DM2 - Mines MP 2019

1 Premiers exemples

1. Le langage reconnu par l'automate A_1 est l'ensemble des $\boxed{\text{mots de taille impaire}}$.
2. Le langage reconnu par l'automate A_2 est l'ensembles des $\boxed{\text{mots contenant un nombre impair de } b}$.
3. $\boxed{L(A_1) = (a \cdot a|b \cdot b|a \cdot b|b \cdot a)^* \cdot (a|b)}$
4. $\boxed{L(A_1) = (a|b \cdot a^* \cdot b)^* \cdot b \cdot a^*}$
5.
```
let a2 = 2, [|0, 1 ; 1, 0|], [|false; true|]  ;;
```

2 États accessibles d'un automate

6.
```
let numero n a =
    let t = Array.make n (-1) in
    let rec aux index = function (*parcours la liste*)
      |[] -> ()
      |h::q -> t.(h) <- index ; aux (index + 1) q in
    aux 0 a ; t ;;
```

7.
```
let etats_accessibles aut =
    let n, delta, f = aut in
    let visited = Array.make n false in
    let parcours = ref [] in
    let aux etat =
        if not visited.(etat) then
            begin
            visited.(etat) <- true ;
            parcours := etat :: !parcours ;
            let succ_a, succ_b = delta.(etat) in
            aux succ_a ;
            aux succ_b
            end in
    aux 0 ;;
    List.rev !parcours
```

Compléxité : La création d'une `Array` de taille n est en $O(n)$, et l'accés à un élément d'une liste, tout comme la concatenation d'un élément avec une liste, sont en $O(1)$. On ne s'intéresse donc qu'aux appels récursifs de la fonction `aux`. Comme $|Q| = n$, et que pour tout $q \in Q$, il existe deux états q_1 et q_2 (potentiellement égaux) tels que $\delta(q,a) = q_1$ et $\delta(q,b) = q_2$, la fonction `aux` fait au plus 2^n appels récursifs, soit une compléxité en $\boxed{O(2^n)}$.

8.
```
let partie_accessible aut =
    let n, delta, f = aut in
    let new_etats = etats_accessibles aut in
    let apparition = numero n new_etats in
    let new_n = List.length new_etats in
    let new_delta = Array.make new_n (0,0) in
    let new_f = Array.make new_n false in
    let rec aux = function
        |[] -> (new_n, new_delta, new_f)
        |h::t -> let s = apparition.(h) in
                new_f.(s) <- f.(h);
                let succ_a, succ_b = delta.(h) in
                new_delta.(s) <- apparition.(succ_a) ,apparition.(succ_b);
                aux t in
    aux new_etats ;;
```

3 Morphismes d'automates

3.1 Exemples de morphismes d'automates

9. $\varphi : \mathcal{A}_3 \to \mathcal{A}_2$ est représentée par

q	$\varphi(q)$
E	C
F	C
G	D

10. $\varphi : \mathcal{A}_4 \to \mathcal{A}_2$ est représentée par

q	$\varphi(q)$
H	C
I	C
J	D
K	D

11. Supposons qu'il existe un morphisme d'automate $\varphi : \mathcal{A}_1 \to \mathcal{A}_2$, alors, pour vérifier (2) : $\varphi(A) = B$, et pour vérifier (4) : $\varphi(B) = D$. Or si φ est un morphisme d'automates, alors d'après (3),

$$\begin{array}{rcl} \varphi(\delta_{\mathcal{A}_1}(A,a)) & = & \varphi(B) \\ & = & D \\ & = & \delta_{\mathcal{A}_2}(\varphi(A),a) \end{array}$$

Mais, $\varphi(A) = C$ d'après (1), donc

$$\begin{array}{rcl} \delta_{\mathcal{A}_2}(\varphi(A),a) & = & \delta_{\mathcal{A}_2}(C,a) \\ & = & D \\ & = & C \\ & \to & \underline{\text{Absurde !}} \end{array}$$

Ainsi, $\boxed{\text{il n'existe pas de morphisme d'automates de } \mathcal{A}_1 \text{ vers } \mathcal{A}_2}$.

