynamique



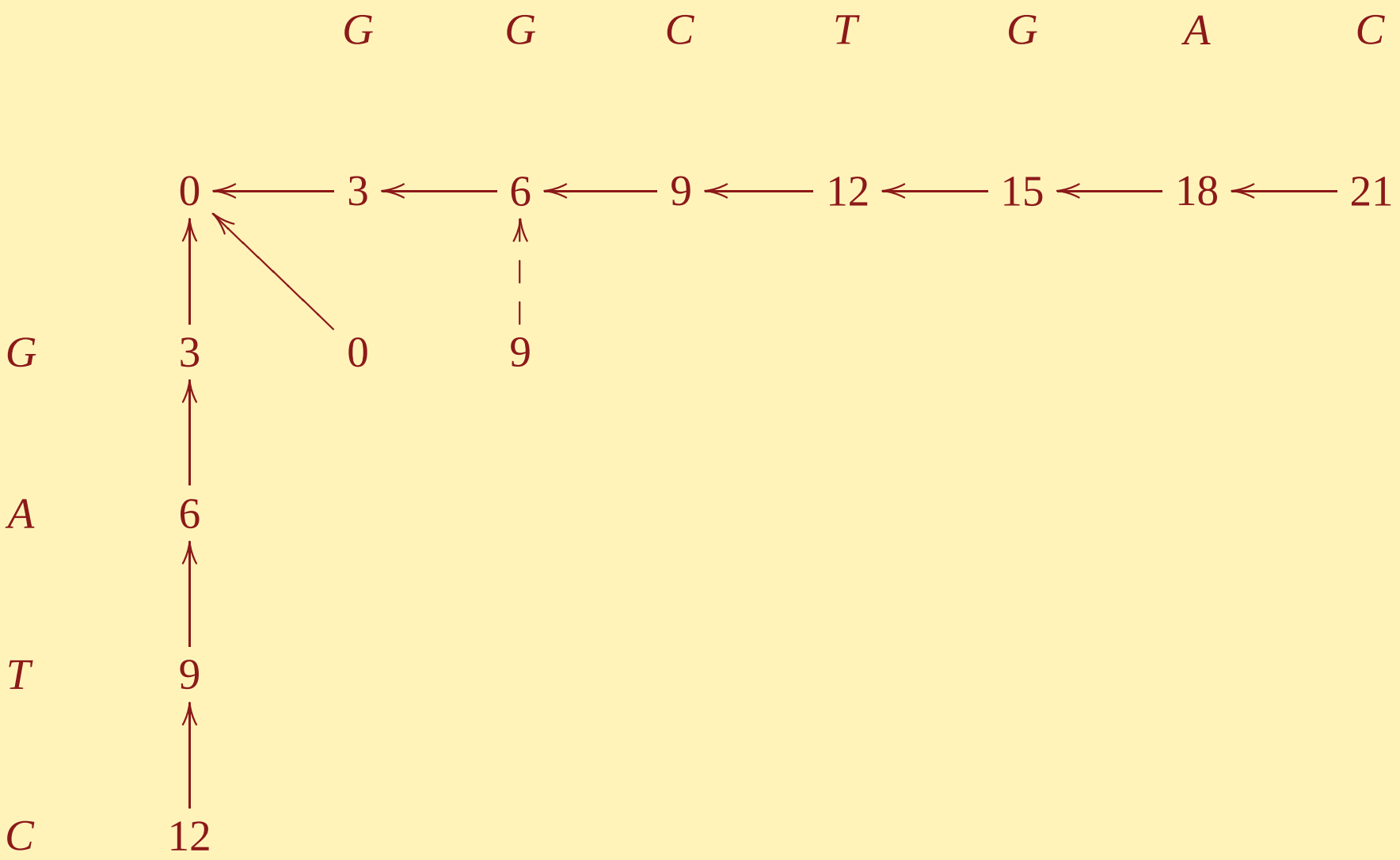
Matrice de programmation dynamique



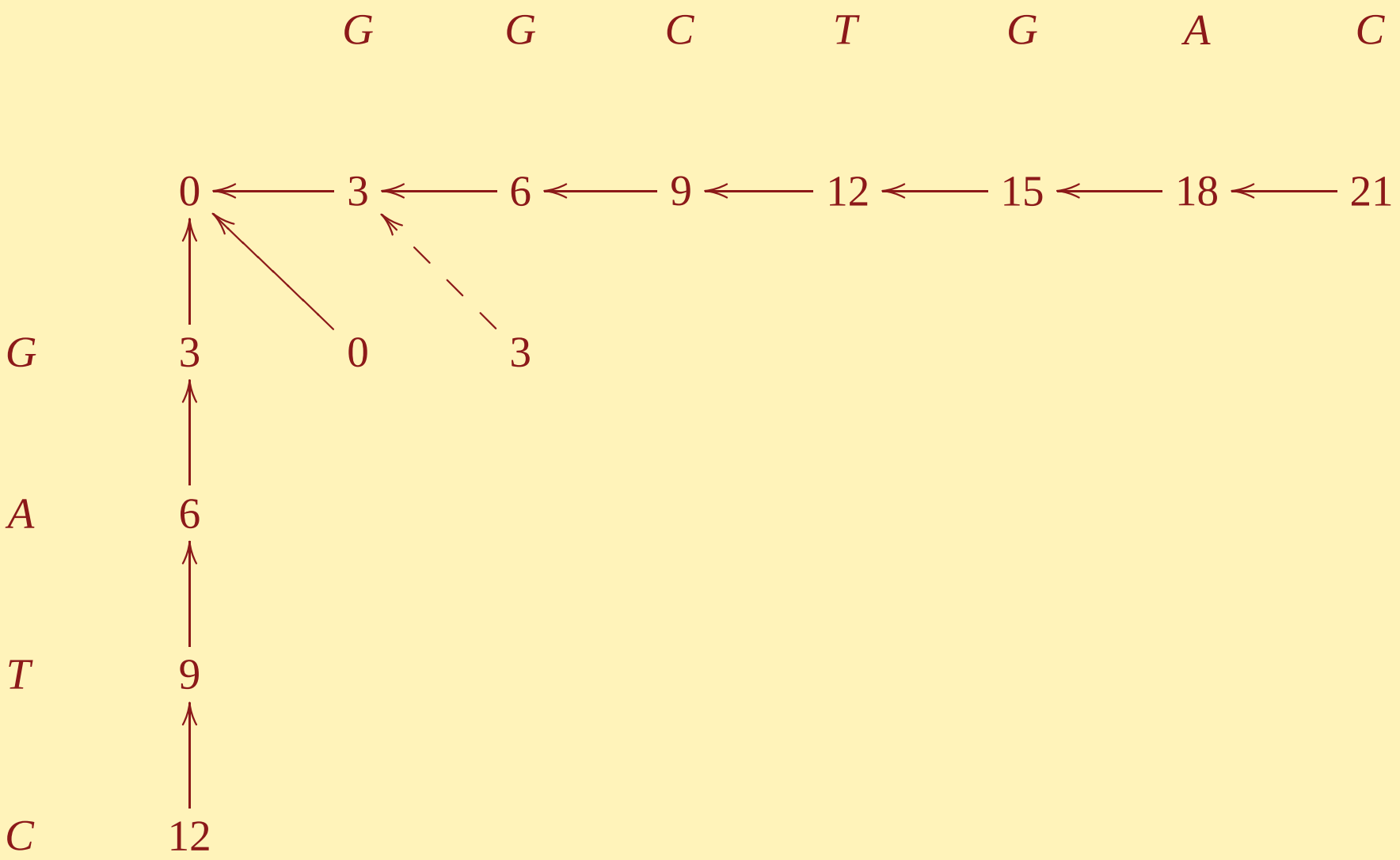
Matrice de programmation dynamique



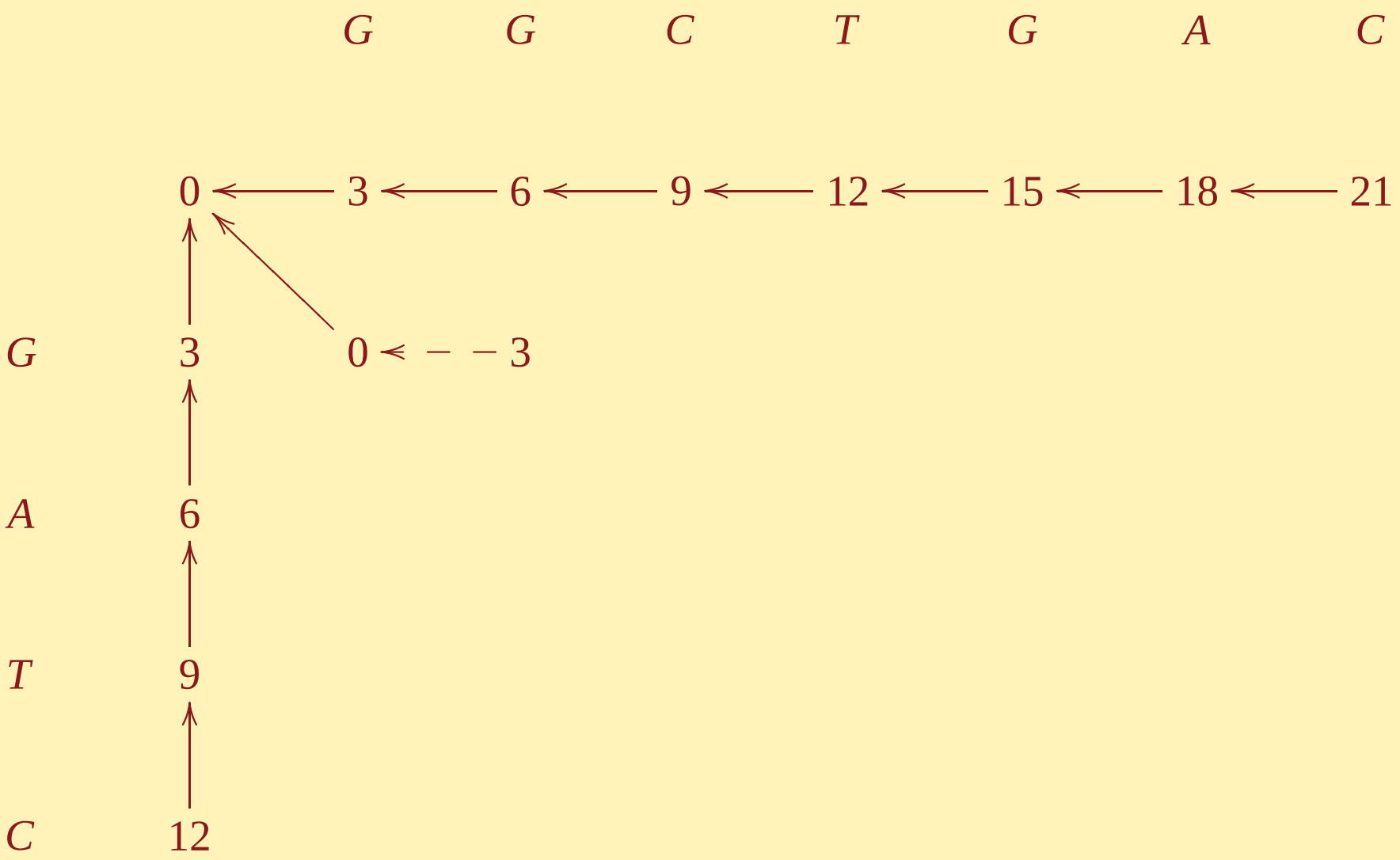
Matrice de programmation dynamique



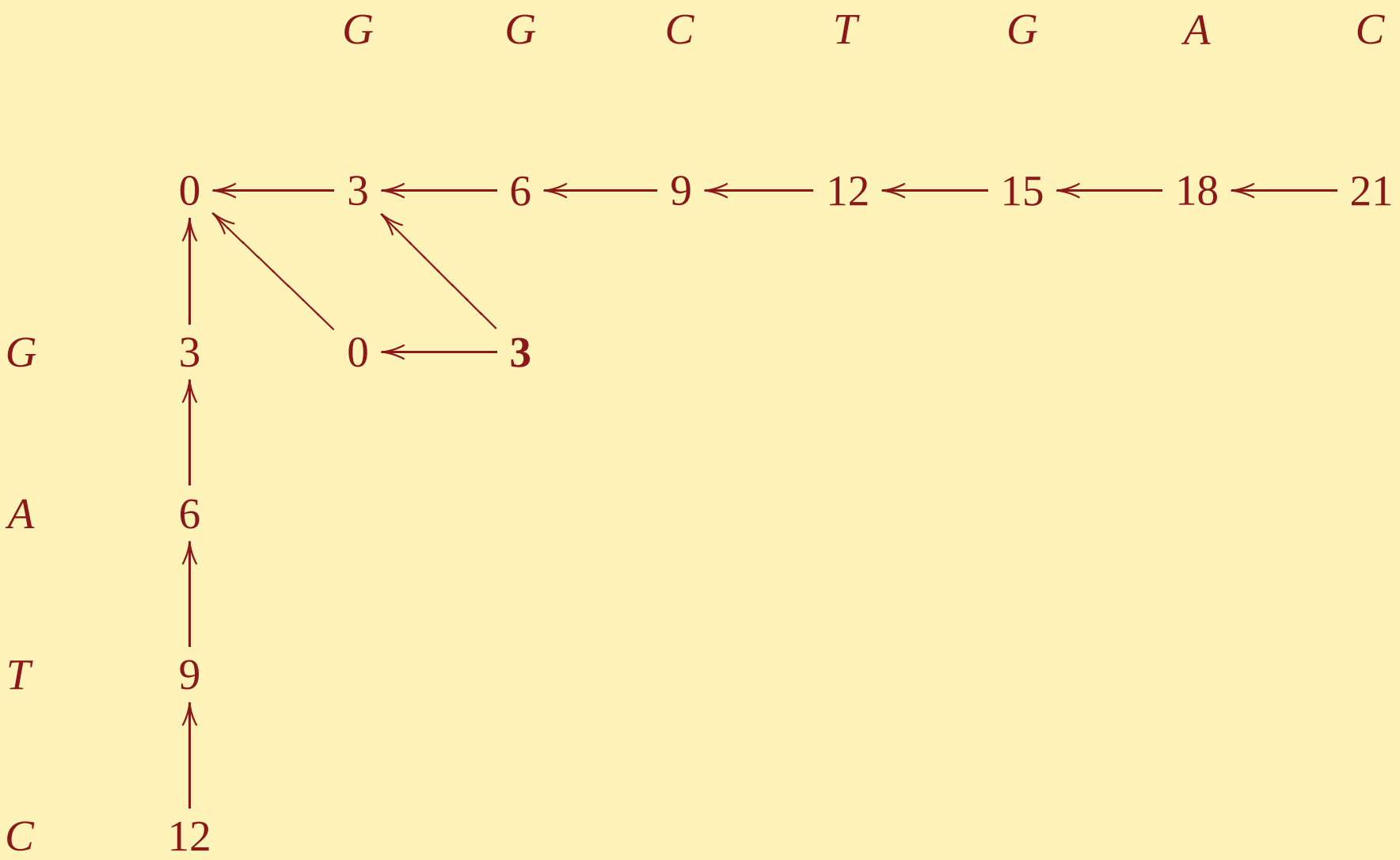
