Machines de Turing

MIASHS L2

(d'après le cours de Julie Dugdale)

Jérôme GENSEL – Jerome.Gensel@univ-grenoble-alpes.fr
Prof. Université Grenoble Alpes
UFR SHS – Laboratoire d'Informatique de Grenoble

Sommaire

- Qu'est-ce qu'une Machine de Turing
- Similarités et différences avec les automates finis et les automates à pile
- Transitions dans une Machine de Turing
- Exemples
- Diagrammes de transition pour les Machines de Turing
- Définition formelle
- Conclusion

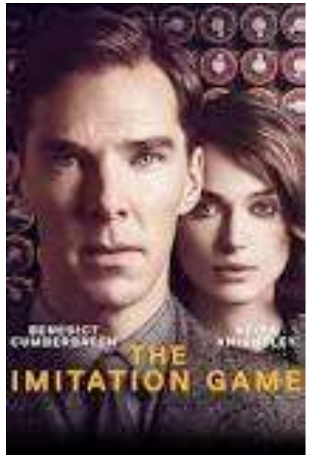
Qu'est-ce qu'une Machine de Turing?

- **Les Machines de Turing** sont des instruments de manipulation de symboles extrêmement basiques qui, malgré leur simplicité, peuvent être adaptés pour simuler la logique de tout ordinateur qui pourrait être construit.

- Elles furent décrites en **1936** par Alan Turing

Petit aparté…
qui était Alan Turing?

- Né en 1912, mathématicien, logicien, cryptographe britannique
- Souvent considéré comme un des fondateurs de l'informatique moderne
  - Il a formalisé le concept d'algorithme et de calcul avec les 'Machines de Turing'
  - Il a mis au point le 'Test de Turing' (Intelligence Artificielle)
    - Qui concerne la question : sera-t-il possible un jour de dire qu'une machine est consciente et peut penser ?
    - Le Test de Turing : brièvement, est un test de la capacité d'une machine à tenir une conversation 'humaine'
    - https://www.youtube.com/watch?v=k0vmuYQAkW4
    - https://www.youtube.com/watch?v=3wLqsRLvV-c
  - 1947 : Il a travaillé sur le développement d'un des premiers ordinateurs électroniques au monde
  - Pendant la 2ème guerre mondiale, il a travaillé sur le décryptage de codes de l'armée nazi…
- Turing mourut après avoir mangé une pomme empoisonnée au Cyanure en 1954 en pleine disgrâce…
- Gracié à titre posthume par la Reine Elisabeth II en 2013 et reconnu depuis comme Héros de guerre



*La machine portable de cryptage allemande **Enigma** utilisée pour encrypter et décrypter les messages secrets*

Qu'est ce qu'une Machine de Turing?

- Les Machines de Turing n'étaient pas supposées être une technologie informatique opérationnelle, pratique et performante, mais plutôt une étude des limites du calcul mécanique

➔ donc elles n'ont pas réellement été construites

- Cependant, l'étude des propriétés abstraites des Machines de Turing aide à comprendre des aspects de l'informatique et de la théorie de la complexité

Qu'est ce qu'une Machine de Turing?

- Elle est basée sur l'idée d'une personne exécutant une procédure bien définie qui change le contenu d'un ruban de papier infini, qui est divisé en cases, chaque case contenant un symbole issu d'un ensemble fini de symboles…

- La procédure est formulée en étapes simples de la forme…

  "Si votre état est 42 et que le symbole que vous voyez est un '0' alors remplacez-le par un '1', déplacez-vous d'un symbole vers la droite, et supposez que l'état 17 est votre nouvel état."

Similarités et différences avec les automates finis et les automates à piles

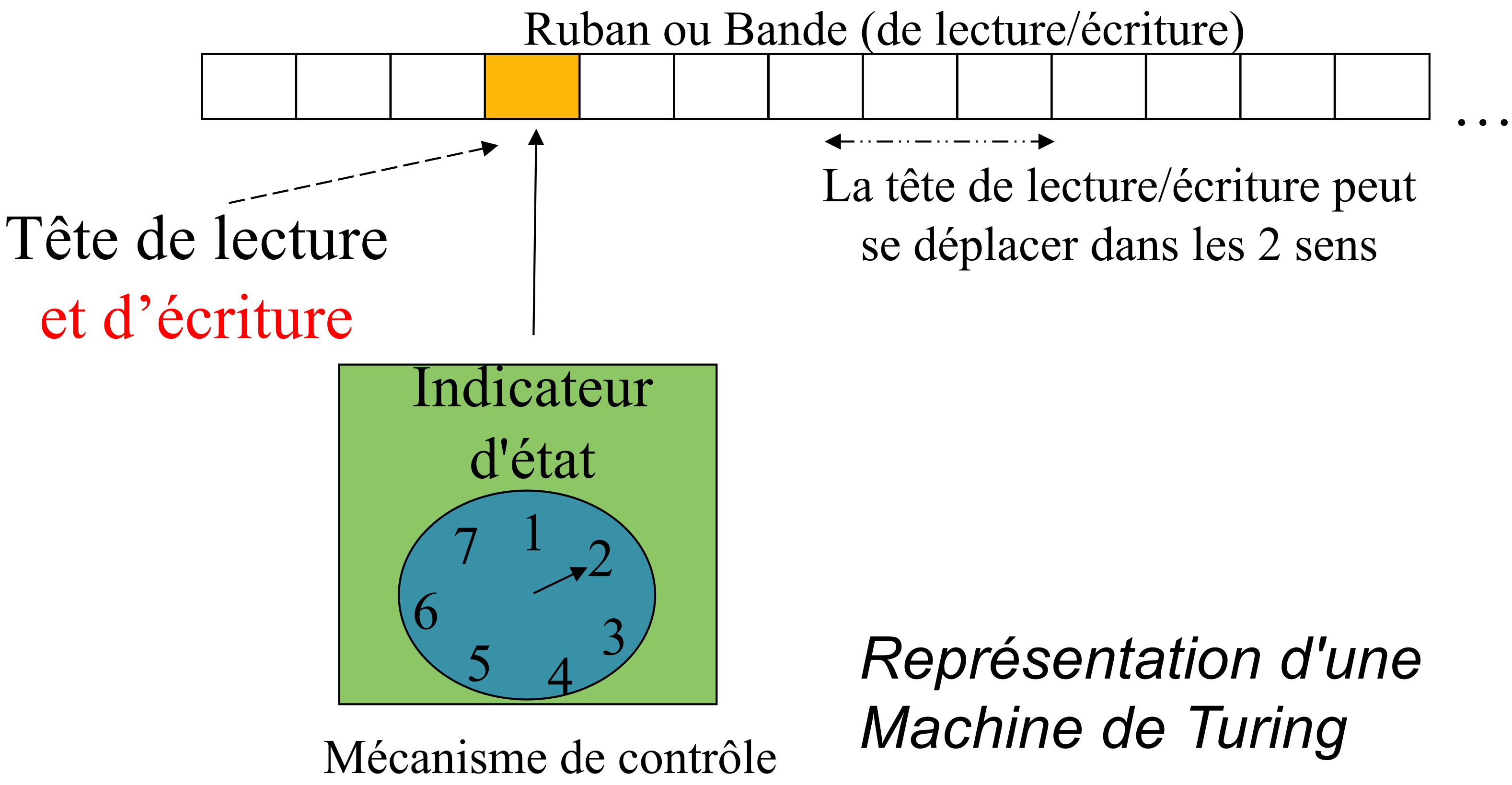
Similarités:

Une Machine de Turing consiste en

1. Un mécanisme de contrôle avec un nombre d'états finis
2. Un ruban avec une extrémité gauche fixe mais qui peut s'étendre indéfiniment vers la droite
3. Une tête de lecture

- Il existe un alphabet fini de symboles dans lequel on trouve les symboles qui sont sur le ruban (ou bande) de la machine
- Il existe un état initial

Similarités et différences avec les automates finis et les automates à piles



Représentation d'une Machine de Turing

<u>Remarque</u> *: la tête peut se déplacer dans les deux directions…*

Similarités et Différences avec les automates finis et les automates à piles

Différences:

- Une Machine de Turing peut écrire sur la bande en plus d'y lire (alors que les automates finis et les automates à pile ne peuvent que lire leur bande de lecture…)
  - Il n'y a donc pas besoin de pile, puisque la bande sert de moyen de lecture et de stockage auxiliaire.
  - De plus, une Machine de Turing n'est pas limitée au dépilage et à l'empilement de symboles sur une pile : elle peut scanner (parcourir) le ruban pour lire et modifier des symboles à n'importe quelle position sur la bande...