12. Comme à la question précédente, supposons qu'il existe un morphisme d'automate $\varphi : \mathcal{A}_5 \to \mathcal{A}_2$, alors (2) et (4) nous donne : $\varphi(L) = C$, $\varphi(M) = D$ et $\varphi(N) = C$ (car $N \notin F_{\mathcal{A}_5}$). (3) impose alors,

$$\begin{array}{rcl} \varphi(\delta_{\mathcal{A}_5}(N,b)) & = & \varphi(L) \\ & = & C \\ & = & \delta_{\mathcal{A}_2}(\varphi(N),b) \\ & = & \delta_{\mathcal{A}_2}(C,b) \quad (\varphi(N) = C) \\ & = & D \\ & \to & \underline{\text{Absurde !}} \end{array}$$

Ainsi, $\boxed{\text{il n'existe pas de morphisme d'automates de } \mathcal{A}_5 \text{ vers } \mathcal{A}_2}$.

3.2 Propriétés des morphismes d'automates

13. Soit $\mathcal{A}, \mathcal{B}$ deux automates. Supposons qu'il existe un morphisme d'automates $\varphi : \mathcal{A} \to \mathcal{B}$.
Notons $\mathcal{P}(n)$ le prédicat : « pour tout $q \in Q_\mathcal{A}$ et pour tout mot m de taille n, $\varphi(\delta^*_\mathcal{A}(q,m)) = \delta^*_\mathcal{B}(\varphi(q),m)$ ».
Montrons par récurrence simple, que pour tout $n \in \mathbb{N}$, $\mathcal{P}(n)$.

<u>Initialisation</u> : Si $n = 0$, alors $m(= \varepsilon)$ est le mot vide, alors pour tout état q,
$\delta^*_\mathcal{A}(q,\varepsilon) = q$ et $\delta^*_\mathcal{B}(\varphi(q),\varepsilon) = \varphi(q)$ donc $\varphi(\delta^*_\mathcal{A}(q,\varepsilon)) = \varphi(q) = \delta^*_\mathcal{B}(\varphi(q),\varepsilon)$.

<u>Hérédité</u> : Soit $n \in \mathbb{N}$, supposons $\mathcal{P}(n)$, montrons $\mathcal{P}(n) \Rightarrow \mathcal{P}(n+1)$,
Soit $q \in Q_\mathcal{A}$ un état quelconque et m un mot de taille $n+1$, on peut alors noter $m = \sigma m_r$ où m_r est un mot de taille n. Ainsi,

$$\begin{array}{rcl} \varphi(\delta^*_\mathcal{A}(q,m)) & = & \varphi(\delta^*_\mathcal{A}(\delta_\mathcal{A}(q,\sigma),m_r)) \text{ par définition de } \delta^*_\mathcal{A} \\ & = & \delta^*_\mathcal{B}(\varphi(\delta_\mathcal{A}(q,\sigma)),m_r) \text{ par l'hypothèse de récurrence} \\ & = & \delta^*_\mathcal{B}(\delta_\mathcal{B}(\varphi(q),\sigma),m_r) \text{ par la propriété (3) d'un morphisme d'automates} \\ & = & \delta^*_\mathcal{B}(\varphi(q),m) \text{ par définition de } \delta^*_\mathcal{B} \end{array}$$

Donc $\forall n \in \mathbb{N}$, $\mathcal{P}(n)$ est vérifié. Ainsi, si $m \in L(\mathcal{A})$, alors $\delta^*_\mathcal{A}(i_\mathcal{A},m) \in F_\mathcal{A}$, et donc $\varphi(\delta^*_\mathcal{A}(i_\mathcal{A},m)) = \delta^*_\mathcal{B}(\varphi(i_\mathcal{A}),m) = \delta^*_\mathcal{B}(i_\mathcal{B},m) \in F_\mathcal{B}$ d'après ce que l'on vient de montrer, (2) et (4). Donc $\delta^*_\mathcal{B}(i_\mathcal{B},m) \in F_\mathcal{B}$, et par conséquent $m \in L(\mathcal{B})$. $\boxed{\text{D'où le résultat}}$.

