

Les machines de Turing (suite)

Une précision

Le temps de calcul d'une machine de Turing est une fonction de complexité $T(N)$, qui dépend de N , la taille des données (nombre de cases de la bande).

Formellement on définit $T(N)$ comme le nombre maximum de transitions pour une donnée de taille N .

Notre simulateur

Vous pouvez le récupérer à l'adresse

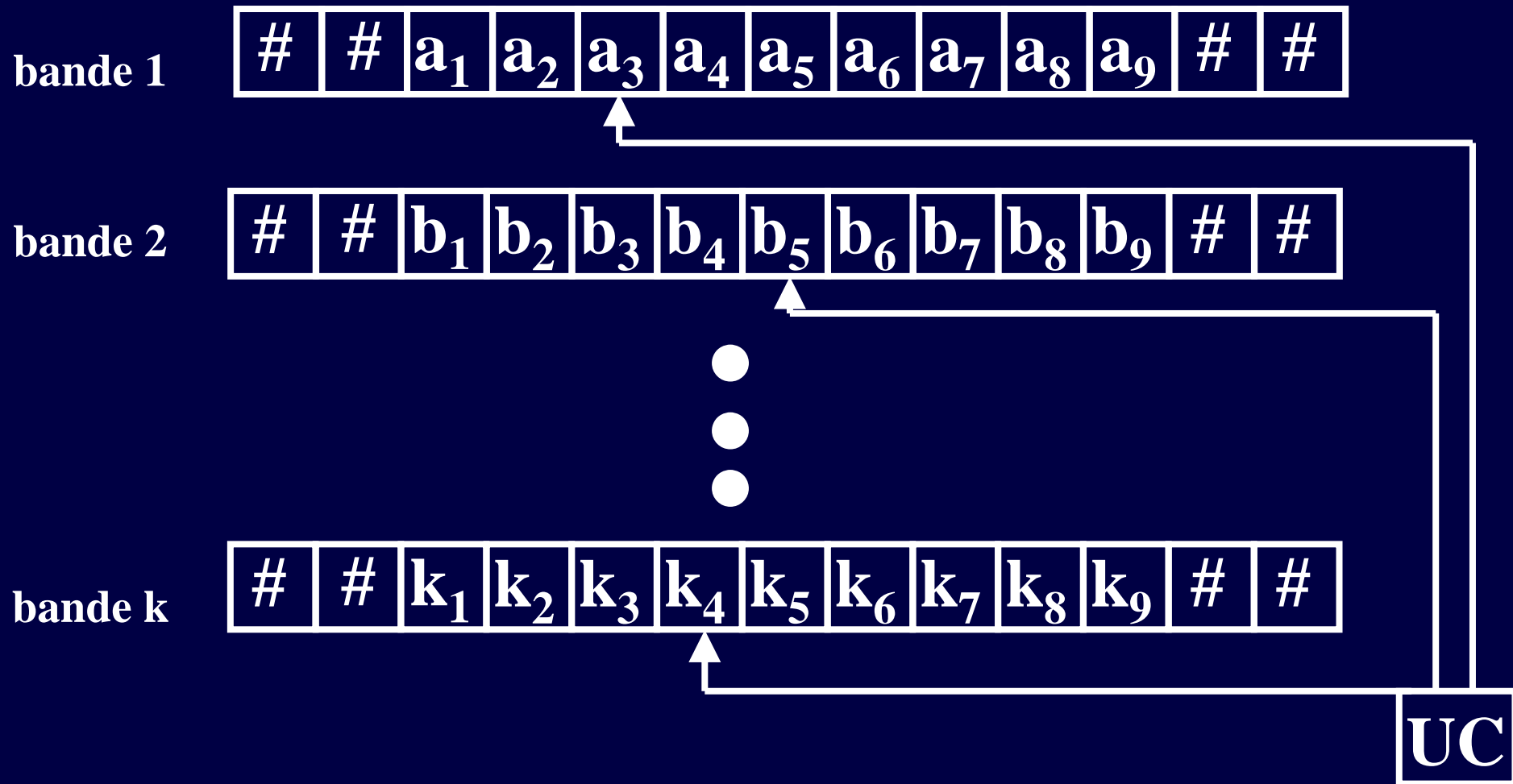
<https://github.com/ChristophePapazian/Automatic-Machines-Simulator>

Equivalences

Théorème :

Si le langage L est accepté par une machine de Turing T à k bandes, alors il existe une machine de Turing T' à une bande pour reconnaître L .

Une machine à k bandes



Preuve

Nous introduisons la notion de piste. Construisons en effet T' à une bande, mais à $2k$ pistes, simulant T .

Ainsi pour décrire ce qui se passe sur la bande i on dispose de deux pistes.

- La piste $2i$, a des blancs partout, à part la case sur laquelle se trouve la tête de lecture/écriture.
- La piste $2i-1$ contient exactement l'information de la bande i .

Pour que la bande de T' puisse contenir cette information sur les $2k$ pistes, on aura un alphabet plus grand, permettant le codage de chaque $2k$ -uplet possible en une seule lettre.

Une machine à k bandes

piste 1

piste 2

piste 3

piste 4

#	#	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	#	#
#	#	#	#	X	#	#	#	#	#	#	#	#
#	#	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	#	#
#	#	#	#	#	#	X	#	#	#	#	#	#

piste $2k-1$

piste $2k$

#	#	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	#	#
#	#	#	#	#	X	#	#	#	#	#	#	#

Pour simuler le fonctionnement de T, la machine T' fonctionne comme suit :

- on «visite» toutes les cases marquées pour lire l'information se trouvant sur chacune des bandes. L'information lue est conservée par l'état.
- on visite à nouveau les cases marquées pour effectuer les changements d'écriture et de marquage dues à la transition.

La même méthode d'utilisation de pistes permet de prouver le résultat suivant :

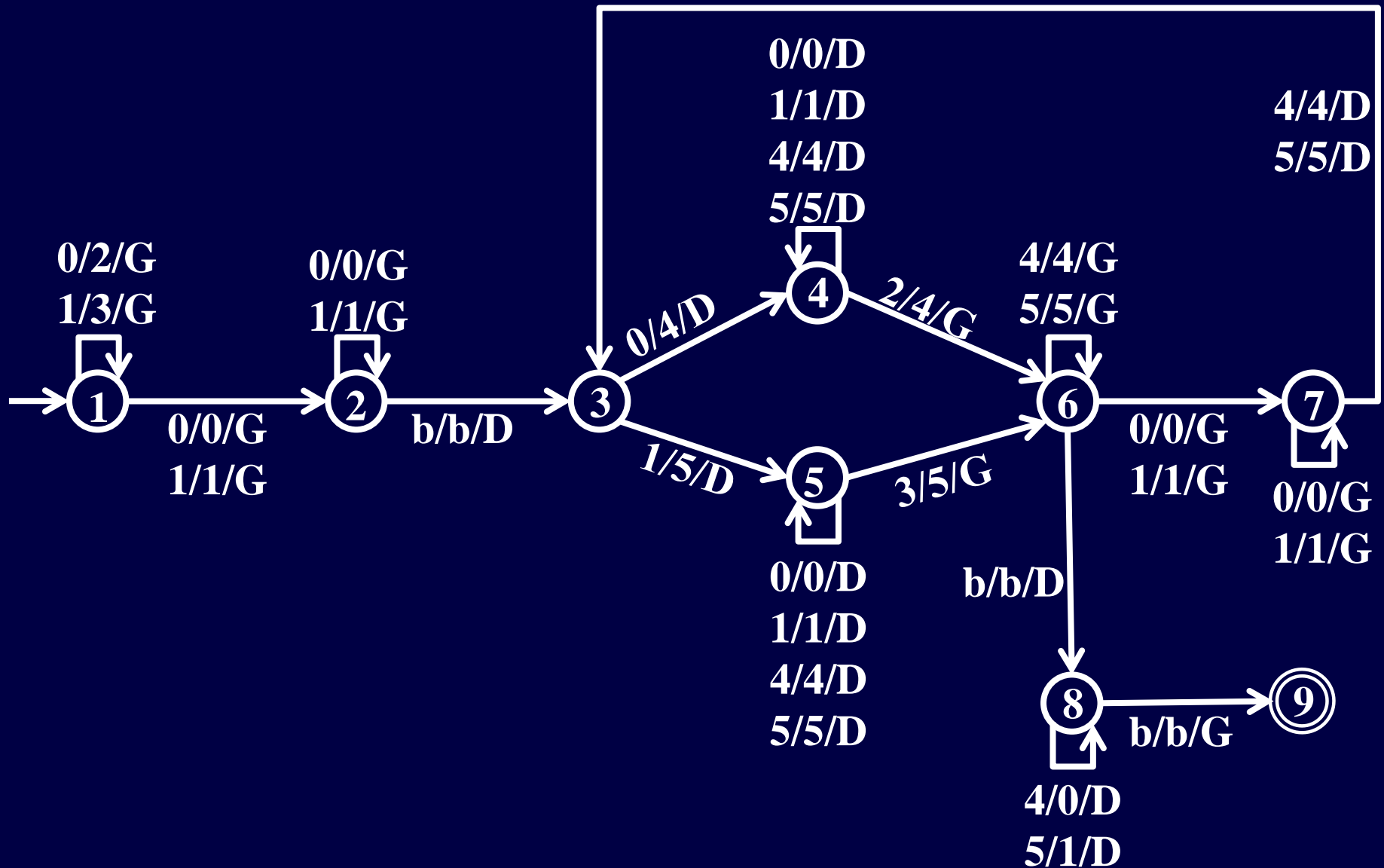
Théorème : Si le langage L est accepté par une machine de Turing T à une bande infinie des deux côtés, alors il existe une machine de Turing T' à une bande infinie d'un seul côté pour reconnaître L .

Preuve : Il suffit d'utiliser une machine à trois pistes
...

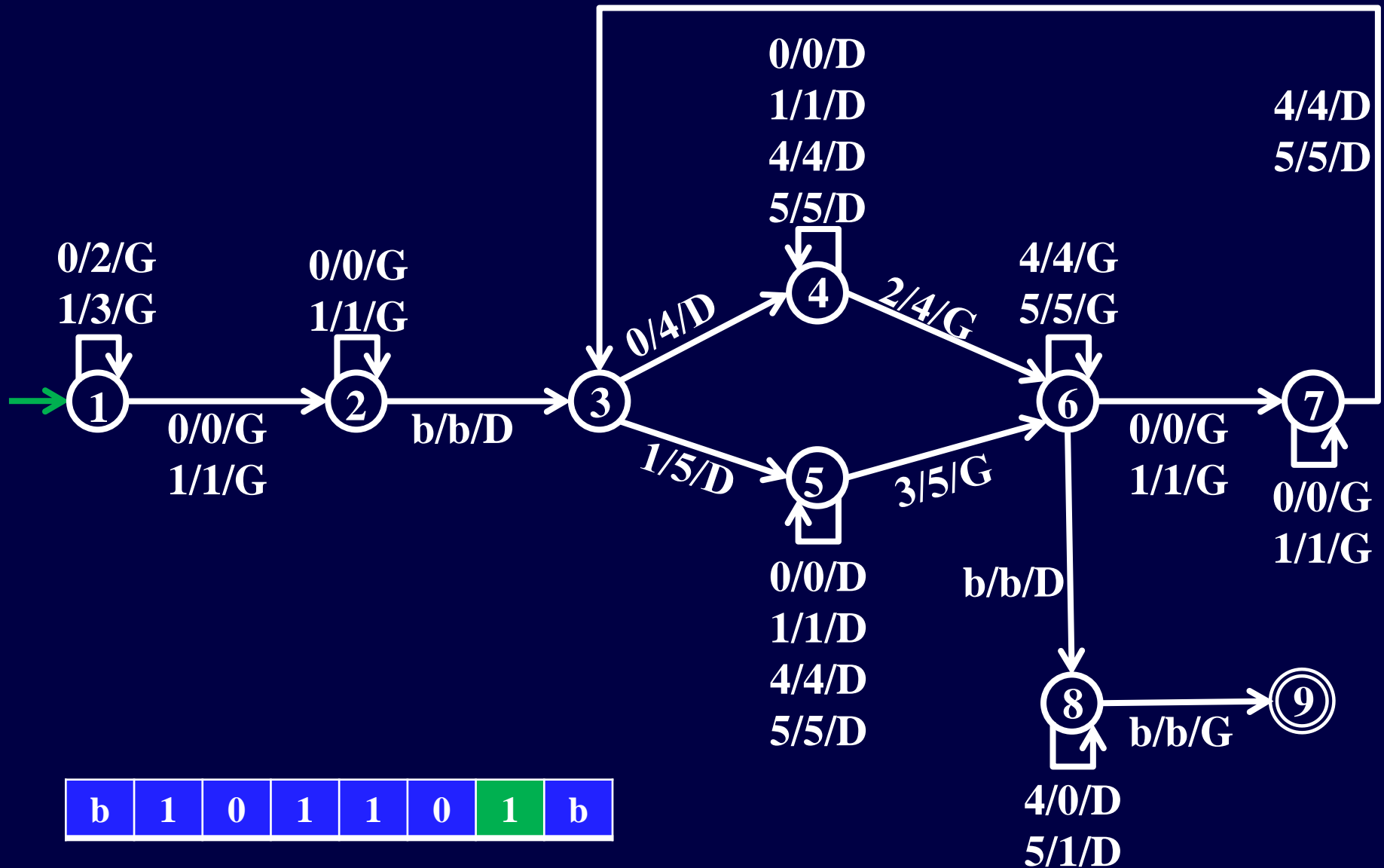
On peut généraliser ce résultat :

Théorème : Si le langage L est accepté par une machine de Turing T à k bandes infinies des deux côtés, alors il existe une machine de Turing T' à k bandes infinies d'un seul côté pour reconnaître L .