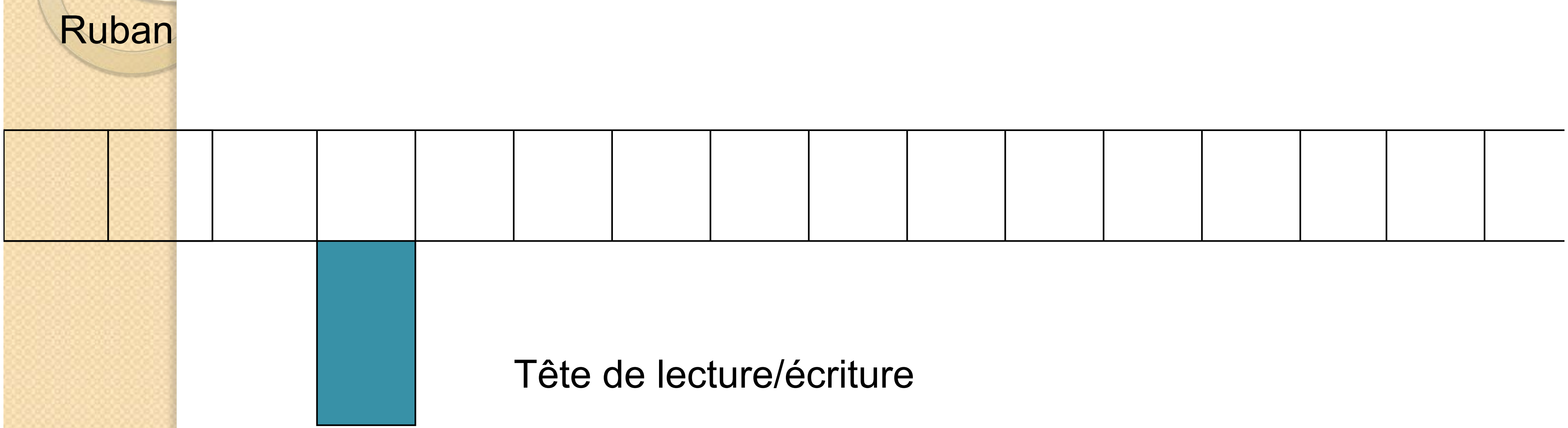
Similarités et Différences avec les automates finis et les automates à piles

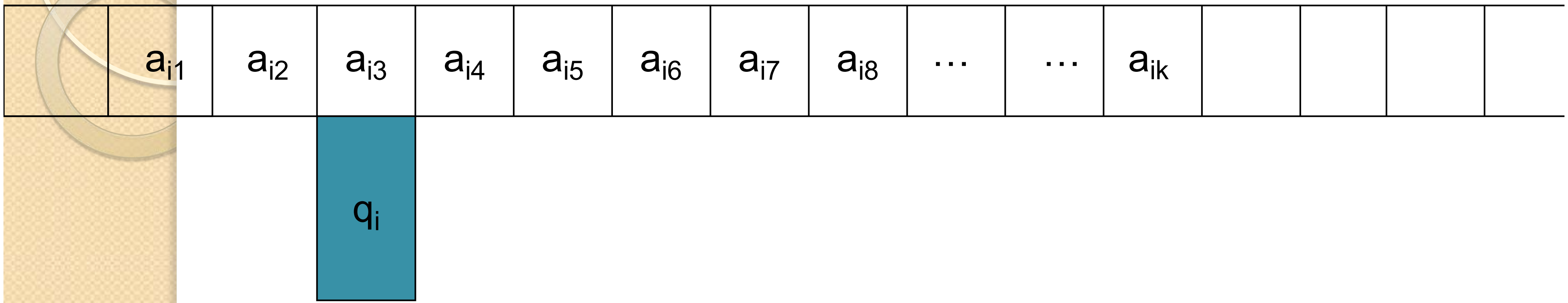
- **Attention !!!**
  - Dans une Machine de Turing, il n'y a qu'**un seul état d'arrêt**, et il est **distinct de l'état initial**.
- Si la machine atteint cet état d'arrêt, le processus s'arrête.
- Par comparaison, rappelez-vous :

1. Les AEF et les AAP ont un **ensemble** d'états d'acceptation, dont chacun peut être identique à l'état initial
2. Les AEF et AAP peuvent arrêter, ou pas, le processus lorsqu'ils atteignent un état d'acceptation.

Similarités et Différences avec les automates finis et les automates à piles

- **L'ensemble des symboles du ruban** inclut l'alphabet d'entrée et des symboles additionnels que la machine peut utiliser à des fins propres (comme les symboles de piles d'un AAP qui incluent les symboles d'entrée et des marqueurs spéciaux de piles).
- En particulier, nous supposons que le symbole vide est dans l'alphabet du ruban, mais pas dans l'alphabet d'entrée, et on le note #.


Ruban
Tête de lecture/écriture



Alphabet : $\Gamma = \{\#, a_0, a_1, a_2, \ldots, a_n\}$: les symboles admis sur le ruban

: le *blanc*

$\Sigma \subseteq \Gamma - \{\#\}$: les symboles d’entrée

$Q = \{q_0, q_1, \ldots, q_m\}$: l’ensemble des états

q_0 : l’état initial

$h \in Q$ *:* l'état d'arrêt

Transitions dans une Machine de Turing

- Les Machines de Turing ont seulement deux types d'opération
  1. des opérations **d'écriture**
     - remplacer un symbole sur un ruban par un autre et

     et

     - passer dans un nouvel état (sans déplacer la tête de lecture/écriture)
  2. des opérations de **déplacement**
     - déplacer la tête de lecture/écriture vers la droite et passer dans un nouvel état

     ou

     - déplacer la tête de lecture/écriture vers la gauche et passer dans un nouvel état.

Transitions dans une Machine de Turing

- À tout moment, le comportement de la machine est déterminé par l'état actuel et le symbole actuel sous la tête de lecture/écriture

Transitions dans une Machine de Turing

- Soit

  Q : l'ensemble des états,

  $h \in Q$: l'état d'arrêt,

  Γ : l'ensemble des symboles du ruban

  G (gauche) et D (droite) : deux symboles, non-membres de Γ

- Alors une **fonction de transition** d'une Machine de Turing est une fonction

  $$\delta : (Q - \{h\} \times \Gamma) \rightarrow (Q \times (\Gamma \cup \{G,D\}))$$

Transitions dans une Machine de Turing

- Il y a donc 3 possibilités qu'on interprète ci-dessous :

1. (p, x) → (q, y) signifie :

   Si la machine est dans l'état p *et si le symbole actuel sous la tête de lecture/écriture est* x *alors remplacer* x *par* y *et passer dans l'état* q

2. (p, x) → (q, G) signifie :

   Si la machine est dans l'état p *et si le symbole actuel sous la tête de lecture/écriture est* x *alors déplacer la tête d'une cellule vers la* ***gauche*** *et passer dans l'état* q

3. (p, x) → (q, D) signifie :

   Si la machine est dans l'état p *et si le symbole actuel sous la tête de lecture/écriture est* x *alors déplacer la tête d'une cellule vers la* ***droite*** *et passer dans l'état* q

Règles de transition

- 2 variantes:
  - la variante « quintuplets »
    - Une règle de transition est un quintuplet
      $(q_i, a_i, q_j, a_j, Dir)$ où $Dir \in \{G, D\}$
      que l'on écrit aussi: $(q_i, a_i) \rightarrow (q_j, a_j, Dir)$
  - la variante « quadruplets »
    - Une règle de transition est un quadruplet
      (qi, ai, qj, A) où A est une « action » ($A \in \Gamma \cup \{G, D\}$)
      que l'on écrit aussi $(q_i, a_i) \rightarrow (q_j, A)$

Remarque

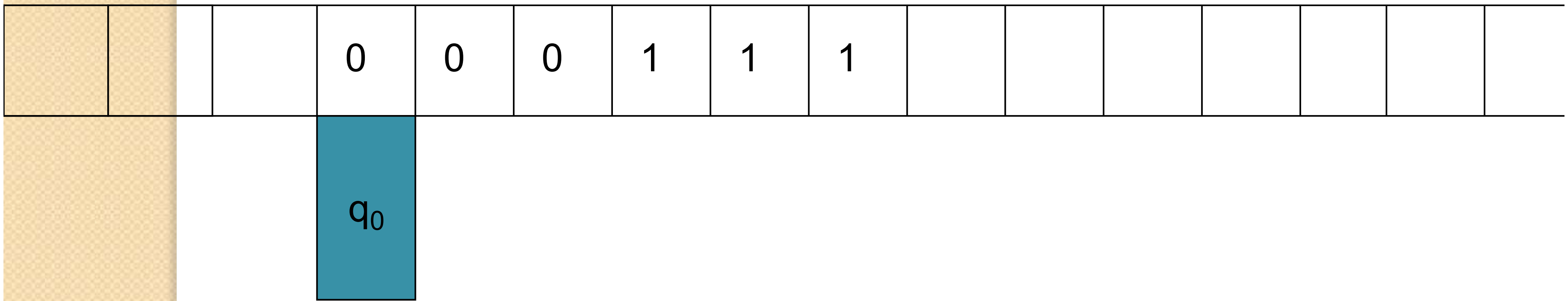
- Pour toute machine « quintuplets » T,
  il existe une machine « quadruplets » T'
  qui exécute la même tâche
  et réciproquement…

Un exemple

- $\Gamma = \{0, 1, X, Y, \#\}$
- $\Sigma = \{0, 1\}$
- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- Transitions (quintuplets) :