14. Soit $\mathcal{A}, \mathcal{B}$ deux automates **finies**, tels que $|Q_\mathcal{A}| = |Q_\mathcal{B}|$. Supposons qu'il existe un mortphisme d'automates $\varphi : \mathcal{A} \to \mathcal{B}$, alors φ est surjective par définition. Or comme $|Q_\mathcal{A}| = |Q_\mathcal{B}|$, φ surjective $\Leftrightarrow$ φ bijective. Donc $\boxed{\varphi \text{ est nécessairement bijective}}$.
   De plus :
   - φ^{-1} est bijective donc surjective (1)
   - $\varphi^{-1}(i_\mathcal{B}) = \varphi^{-1}(\varphi(i_\mathcal{B})) = i_\mathcal{A}$ (2)
   - Soit $q \in F_\mathcal{B}$, d'après la définition de $\varphi, \forall \sigma \in \{a, b\}$, $\varphi(\delta_\mathcal{A}(\varphi^{-1}(q), \sigma)) = \delta_\mathcal{B}(q, \sigma)$, et donc en appliquant φ^{-1} à l'égalité :
     $\forall q \in Q_\mathcal{B}$, $\forall \sigma \in \{a, b\}$, $\varphi^{-1}(\delta_\mathcal{B}(q, \sigma)) = \delta_\mathcal{A}(\varphi^{-1}(q), \sigma)$ (3)
   - $\forall q \in Q_\mathcal{B}$, (encore une fois d'après la définition de φ),

   $$\begin{aligned} \varphi^{-1}(q) \in F_\mathcal{A} &\iff \varphi(\varphi^{-1}(q)) \in F_\mathcal{B} \\ &\iff q \in F_\mathcal{B} \ (4) \end{aligned}$$

   Donc $\boxed{\varphi^{-1} \text{ est bien un morphisme d'automates}}$.

15. Soit $\mathcal{A}, \mathcal{B}, \mathcal{C}$ trois automates et $\varphi : \mathcal{B} \to \mathcal{C}$, $\psi : \mathcal{A} \to \mathcal{B}$ deux morphismes d'automates.
   - φ et ψ surjectives donc $\varphi \circ \psi$ est surjective (1)
   - $(\varphi \circ \psi)(i_\mathcal{A}) = \varphi(i_\mathcal{B}) = i_\mathcal{C}$ (2)
   - $\forall q \in Q_\mathcal{A}$, $\forall \sigma \in \{a, b\}$, $(\varphi \circ \psi)(\delta_\mathcal{A}(q, \sigma)) = \varphi(\delta_\mathcal{B}(\psi(q), \sigma)) = \delta_\mathcal{A}((\varphi \circ \psi)(q), \sigma)$ (3)
   - $\forall q \in Q_\mathcal{A}$, $q \in F_\mathcal{A} \iff \psi(q) \in F_\mathcal{B} \iff (\varphi \circ \psi)(q) \in F_\mathcal{C}$ (4)

   Donc $\varphi \circ \psi : \mathcal{A} \to \mathcal{C}$ est un morphisme d'automates, $\boxed{\text{d'où le résultat}}$.

3.3 Existence de morphismes d'automates entre automates accessibles

16. Soit $\mathcal{A}, \mathcal{B}$ deux automates **accessibles**, soit $\varphi : \mathcal{A} \to \mathcal{B}$ une application verifiant les propriétés (2), (3) et (4). Le résulat montré par récurrence à la question 13. est toujours valable puisque les seuls hypothèses utilisées ((2), (3) et (4)) sont vérfiées.
   Soit $q \in Q_\mathcal{B}$, comme q est accesible, il existe $m \in L(\mathcal{B})$ tel que :