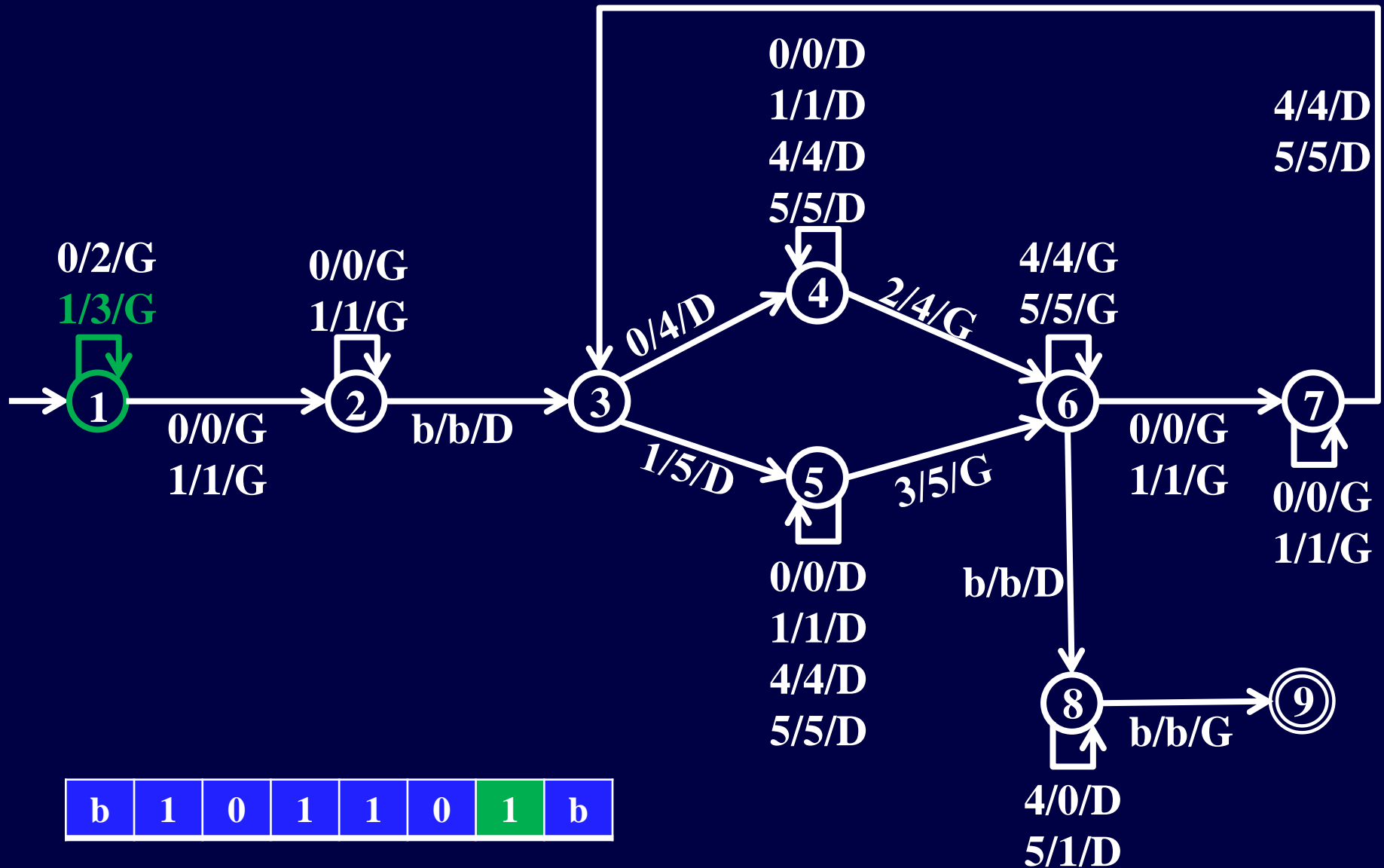
Un exemple de machine non-déterministe



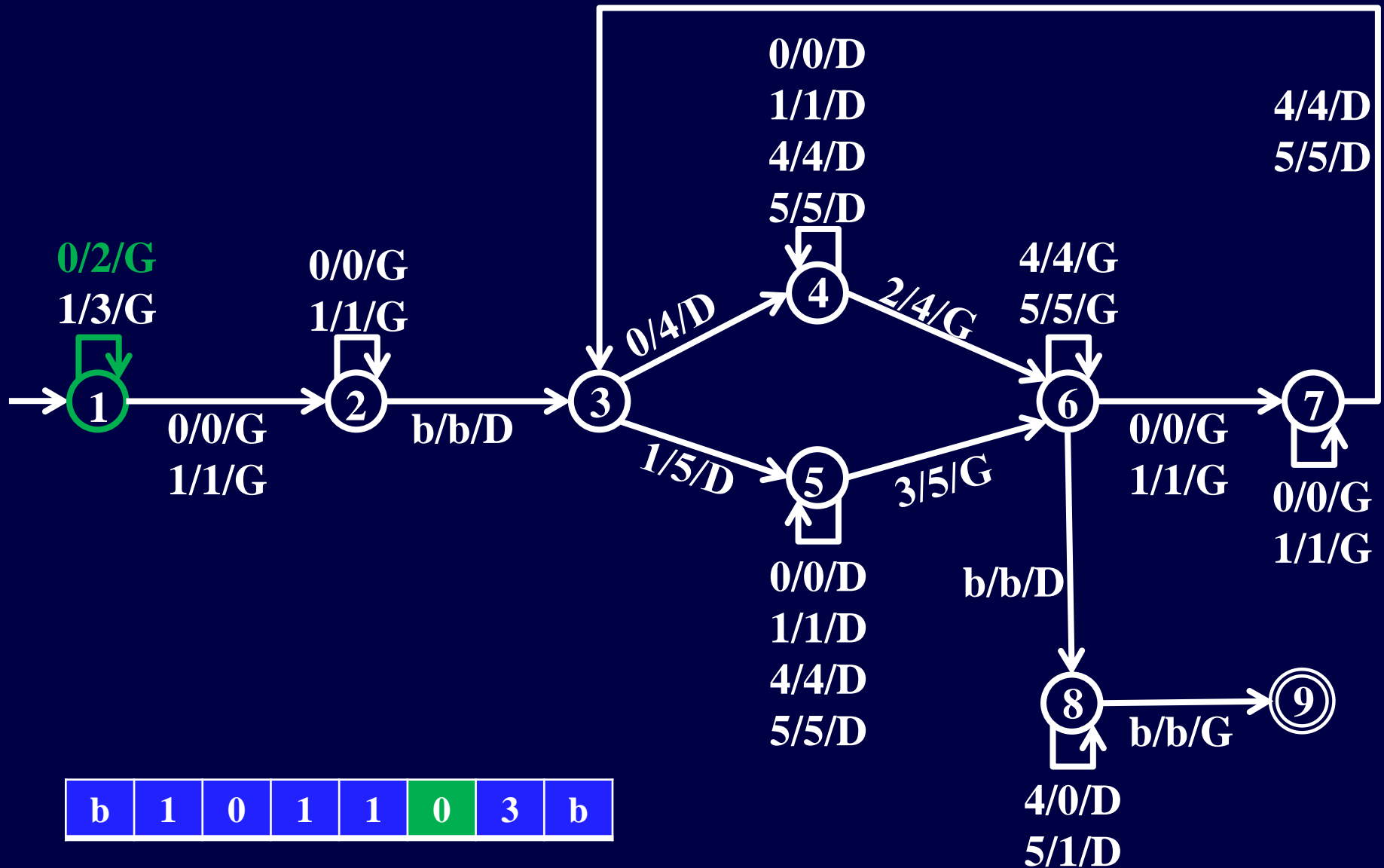
Un exemple de machine non-déterministe



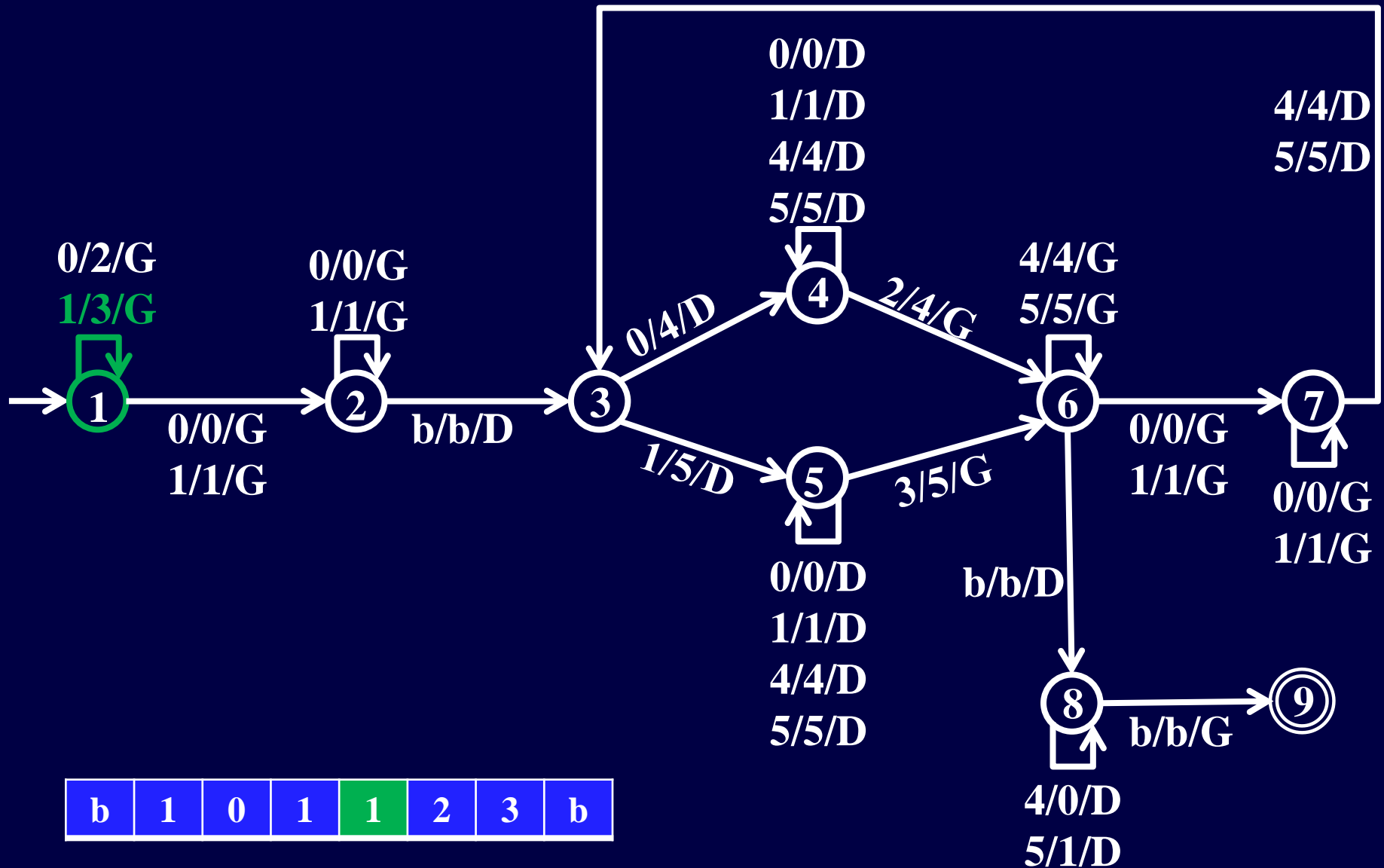
Un exemple de machine non-déterministe



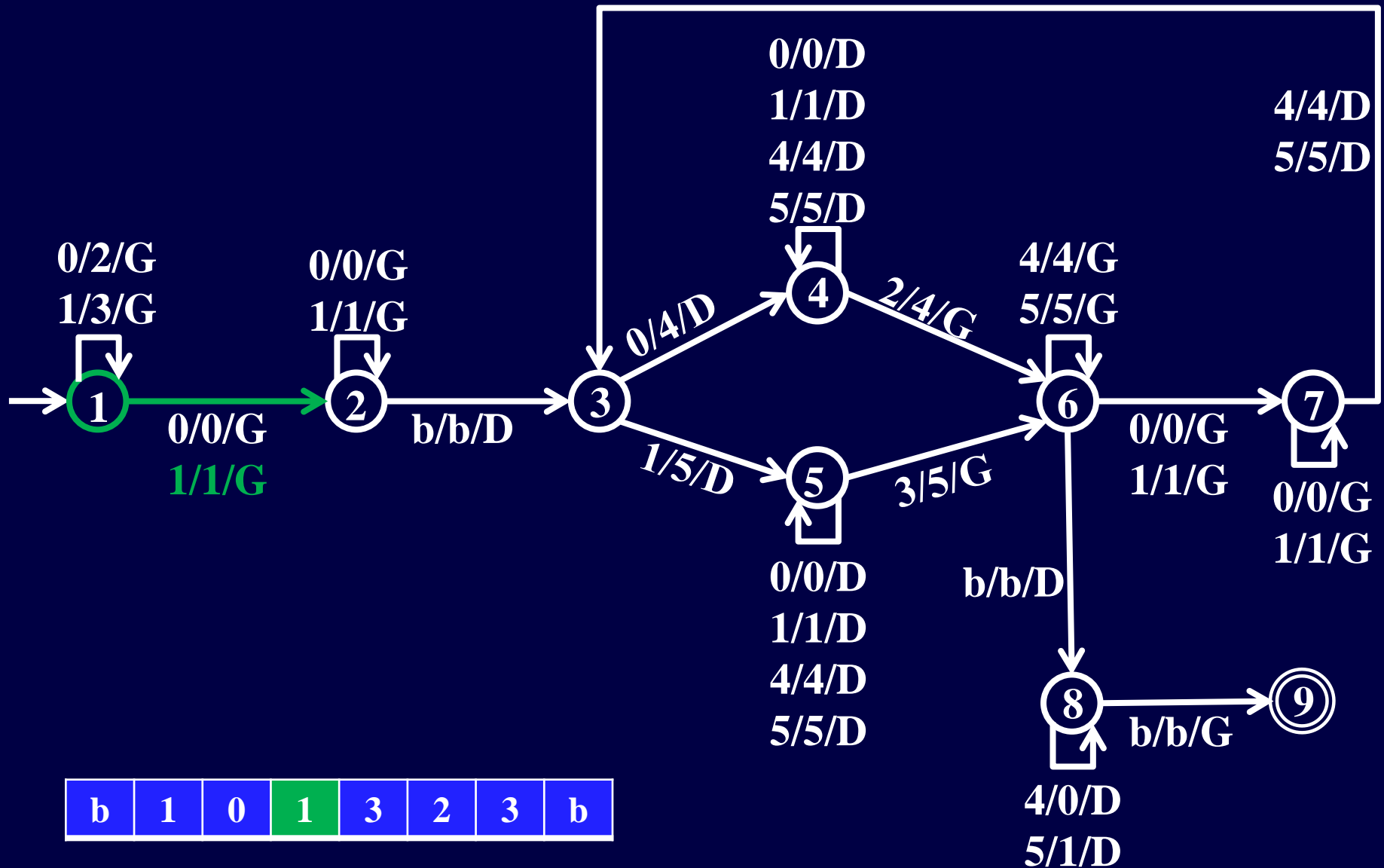
Un exemple de machine non-déterministe



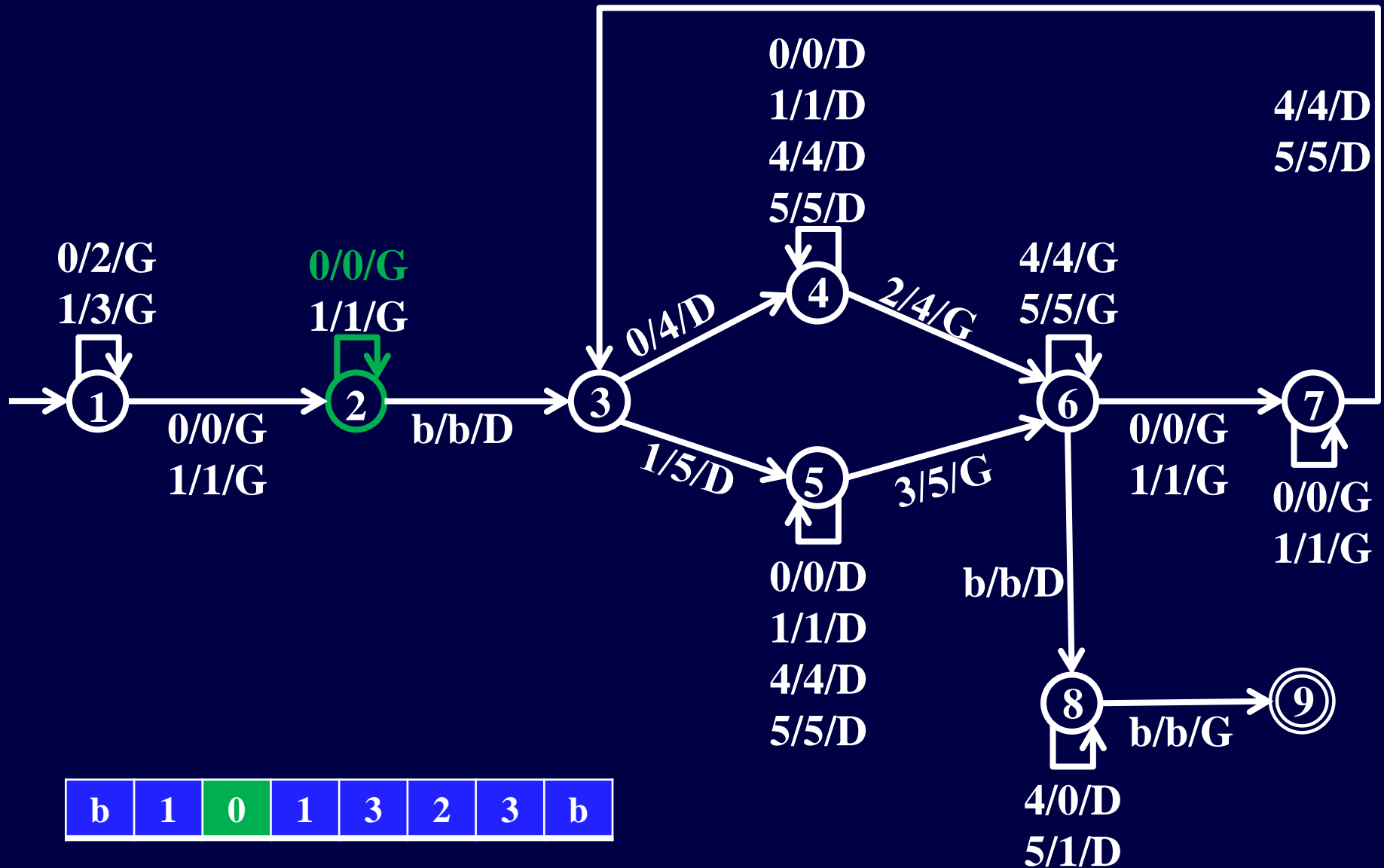
Un exemple de machine non-déterministe



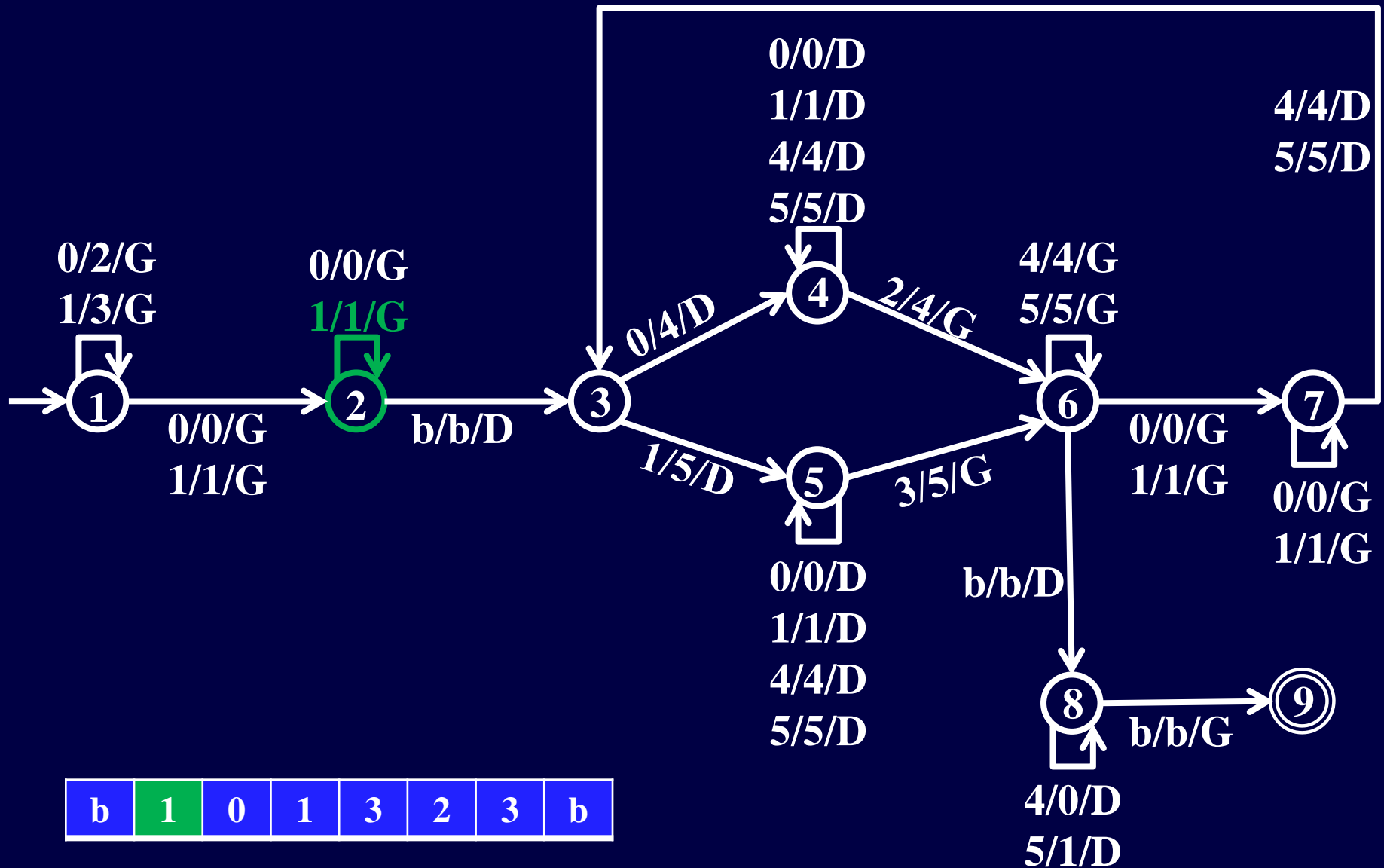
Un exemple de machine non-déterministe



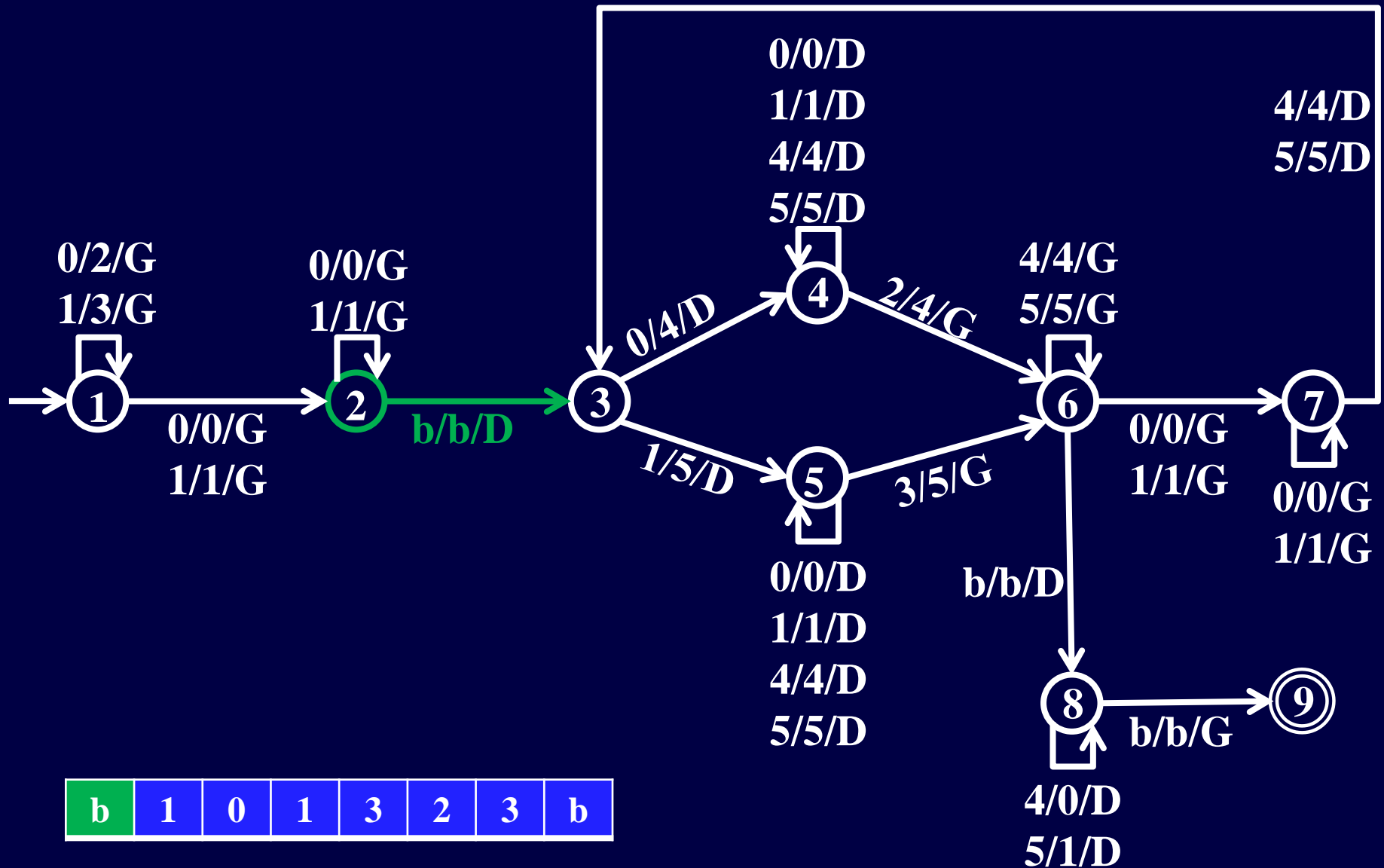
Un exemple de machine non-déterministe



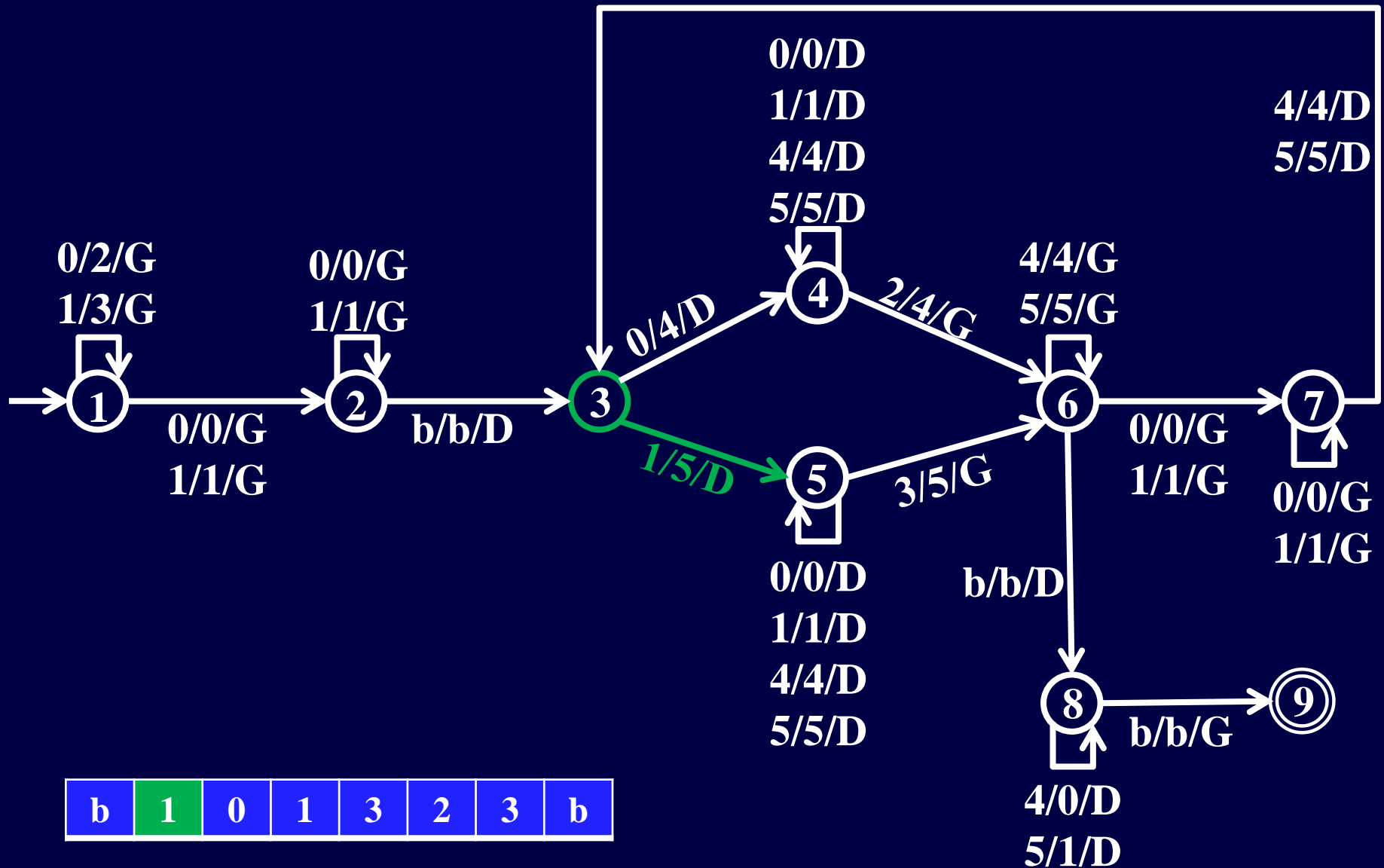
Un exemple de machine non-déterministe



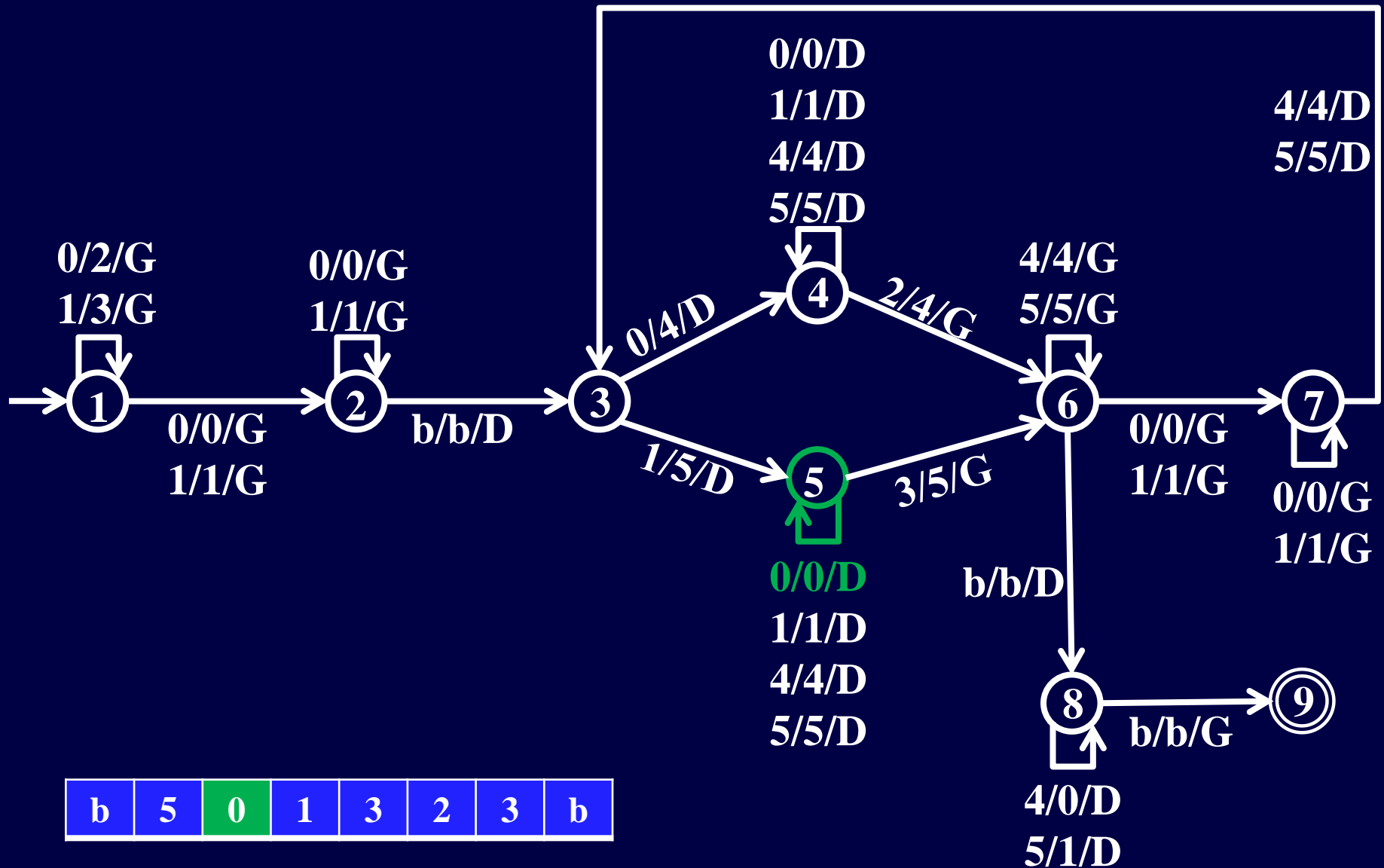
Un exemple de machine non-déterministe



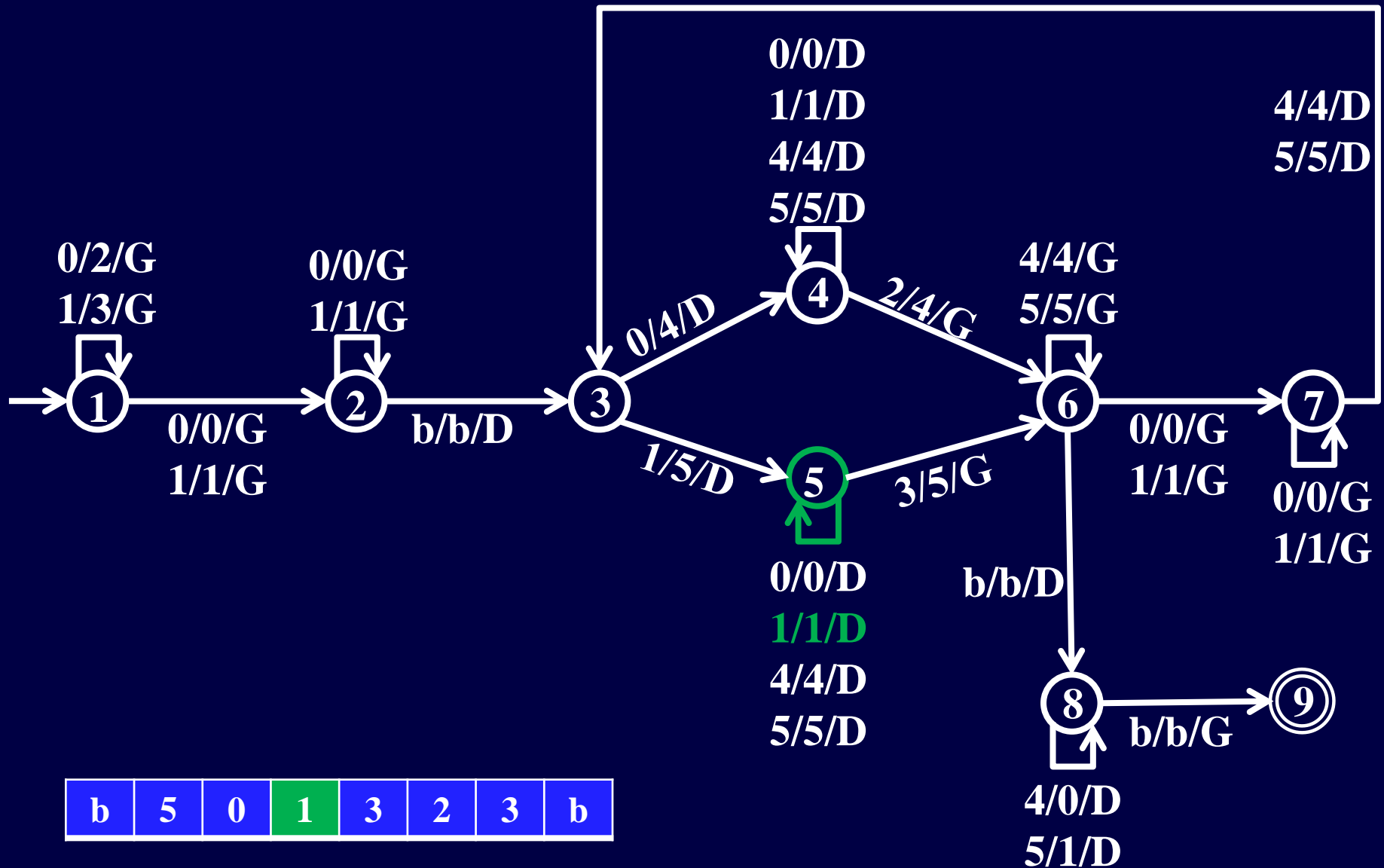
Un exemple de machine non-déterministe



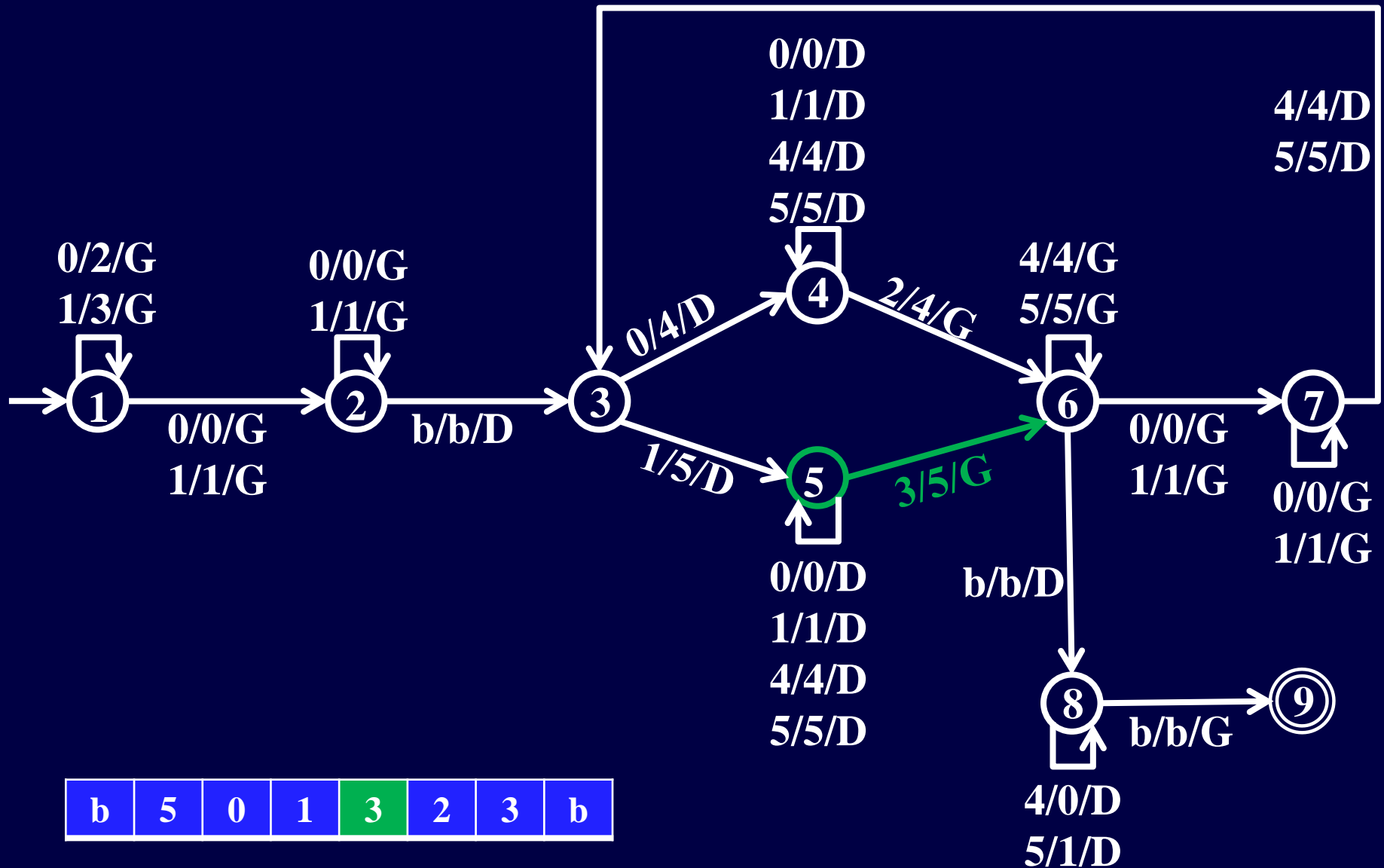
Un exemple de machine non-déterministe



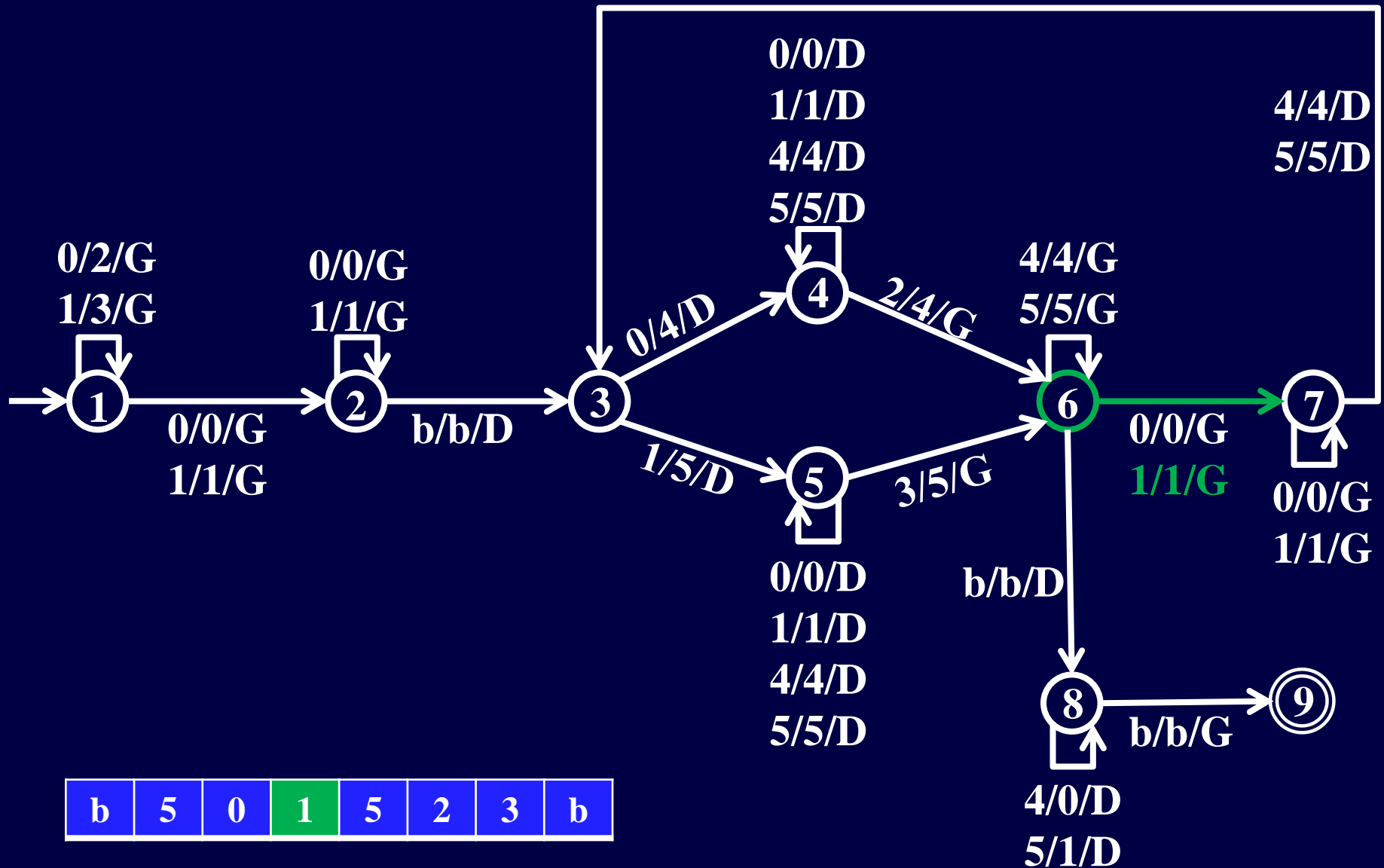
Un exemple de machine non-déterministe



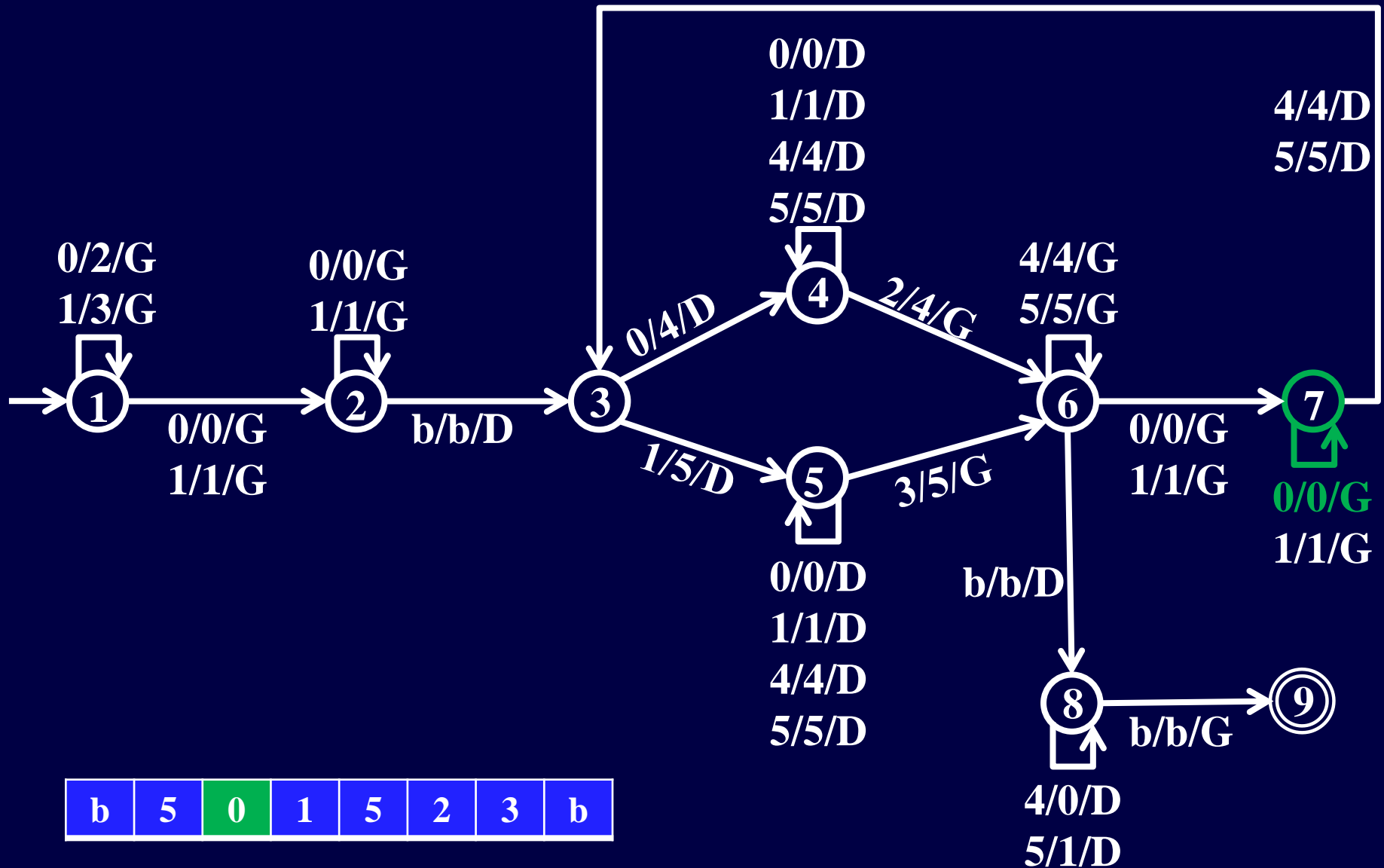
Un exemple de machine non-déterministe



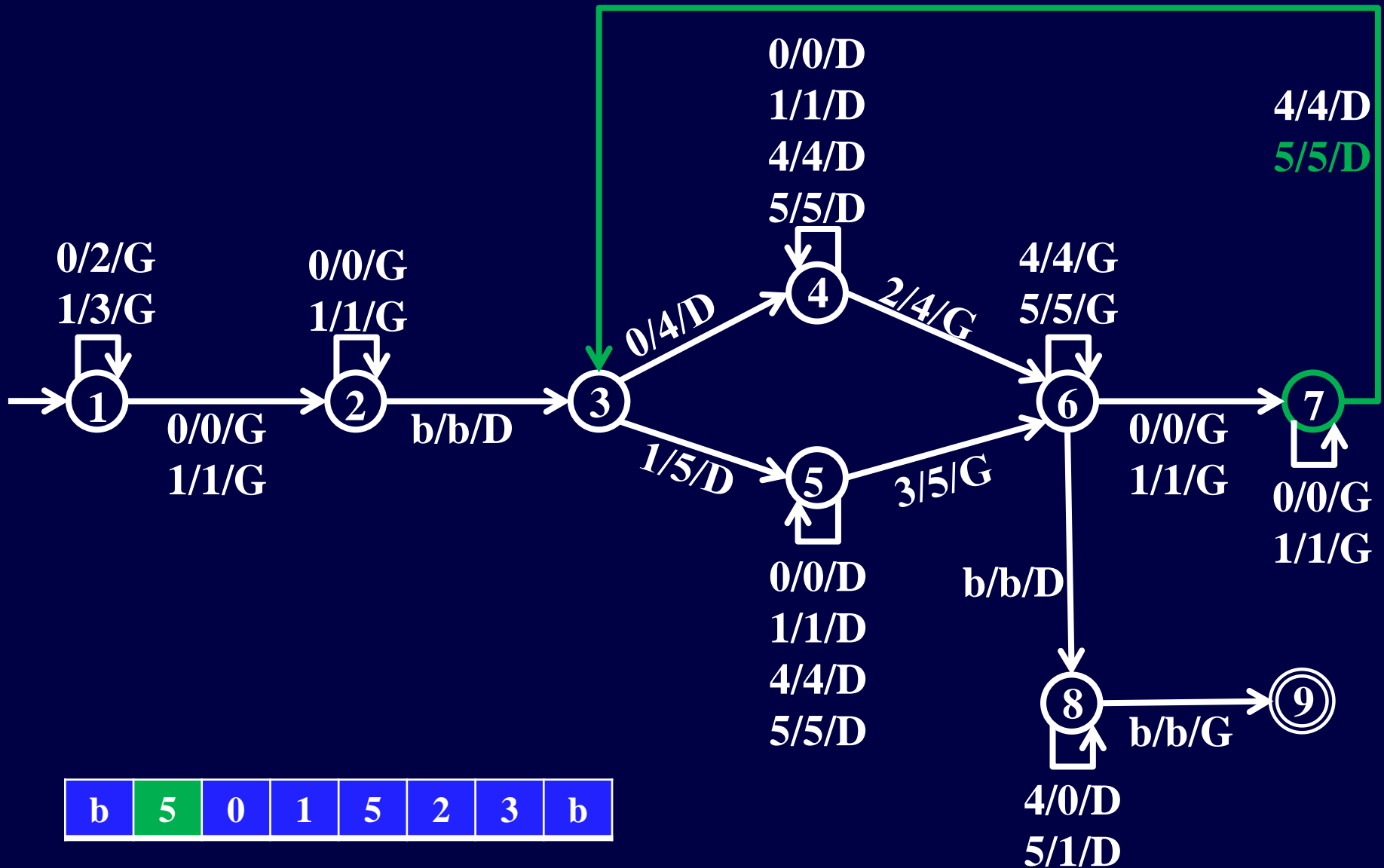
Un exemple de machine non-déterministe