Matrice de programmation dynamique



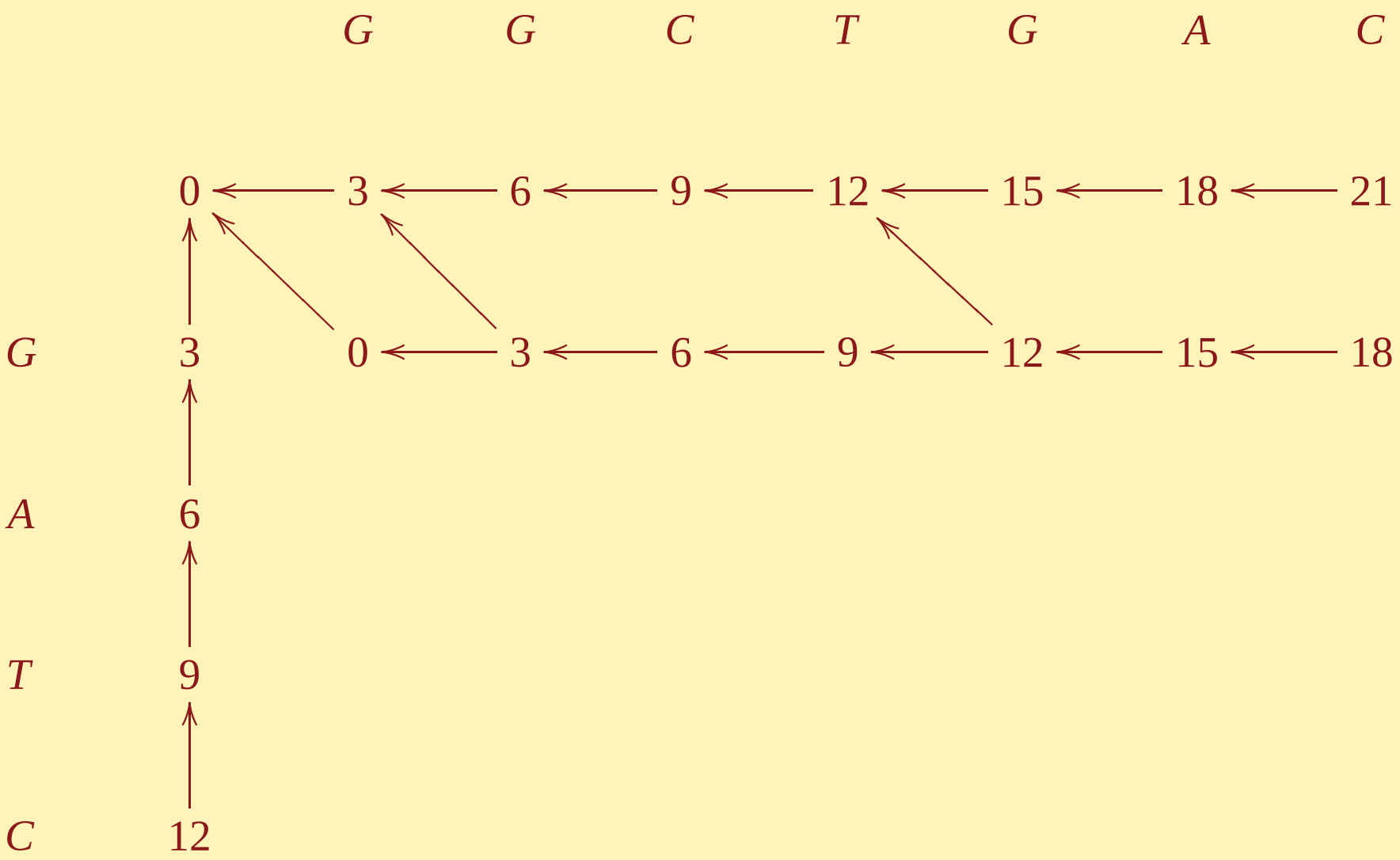
Matrice de programmation dynamique



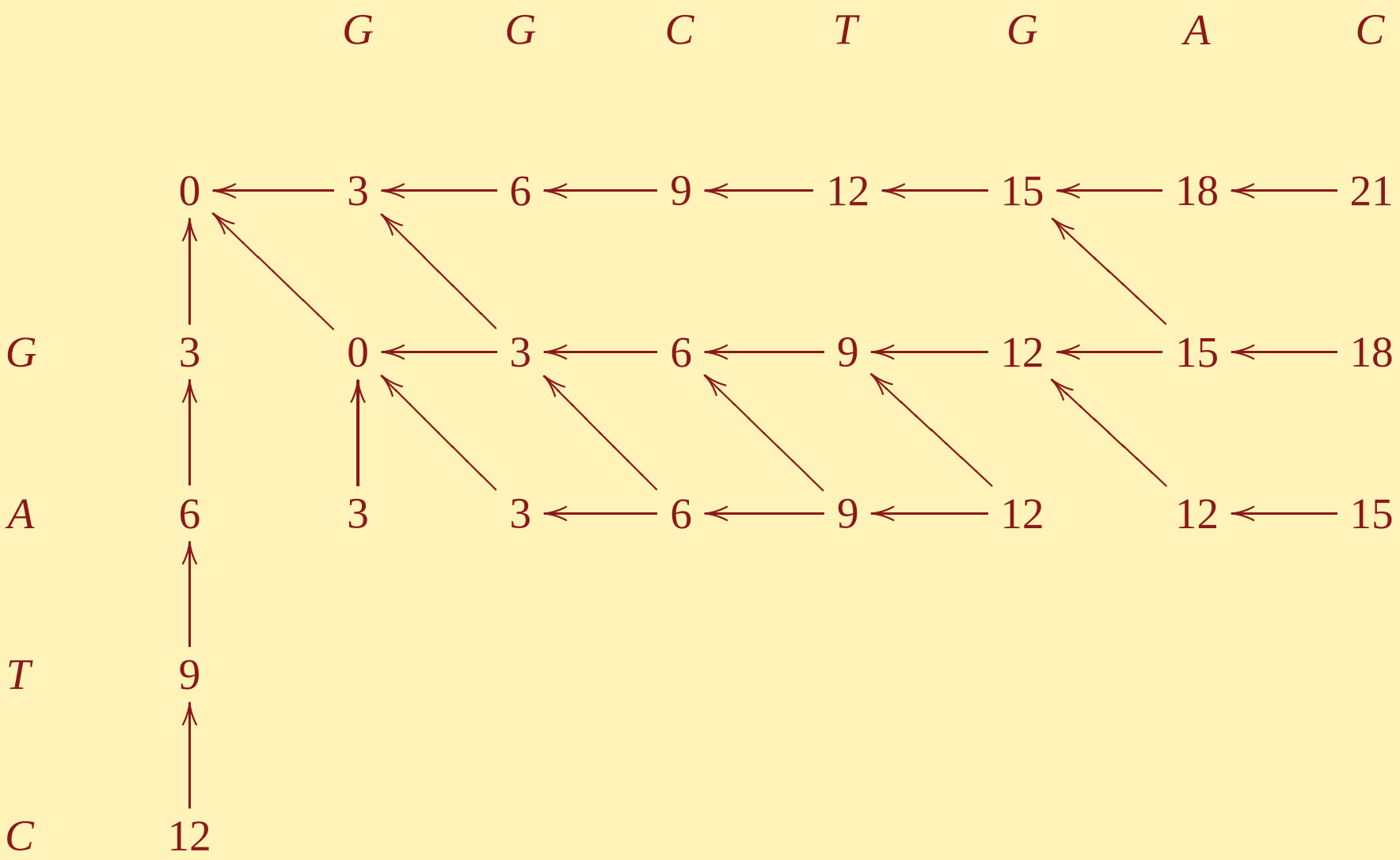
Matrice de programmation dynamique



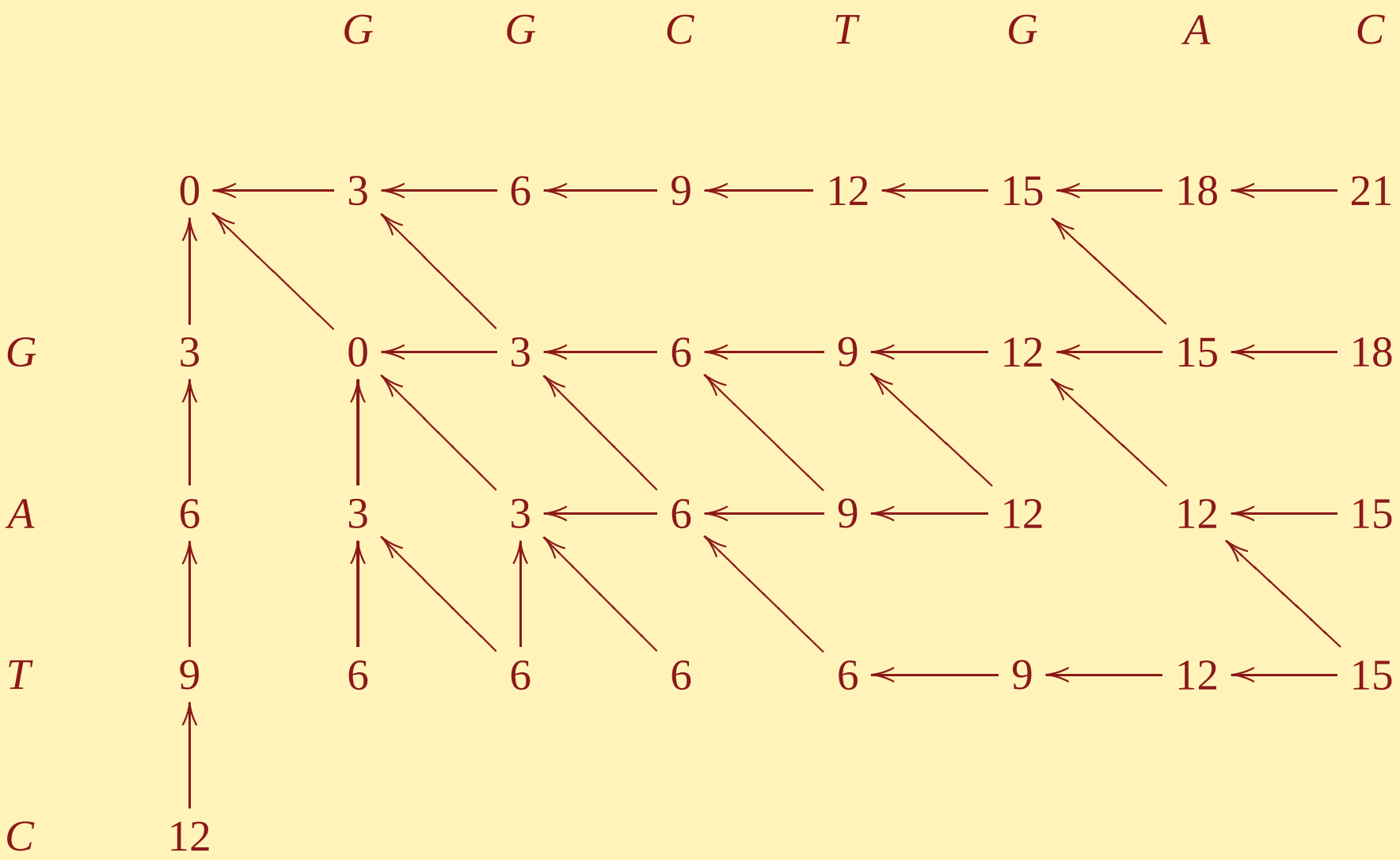
Matrice de programmation dynamique