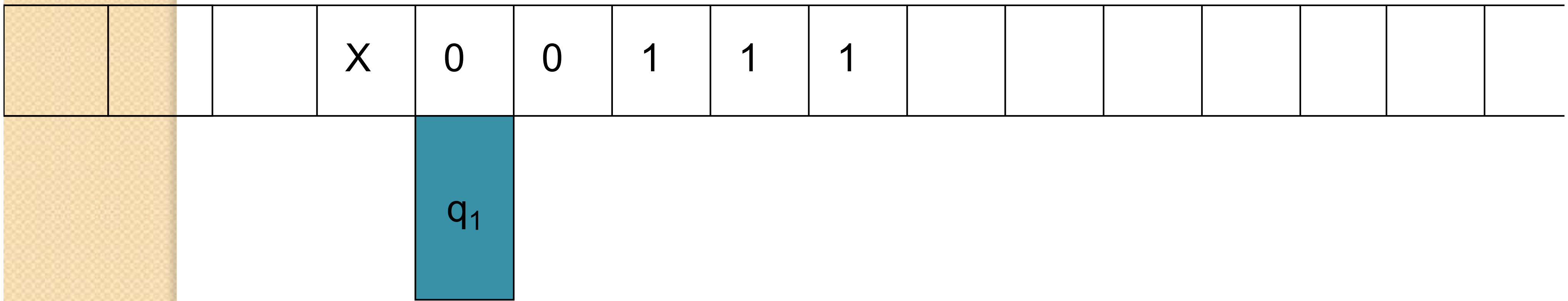
  $(q_0, 0, q_1, X, D)$, (q_0, Y, q_3, Y, D), $(q_1, 0, q_1, 0, D)$, $(q_1, 1, q_2, Y, G)$, (q_1, Y, q_1, Y, D), $(q_2, 0, q_2, 0, G)$, (q_2, X, q_0, X, D), (q_2, Y, q_2, Y, G), (q_3, Y, q_3, Y, D), $(q_3, \#, q_4, \#, D)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G),
(q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G),
(q3, Y, q3, Y, D), (q3, #, q4, #, D)



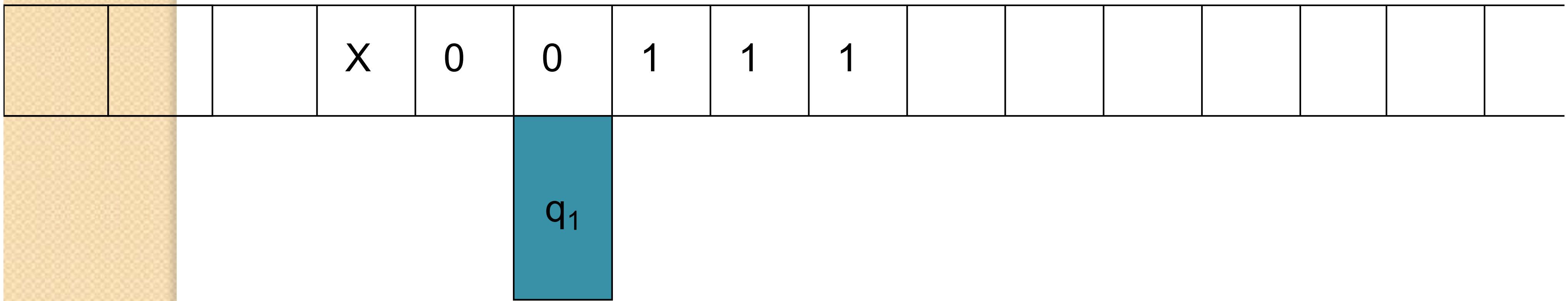
$(q_0, 0, q_1, X, D)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G),
(q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G),
(q3, Y, q3, Y, D), (q3, #, q4, #, D)



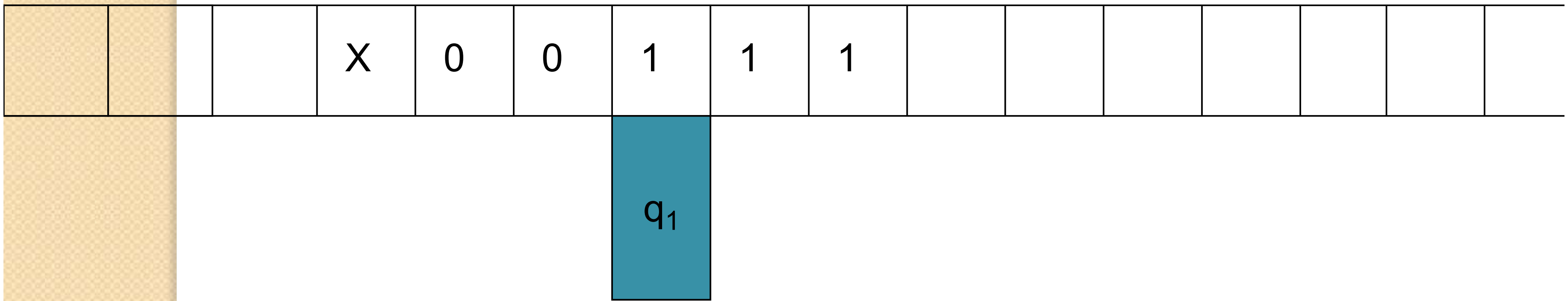
$(q_1, 0, q_1, 0, D)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



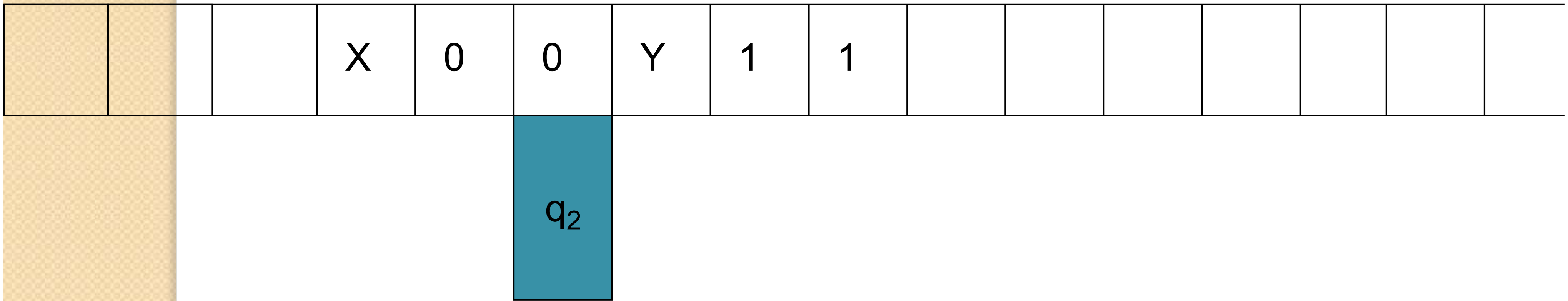
$(q_1, 0, q_1, 0, D)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G),
(q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G),
(q3, Y, q3, Y, D), (q3, #, q4, #, D)



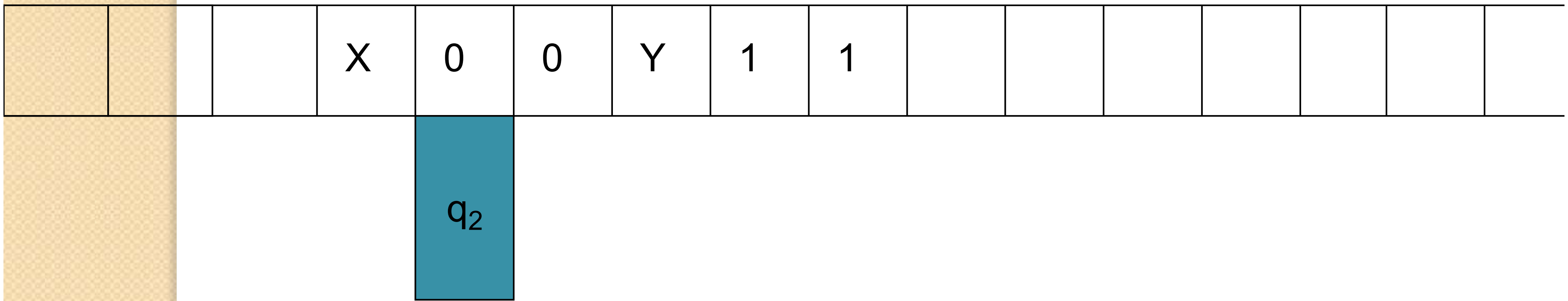
$(q_1, 1, q_2, Y, G)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G),
(q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G),
(q3, Y, q3, Y, D), (q3, #, q4, #, D)



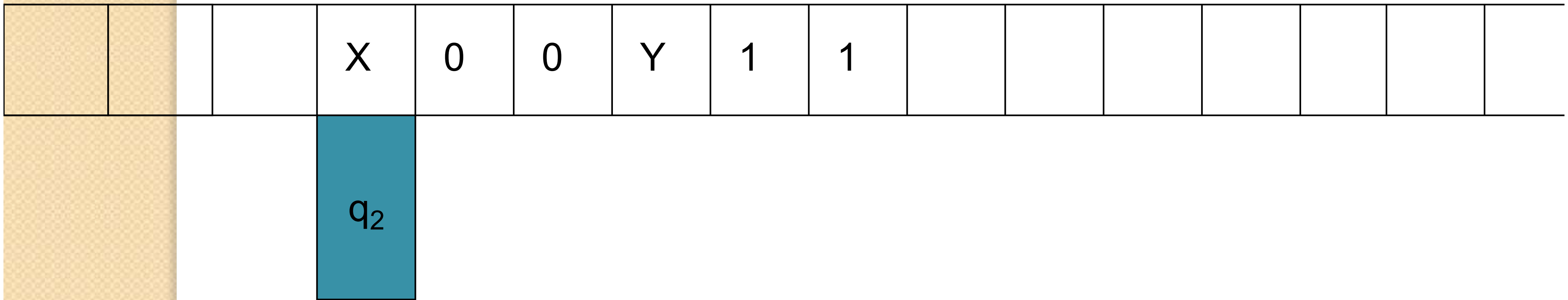
$(q_2, 0, q_2, 0, G)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