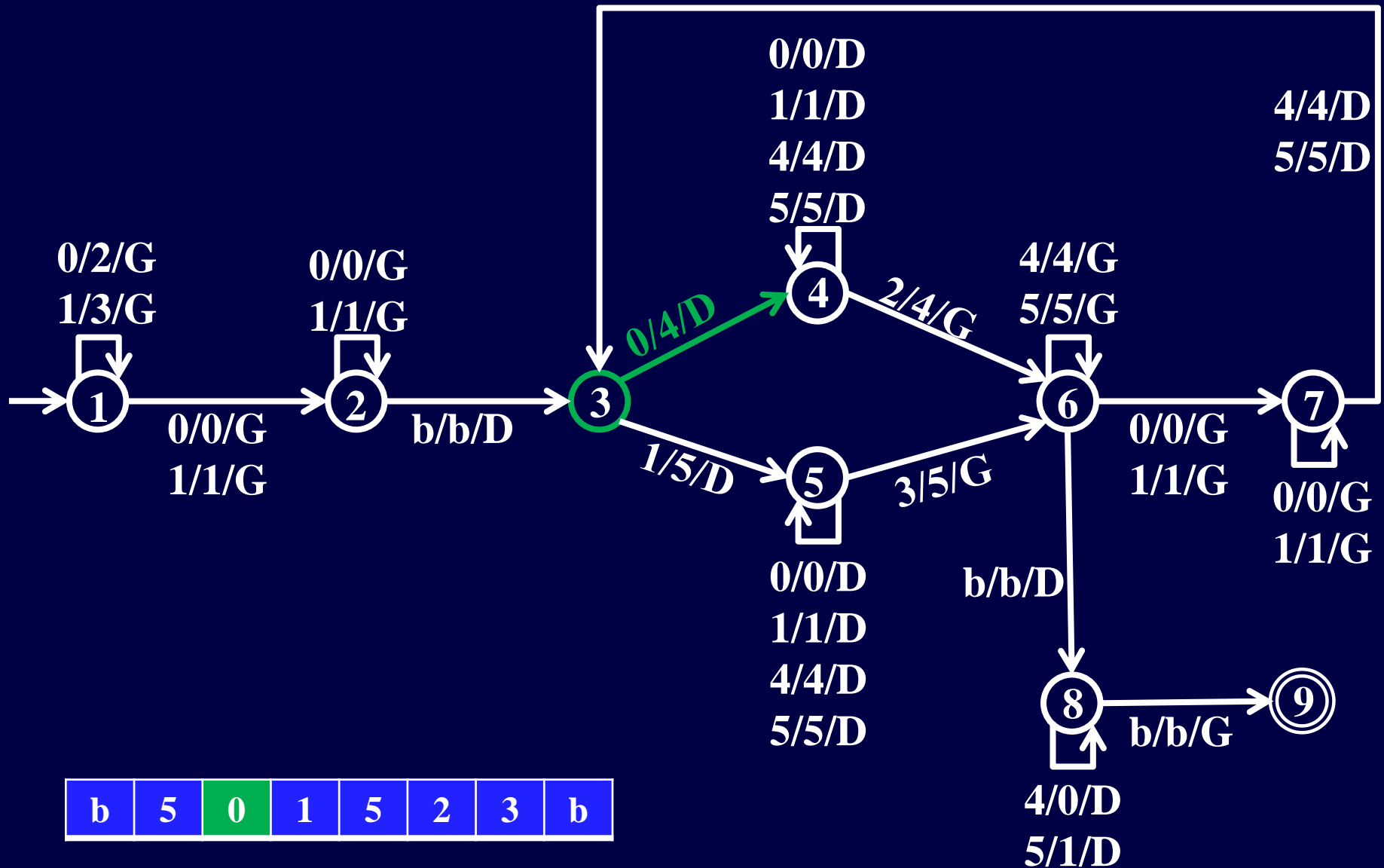
Un exemple de machine non-déterministe



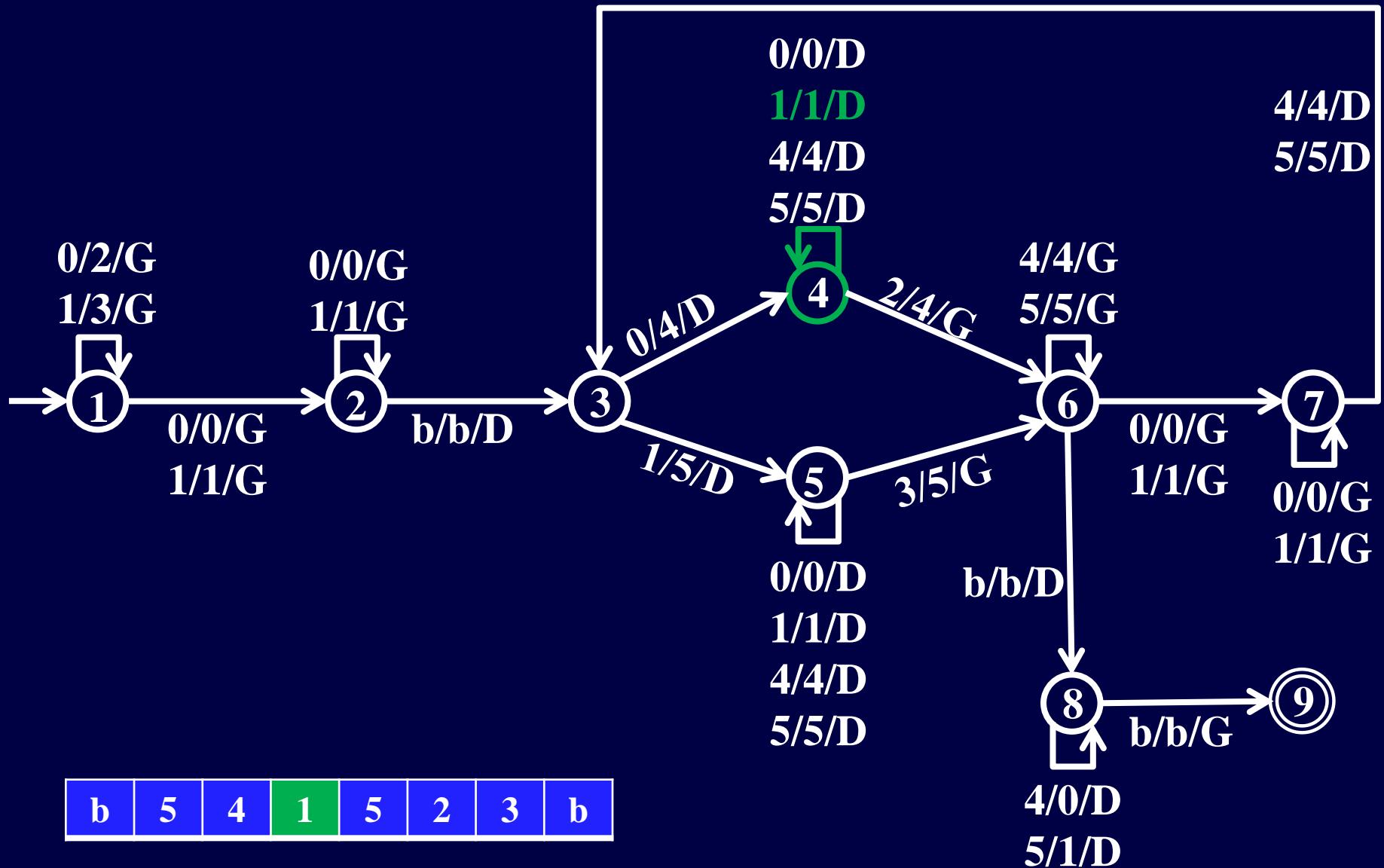
Un exemple de machine non-déterministe



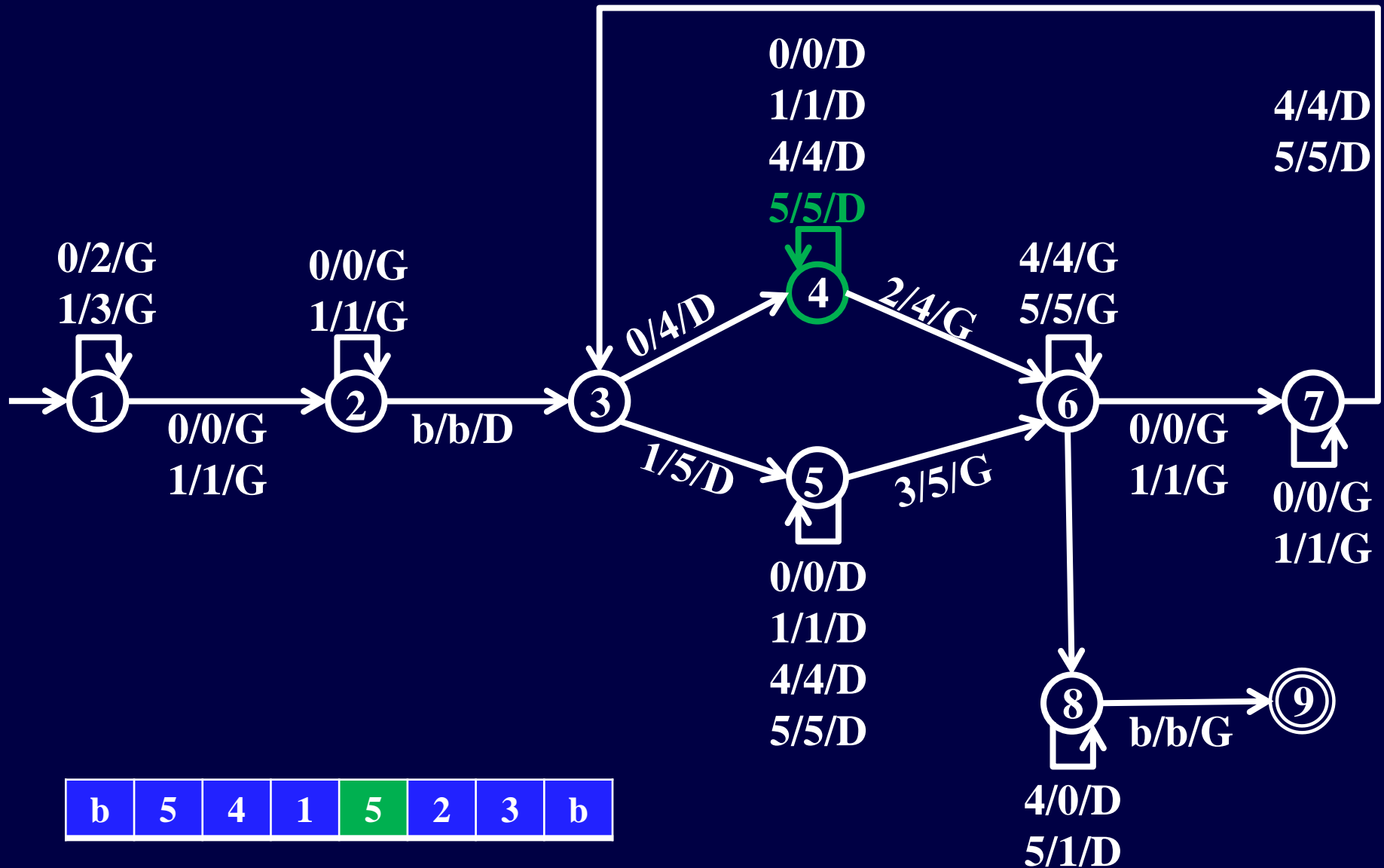
Un exemple de machine non-déterministe



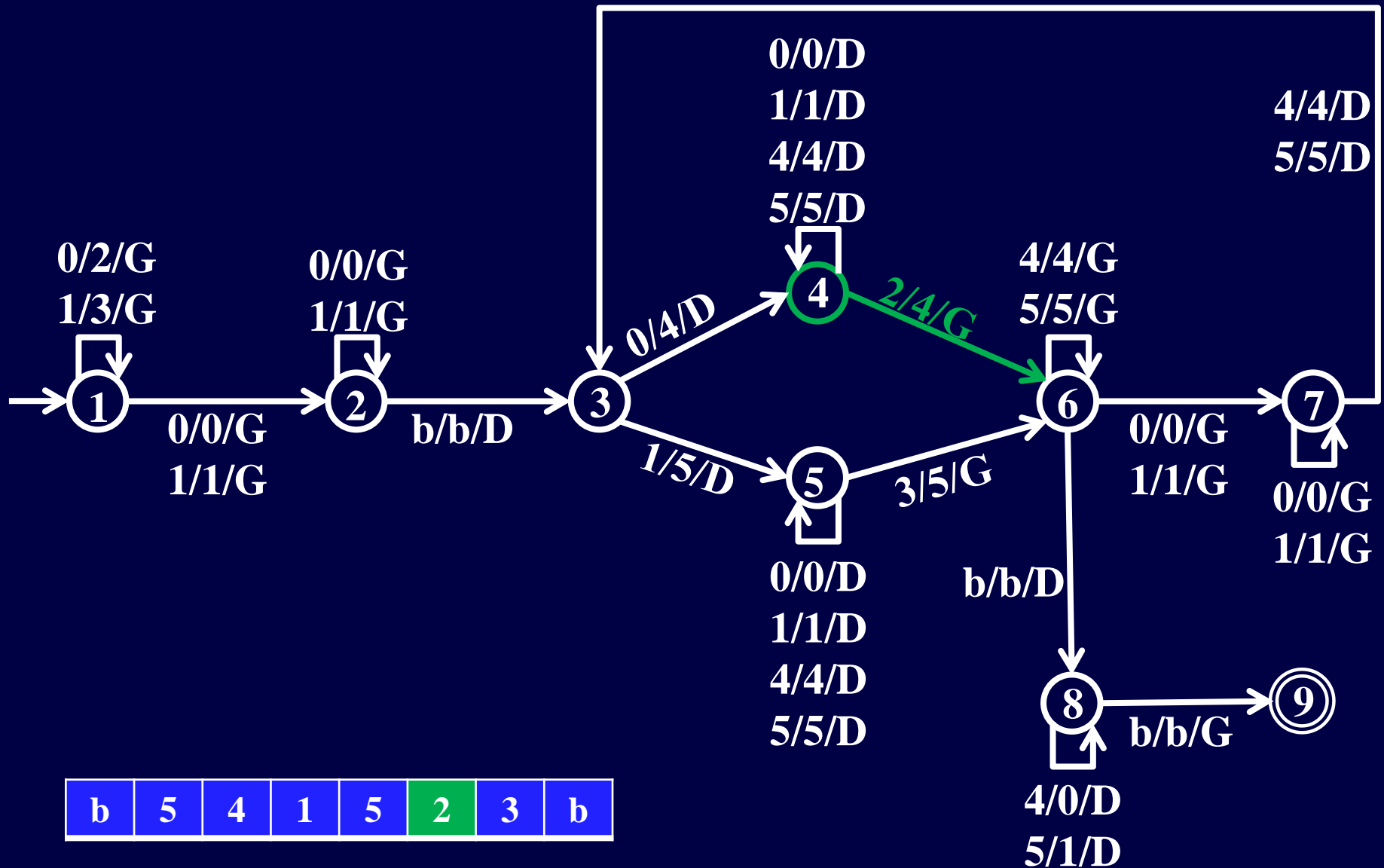
Un exemple de machine non-déterministe



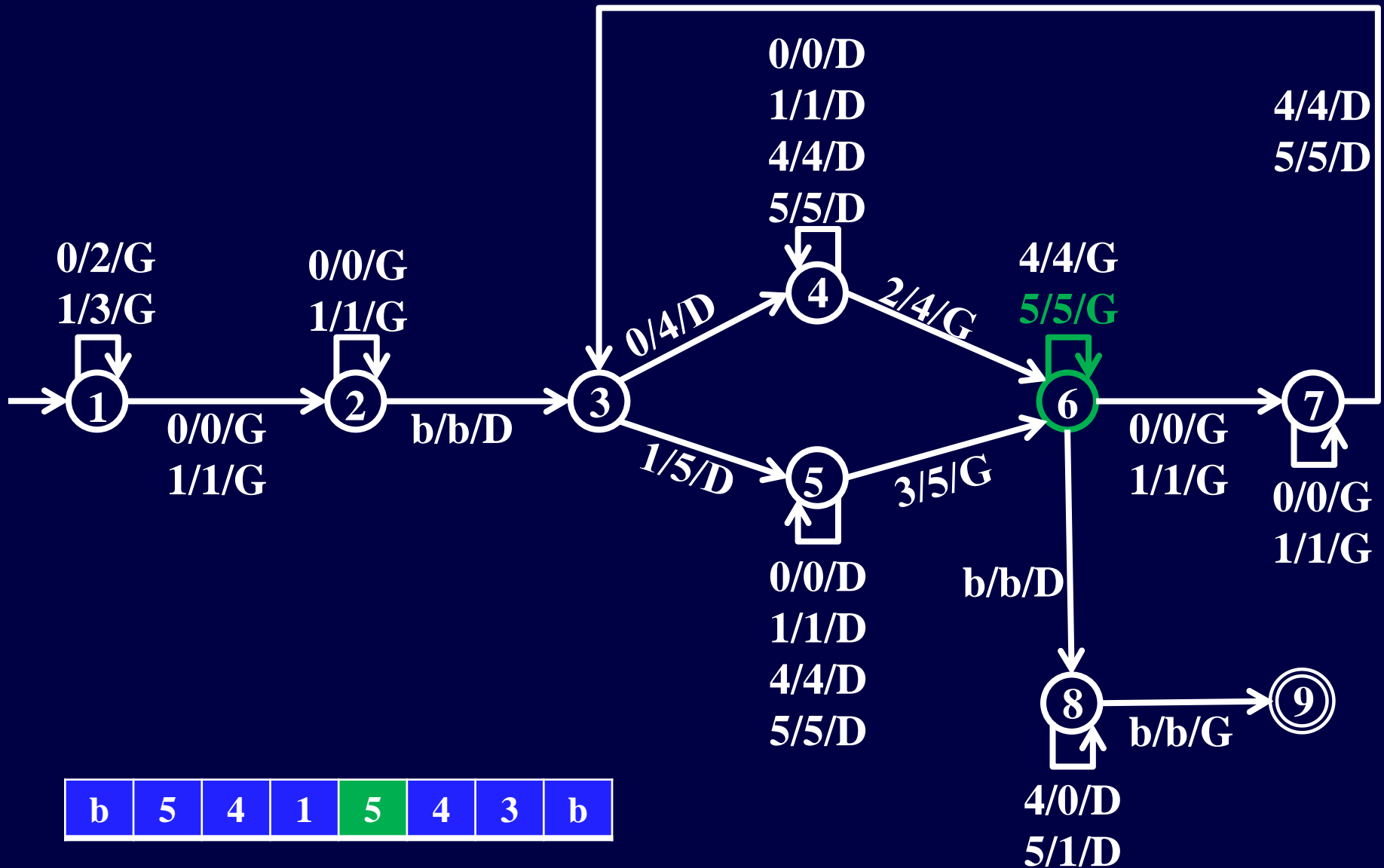
Un exemple de machine non-déterministe



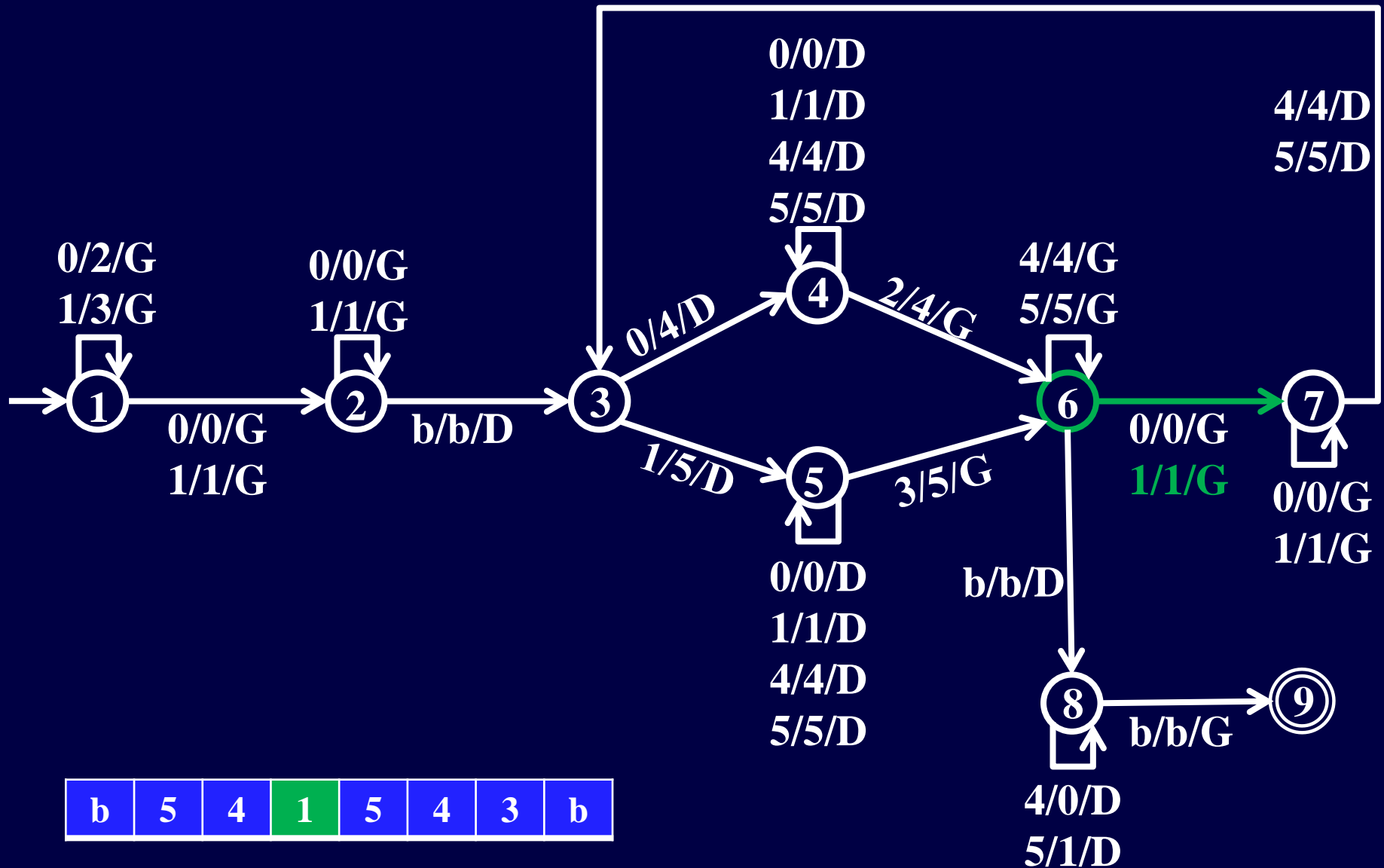
Un exemple de machine non-déterministe



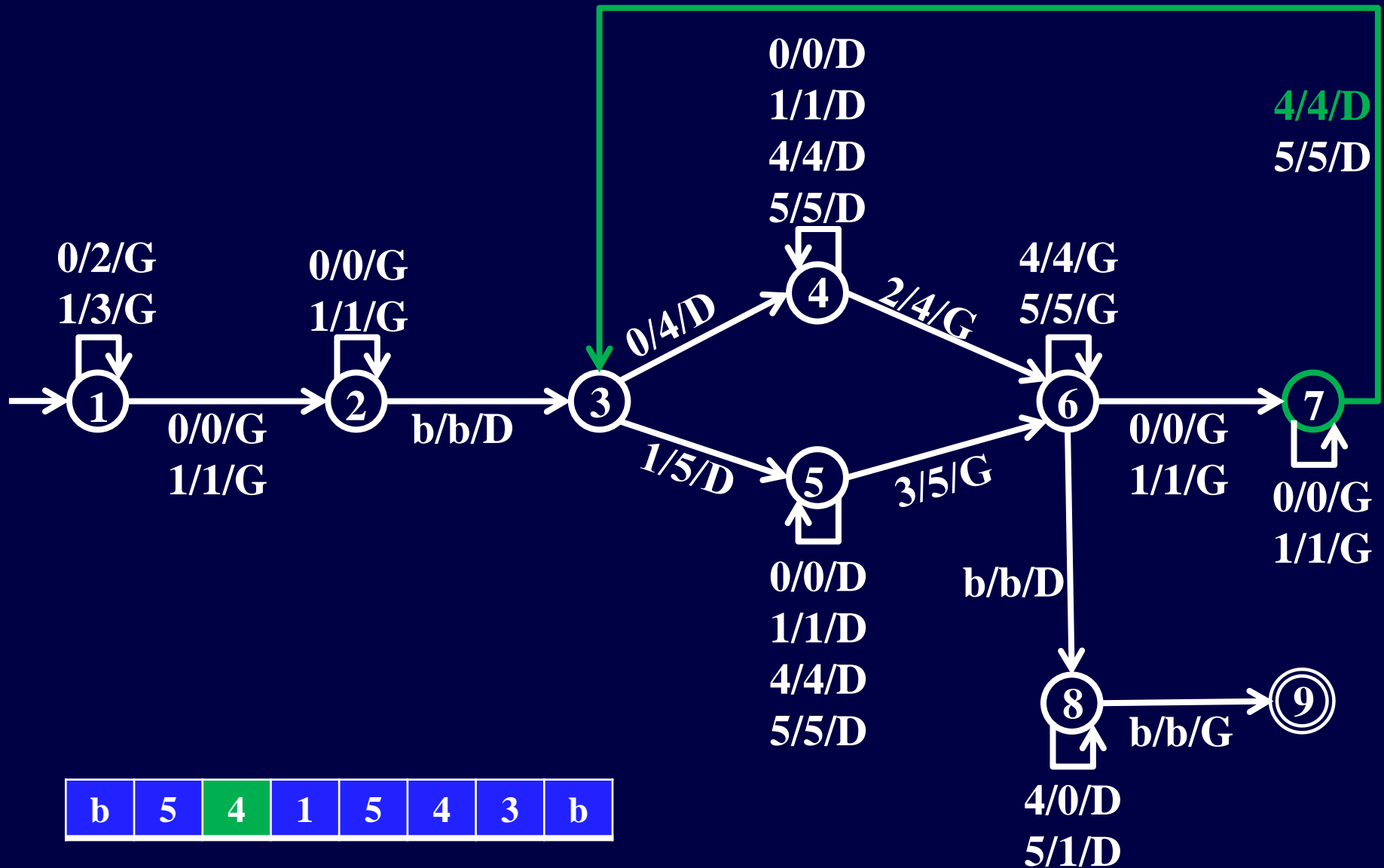
Un exemple de machine non-déterministe



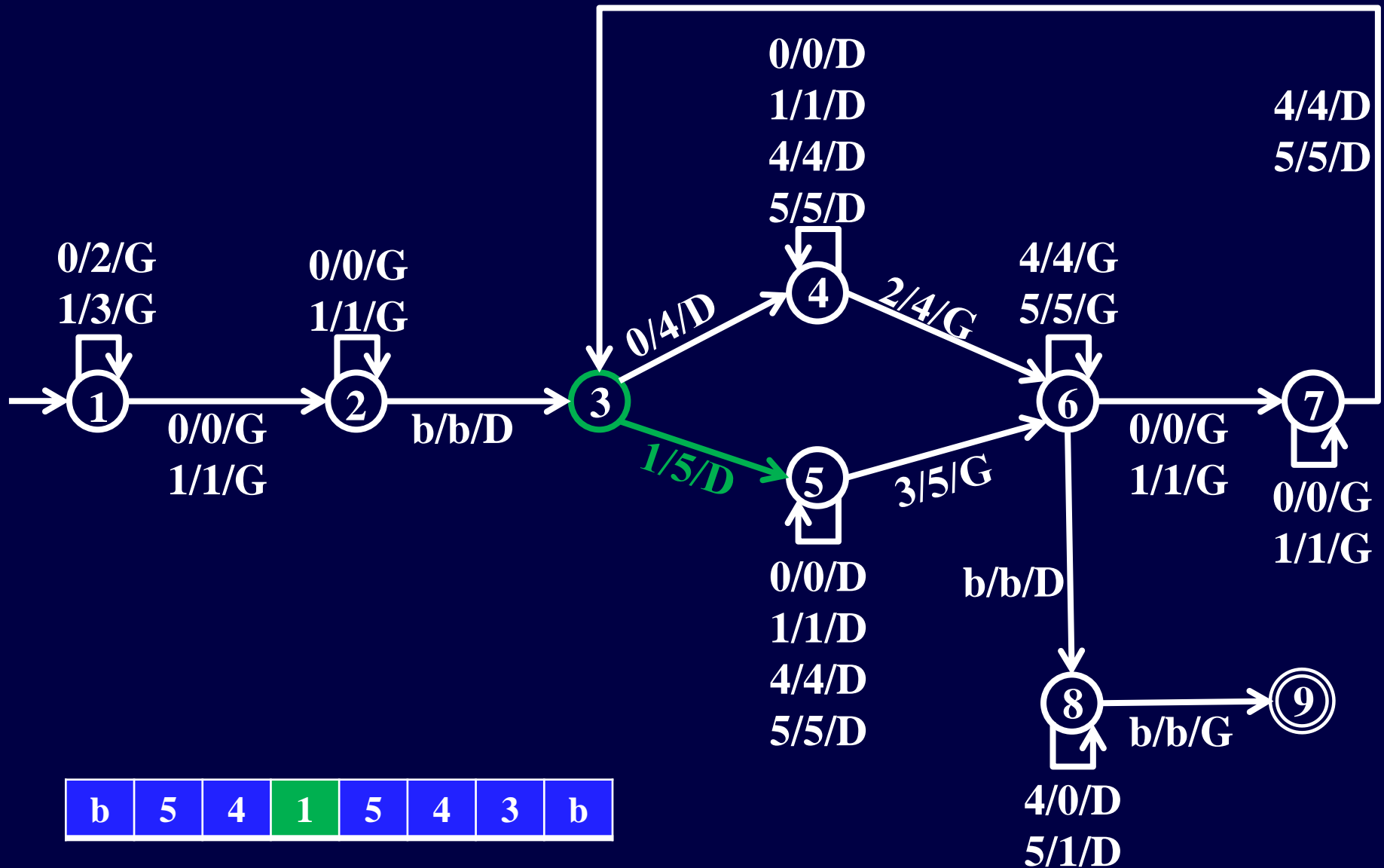
Un exemple de machine non-déterministe



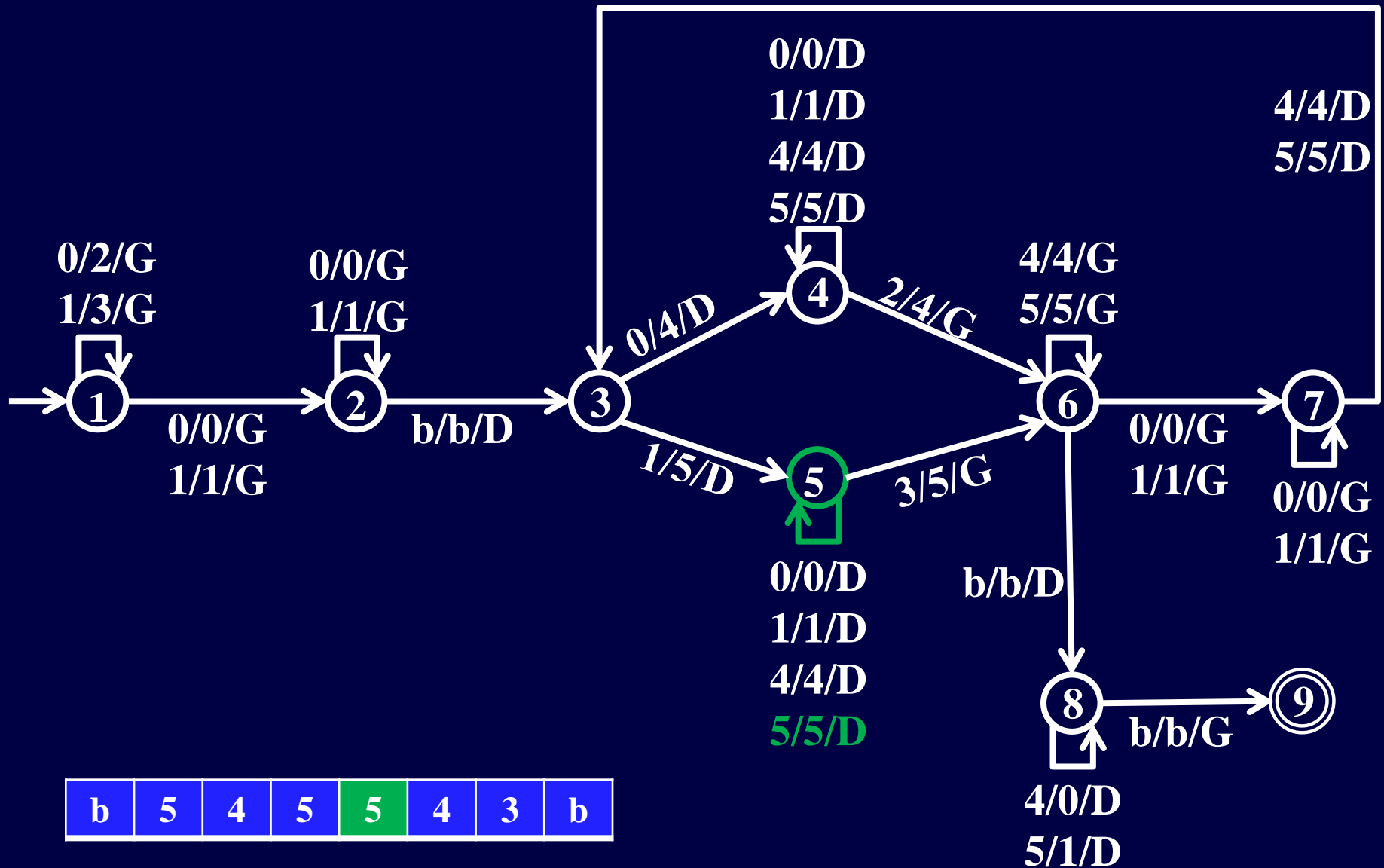
Un exemple de machine non-déterministe



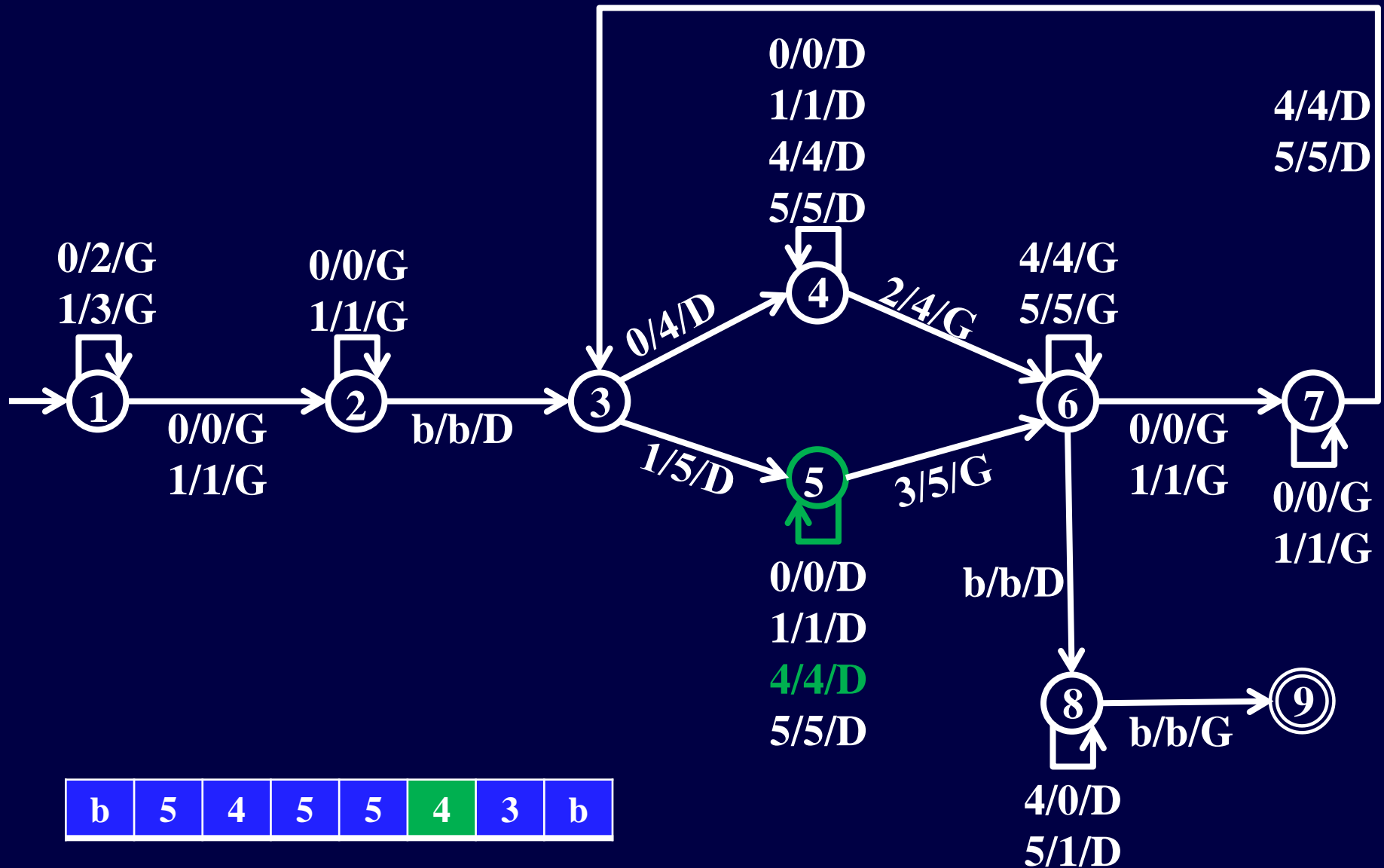
Un exemple de machine non-déterministe



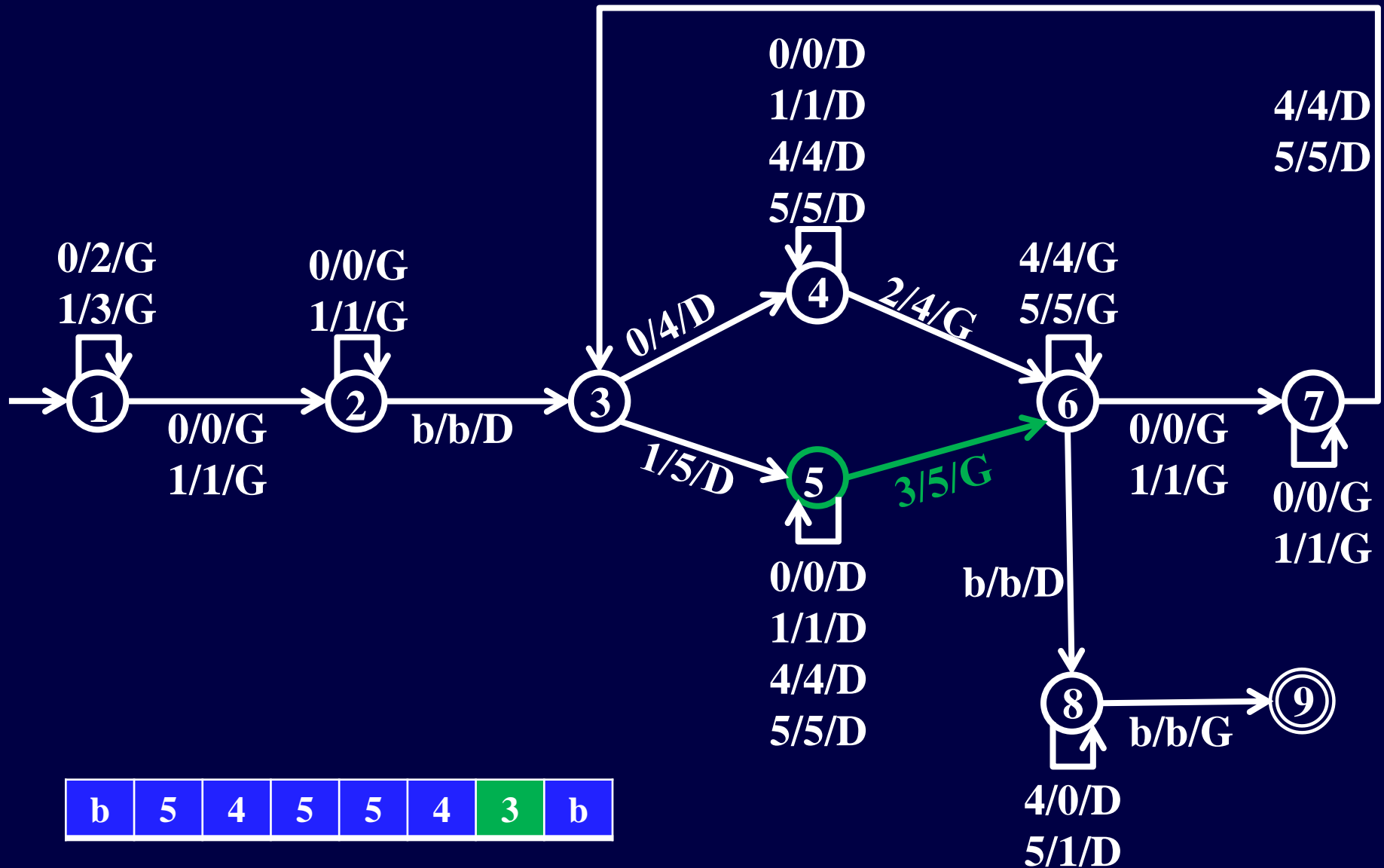
Un exemple de machine non-déterministe



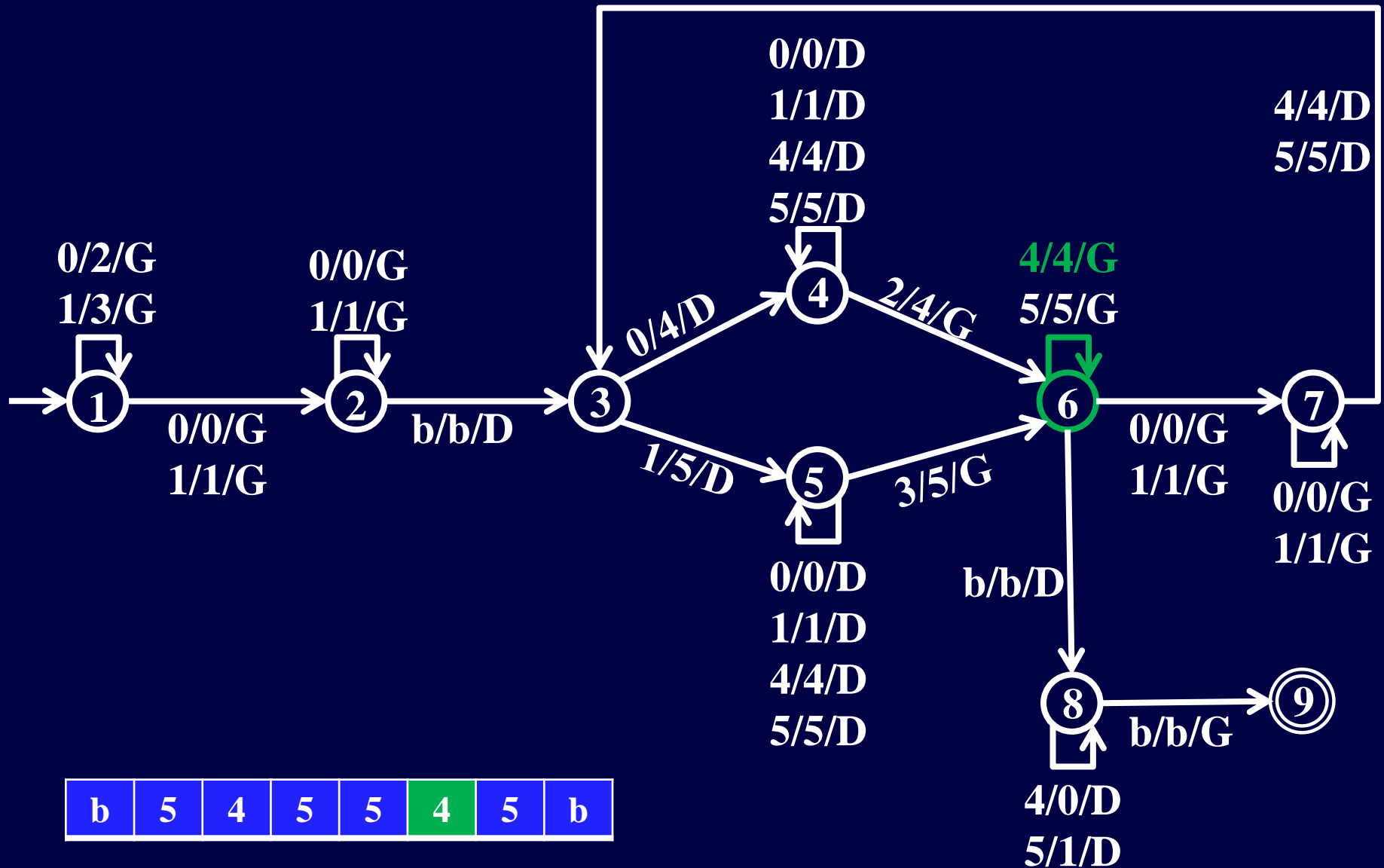
Un exemple de machine non-déterministe



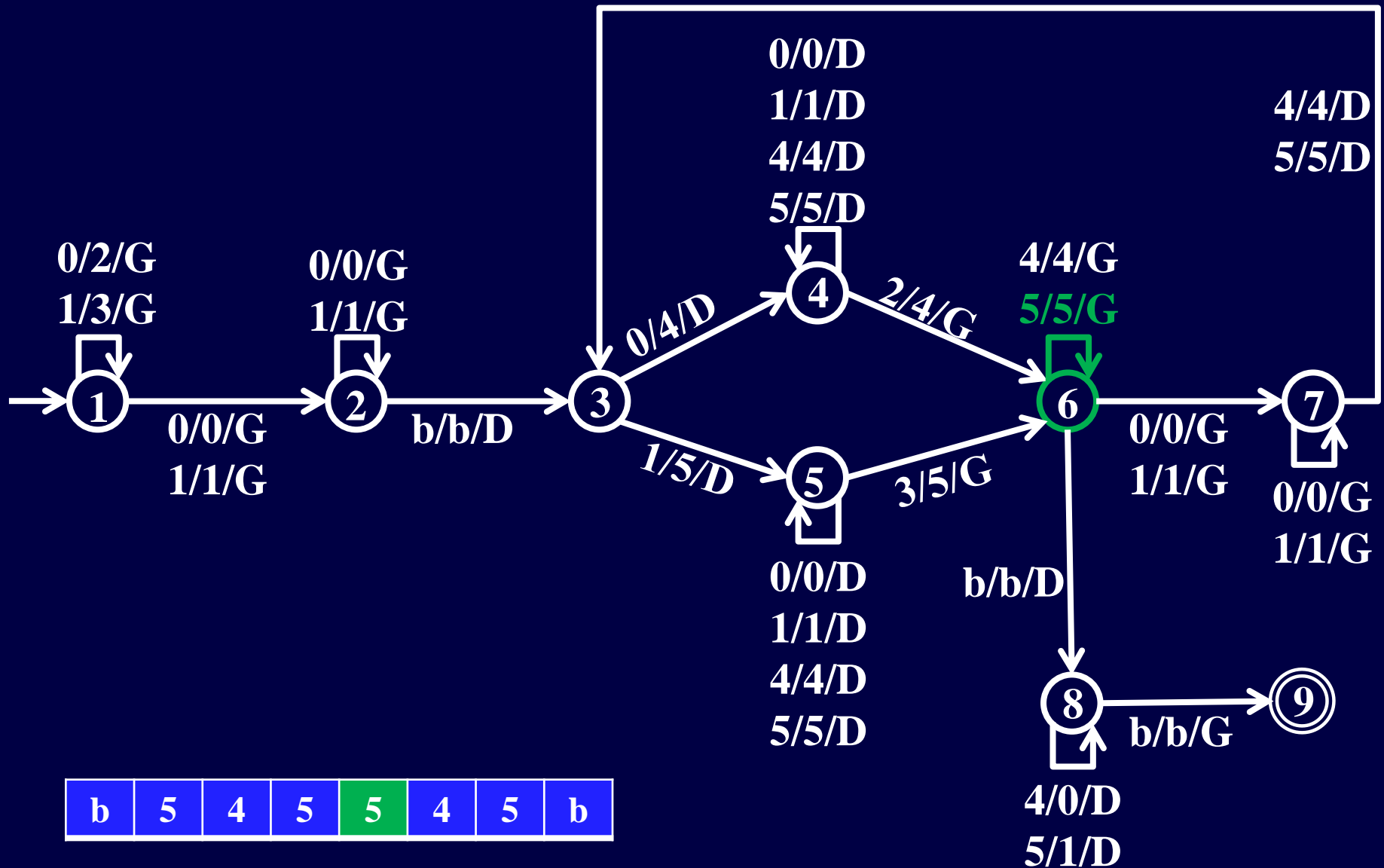
Un exemple de machine non-déterministe



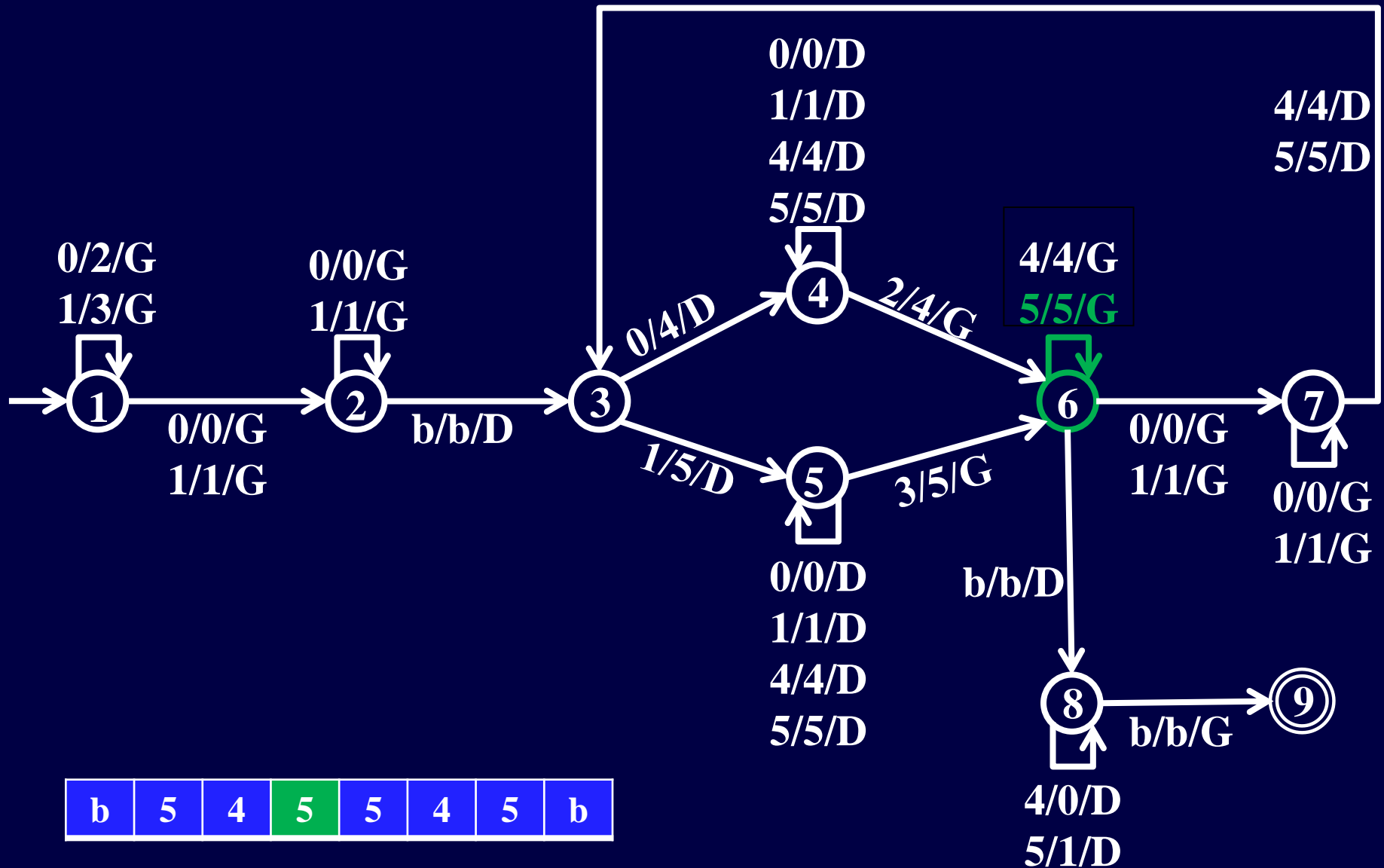
Un exemple de machine non-déterministe



