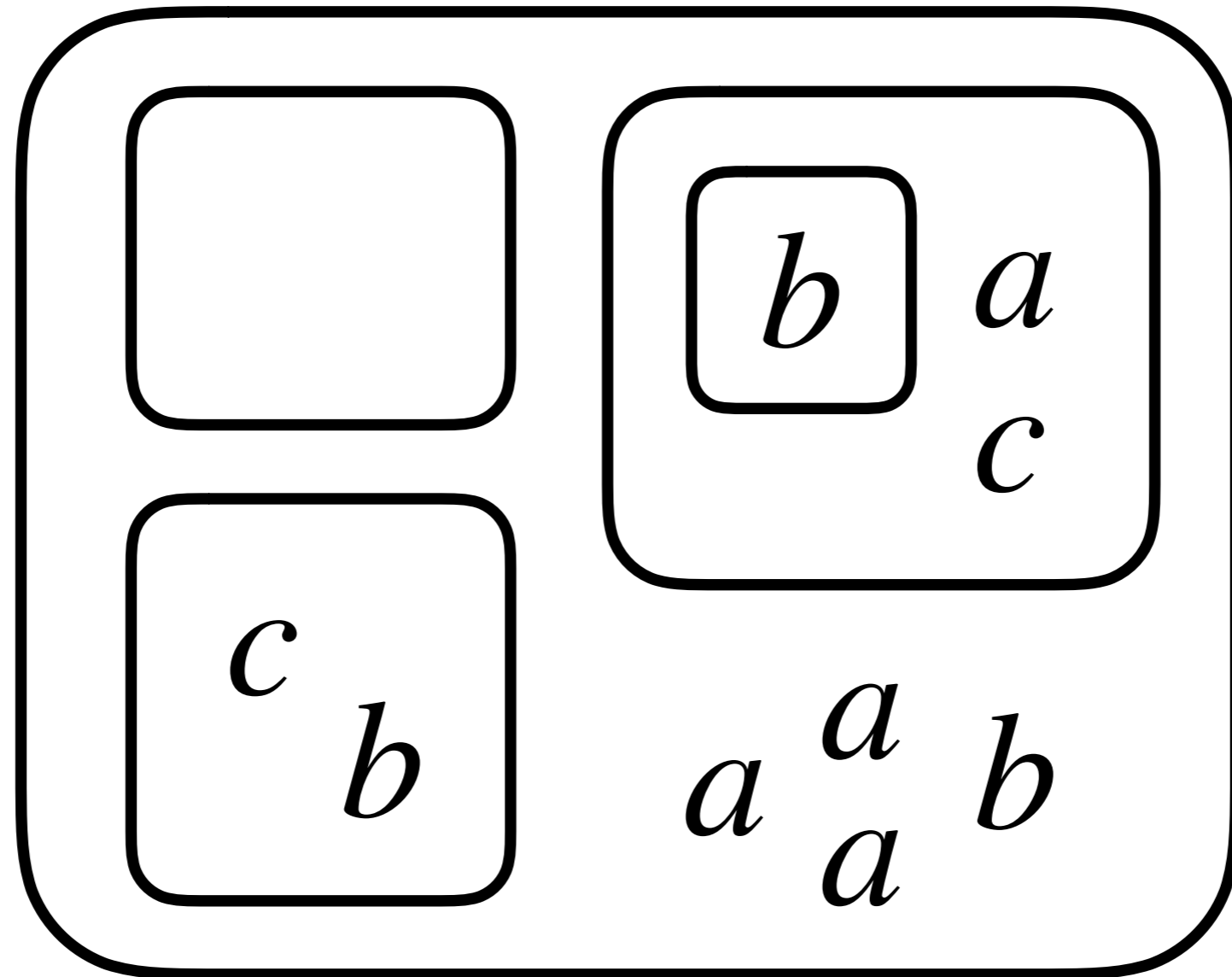


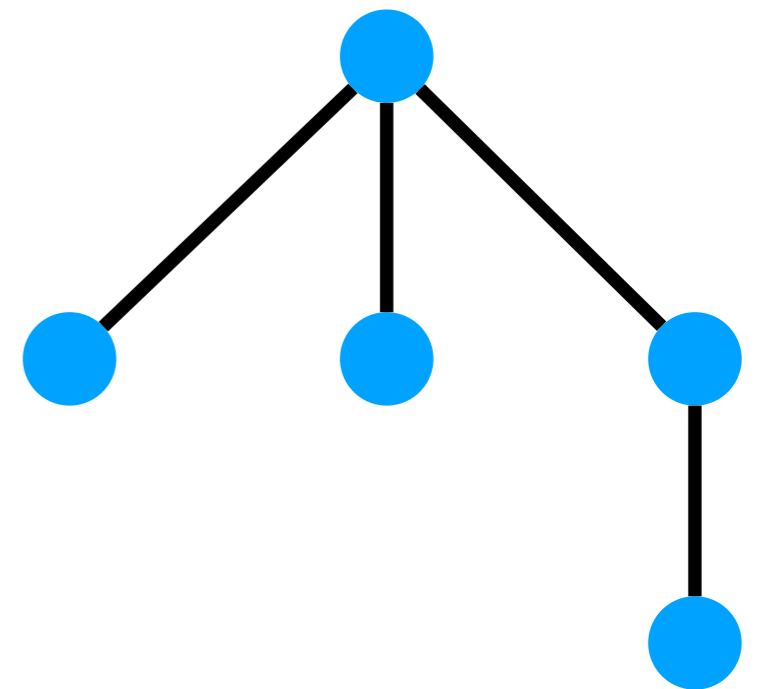
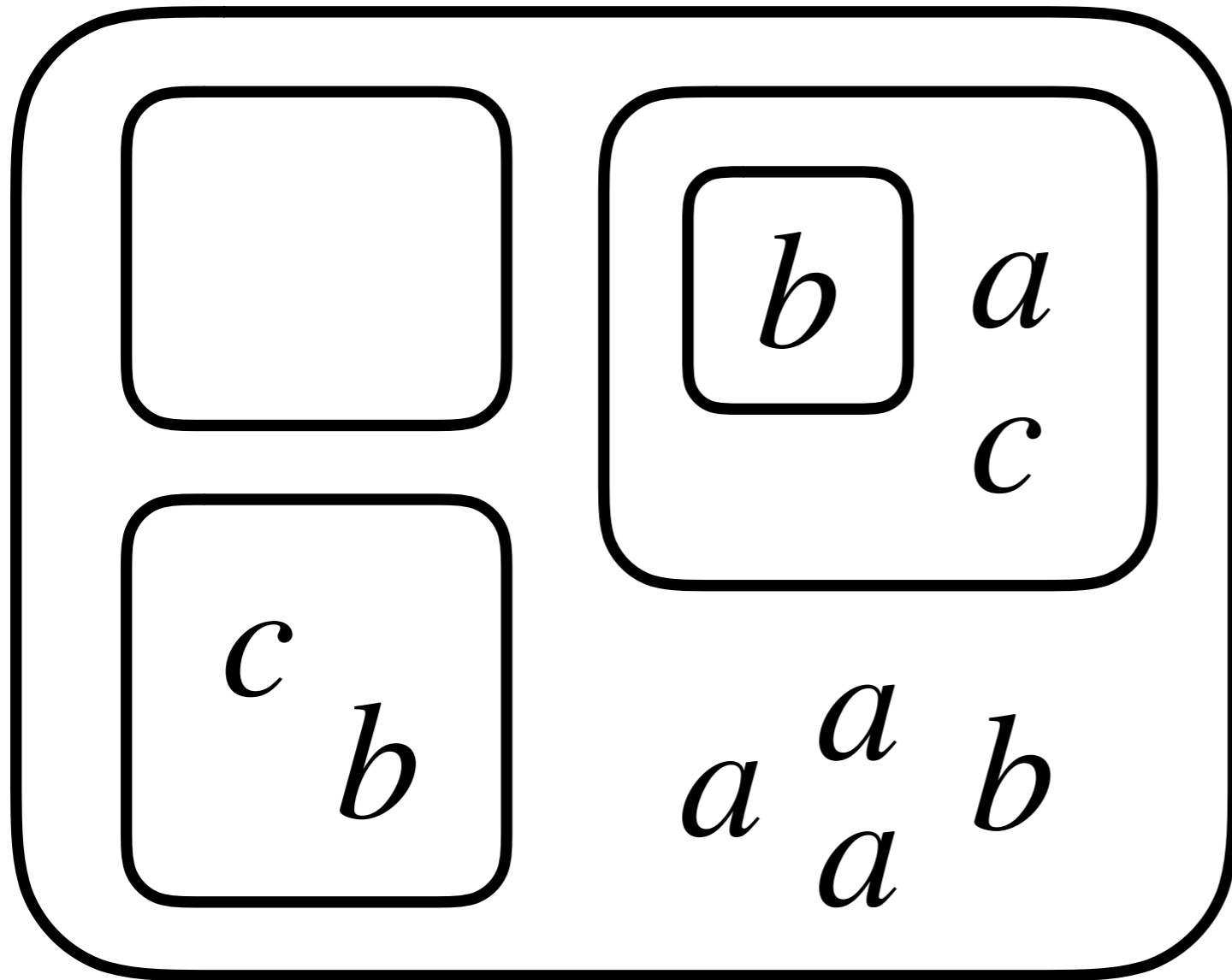
Systemes à membranes (et calcul naturel)