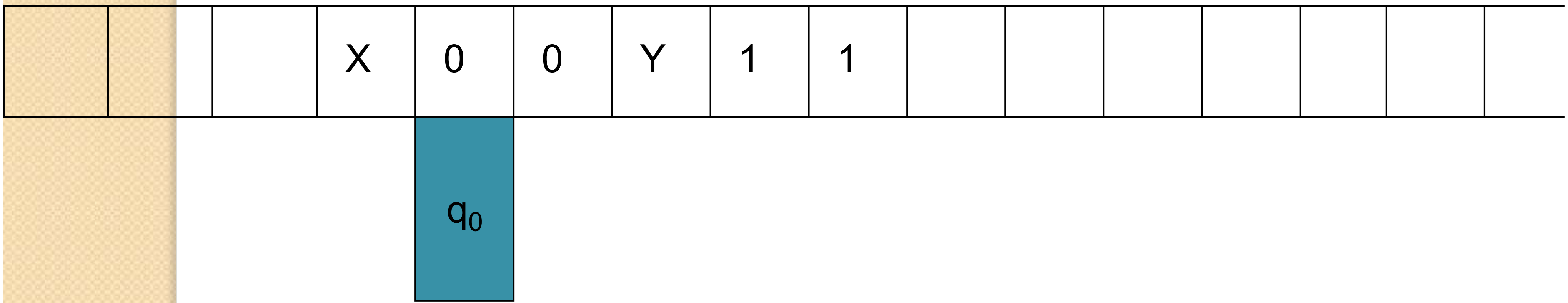
(q_2, 0, q_2, 0, G)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



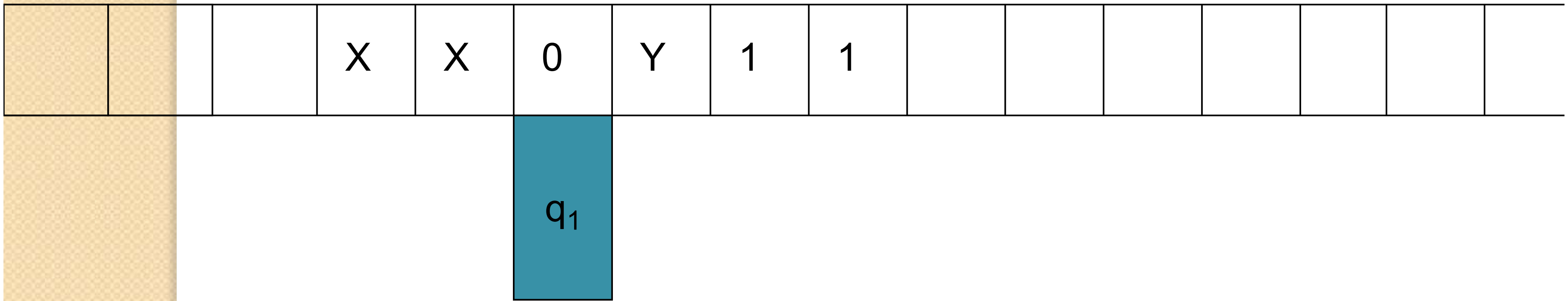
(q_2, X, q_0, X, D)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



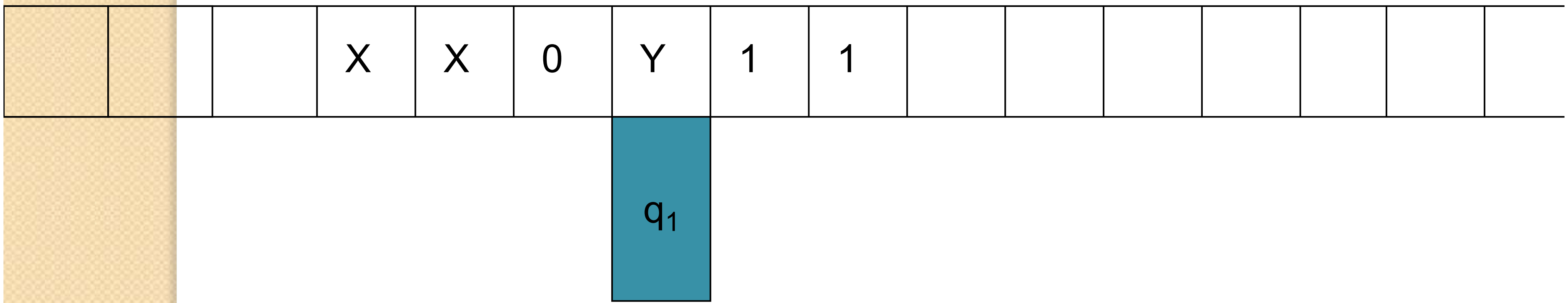
$(q_0, 0, q_1, X, D)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G),
(q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G),
(q3, Y, q3, Y, D), (q3, #, q4, #, D)



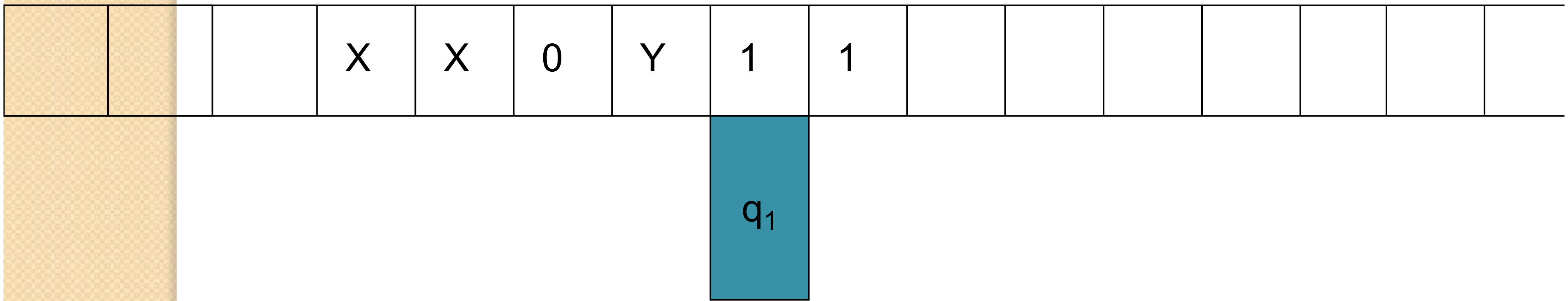
$(q_1, 0, q_1, 0, D)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



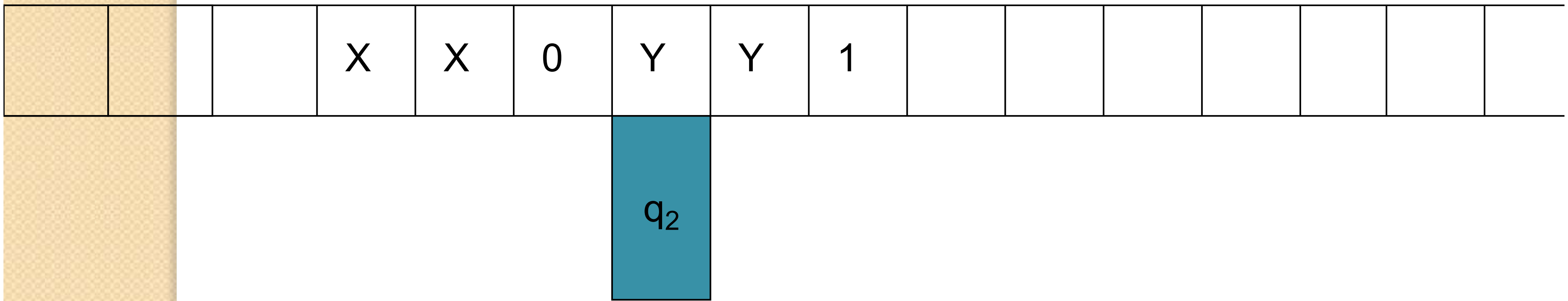
(q_1, Y, q_1, Y, D)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G),
(q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G),
(q3, Y, q3, Y, D), (q3, #, q4, #, D)