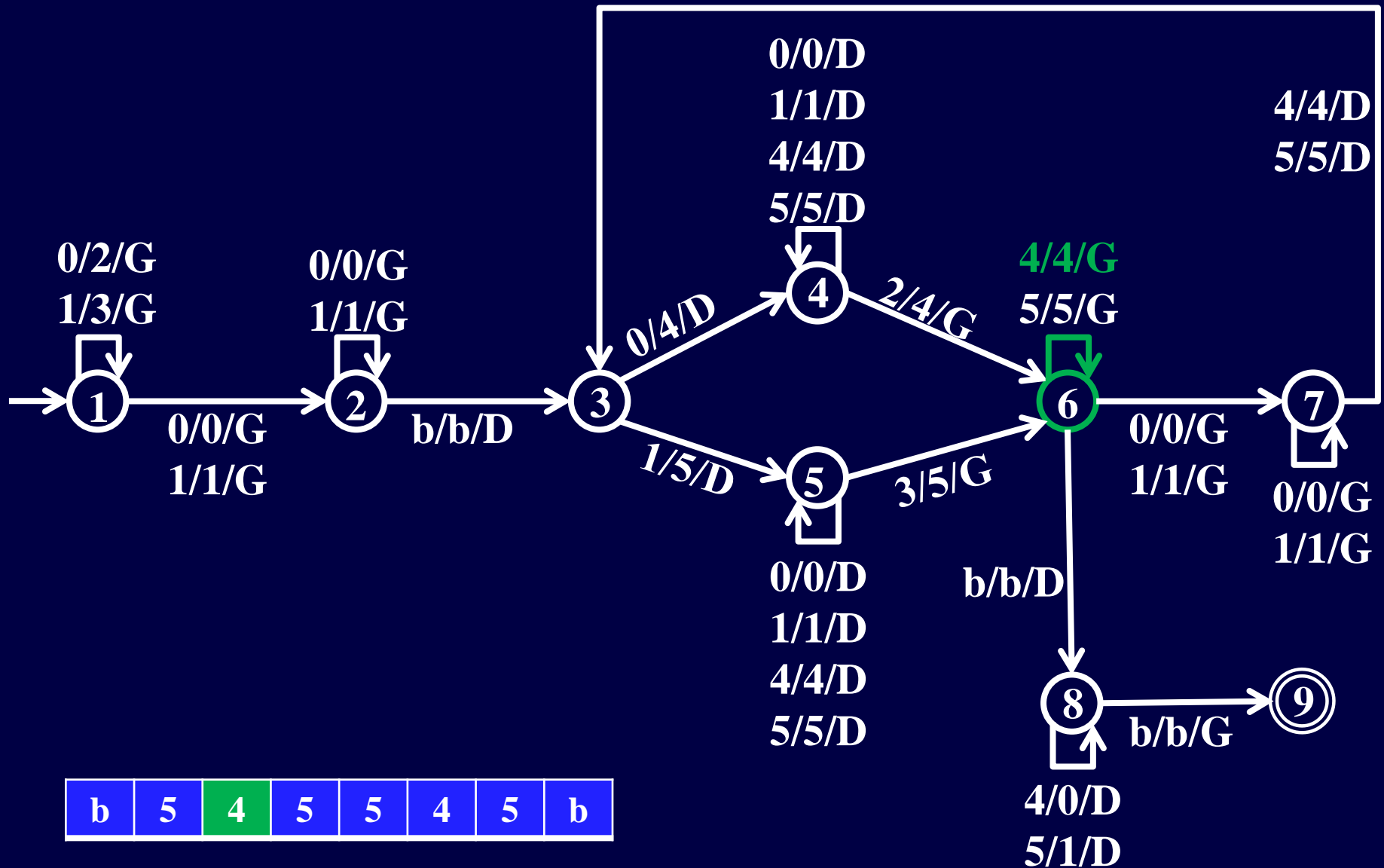
Un exemple de machine non-déterministe



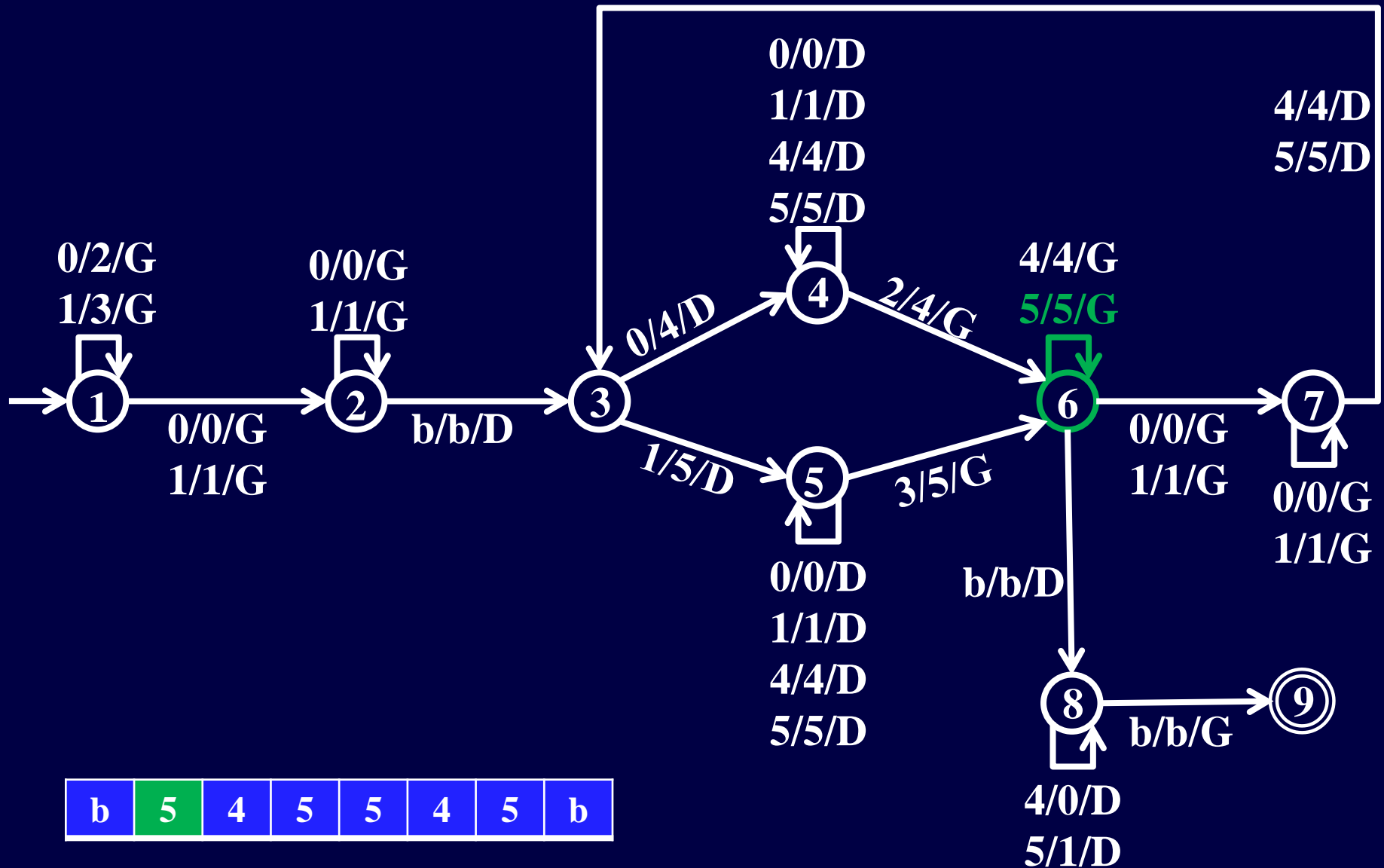
Un exemple de machine non-déterministe



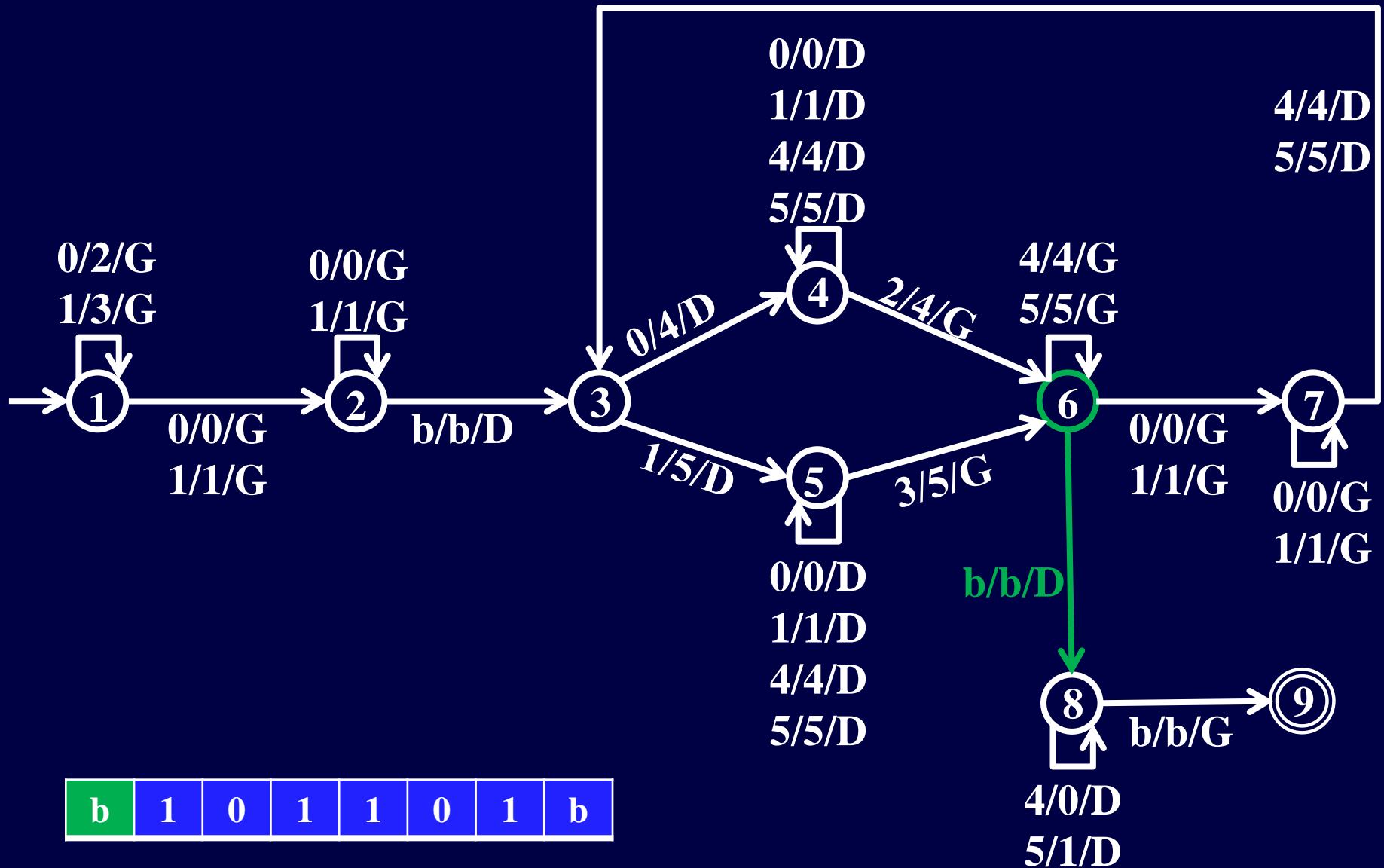
Un exemple de machine non-déterministe



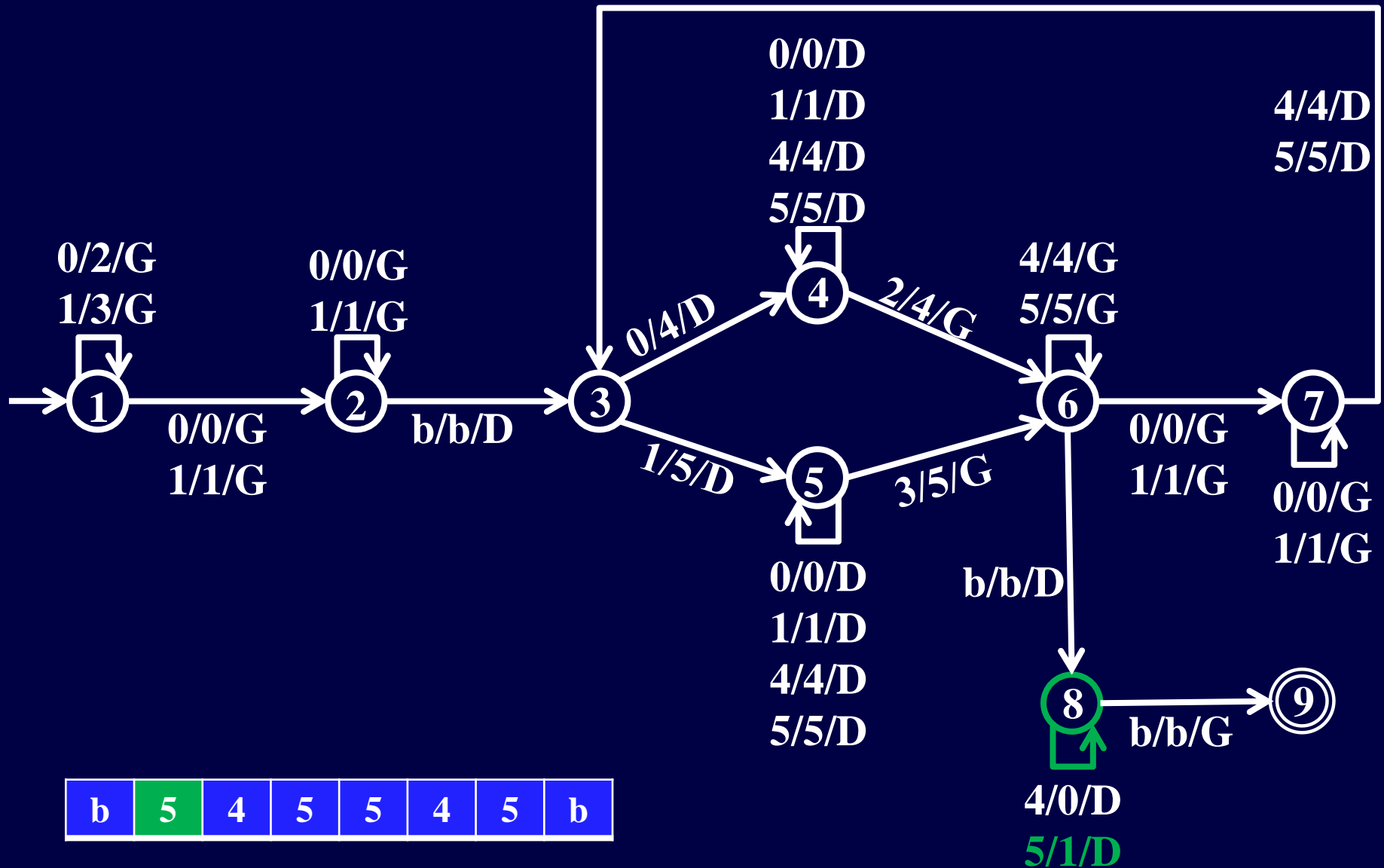
Un exemple de machine non-déterministe



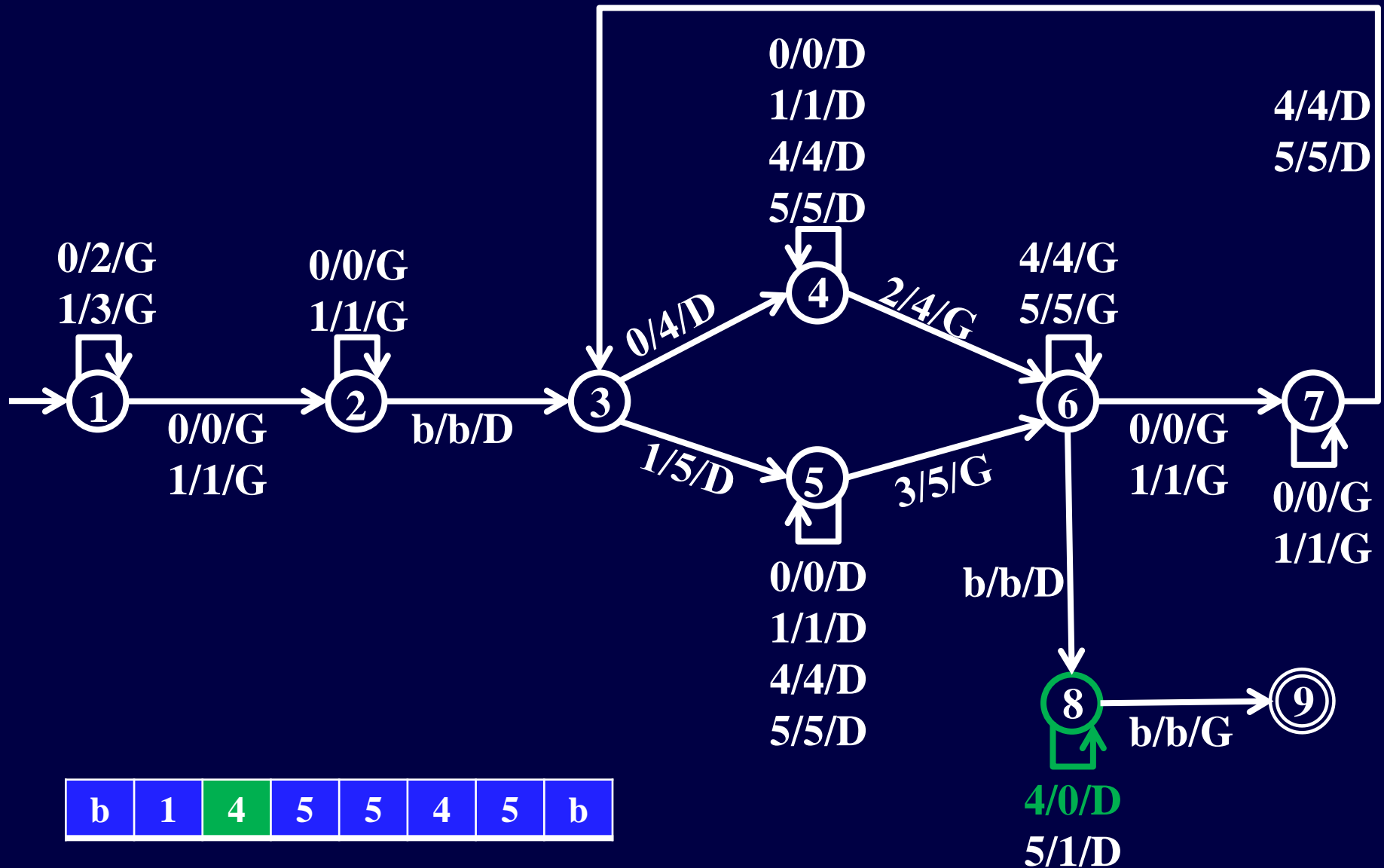
Un exemple de machine non-déterministe



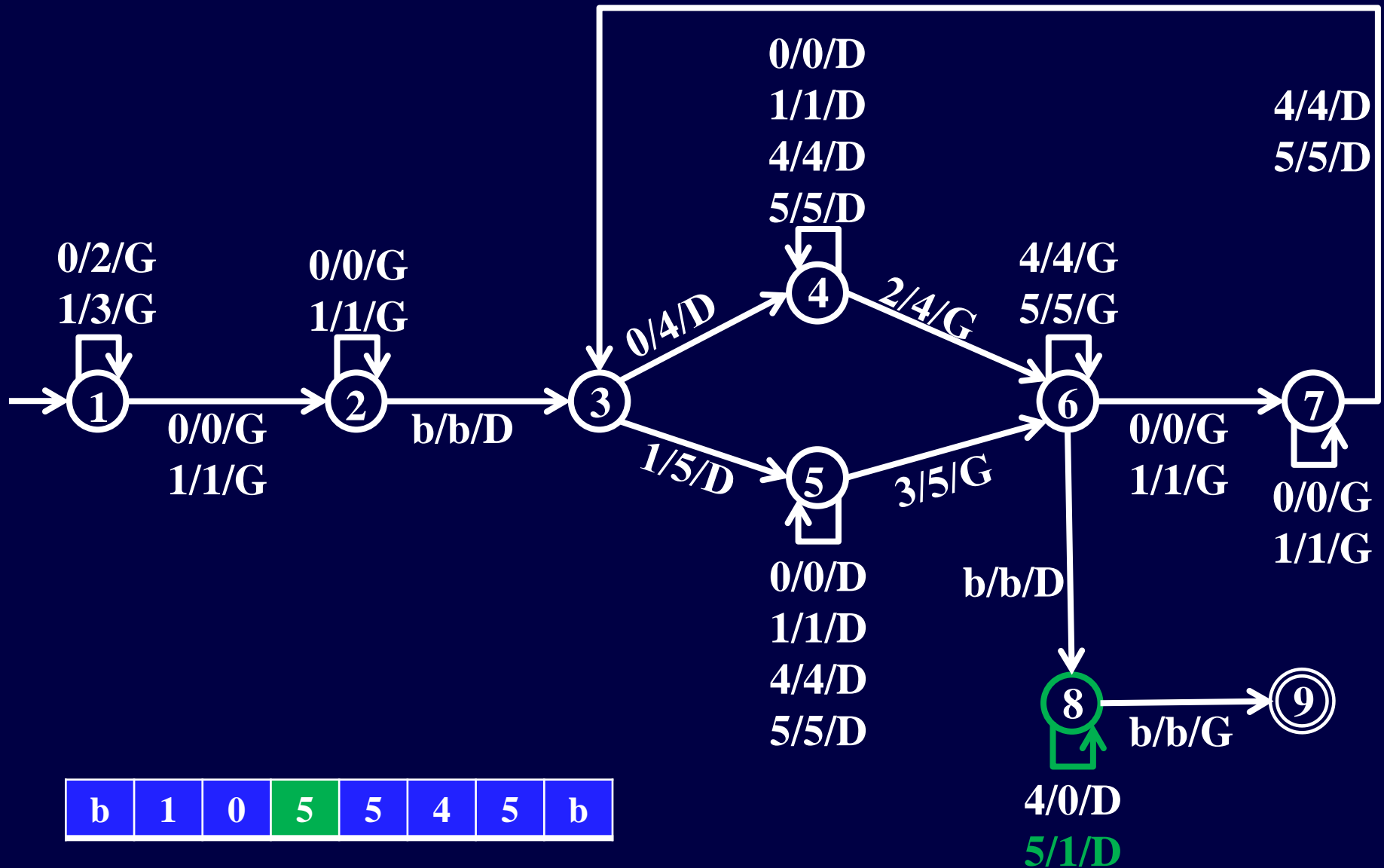
Un exemple de machine non-déterministe



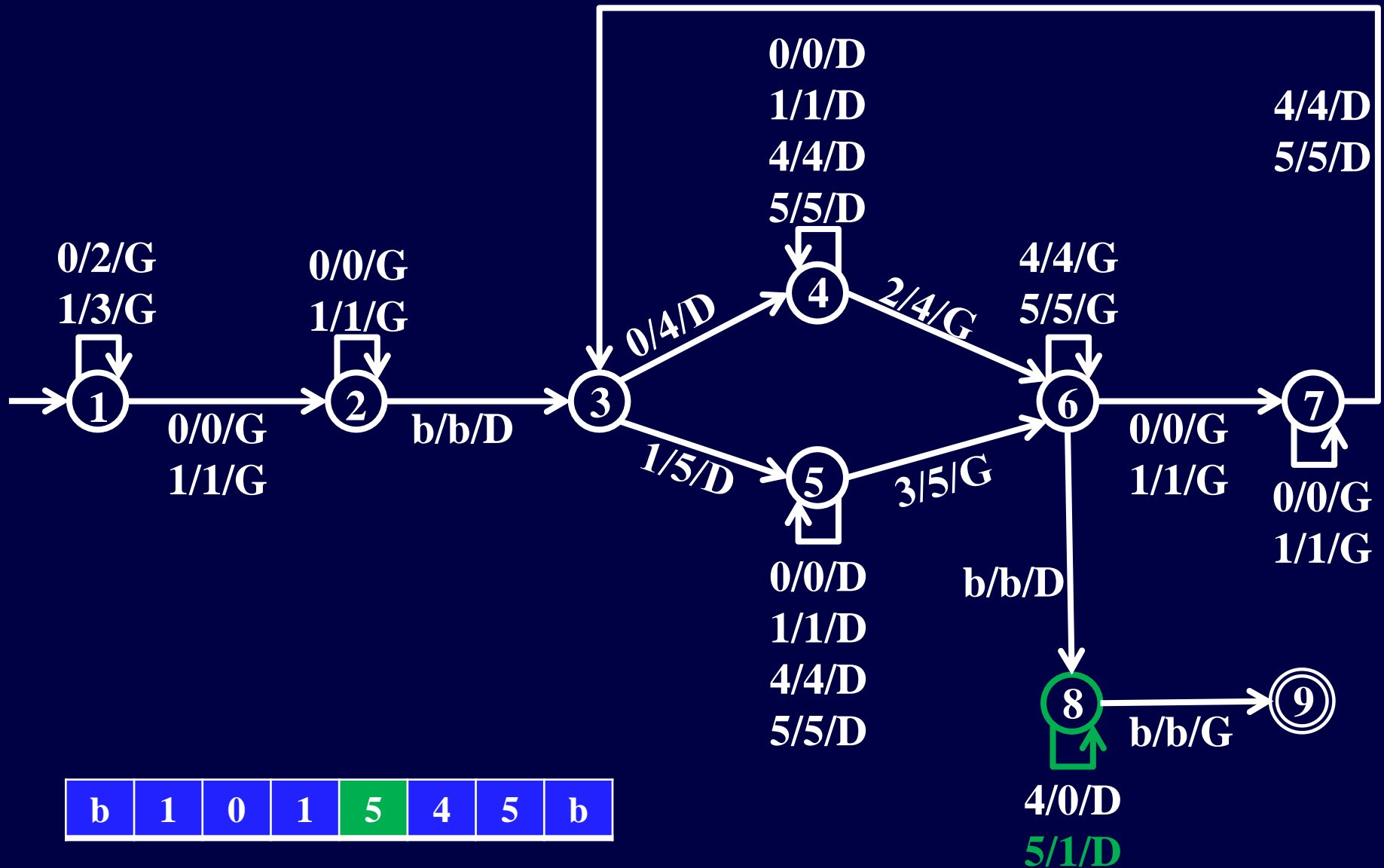
Un exemple de machine non-déterministe



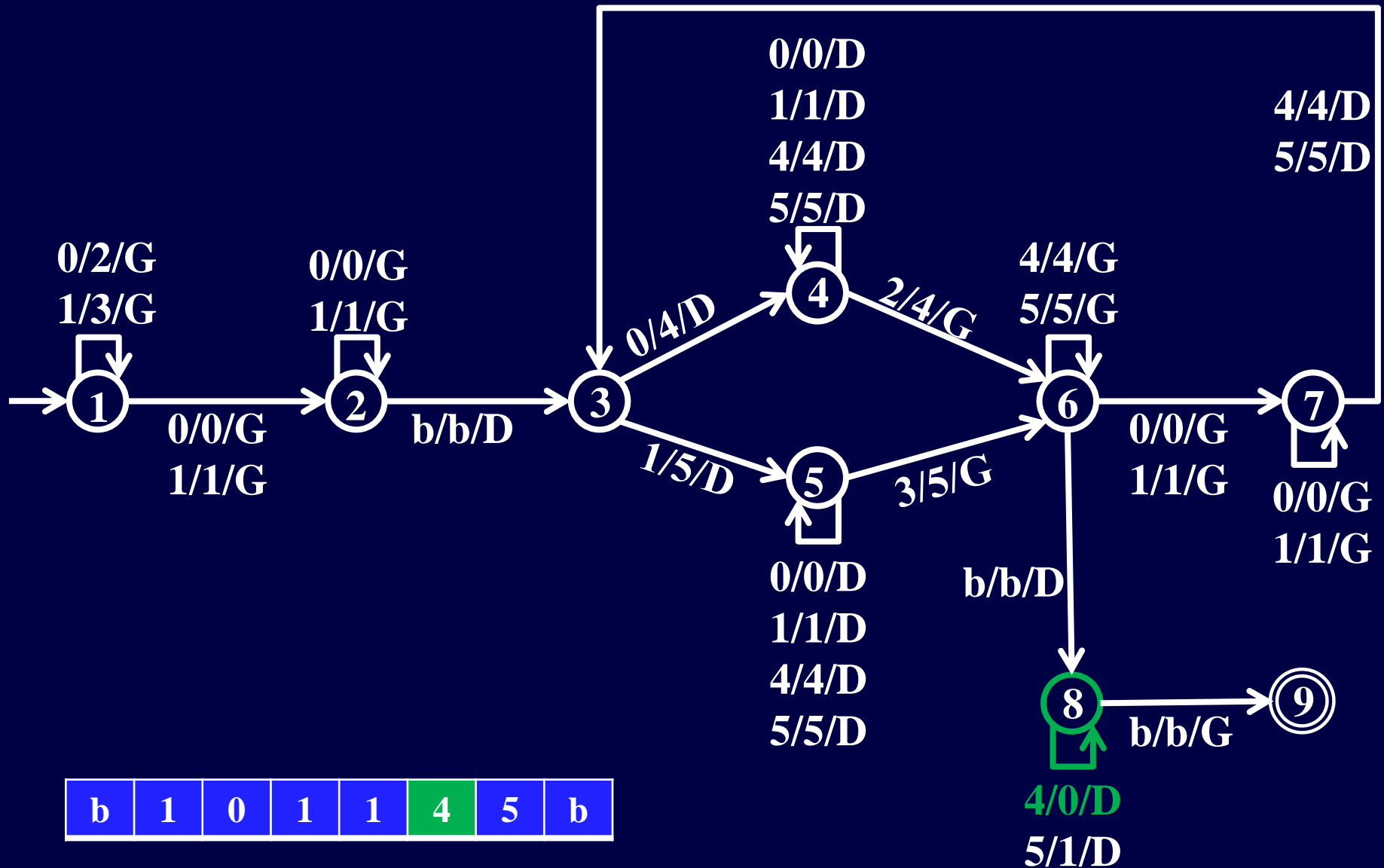
Un exemple de machine non-déterministe



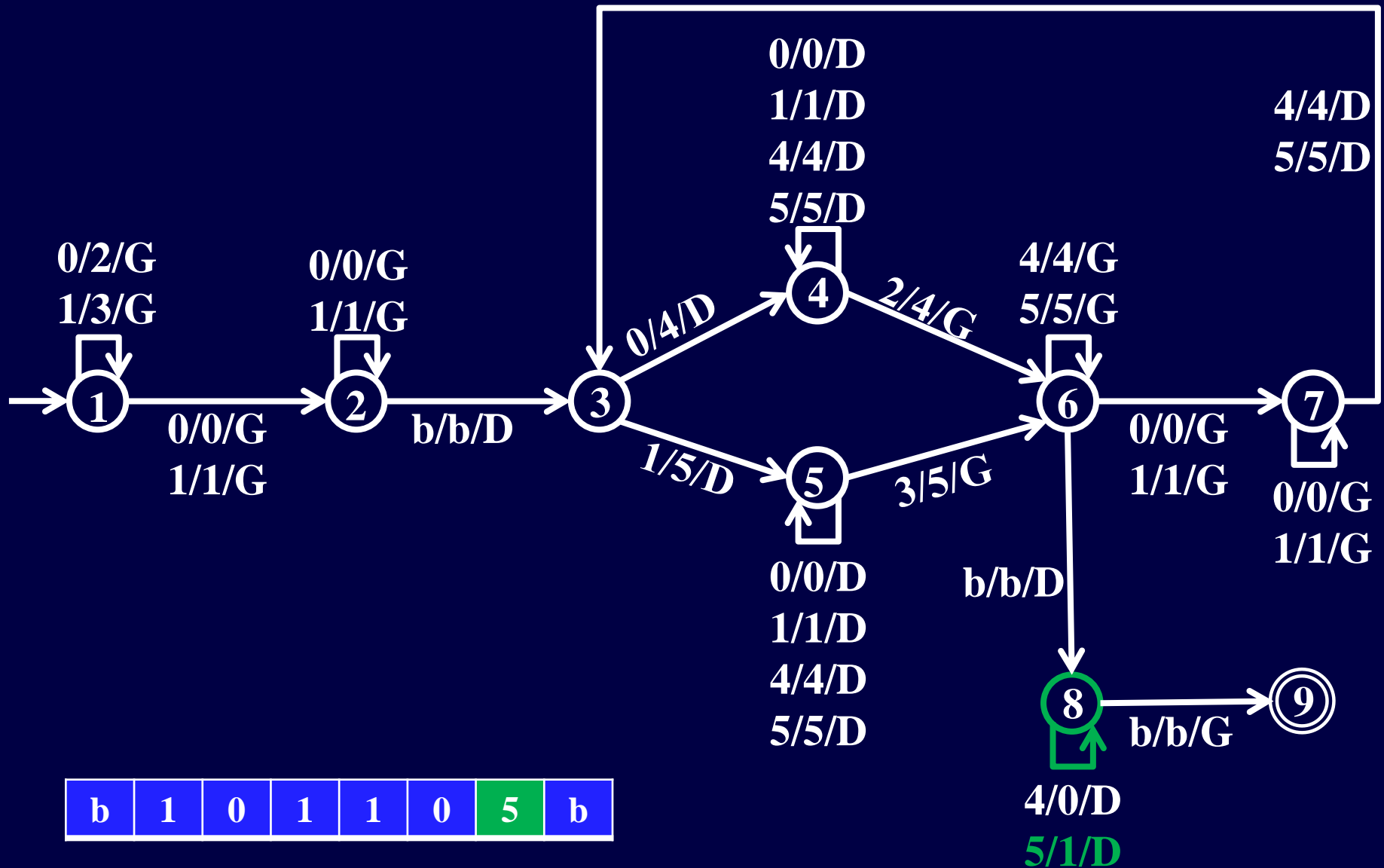
Un exemple de machine non-déterministe



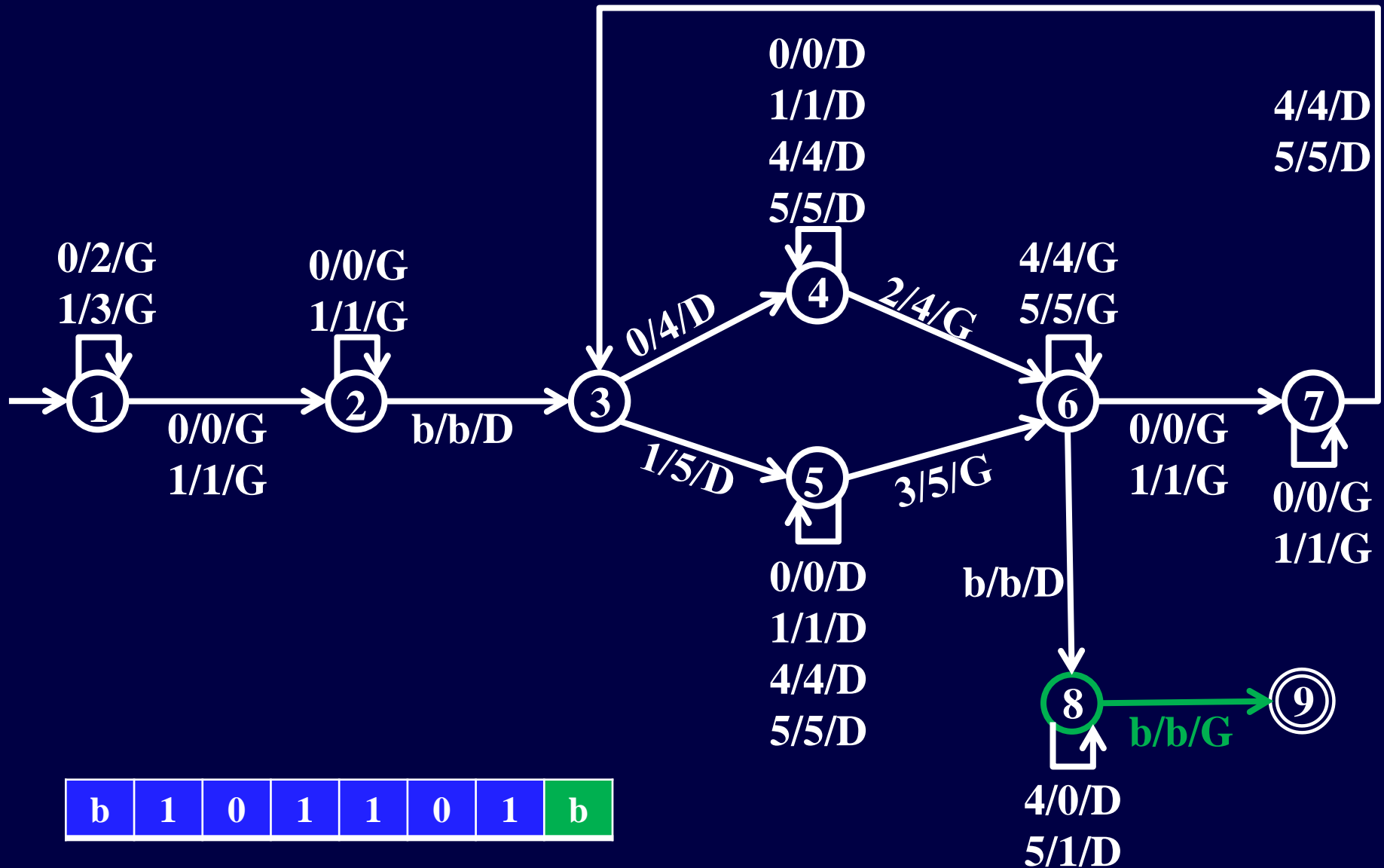
Un exemple de machine non-déterministe



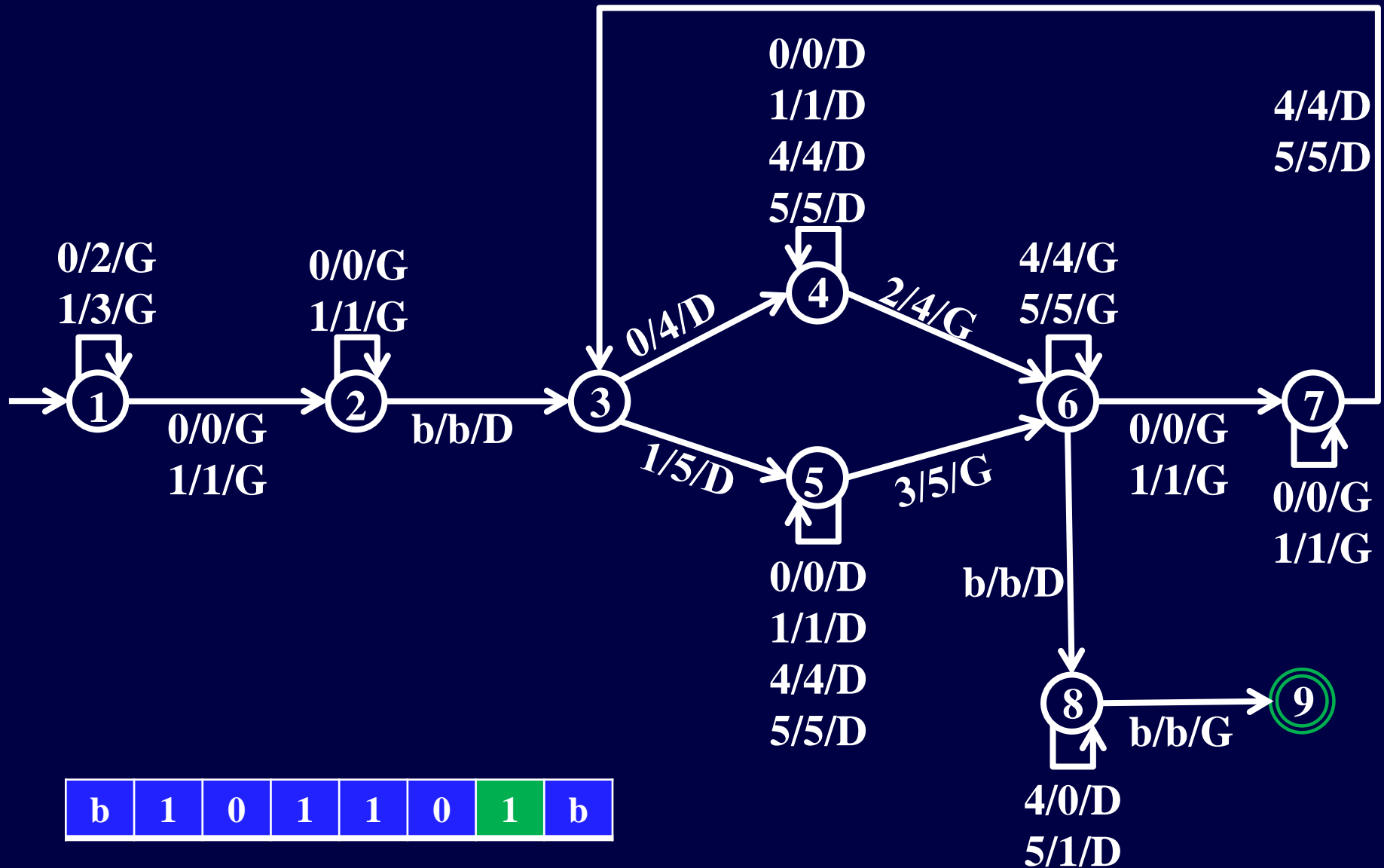
Un exemple de machine non-déterministe



Un exemple de machine non-déterministe



Un exemple de machine non-déterministe



Terminé par acceptation

Le déterminisme

Théorème :

Si le langage L est accepté par une machine de Turing non-déterministe T à une bande, alors il existe une machine de Turing déterministe T' à 3 bandes pour accepter L .