Antonio E. Porreca
<https://aeporreca.org>

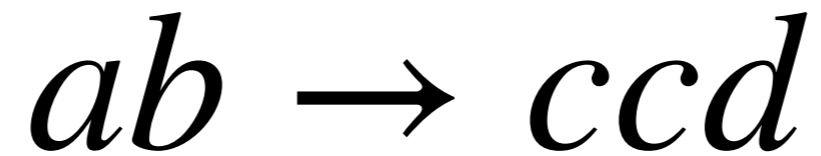
Systemes à membranes (ou P systems)



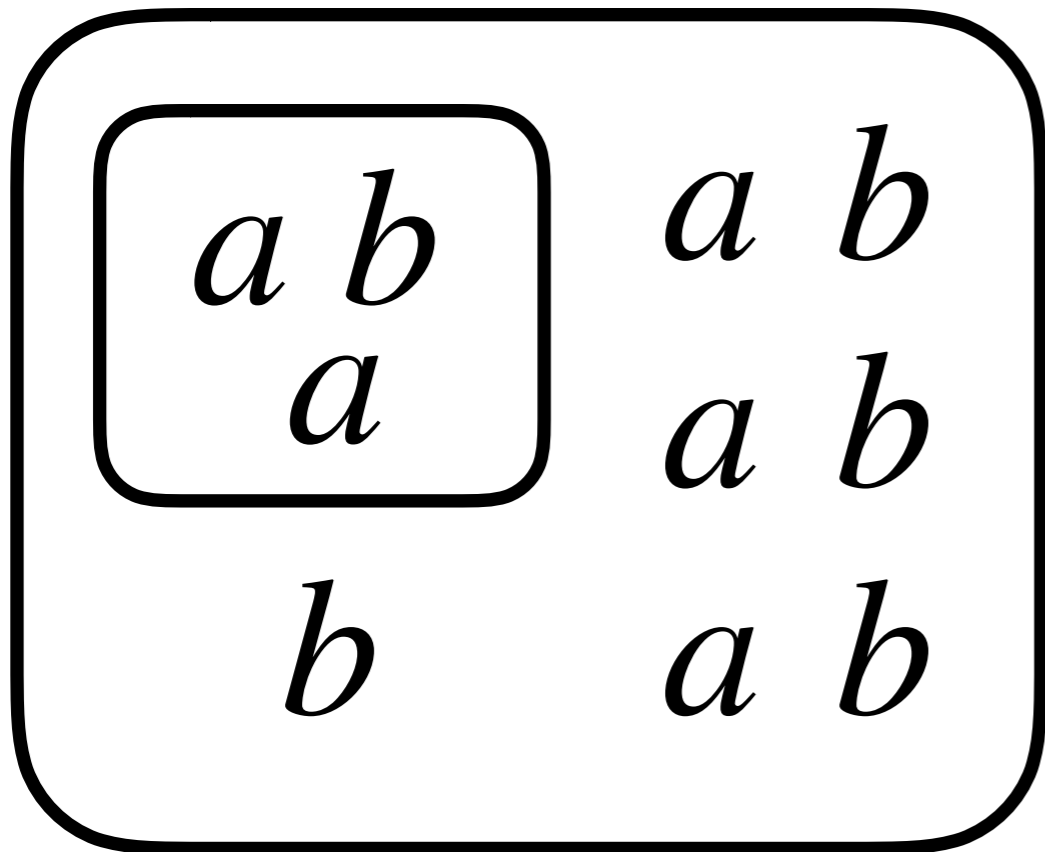
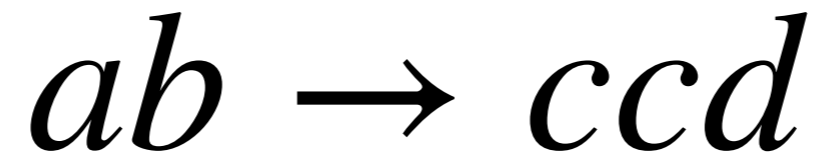
Systemes à membranes (ou P systems)



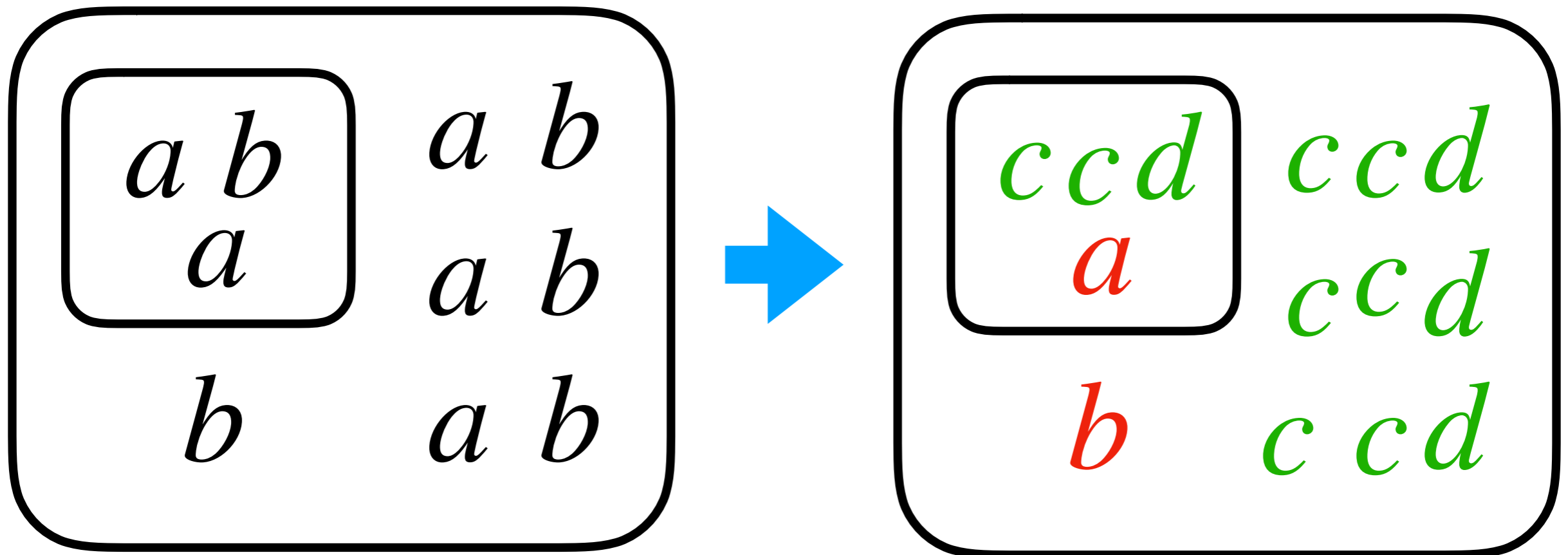
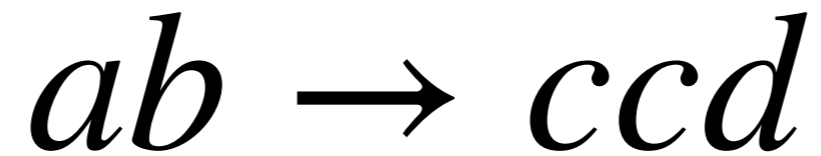
Réactions chimiques