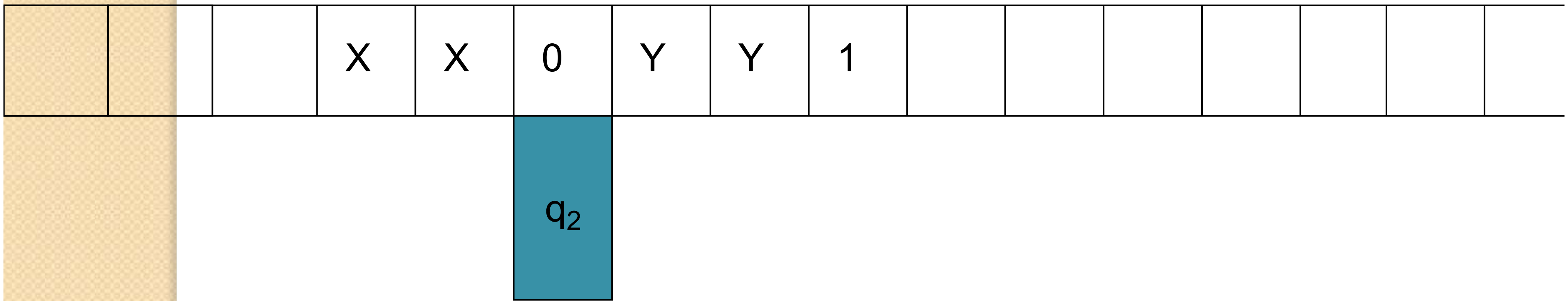
$(q_1, 1, q_2, Y, G)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



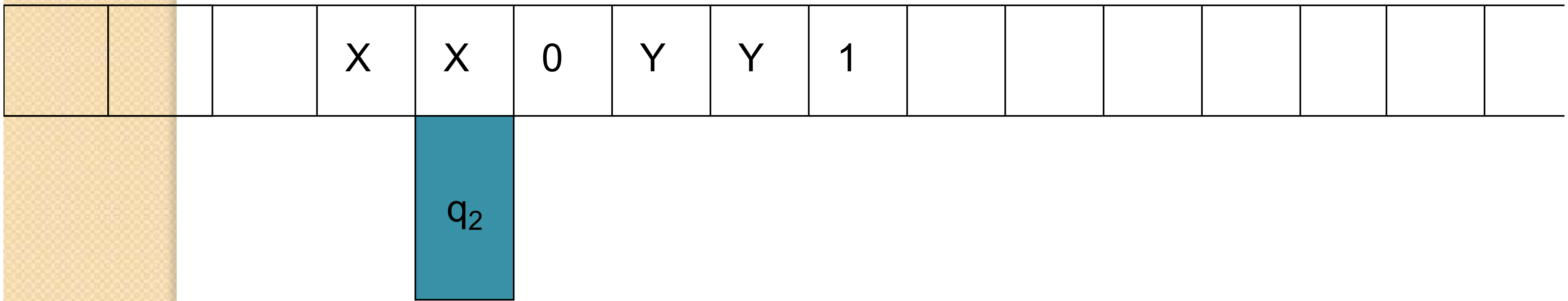
(q_2, Y, q_2, Y, G)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



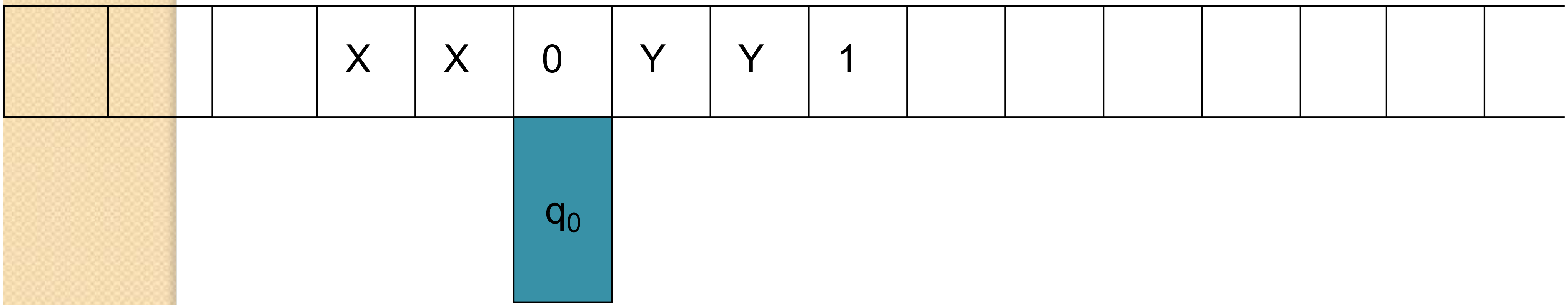
$(q_2, 0, q_2, 0, G)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



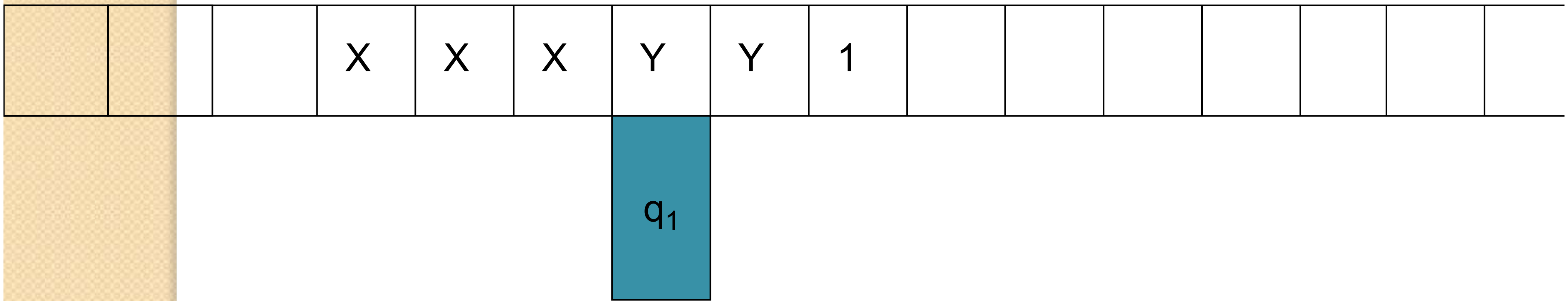
(q_2, X, q_0, X, D)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



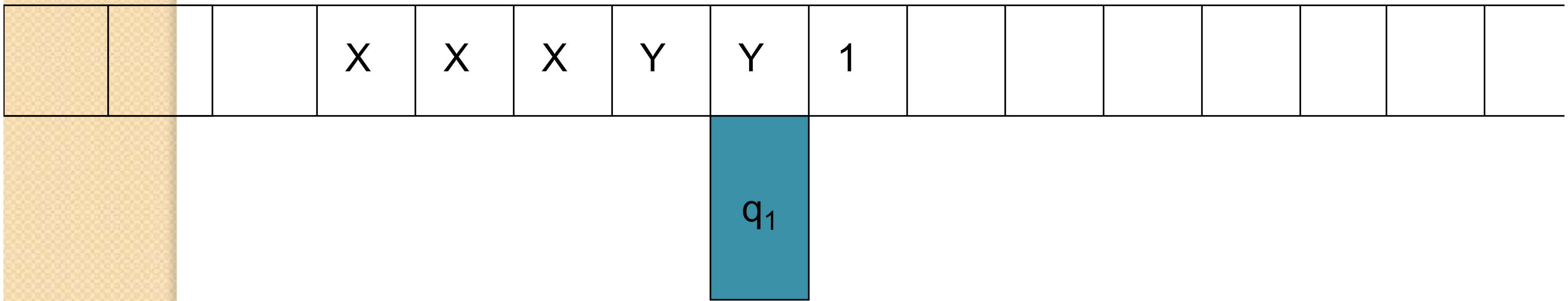
$(q_0, 0, q_1, X, D)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



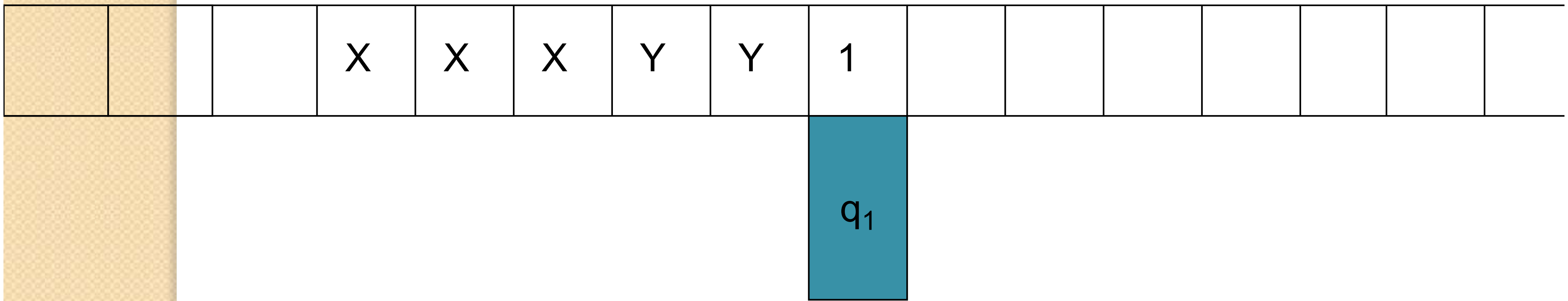
(q_1, Y, q_1, Y, D)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



