

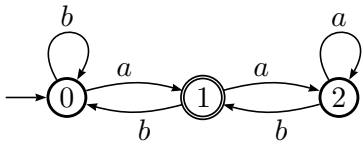
Automates Avancés

Travaux Dirigés n°9

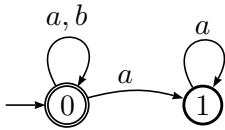
► Exercice 1. Automates de co-Büchi

Un automate de co-Büchi est un quintuplet $(Q, \Sigma, \delta, Q_0, F)$, où Q est un ensemble fini d'états, Σ un alphabet fini, δ la fonction de transition, $Q_0 \subseteq Q$ un ensemble fini d'états initiaux et $F \subseteq Q$ un ensemble fini d'états. Un mot infini $w \in \Sigma^\omega$ est accepté par l'automate, si il y a un calcul de l'automate sur w qui **ne passe pas** infiniment souvent par des états de F .

1. Donnez le langage accepté par l'automate de co-Büchi suivant:



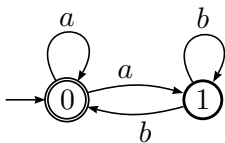
2. Donnez le langage accepté par l'automate de co-Büchi suivant:



Donnez un automate de co-Büchi déterministe pour ce langage.

3. Soit $A = (Q, \Sigma, \delta, Q_0, F)$ un automate de Büchi déterministe. Donnez un automate de co-Büchi qui accepte $\Sigma^\omega \setminus L^\omega(A)$.
4. On peut déterminer un automate de co-Büchi. Soit $A = (Q, \Sigma, \delta, Q_0, F)$. On définit A_{det} comme suit: $A_{det} = (Q', \Sigma, \delta', Q'_0, F')$ avec $Q' = 2^Q \times 2^Q$, $Q'_0 = (Q_0, \emptyset)$, $F' = 2^Q \times \{\emptyset\}$ et pour tout $a \in \Sigma$,
 - si $S = \emptyset$, alors $\delta'((R, S), a) = (T, T \setminus F)$ avec $T = \{q \in Q \mid \exists p \in R \text{ et } q \in \delta(p, a)\}$
 - si $S \neq \emptyset$, alors $\delta'((R, S), a) = (T, T')$ avec $T = \{q \in Q \mid \exists p \in R \text{ et } q \in \delta(p, a)\}$ et $T' = \{q \in Q \mid \exists p \in S \text{ et } q \in \delta(p, a)\} \setminus F$

Appliquez la construction à l'automate suivant:



► **Exercice 2.**

On considère le problème d'appartenance d'un mot $w \in \Sigma^\omega$ dans le langage donné par un automate de Büchi A non-déterministe.

1. Donnez un algorithme (le plus efficace possible) pour décider l'appartenance d'un mot $w = w_1.w_2^\omega$ avec $w_1, w_2 \in \Sigma^*$ dans $L^\omega(A)$ où A est un automate de Büchi non-déterministe avec n états. Indication: On peut considérer d'abord le problème du vide d'un automate de Büchi.
2. Donnez la complexité en temps de votre algorithme en fonction de la longueur de w_1 et de w_2 et de n .