Réactions chimiques



Réactions chimiques

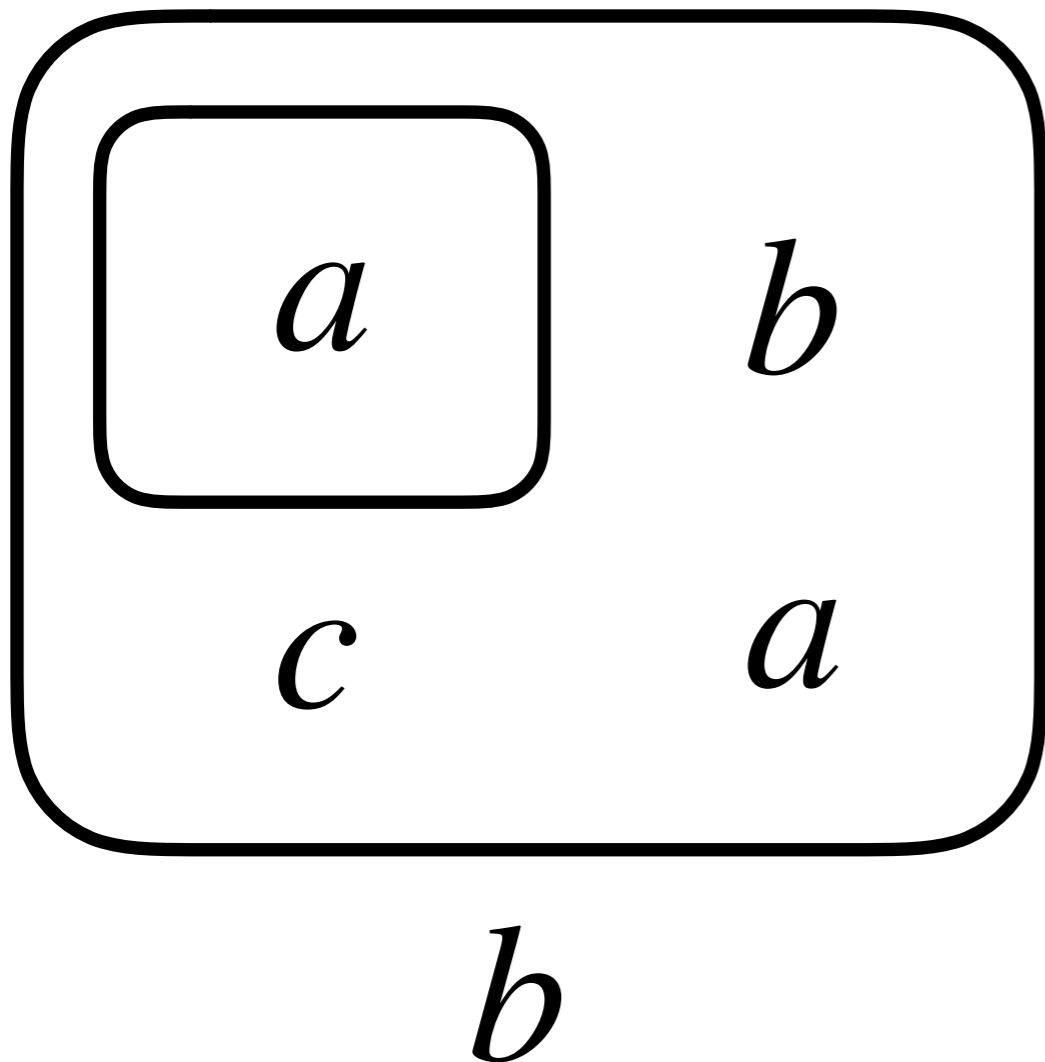


Communication

$$a[b] \rightarrow c[d]$$

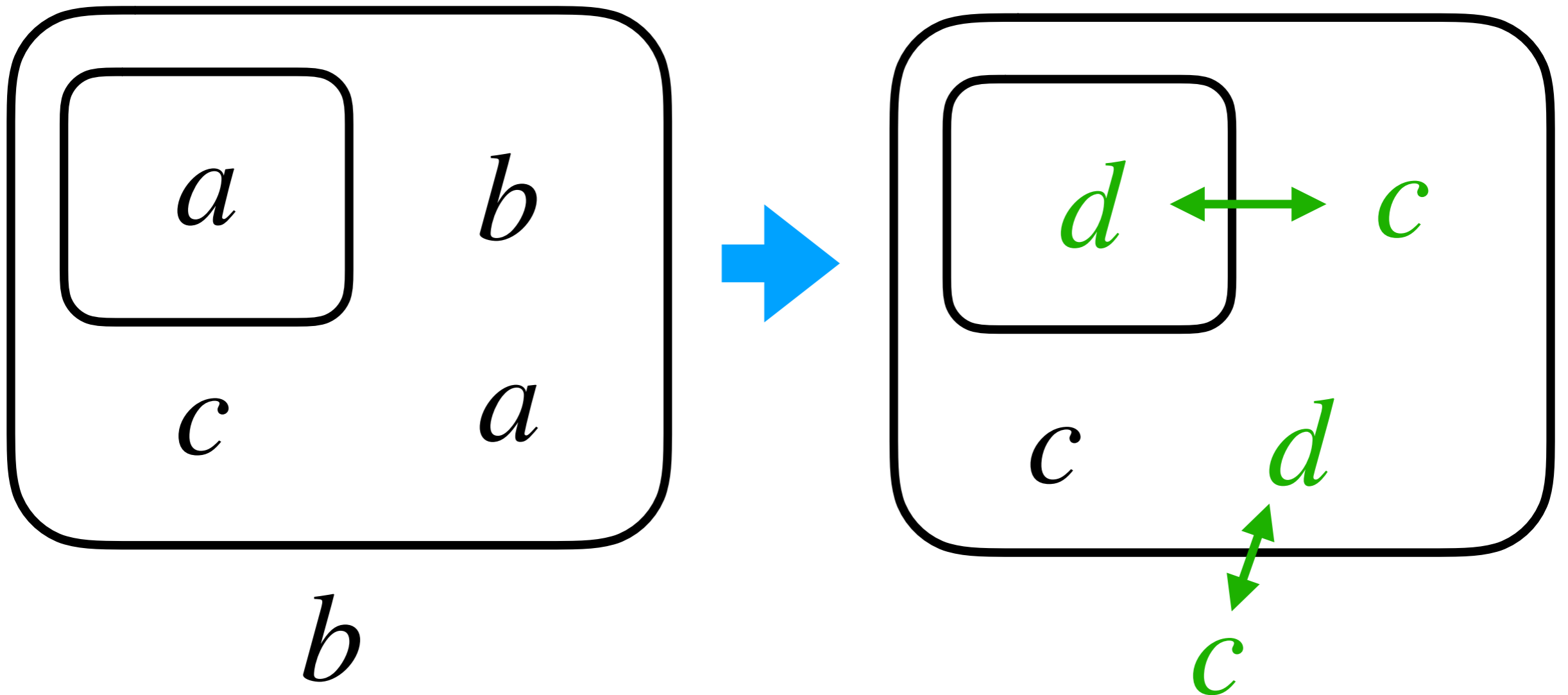
Communication

$$a[b] \rightarrow c[d]$$



Communication

$$a[b] \rightarrow c[d]$$

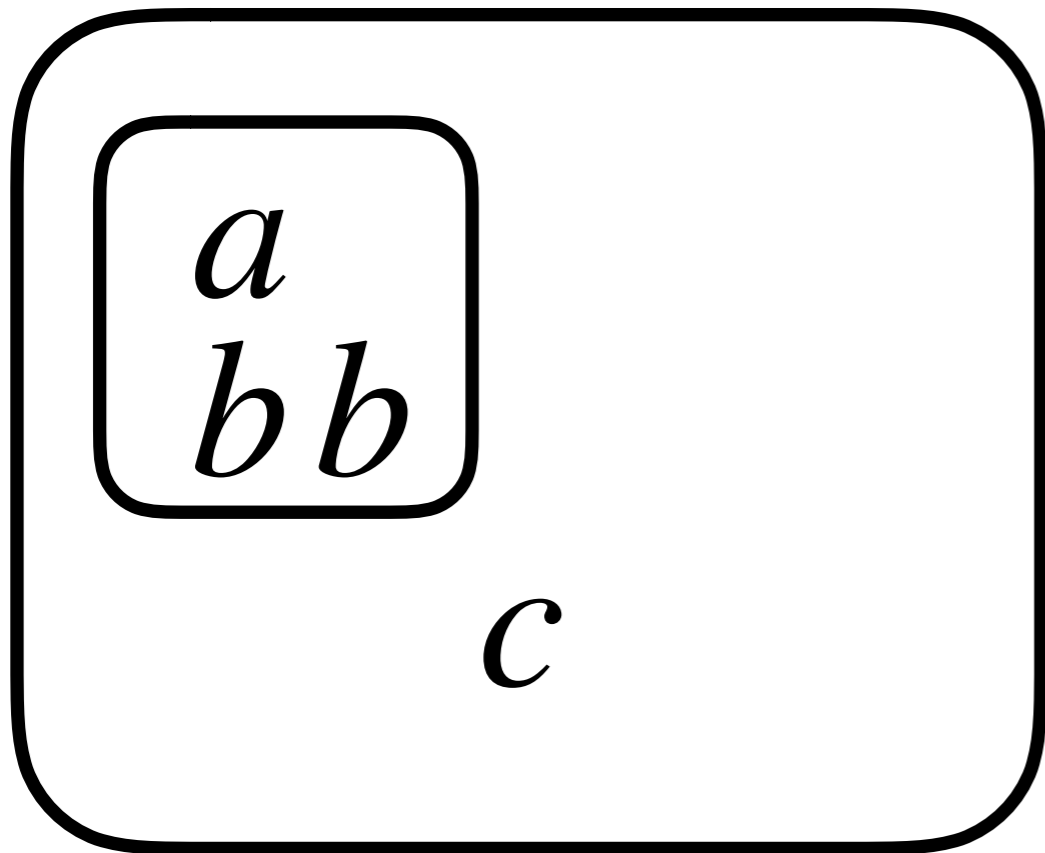


Division de membranes

$$[a] \rightarrow [bc] [d]$$

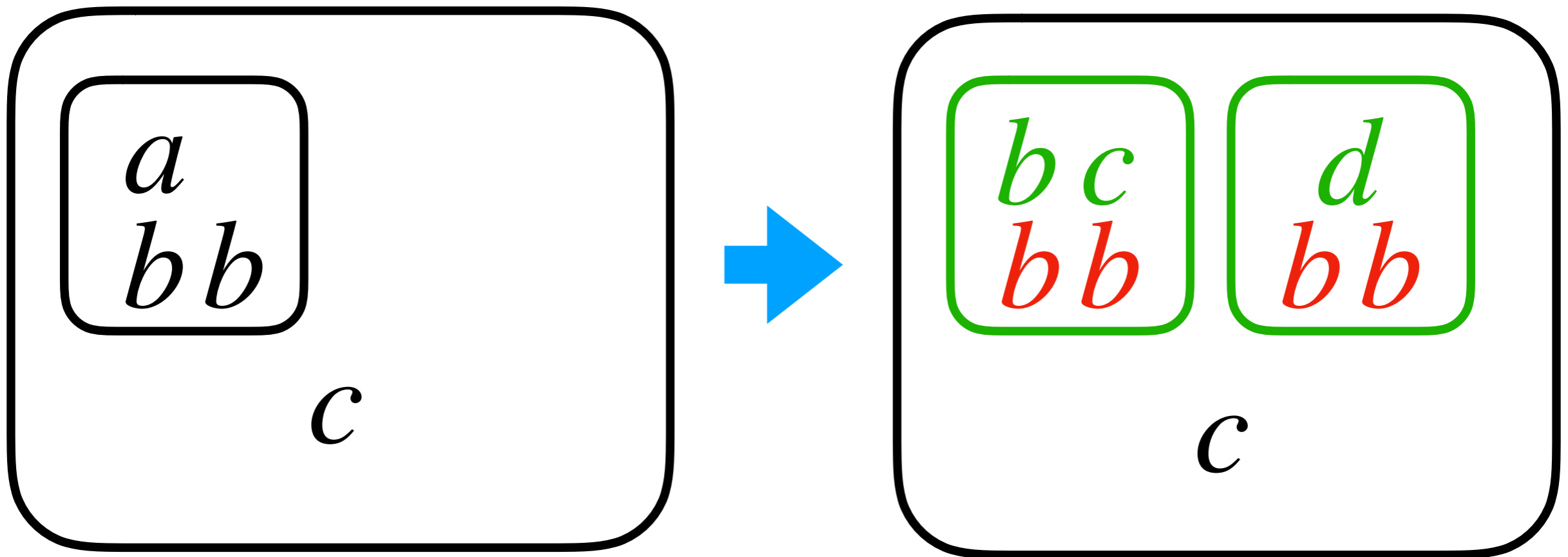
Division de membranes

$$[a] \rightarrow [bc] [d]$$

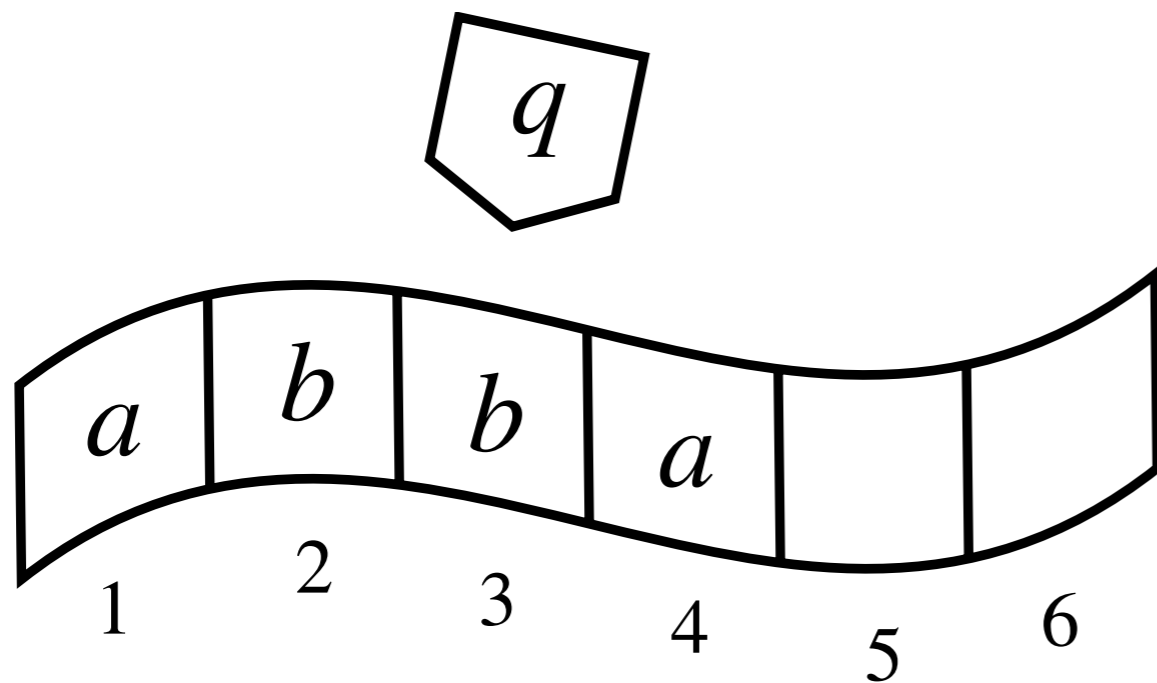


Division de membranes

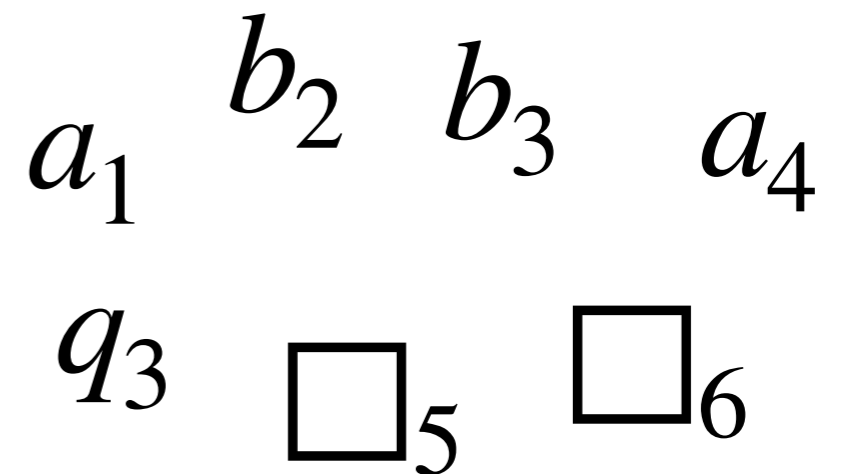
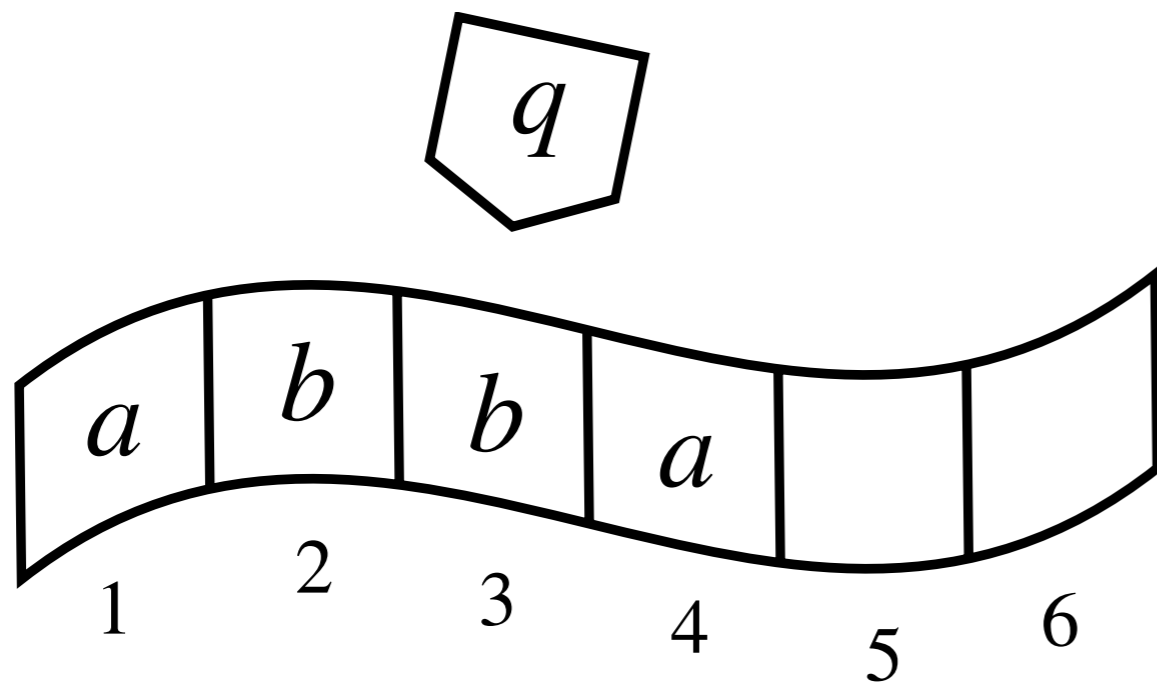
$$[a] \rightarrow [bc] [d]$$



Codage des configurations des machines de Turing



Codage des configurations des machines de Turing



Simulation des transitions

$$\delta(q, b) = (r, a, + 1)$$

Simulation des transitions

$$\delta(q, b) = (r, a, + 1)$$