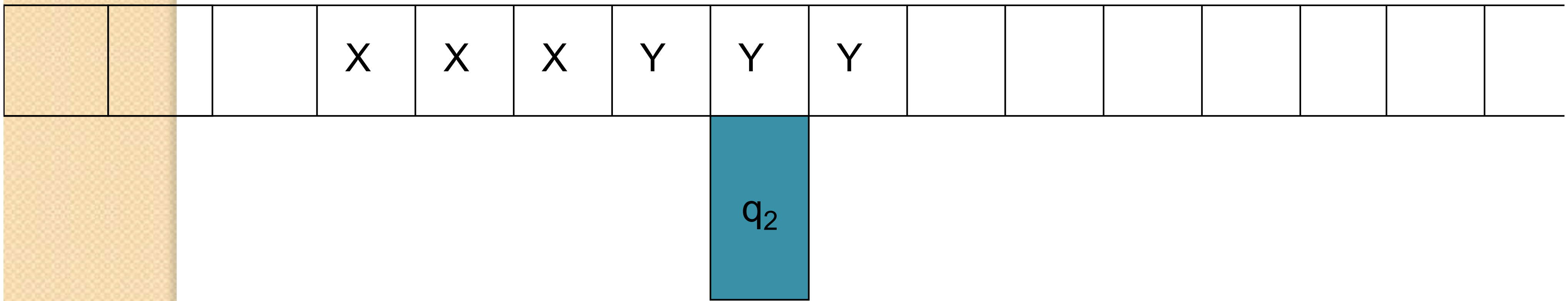
(q_1, Y, q_1, Y, D)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G),
(q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G),
(q3, Y, q3, Y, D), (q3, #, q4, #, D)



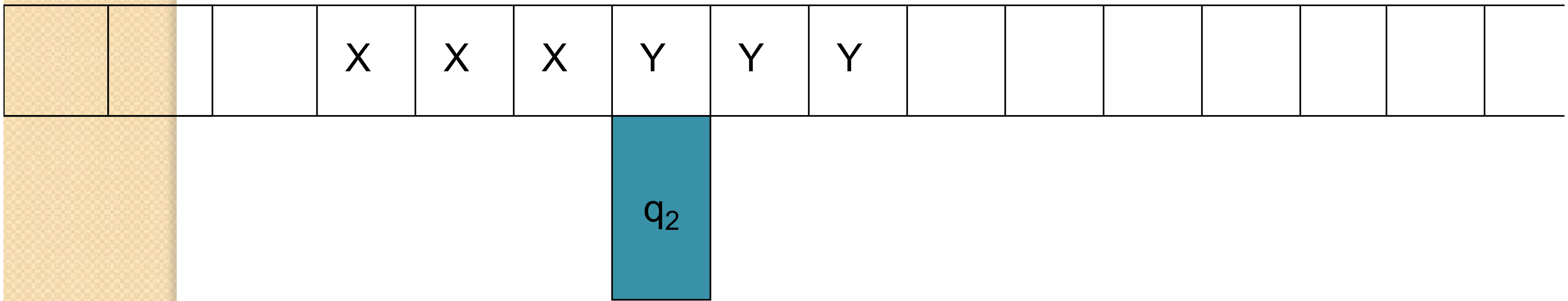
$(q_1, 1, q_2, Y, G)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G),
(q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G),
(q3, Y, q3, Y, D), (q3, #, q4, #, D)



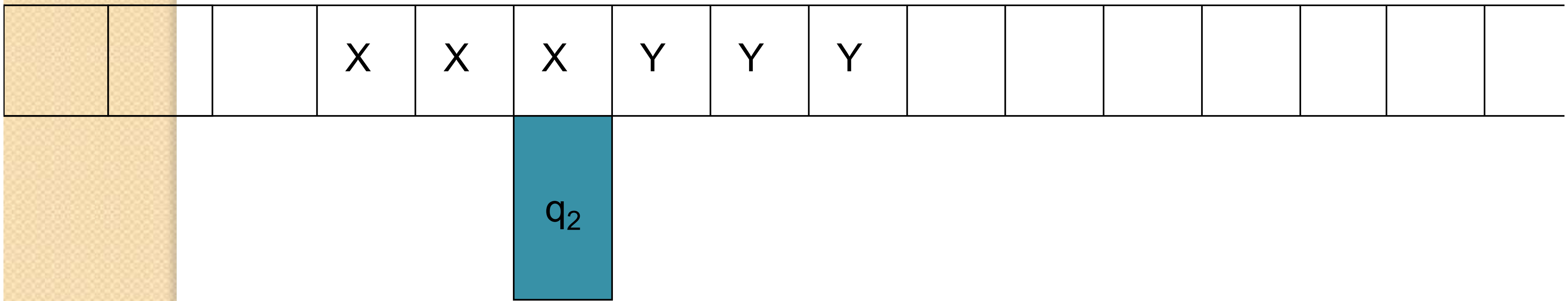
(q_2, Y, q_2, Y, G)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



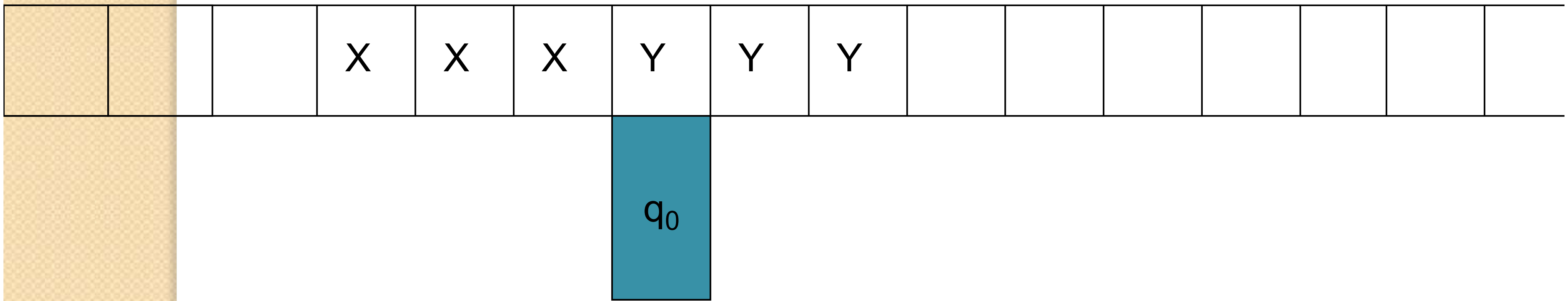
(q_2, Y, q_2, Y, G)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



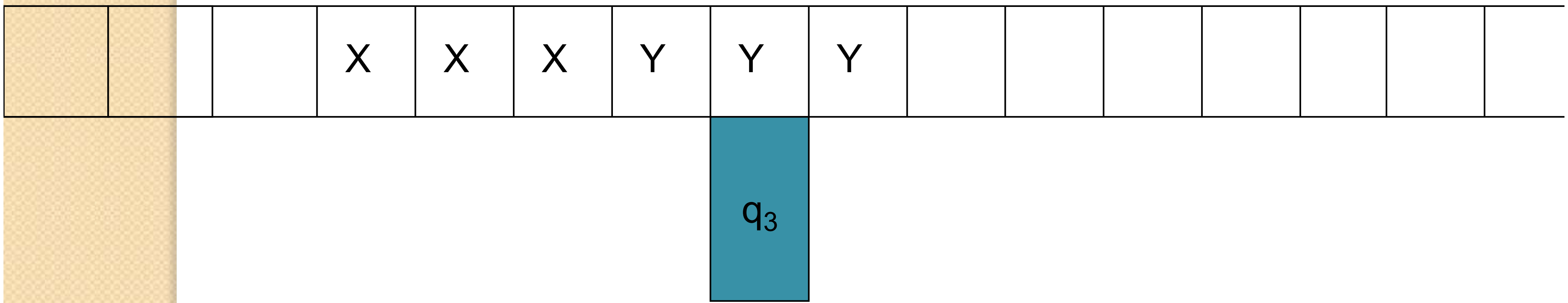
(q_2, X, q_0, X, D)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



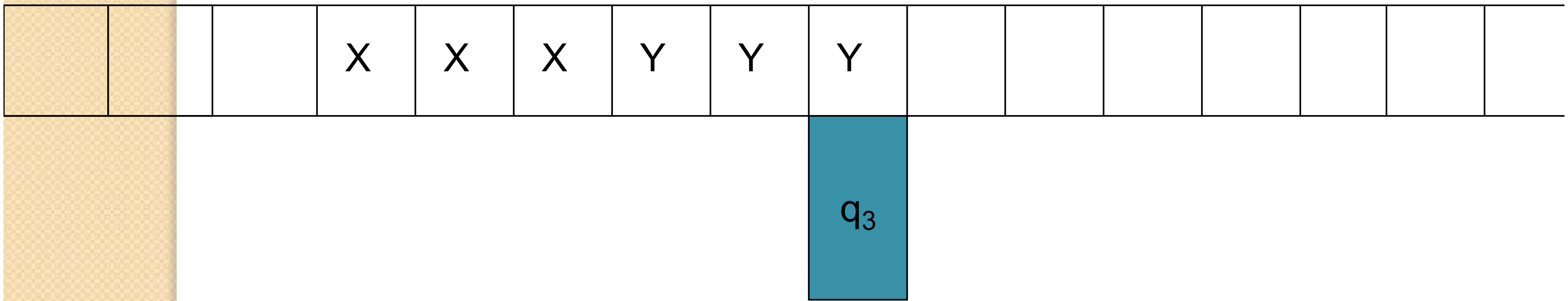
(q_0, Y, q_3, Y, D)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