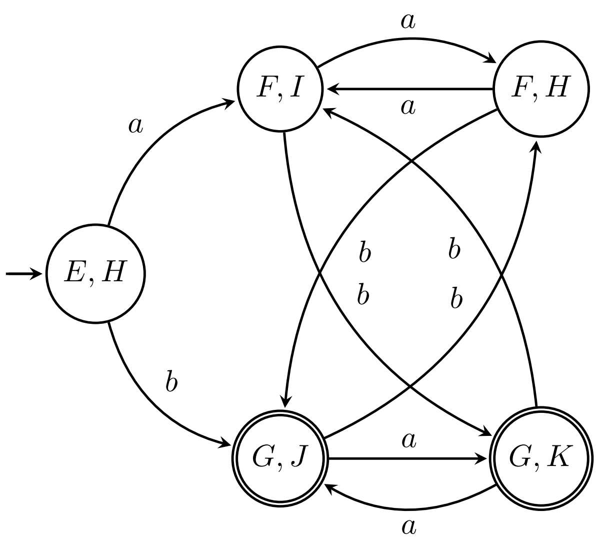
   $$\begin{aligned} \delta^*_\mathcal{B}(i_\mathcal{B}, m) &= q \\ &= \delta^*_\mathcal{B}(\varphi(i_\mathcal{A}), m) \text{ d'après (2)} \\ &= \varphi(\delta^*_\mathcal{A}(i_\mathcal{A}, m)) \text{ d'après la question 13.} \end{aligned}$$

   Or, comme $\delta^*_\mathcal{A}(i_\mathcal{A}, m) \in Q_\mathcal{A}$, alors il existe $q' \in Q_\mathcal{A}$ tel que $\varphi(q') = q$ donc φ est surjective, d'où $\boxed{(\text{automates accessibles}) \wedge (2) \wedge (3) \wedge (4) \implies (1)}$

17.
```
let existe_morphismes aut1 aut2 =
    let n1, delta1, f1 = aut1 in
    let n2, delta2, f2 = aut2 in
    let def = ref true in (*variable indiquant si un morphisme existe ou non*)
    let visited = Array.make n1 false in
    let morphisme = Array.make n1 (-1) in
    let construire etat1 etat2 = (*fonction qui construit le morphisme*)
        if morphisme.(etat1) <> 1 (*si phi(q) ne s'est pas encore vu etre associe
        ↪  une image, on la definit*)
            then begin
                if ( f1.(etat1) = f2.(etat2) ) || ( (not f1.(etat1)) = (not
                ↪  f2.(etat2)) ) (*test de la condition (4)*)
                then morphisme.(etat1) <- etat2 (*si (4) est respectee, alors on
                ↪  peut definir phi(q)*)
                else def := false (*sinon il n'existe pas de morphisme*)
            end
        else if morphisme.(etat1) <> etat2 (*si phi(q) est deja defini, mais que la
        ↪  condition (3) n'est pas respectee...*)
             then def := false (*... alors il n'existe pas de morphisme*)
    in
    let rec aux etat1 = (*fonction qui parcours l'automate afin de constuire le
    ↪  morphisme*)
```

```ocaml
        if not visited.(etat1) (*on verifie que le sommet n'a pas deja ete visite*)
        then begin
            let etat2 = morphisme.(etat1) in
            let succ1_a, succ1_b = delta1.(etat1) in (*on construit le morphisme recursivement, en partant du sommet que l'on visite*)
            let succ2_a, succ2_b = delta2.(etat2) in
            visited.(etat1) <- true ;
            construire succ1_a succ2_a ; (*on definit les images de phi pour sigma = a*)
            construire succ1_b succ2_b ; (*on definit les images de phi pour sigma = b*)
            if !def then (aux succ1_a ; aux succ1_b) (*si le morphisme existe (i.e les etapes de construction se sont achevees), alors on continue jusqu'a ce que tous les sommets soient visites*)
            end
    in
    morphisme.(0) <- 0 ; (*initialisation de la constuction avec la condition (2)*)
    aux 0 ; (*début du parcours*)
    !def , morphisme ;;
```

18.