$$q_1 b_1 \rightarrow r_2 a_1$$

Simulation des transitions

$$\delta(q, b) = (r, a, + 1)$$

$$q_1 b_1 \rightarrow r_2 a_1$$

$$q_2 b_2 \rightarrow r_3 a_2$$

⋮

$$q_i b_i \rightarrow r_{i+1} a_i$$

Simulation des transitions

a_1 b_2 q_3 b_3 a_4 \square_5 \square_6

$$\delta(q, b) = (r, a, + 1)$$

Simulation des transitions

a_1 b_2 q_3 b_3 a_4 \square_5 \square_6

$$\delta(q, b) = (r, a, + 1)$$



a_1 b_2 a_3 r_4 a_4 \square_5 \square_6

Résultat du calcul

a_1 b_2 b_3 a_4 \square_5 \square_6 q_3

si q est un
état acceptant

Résultat du calcul

a_1 b_2 b_3 a_4 \square_5 \square_6
 q_3

si q est un
état acceptant

$q_i \rightarrow \text{yes}$ 

a_1 b_2 b_3 a_4 \square_5 \square_6
 yes

Résultat du calcul

a_1 b_2 b_3 a_4 \square_5 \square_6 q_3


si q est un
état de refus

Résultat du calcul

a_1 b_2 b_3 a_4 \square_5 \square_6