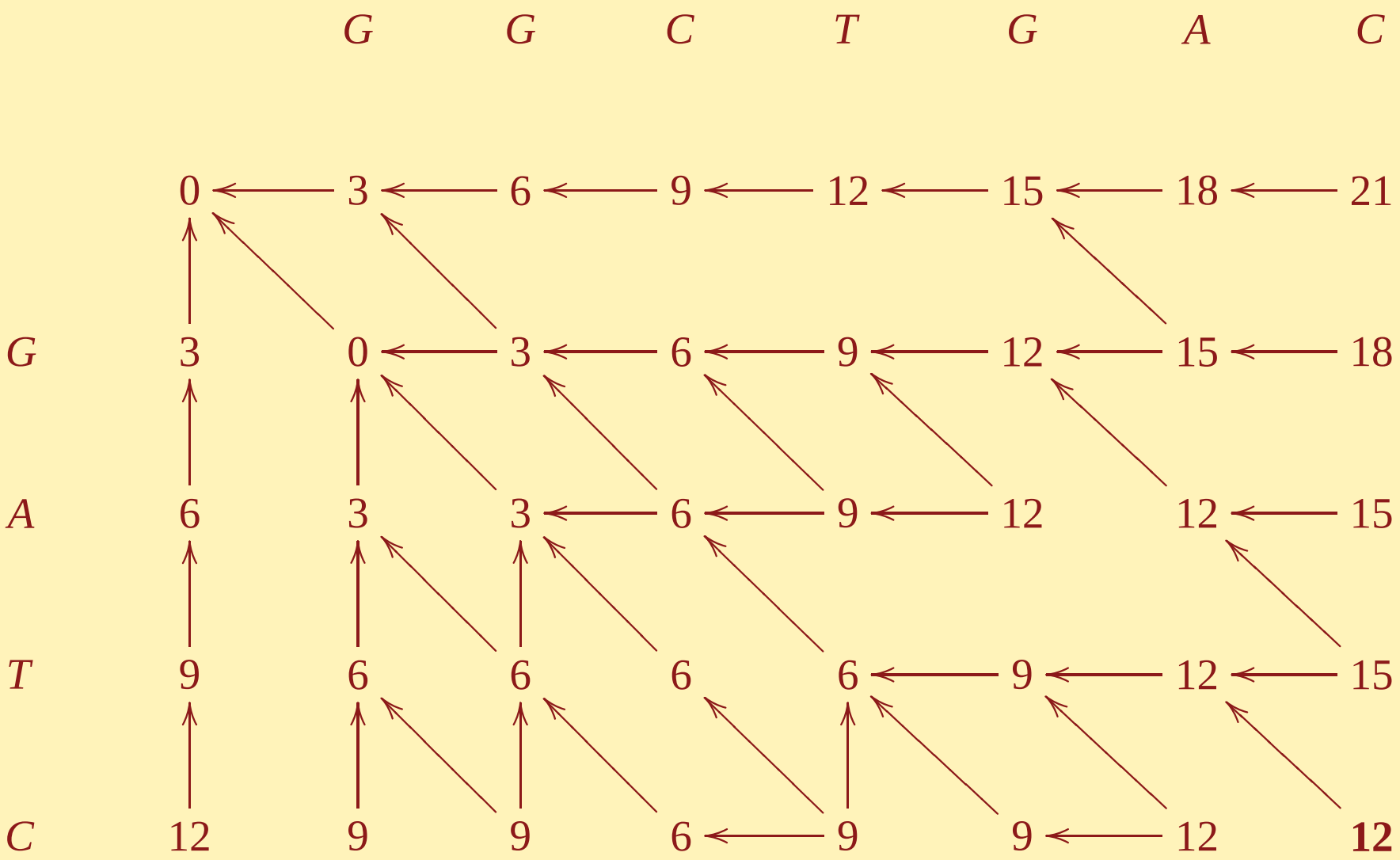
Matrice de programmation dynamique



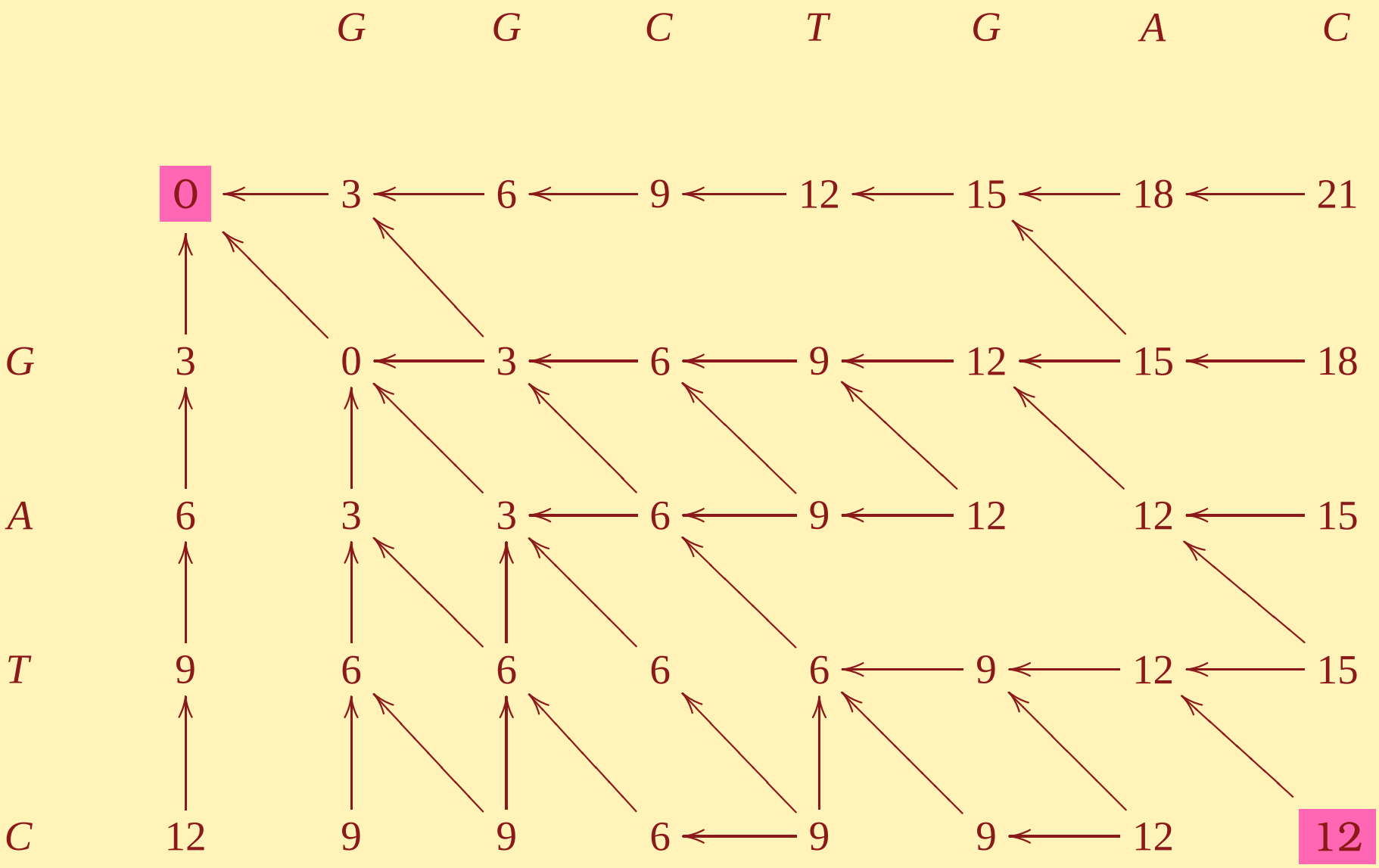
Matrice de programmation dynamique



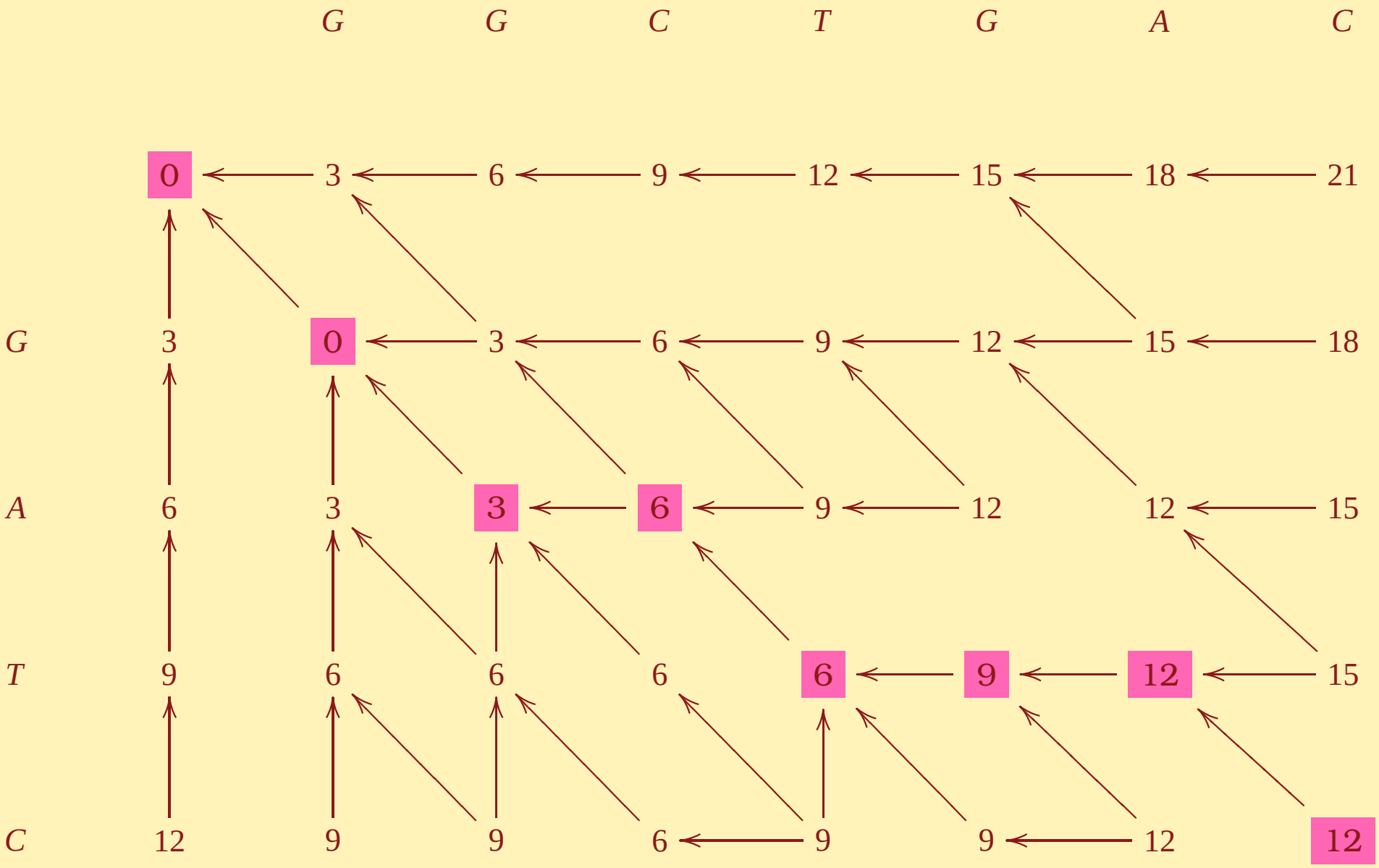
Matrice de programmation dynamique



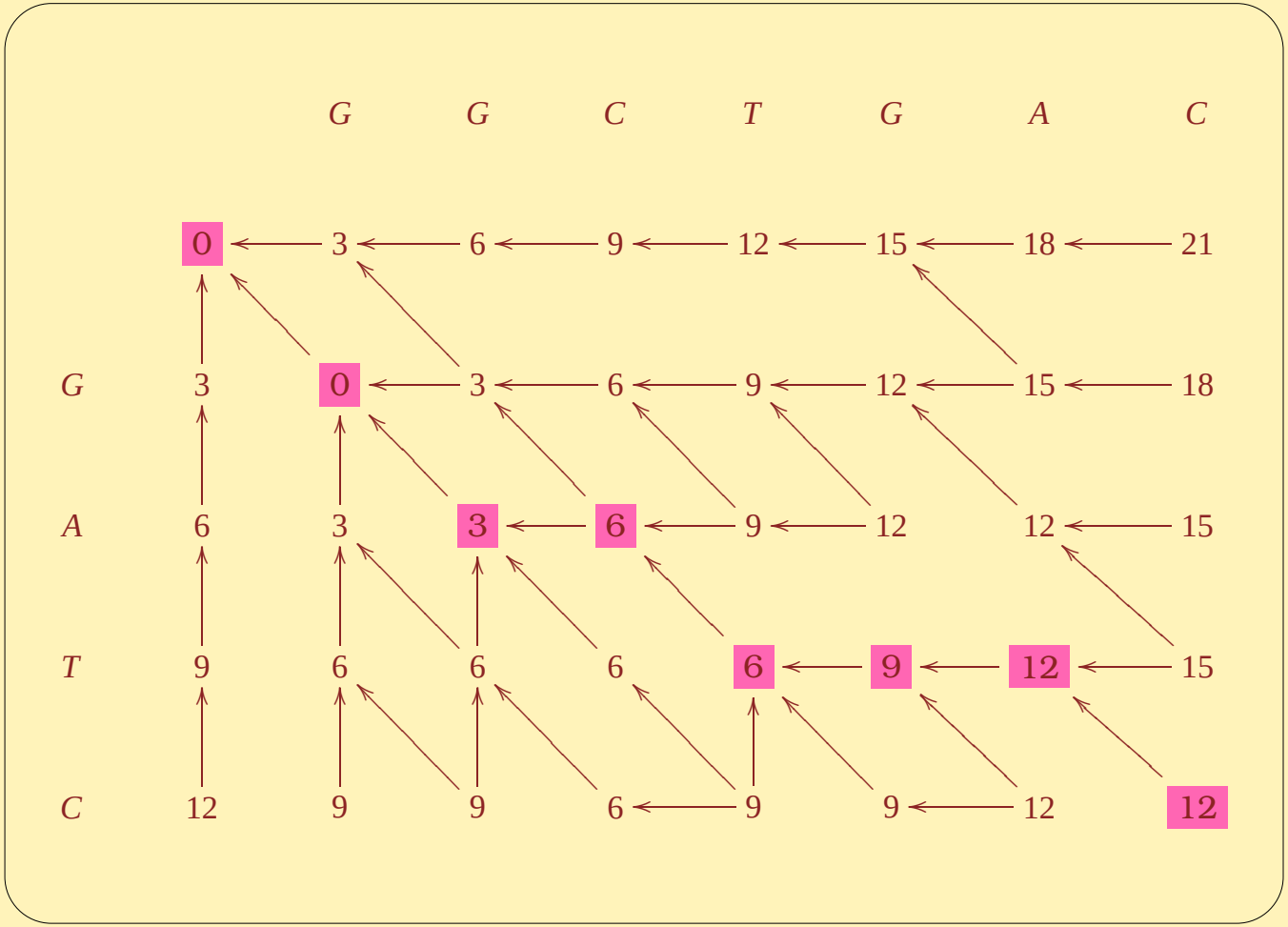
Matrice de programmation dynamique



