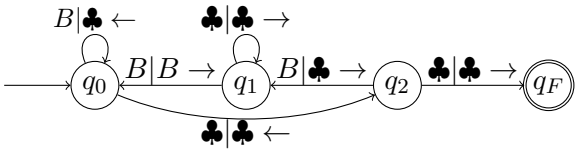
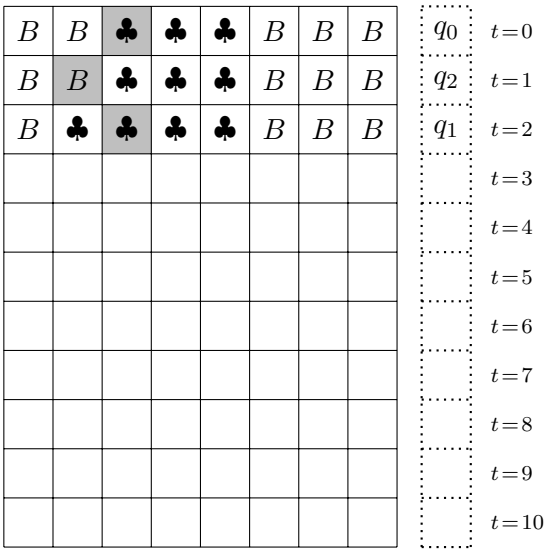
TD 02 – Machines de Turing et codage

Définition. Une **machine de Turing** déterministe est définie par $M = (Q, \Gamma, \Sigma, \delta, q_0, B, q_F)$, où :
- Q est un ensemble fini d'**états**,
- Γ est un **alphabet de ruban** fini,
- $\Sigma \subset \Gamma$ est l'**alphabet d'entrée**,
- $\delta : (Q \setminus \{q_F\}) \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ est la **fonction de transition**,
- $q_0 \in Q$ est l'**état initial**,
- $B \in \Gamma \setminus \Sigma$ est le **symbole blanc**,
- $q_F \in Q$ est l'**état final**.

Exemple. $Q = \{q_0, q_1, q_2, q_F\}$, $\Gamma = \{B, ♣\}$, $\Sigma = \{♣\}$.



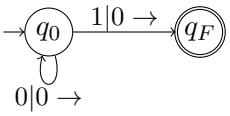
Calcul sur le mot d'entrée ♣♣♣ :



Exercice 1. *MT de l'école primaire*

Soit $\Sigma = \{0, 1\}$. Les mots en entrées sont interprétés en binaire comme des entiers naturels.

1. Donner une machine de Turing qui calcule la fonction $f_1 : \mathbb{N} \to \mathbb{N}$ définie par $f_1(n) = n + 1$.
2. Donner une machine de Turing qui calcule la fonction $f_2 : \mathbb{N} \to \mathbb{N}$ définie par $f_2(n) = 2n$.
3. Donner une machine de Turing qui calcule la fonction $f_3 : \mathbb{N} \to \mathbb{N}$ définie par $f_3(n) = \lfloor n/2 \rfloor$.
4. Quelle est la fonction $f_4 : \mathbb{N} \to \mathbb{N}$ calculée par la machine de Turing suivante ?



Exercice 2. *États d'une MT = mémoire finie*

Objectif : voir que l'on peut sauvegarder des informations (en quantité finie) dans les états.

Soit $M = (Q, \Gamma, \Sigma, \delta, q_0, B, q_F)$ la machine de Turing où
- $Q = \{q_0, q_a, q_b, q'_a, q'_b, q_F\}$,
- $\Sigma = \{a, b\}$, $\Gamma = \{a, b, B\}$,
- δ est donnée par :

$\delta(q_a, a) = (q_a, a, \rightarrow)$ $\quad$ $\delta(q_b, a) = (q_b, a, \rightarrow)$ $\quad$ $\delta(q_0, a) = (q_a, a, \rightarrow)$
$\delta(q_a, b) = (q_a, b, \rightarrow)$ $\quad$ $\delta(q_b, b) = (q_b, b, \rightarrow)$ $\quad$ $\delta(q_0, b) = (q_b, b, \rightarrow)$
$\delta(q_a, B) = (q'_a, B, \leftarrow)$ $\quad$ $\delta(q_b, B) = (q'_b, B, \leftarrow)$ $\quad$ $\delta(q'_a, a) = (q_F, a, \rightarrow)$
$\delta(q'_b, b) = (q_F, b, \rightarrow)$