q_3

si q est un
état de refus

$q_i \rightarrow$ no 

no

a_1 b_2 b_3 a_4 \square_5 \square_6

Théorème : le systèmes à membranes simulent les machine de Turing d'une façon efficace

Simulation des transitions non déterministes

$$\delta(q, b) = \begin{cases} (r, a, + 1) \\ (s, b, - 1) \end{cases}$$

Simulation des transitions non déterministes

$$\delta(q, b) = \begin{cases} (r, a, +1) \\ (s, b, -1) \end{cases}$$

$$[q_2 \ b_2] \rightarrow [r_3 \ a_2] [s_1 \ b_2]$$

Simulation des transitions non déterministes

$$\delta(q, b) = \begin{cases} (r, a, + 1) \\ (s, b, - 1) \end{cases}$$

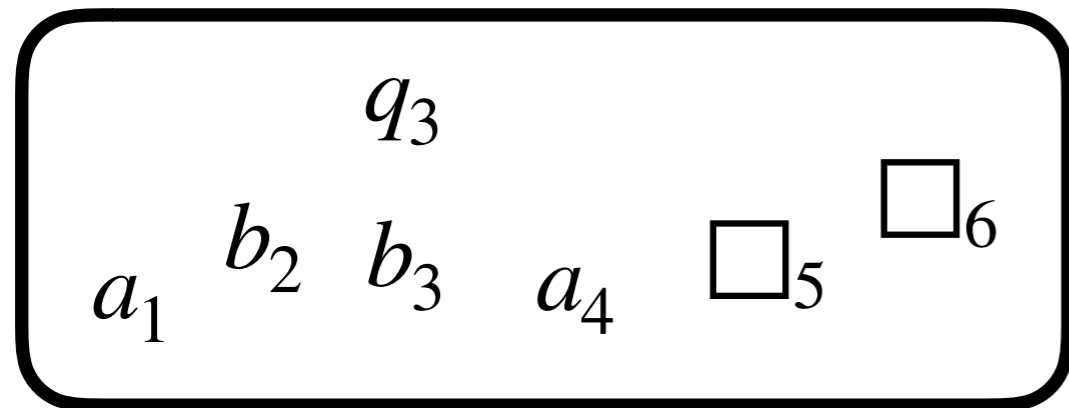
$$[q_2 \ b_2] \rightarrow [r_3 \ a_2] [s_1 \ b_2]$$

$$[q_3 \ b_3] \rightarrow [r_4 \ a_3] [s_2 \ b_3]$$

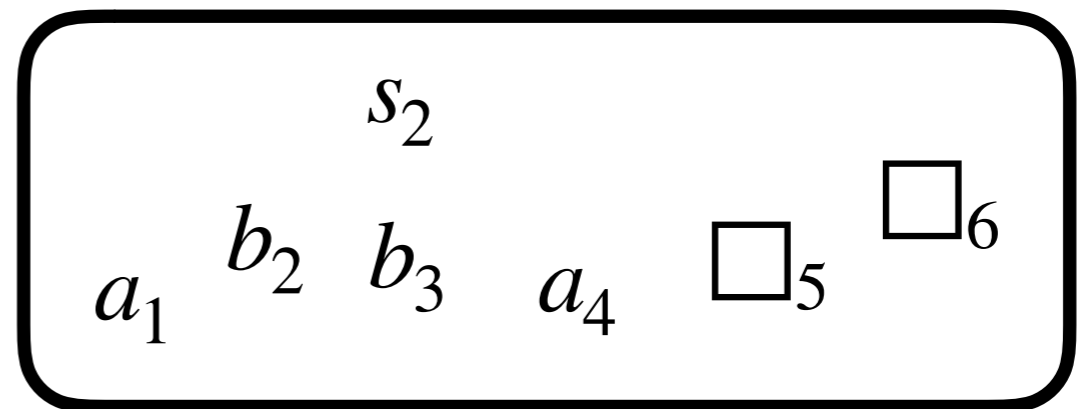
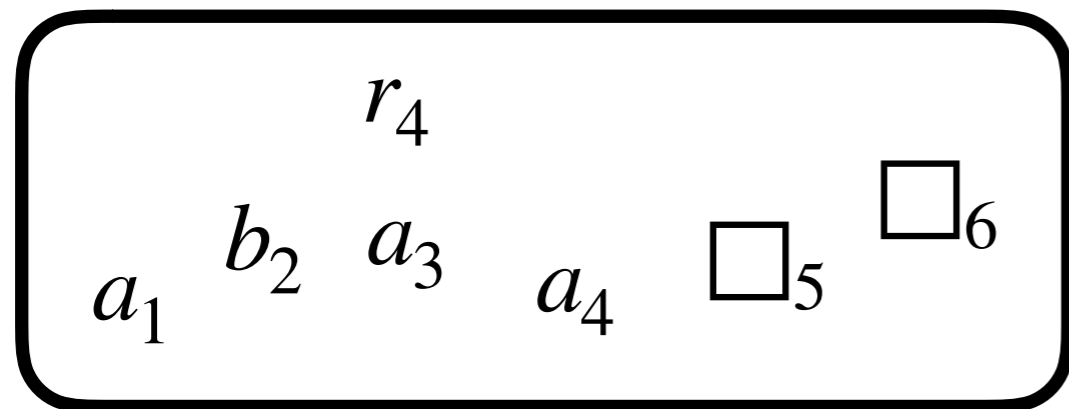
⋮