Matrice de programmation dynamique



Alignement correspondant



S : G A - T - - C

R : G G C T G A C

Modèle Évolutif Particulier

- Opérations unitaires
  - **Amplification(A)/Contraction(C)** duplique/supprime un caractère se trouvant à côté d'un caractère identique.

$$a\ b\ c \xrightarrow{\text{Amplification}} a\ b\ b\ c \xrightarrow{\text{Contraction}} a\ b\ c$$

  - **Mutation(M)** mute un caractère en un autre.

$$a\ b\ c \xrightarrow{\text{Mutation}} a\ d\ c$$

  - **Insertion(I)/Délétion(D)** insère/supprime un caractère mais sans contrainte.

$$a\ b\ c \xrightarrow{\text{Insertion}} a\ b\ c\ d \xrightarrow{\text{Délétion}} b\ c\ d$$

- Opérations non unitaires
  - **Génération(G)/Compression(K)** génère/compresse une **arche** à partir/en son caractère ancêtre.

$$a \xrightarrow{\text{Génération}} a\ b\ c\ d\ a \xrightarrow{\text{Compression}} a$$

Exemple d'alignement

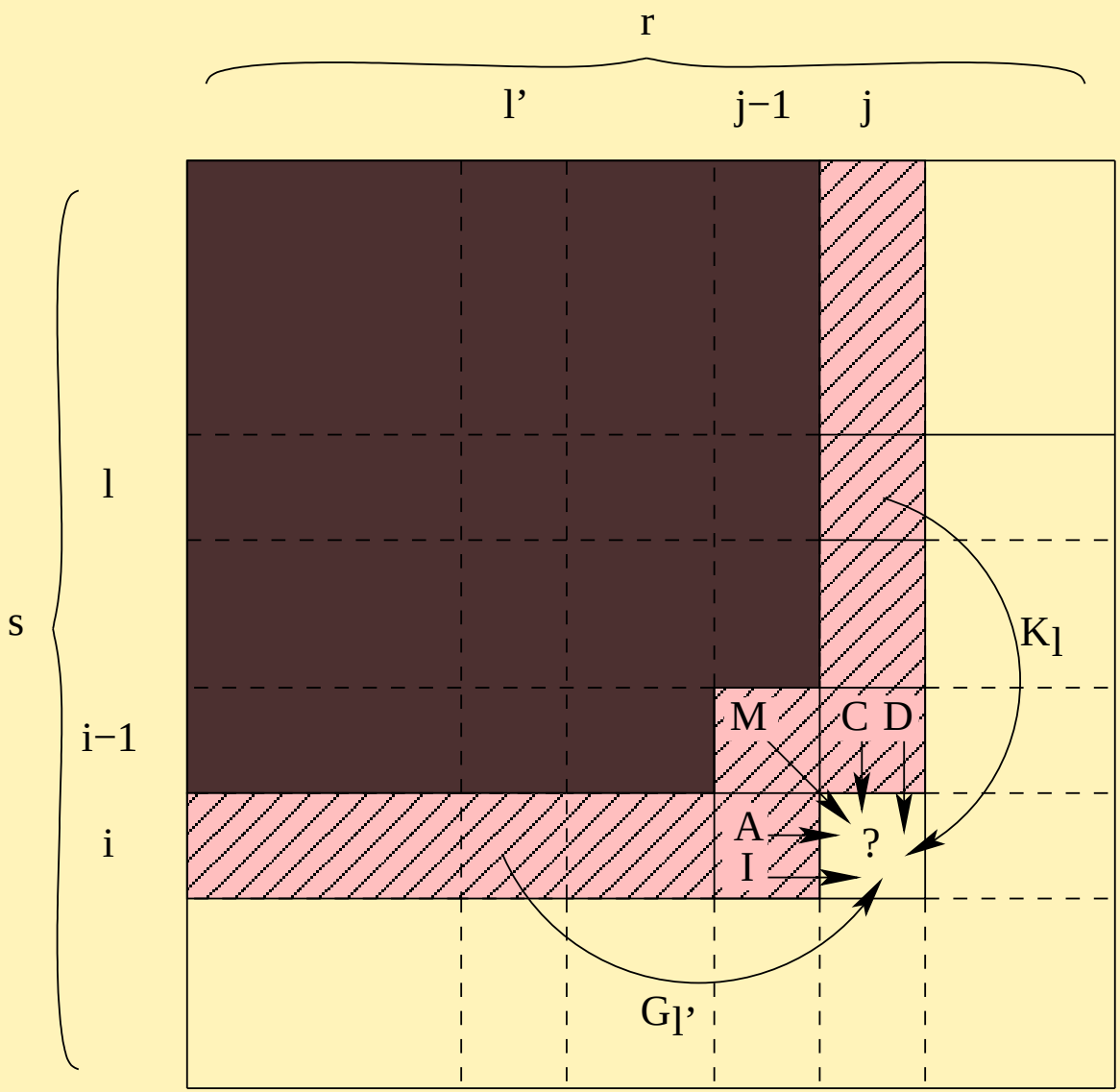
Seq1 : A A A B B C B D D B A

Seq2 : A E A A A

Nous voulons obtenir l'alignement optimal suivant :

```
Seq1 : A A A B B C B D D B A
       |   |   /   /   /  /  |
Seq2 : A E A A -  -  -  -  -  -  A
```