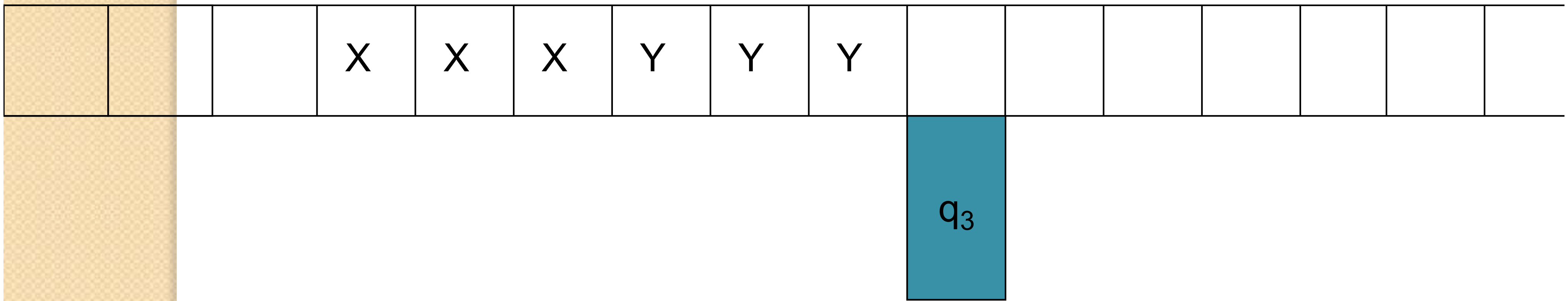
(q_3, Y, q_3, Y, D)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G),
(q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G),
(q3, Y, q3, Y, D), (q3, #, q4, #, D)



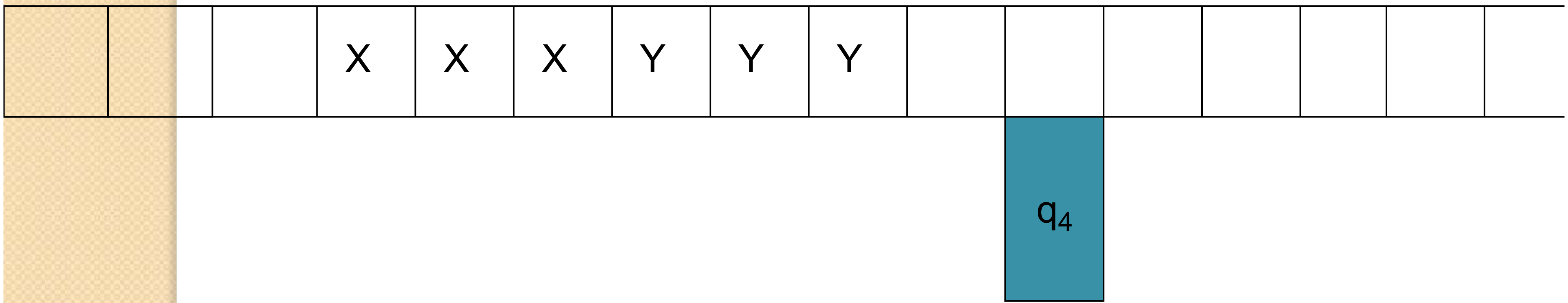
(q_3, Y, q_3, Y, D)

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G),
(q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G),
(q3, Y, q3, Y, D), (q3, #, q4, #, D)



$(q_3, \#, q_4, \#, D)$

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



$(q_3, \#, q_4, \#, D)$

Diagrammes de transitions pour les Machines de Turing

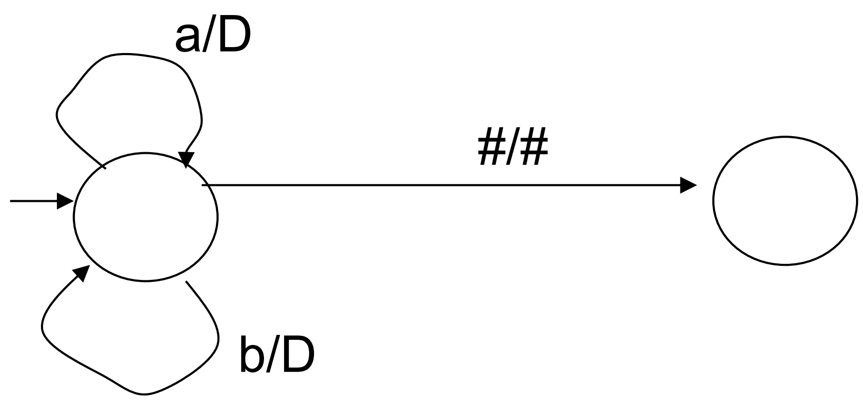
- Les transitions d'une Machine de Turing peuvent être représentées graphiquement par un diagramme de transitions

- Les étiquettes des arcs dans un diagramme de transitions prennent la forme s/A où
  - s est le symbole lu par la tête
  - A est l'action à effectuer

- Si A est un symbole *s'* alors le symbole courant *s* doit être remplacé par *s'* sur le ruban
- Si A est le symbole G alors la tête de lecture/écriture doit se déplacer vers la gauche;
- Si A est le symbole D alors la tête doit se déplacer vers la droite

Diagrammes de transition pour les Machines de Turing

- Exemple
  - Un diagramme de transition pour une machine de Turing qui lit des 'a' et des 'b', en se déplaçant vers la droite après chaque lecture, jusqu'à ce qu'elle rencontre un vide (#)
  - À ce moment là, elle remplace le vide par un vide et s'arrête sur cette case



Diagrammes de transition pour les Machines de Turing

- **Attention** : d'autres notations existent…

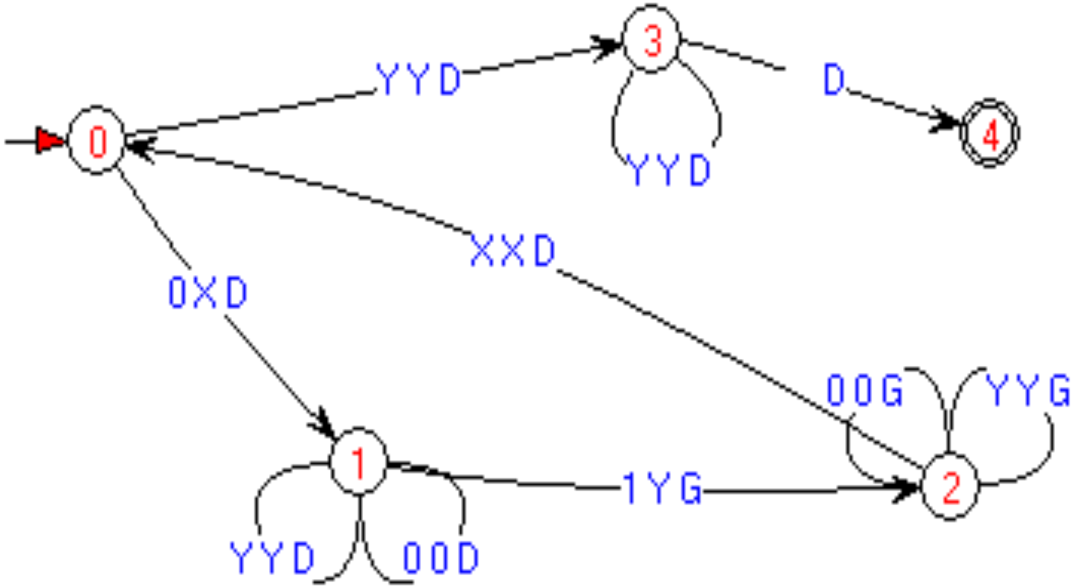
Diagramme de transitions – sur le long exemple précédent

- La MT est définie par :

  Γ = {0, 1, X, Y, #}
  Σ = {0, 1}
  Q = {q0, q1, q2, q3, q4}
  F = {q4}

- Notation adoptée pour représenter une transition dans le diagramme :
  - Par exemple, on labellise l'arête entre q0 et q1 par 0XD : on lit '0' sur le ruban, on écrit X et on se déplace à droite

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)



Γ = {0, 1, X, Y, #}
Σ = {0, 1}
Q = {q0, q1, q2, q3, q4}
F = {q4}

(q0, 0, q1, X, D), (q0, Y, q3, Y, D), (q1, 0, q1, 0, D), (q1, 1, q2, Y, G), (q1, Y, q1, Y, D), (q2, 0, q2, 0, G), (q2, X, q0, X, D), (q2, Y, q2, Y, G), (q3, Y, q3, Y, D), (q3, #, q4, #, D)

Définition formelle

- Normalement, une Machine de Turing démarre dans son état initial et exécute des transitions jusqu'à ce que son état d'arrêt soit atteint

Notes :

1. l'état d'arrêt peut ne jamais être atteint, lorsque la machine entre dans une boucle sans fin
2. la machine peut terminer anormalement si elle essaye de déplacer la tête de lecture/écriture au-delà de l'extrémité gauche du ruban

Définition formelle

- Formellement, une Machine de Turing est définie comme un sextuplet :

$$(Q, \Sigma, \Gamma, \delta, i, h)$$

où :

- Q est un ensemble fini d'états
- Σ est un ensemble fini de symboles non-vides (alphabet de la machine)
- Γ est un ensemble fini de symboles, incluant les symboles de Σ (les symboles de la tête de lecture/écriture)
- δ est la fonction de transition de la machine
- $i \in Q$ est l'état initial de la machine
- $h \in Q$ est l'état d'arrêt de la machine