$$[q_i \ b_i] \rightarrow [r_{i+1} \ a_i] [s_{i-1} \ b_i]$$

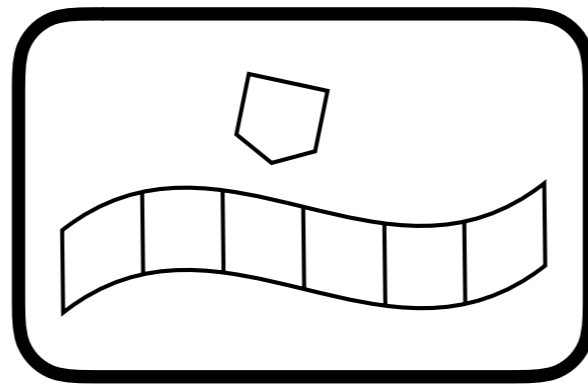
Simulation des transitions non déterministes



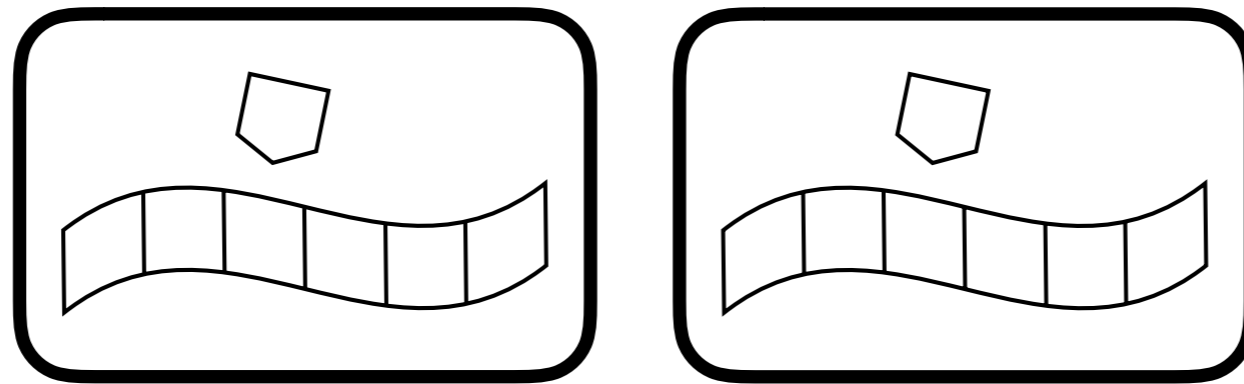
$$\delta(q, b) = \begin{cases} (r, a, +1) \\ (s, b, -1) \end{cases}$$