Matrice de programmation dynamique



Récurrence

$$\mathscr{A}[i,j] = min \begin{cases} \mathscr{A}[i-1, j-1] + M(s[i], r[j]) & \text{Mutation} \\ \mathscr{A}[i-1, j] + C \quad \textbf{iff } s[i-1] = s[i] & \text{Contraction} \\ \mathscr{A}[i-1, j] + D & \text{Délétion} \\ \mathscr{A}[i, j-1] + A \quad \textbf{iff } r[j-1] = r[j] & \text{Amplification} \\ \mathscr{A}[i, j-1] + I & \text{Insertion} \\ \mathscr{A}[l, j] + K(s[l, i]) \quad \forall\, \textbf{l} \in \textbf{[1, i - 2] tel que } s[l] = s[i] & \text{Compression d'arche} \\ \mathscr{A}[i, l'] + G(r[l', j]) \quad \forall\, \textbf{l'} \in \textbf{[1, j - 2] tel que } r[l'] = r[j] & \text{Génération d'arche} \end{cases}$$

Les minisatellites (ms)

- Séquences d'ADN appartenant à la classe des séquences répétées en tandem :

  $\cdots$ cggcgat cggcgac cggagat cggcgat cggcgat cggagat cgacgat $\cdots$

- ms évoluent principalement grâce aux amplifications en tandem et contractions en tandem

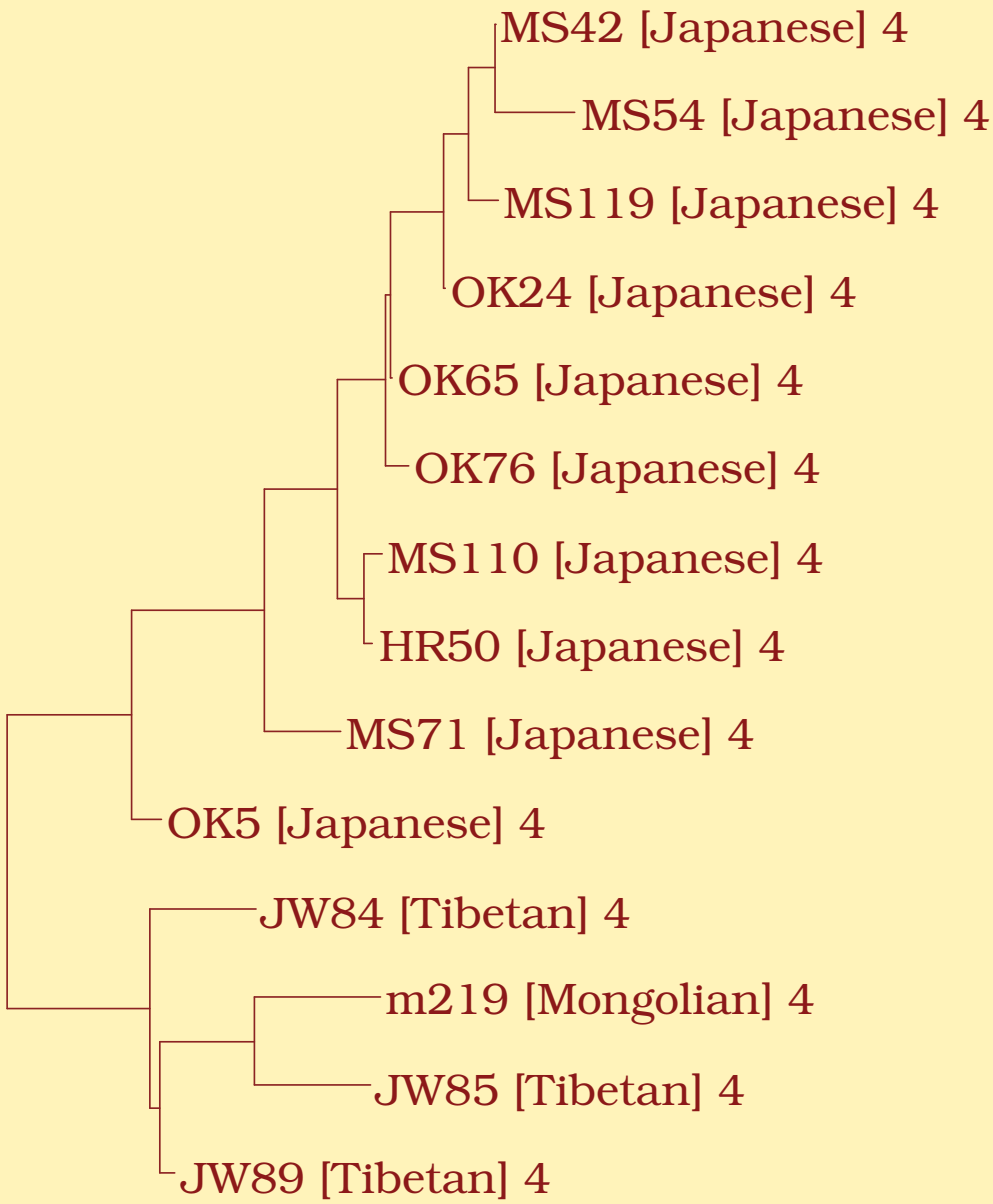
- Les minisatellites sont difficiles à séquencer $\Rightarrow$ une méthode spécifique : **Minisatellite Variant Repeat PCR** [Jeffreys et al. 91].
  MVR-PCR fournit une **carte de ms**:

  - Nouvel alphabet : A = cggcgat B = cggcgac C = cggagat D = cgacgat

  - Carte correspondante : A B C A A C D

- Expériences sur un jeu de 690 cartes du minisatellite humain MSY1 : Calcul de tous les alignements deux à deux, construction de l'arbre phylogénétique avec BioNJ à partir de notre matrice de distance.

Exemple d'arbre phylogénétique



Arbre phylogénétique de l'haplogroupe 4.