Exemple : nombre pair de 1

- Soit une séquence de 0 et de 1 écrite sur le ruban
- La Machine de Turing doit découvrir si cette séquence comporte un nombre pair de 1
- Si oui, elle écrira un 1 à droite de la séquence (après un blanc écrit après le dernier chiffre de la séquence)
- Sinon elle écrira un 0

Exemple 2 : nombre pair de 1

- Si initialement, on a :

			0	1	0	1	0	1							

- La réponse de la Machine de Turing doit conduire au ruban

			0	1	0	1	0	1		0					

Exemple : nombre pair de 1

- <u>L'idée</u> : la tête va parcourir la séquence en passant alternativement de l'état 1 à l'état 2 et réciproquement à chaque fois qu'elle rencontre un 1

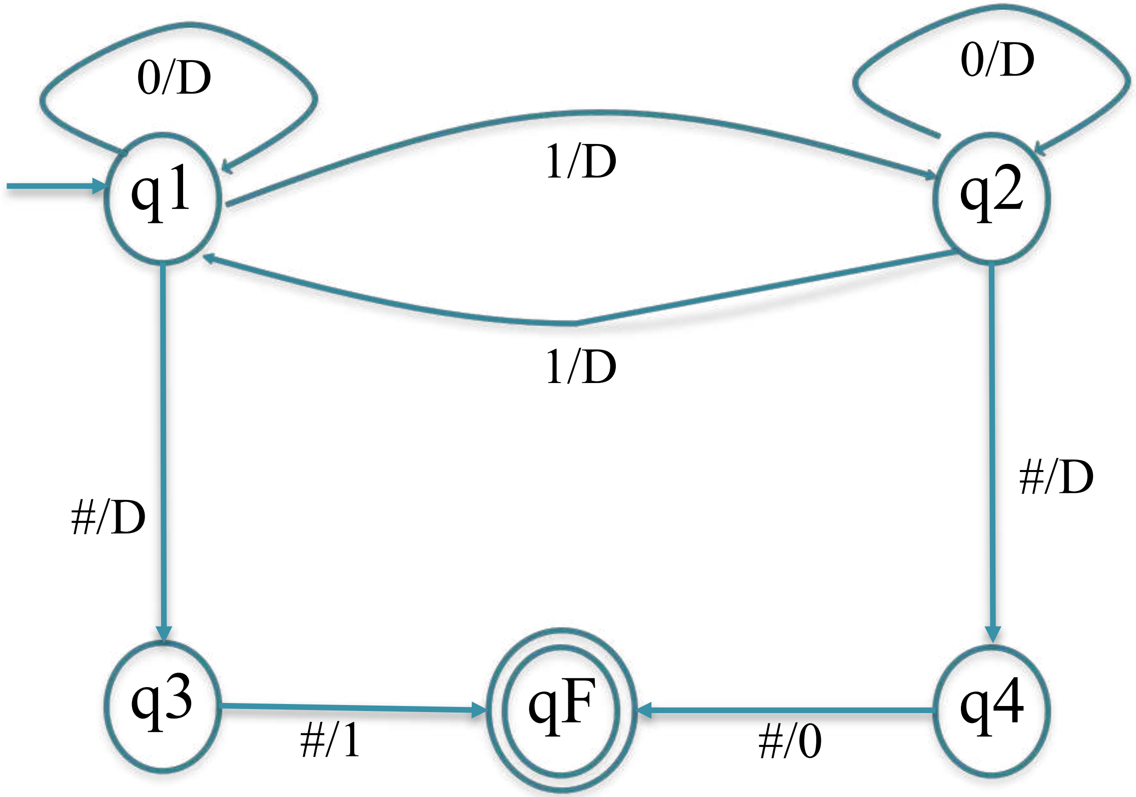
- L'état 1 correspond à l'état dans lequel se trouve la machine si elle a lu un nombre pair de 1
- Dans les états 1 et 2, si la machine rencontre un 0, elle se déplace à droite et ne fait rien d'autre.
- La machine « comprend » qu'elle a fini la lecture de la séquence quand elle rencontre le premier blanc
- À l'état 3 (réponse OUI), elle écrit 1 sur un blanc puis s'arrête
- À l'état 4 (réponse NON) elle écrit 0 sur un blanc puis s'arrête

Exemple 2 : nombre pair de 1

- D'où la table de transitions :

état	lecture	écriture	déplacement	Nouvel état
q1	b (blanc) 0 1		D D D	q3 q2
q2	b (blanc) 0 1		D D D	q4 q1
q3	b (blanc) 0 1	1		qF
q4	b (blanc) 0 1	0		qF

Exemple : nombre pair de 1

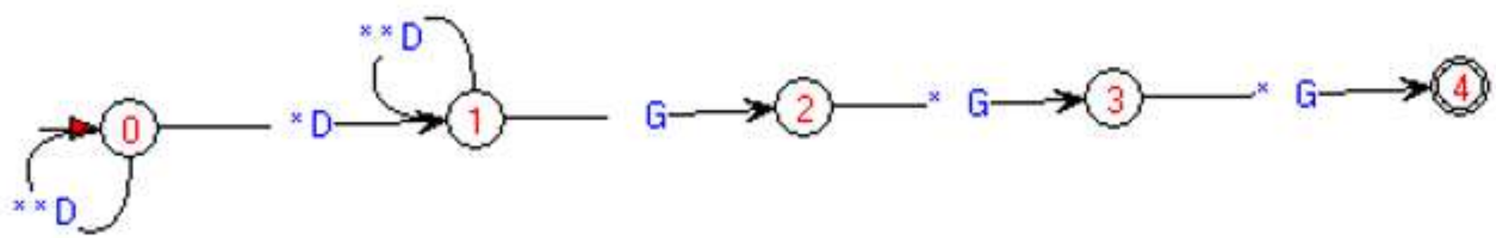


Exemple : calcul sur des entiers

- Conventions:
  - L'entier n est représenté par une suite de n+1 '*' (de façon à pouvoir représenter 0)
  - Deux entiers à additionner sont séparés sur le ruban par un blanc
  - La machine commence toujours son calcul en étant positionnée sur le premier caractère non blanc à partir de la gauche

Exemple : addition d'entiers

- Donner l'ensemble de transitions de l'addition entière 2 + 3 étant donnée le diagramme de transitions :



Opérations

- On peut, à l'aide d'une Machine de Turing, réaliser :
  - Multiplication de deux entiers
  - Exponentiation (calcul de 2^n par exemple)
  - Soustraction entière :
    - $x - y$ = l'entier positif z tel que $z+y = x$ si $y \leq x$

      $x - y = 0$ sinon
  - etc., etc., etc.

Grammaire & Automates

Type	***Grammaire***	***Automate***
Type 0	Non-restreintes	Machines de Turing
Type 1	Sensibles au contexte	Machines de Turing linéaires bornées
Type 2	Hors-contexte	Automates à Piles
Type 3	Regulières	AEF

Un peu d'histoire : qu'est ce que Turing essayait de faire ?

- Turing a présenté ces idées sur les machines de calcul en 1936
  ➔ ce travail précède les travaux sur les automates finis et automates à piles.
- Il n'essayait pas d'étendre des modèles plus simples (comme nous), mais il essayait d'exprimer de la manière la plus simple l'essence même du calcul.
- Le modèle de Turing était celui d'un humain essayant, avec un crayon et du papier, d'effectuer un calcul
- Personne **n'a encore découvert de modèle de calcul plus puissant** et le consensus est qu'il **n'en existe pas** :

Thèse de Turing : *la puissance de calcul des Machines de Turing est au moins égale à celle de tout système de calcul possible*

Un peu d'histoire : qu'est ce que Turing essayait de faire ?

- Il est intéressant de remarquer que les idées de Turing ont précédé l'invention des ordinateurs digitaux, et que les premiers ordinateurs ont été construits à partir de ses modèles mathématiques.

- C'est un exemple remarquable de la valeur du travail théorique…

Finalement..

- Revue du cours sur les automates et les langages:
- C'était un cours d'introduction sur la théorie du calcul
- Objectifs:
  - Développer la compréhension des notions de mots, langages
  - Introduire et étudier les automates de calcul abstraits (tels que les machines à états finis, les machines à piles, et les Machines de Turing),
  - Développer la compréhension des différents types de grammaires formelles
  - Comprendre les relations entre ces différents automates, les grammaires formelles et les langages.

Finalement…

Cours 1 : Mots, langages, expressions régulières (Général)

Objectifs:

- Développer la compréhension des notions de mots, langages

Cours 2 & 3 : AEF

- Introduire et étudier les automates de calcul abstraits (tels que les machines à états finis, les machines à piles, et les Machines de Turing),

Cours 5 : AAP

Cours 6 : Machines de Turing

- Développer la compréhension des différents types de grammaires formelles

Cours 4 : Grammaires

- Comprendre les relations entre ces différents automates, les grammaires formelles et les langages.

A partir du cours 2

Machines de Turing

MIASHS L2

(d'après le cours de Julie Dugdale)

Jérôme GENSEL – Jerome.Gensel@univ-grenoble-alpes.fr
Prof. Université Grenoble Alpes
UFR SHS – Laboratoire d'Informatique de Grenoble