1. Dessiner cette machine sous la forme d'un automate.
2. Quel est le langage $L(M)$ reconnu par cette machine de Turing, c'est-à-dire l'ensemble des mots de Σ^* pour lesquels la machine atteind son état final ?
3. Est-ce que le calul de cette machine de Turing s'arrête à partir de toute entrée dans Σ^* ?

Exercice 3. *MT*

Pour chacun des langages suivants, donner une machine de Turing déterministe qui le reconnaît, et qui de plus s'arrête sur toute entrée (argumenter brièvement).

1. $L = \{wbac \mid w \in \Sigma^*\}$ avec $\Sigma = \{a, b, c\}$.
2. $L = \{w \in \Sigma^* \mid |w| \equiv 0 \mod 3\}$ avec $\Sigma = \{a\}$.
3. $L = \{a^n b^n \mid n \in \mathbb{N}\}$ avec $\Sigma = \{a, b\}$.

Exercice 4. *MT peu utiles...*

Pour l'alphabet d'entrée $\Sigma = \{a, b\}$, donner des machines de Turing ayant chacun des comportements suivants.

1. Une machine de Turing qui ne s'arrête sur aucune entrée.
2. Une machine de Turing qui s'arrête uniquement sur l'entrée $abba$.

Exercice 5. *Espace et temps*

Supposons qu'une machine de Turing s'arrête au bout de t étapes de calcul en visitant (possiblement plusieurs fois chacune) s cases du ruban. On a évidement $s \leq t$ car au plus une nouvelle case est visitée par étape de temps, mais peut-on borner supérieurement t par une fonction de s ?

Exercice 6. *MT*

Donner des machines de Turing pour décider les langages suivants.

1. $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ avec $\Sigma = \{a, b, c\}$.
2. $L = \{uav \mid u, v \in \{a, b\}^* \text{ et } |u| = |v|\}$ avec $\Sigma = \{a, b\}$.

Exercice 7. *Codage*

On définit une fonction de codage $code_1$ des couples d'entiers de la manière suivante : étant donnés deux entiers x et y, on pose

$$code_1(x, y) = x_1 0 x_2 0 \dots x_{n-1} 0 x_n \, 1 \, y_1 y_2 \dots y_m,$$

où $x_1 x_2 \dots x_n$ et $y_1 y_2 \dots y_m$ sont les représentations binaires respectives de x et y.

1. Ce codage est-il injectif ? (Autrement dit, un tel code correspond-il bien à un unique couple d'entiers ?)
2. Donnez le codage du couple $(10, 5)$.
3. Quel est le couple associé au codage 100010100010111001110 ?
4. Quel est le nombre de bits utilisés pour coder un couple (x, y) ?

On définit un nouveau codage $code_2$ en posant maintenant

$$code_2(x, y) = t_1 0 t_2 0 \dots t_{\ell-1} 0 t_\ell \, 1 \, x_1 x_2 \dots x_n \, y_1 y_2 \dots y_m,$$

où $x_1 x_2 \dots x_n$ et $y_1 y_2 \dots y_m$ sont les représentations binaires respectives de x et y, et où $t_1 t_2 \dots t_\ell$ est la représentation binaire de l'entier n (qui est lui-même la taille de la représentation binaire de x).

5. Ce codage est-il injectif ?
6. Donnez le codage du couple $(10, 5)$.
7. Quel est le couple (x, y) associé au codage 100011100011001 ?
8. Quel est le nombre de bits utilisés pour coder un couple (x, y) ?

On souhaite généraliser ce codage aux tuples d'entiers $(x^1, \dots, x^k)$ pour k quelconque.

9. Proposez un tel codage.
10. Donnez le codage du tuple $(6, 11, 10, 3)$.
11. Donnez le nombre de bits utilisés pour coder un tuple arbitraire $(x^1, \dots, x^k)$.