Preuve

- Il faut simuler le non-déterminisme.
- générer dans un ordre fixé à l'avance tous les choix possibles, jusqu'à ce qu'on arrive à l'état d'acceptation.

Parcours de l'arborescence qui comporte tous les choix.

- parcours en profondeur ne peut pas être retenue, à cause des choix qui font «boucler» la machine.
- un parcours en largeur peut être utilisé.

La machine à 3 bandes

bande 1

#	#	d	o	n	é	e	s	x	x	x	#	#
---	---	---	---	---	---	---	---	---	---	---	---	---

bande 2

#	#	#	#	#	#	#	#	#	#	0	#	#
---	---	---	---	---	---	---	---	---	---	---	---	---

bande 3

#	#	s	a	u	v	e	g	a	r	d	e	#
---	---	---	---	---	---	---	---	---	---	---	---	---

Première essai : terminaison au bout d'une seule transition ...

- l'impossibilité de revenir en arrière.
- La simulation d'un parcours en largeur dans ces conditions devient une suite de parcours partiels, de la racine vers les différents sommets, dans l'ordre qui correspond à l'ordre d'un parcours en largeur.
- Ainsi, si on note les suites de choix retenus, on obtient des mots rangés selon l'ordre lexicographique.

La simulation se déroule comme suit : Supposons que la machine T ait au maximum r choix. Le fonctionnement de la machine T' est comme suit :

- On **génère** un mot de $\{0,1,2,\dots,r-1\}^*$ sur la bande 2.
- On **recopie** les données de la bande 3 sur la bande 1.
- On **exécute** le programme de T en travaillant sur la bande 1, en prenant comme choix celui indiqué par le bit correspondant sur la bande 2 (sur laquelle on avance). Si on arrive à un état d'acceptation on arrête, sinon on génère une nouvelle suite sur la bande 2, on efface la bande 1 et on recommence en ii).

Le théorème de l'accélération

Théorème (de l'accélération - speed up) :

Si L est accepté par une machine de Turing à k bandes ($k > 1$) en temps $T(n)$ (avec $\lim T(n)/n = \infty$), alors pour tout nombre réel $c > 0$ il existe une machine de Turing à k bandes acceptant L en temps $c T(n)$.

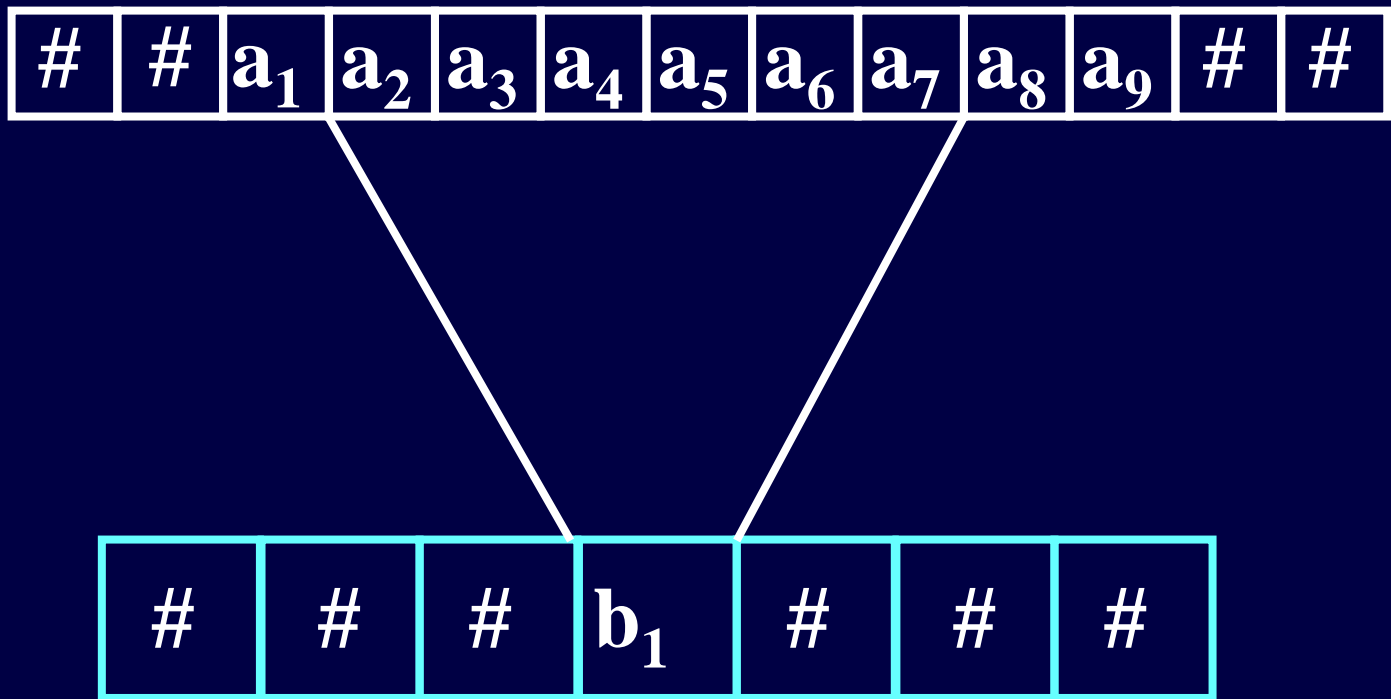
Ce théorème « justifie »

l'utilisation de la notation O !

Preuve :

La preuve se fait par construction d'une machine de Turing qui simule la machine originale. La machine M' est construite comme suit :