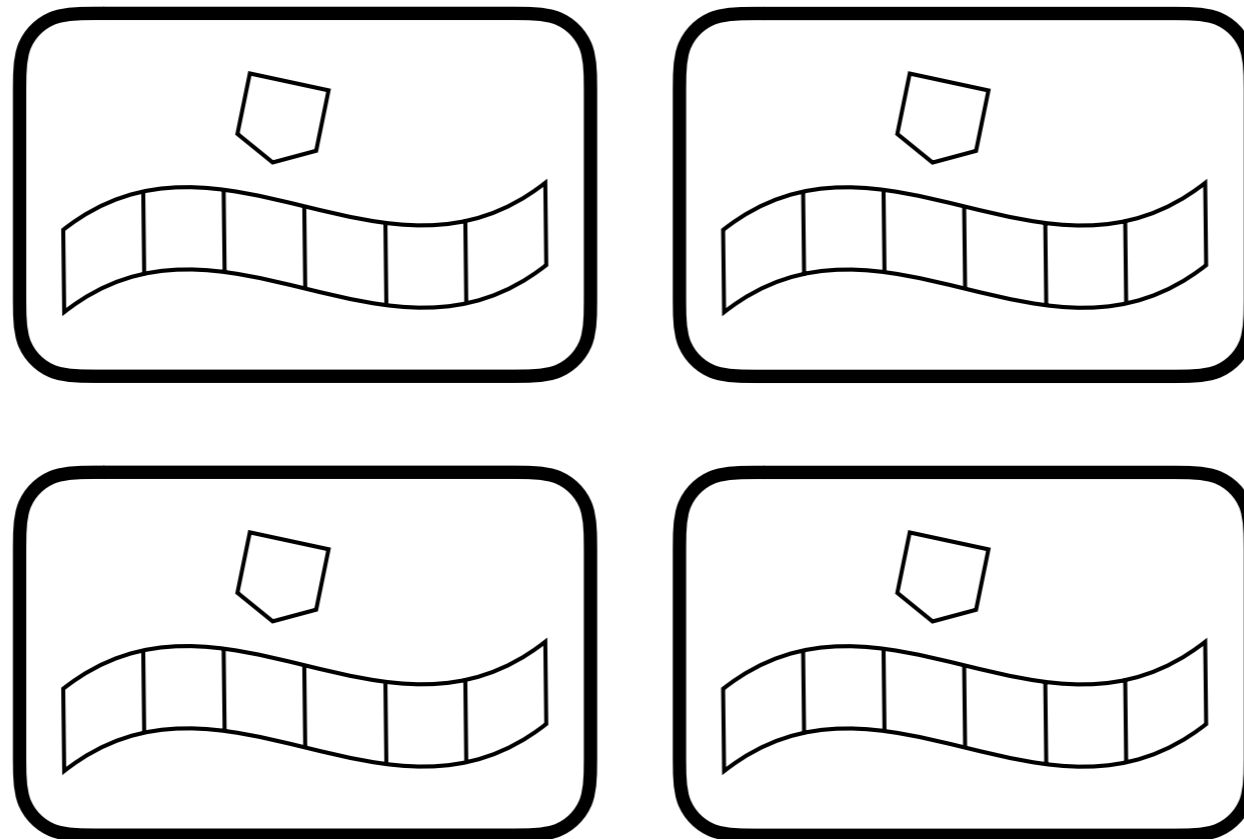
Simulation d'un calcul non déterministe



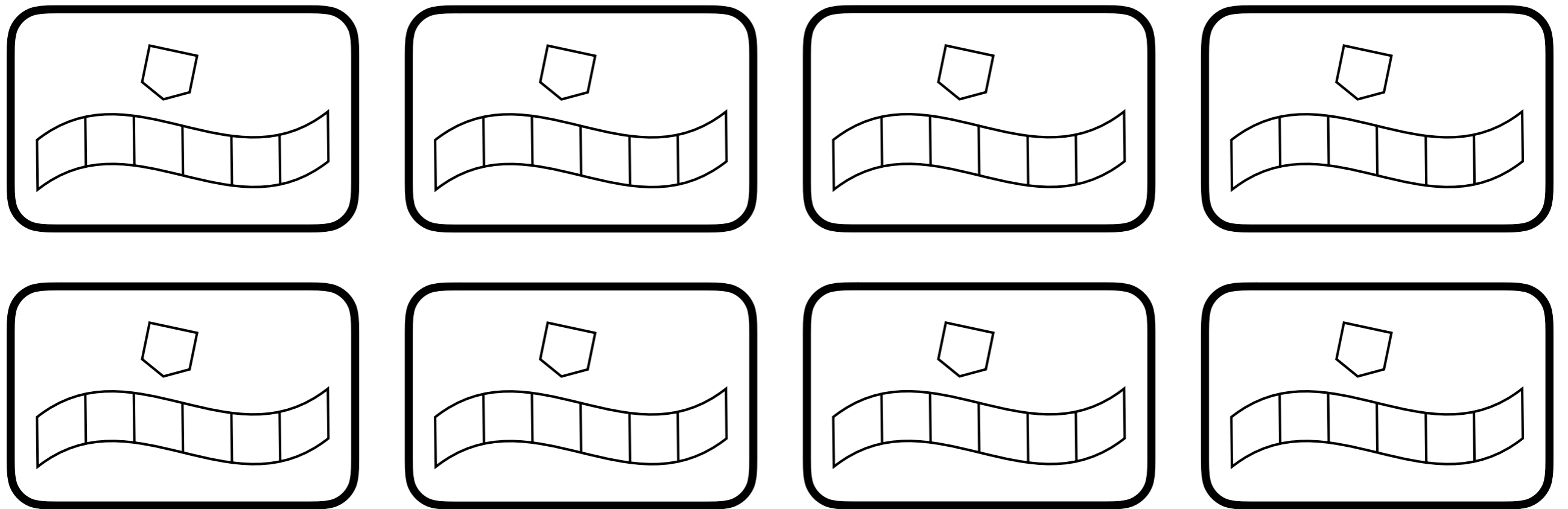
Simulation d'un calcul non déterministe



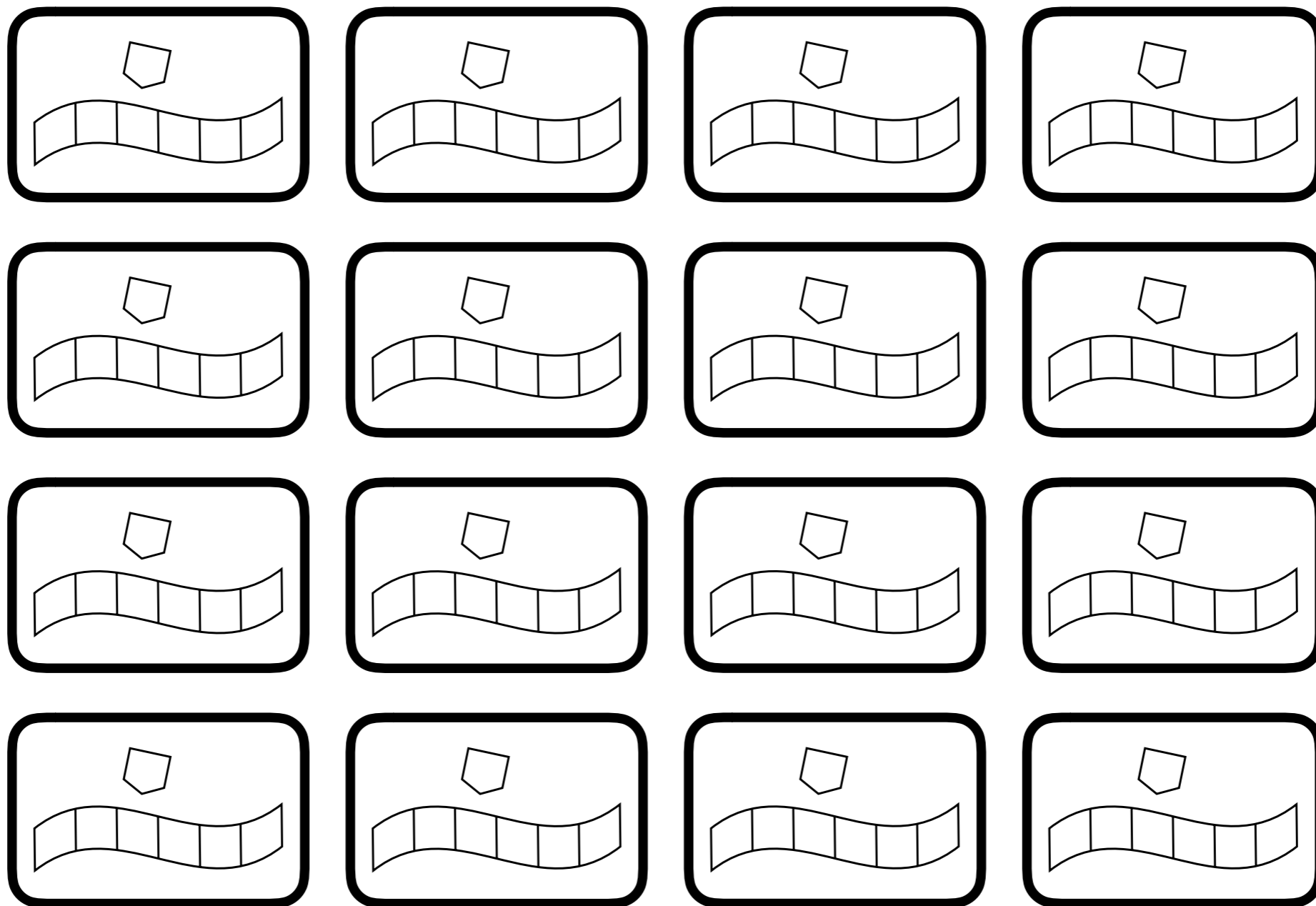
Simulation d'un calcul non déterministe



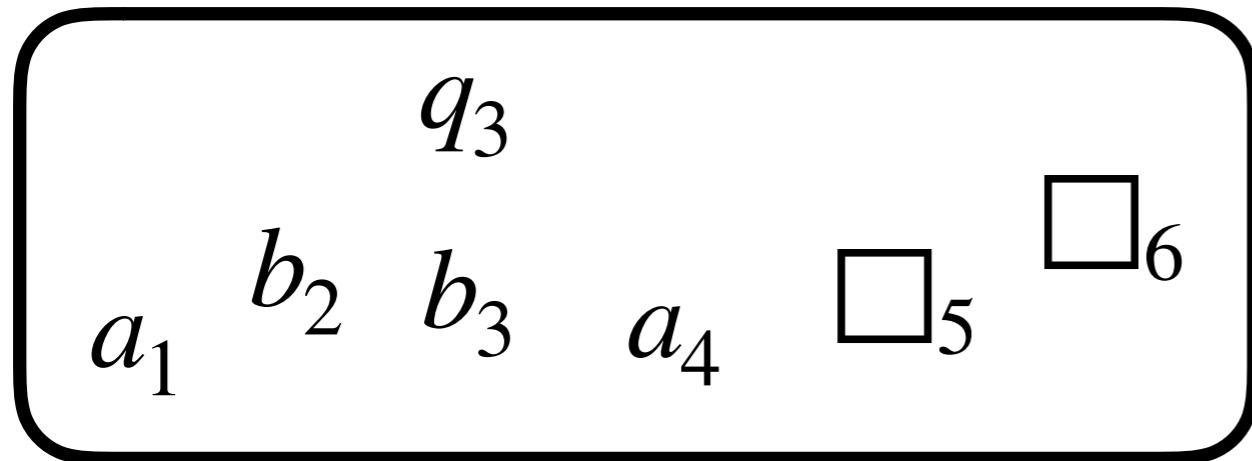
Simulation d'un calcul non déterministe



Simulation d'un calcul non déterministe

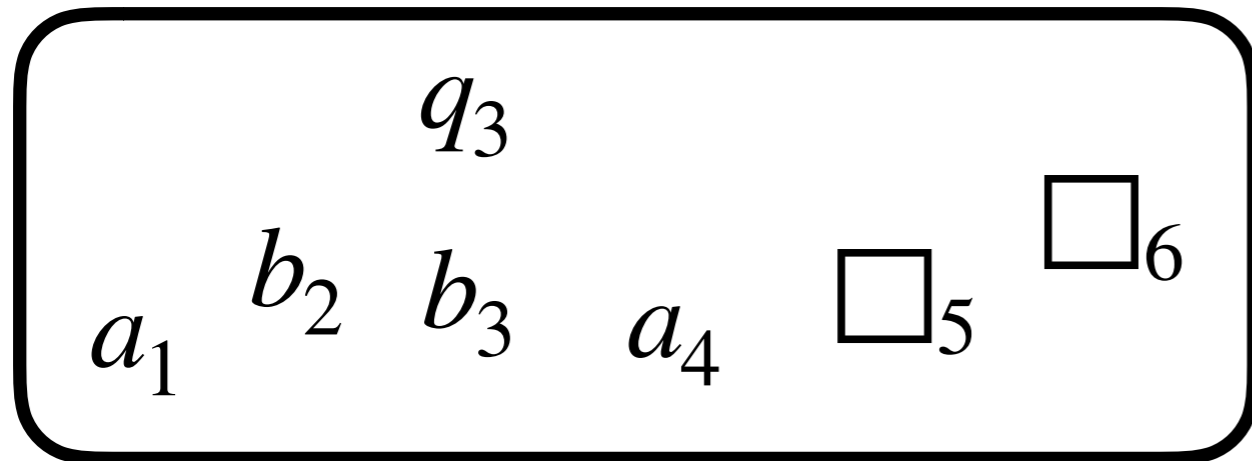


Résultat du calcul



**si q est un
état acceptant**

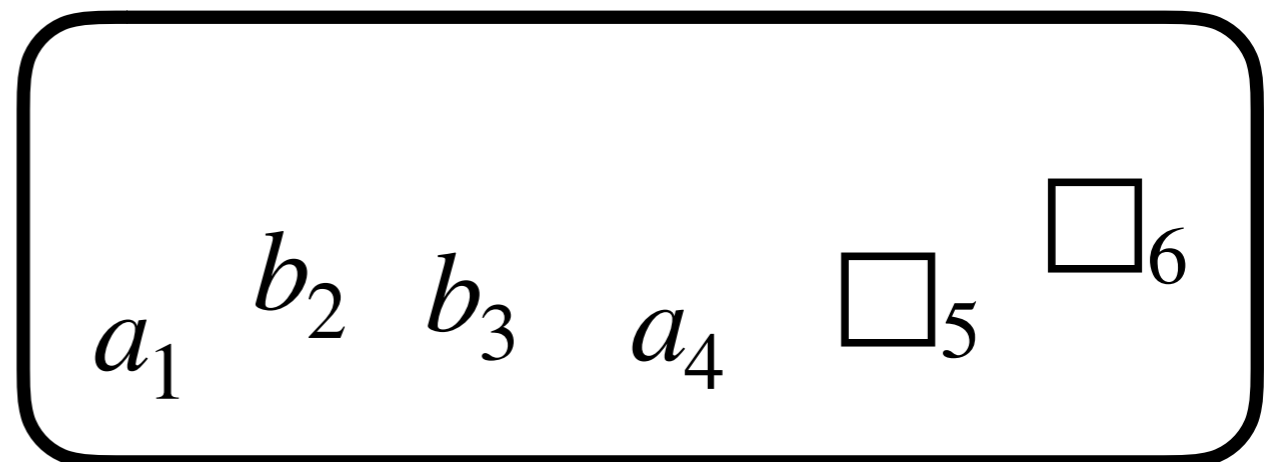
Résultat du calcul



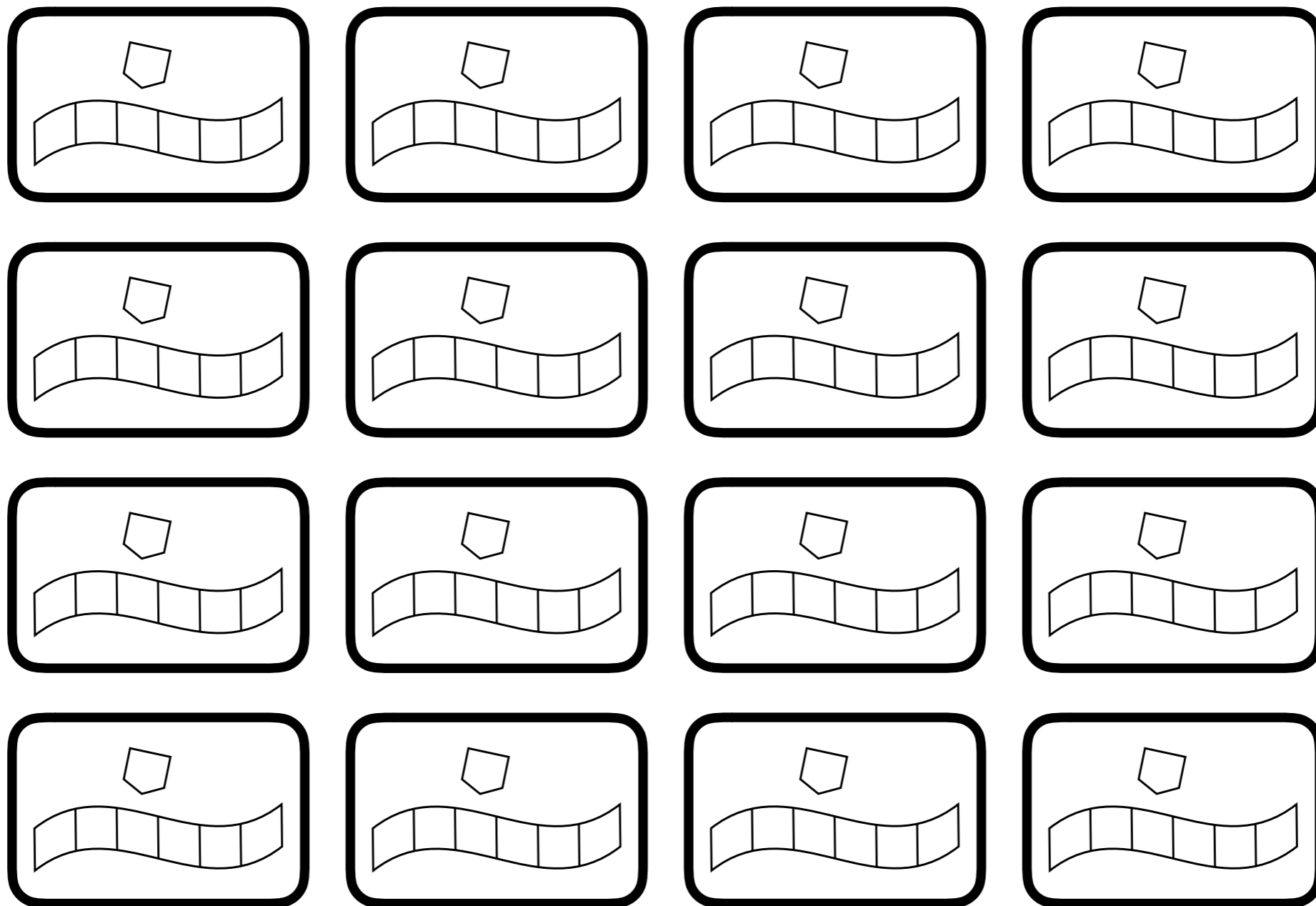
si q est un
état acceptant

$[q_i] \rightarrow []$ yes 

yes



Simulation d'un calcul non déterministe



Théorème : le systèmes à membranes simulent les machine de Turing **non déterministes** d'une façon efficace !

Calcul naturel