I) On encode les données, de façon que m cases de la machine originale soient codées par une seule lettre - donc une seule case, dans la nouvelle machine. Cette opération se fait en temps linéaire ($2n$ par exemple). Notons que le temps linéaire est possible, car il s'agit d'une machine à plusieurs bandes !



II) Le codage obtenu permet d'effectuer plusieurs transitions à la fois, car une lecture donne l'information de m cases de la machine simulée et donc on peut effectuer tout le travail imposé par ces m lectures. Par contre, il est impossible d'éliminer ainsi les effets de bords, c.a.d. le cas où la machine simulée effectue les aller - retours entre deux cases qui impliquent des allers - retours entre deux «blocs» de m cases.

Pour éliminer ce problème, nous allons effectuer la lecture de trois cases voisines (ce qui correspond à la lecture de $3m$ cases de la machine simulée), et ainsi, nous pouvons assurer que chaque «phase de base» correspond à au moins m transitions de la machine simulée.

Pendant la simulation, la machine M' aura les «phases de base» suivantes :

- Lecture et déplacement à gauche
- Lecture et déplacement à droite
- Déplacement à droite
- Lecture et déplacement à gauche

On se trouve sur la case de départ et on connaît le contenu des deux cases voisines.

On connaît le contenu de $3m$ cases de la machine simulée, ce qui permet d'effectuer d'un coup toutes les transitions que la machine originale effectue, jusqu'au moment où elle quitte ces $3m$ cases.

Selon que l'on quitte les $3m$ cases vers la droite ou vers la gauche on effectue les opérations suivantes.

- Écriture et déplacement à gauche (resp. droite)
- Écriture et déplacement à droite (resp. gauche)
- Déplacement à droite (resp. gauche)
- Écriture et déplacement à droite (resp. gauche)

Une phase de base comprend 8 pas de calcul de la nouvelle machine et correspond à plus de m pas de calculs de la machine originale.

La complexité de la nouvelle machine sera

$$T'(n) < 2n + 8 T(n) / m$$

Il faut choisir m de façon à obtenir $T'(n) \leq cT(n)$.

Soit $m > 16/c$. Alors on a

$$8T(n) / m < c/2 T(n) < c T(n) - 2n$$

(pour n assez grand).

Donc : $T'(n) < c T(n)$

Théorème :

Si le langage L est accepté par une machine de Turing à k ($k > 1$) bandes en temps $T(n)$,
 \lim $T(n)/n = \infty$, alors L est accepté par une machine de Turing à une bande M' en temps au plus $T^2(n)$.