L'étude des modèles de calcul d'inspiration naturelle

- Biologique
- Chimique
- Physique

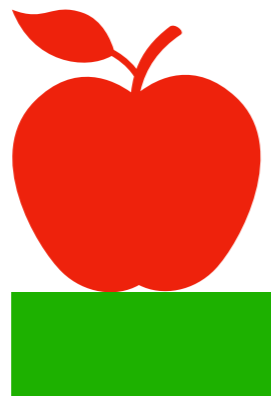
Exemple physique



d

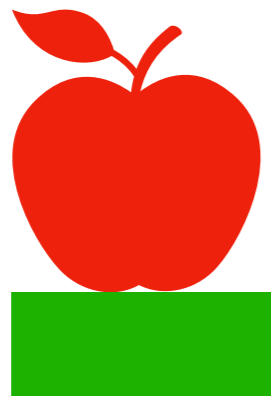


Exemple physique



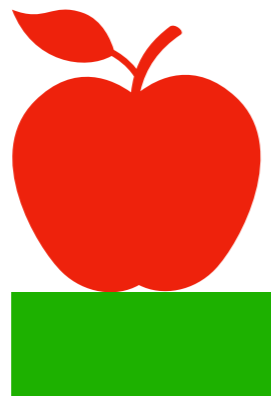
Exemple physique

$$t = \sqrt{\frac{2d}{g}}$$

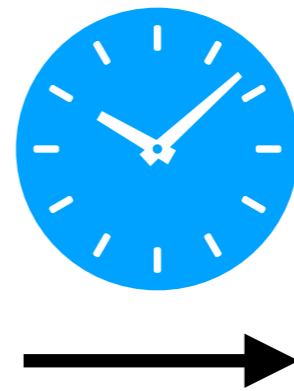


Exemple physique

$$t = \sqrt{\frac{2d}{g}}$$



$$d = \frac{xg}{2}$$



$$t = \sqrt{x}$$

**D'un certain point de vue, même
les ordinateurs électroniques,
c'est du calcul naturel !**

Ça sert à quoi, d'étudier la complexité en calcul naturel ?

- Établir la complexité des simulations pour applications pratiques
- Construire d'ordinateurs non électroniques (calculateurs à ADN d'Adleman)
- Étudier la nature du calcul en soi
- Essayer de résoudre les problèmes ouverts en informatique théorique (**P** vs **NP**)