Partie accessible de $\mathcal{A}_3 \times \mathcal{A}_4$

19.

```ocaml
let produit aut1 aut2 =
    let n1, delta1, f1 = aut1 in
    let n2, delta2, f2 = aut2 in
    let n = n1 * n2 in
    let f = Array.make n false in
    let delta = Array.make n (0, 0) in
    let prod_etats n x1 x2 = (*fonction auxiliaire pour calculer delta d'un produit d'automates*)
    match x1, x2 with
    |(a, b), (c, d) -> n*a + c, n*b + d in
    for i = 0 to n1 - 1 do
        for j = 0 to n2 - 1 do
            begin
            if f1.(i) && f2.(j) then
            f.(n2 * i + j) <- true ;
            delta.(n2 * i + j) <- prod_etats n2 delta1.(i) delta2.(j) ;
            end
        done; done;
    partie_accessible (n, delta, f) ;;
```

20. Soit $\mathcal{A}, \mathcal{A}'$ deux automates.
Notons $\mathcal{P}(n)$ le prédicat : « pour tout état $(q, q') \in Q_{\mathcal{A}} \times Q_{\mathcal{A}'}$ et pour tout mot m de taille n, $\delta^*_{\mathcal{A}\times\mathcal{A}'}((q,q'), m) = (\delta^*_{\mathcal{A}}(q,m), \delta^*_{\mathcal{A}'}(q',m))$ ». Montrons par récurrence simple, que pour tout $n \in \mathbb{N}$, $\mathcal{P}(n)$.

Initialisation : Si $n = 0$, alors $m(=\varepsilon)$ est le mot vide, alors pour tout état (q,q'), $\delta^*_{\mathcal{A}\times\mathcal{A}'}((q,q'),\varepsilon) = (q,q')$
Or $\delta^*_{\mathcal{A}}(q,\varepsilon) = q$ et $\delta^*_{\mathcal{A}'}(q',\varepsilon) = q'$ donc $\delta^*_{\mathcal{A}\times\mathcal{A}'}((q,q'),\varepsilon) = (q,q') = (\delta^*_{\mathcal{A}}(q,\varepsilon), \delta^*_{\mathcal{A}'}(q',\varepsilon))$

Hérédité : Soit $n \in \mathbb{N}$, supposons $\mathcal{P}(n)$, montrons $\mathcal{P}(n) \Rightarrow \mathcal{P}(n+1)$,
Soit $(q,q') \in Q_{\mathcal{A}} \times Q_{\mathcal{A}'}$ un état quelconque et m un mot de taille $n+1$, on peut alors noter $m = \sigma m_r$ où m_r est un mot de taille n. Ainsi,

$$\begin{aligned}
\delta^*_{\mathcal{A}\times\mathcal{A}'}((q,q'),m) &= \delta^*_{\mathcal{A}\times\mathcal{A}'}(\delta_{\mathcal{A}\times\mathcal{A}'}((q,q'),\sigma), m_r) \text{ par définition de } \delta^*_{\mathcal{A}\times\mathcal{A}'} \\
&= \delta^*_{\mathcal{A}\times\mathcal{A}'}((\delta_{\mathcal{A}}(q,\sigma), \delta_{\mathcal{A}'}(q',\sigma)), m_r) \text{ par par définition de } \delta_{\mathcal{A}\times\mathcal{A}'} \\
&= (\delta^*_{\mathcal{A}}(\delta_{\mathcal{A}}(q,\sigma), m_r), \delta^*_{\mathcal{A}'}(\delta_{\mathcal{A}'}(q',\sigma), m_r)) \text{ d'après l'hypothèse de récurrence} \\
&= (\delta^*_{\mathcal{A}}(q,m), \delta^*_{\mathcal{A}'}(q',m)) \text{ par définition de } \delta^*_{\mathcal{A}} \text{ et } \delta^*_{\mathcal{A}'}
\end{aligned}$$

Donc $\forall n \in \mathbb{N}$, $\mathcal{P}(n)$ est vérifié. Ainsi, si $(q,q') \in Q_{\mathcal{A}} \times Q_{\mathcal{A}'}$ est un état accessible, alors il existe m tel que $(q,q') = \delta^*_{\mathcal{A}\times\mathcal{A}'}((i_{\mathcal{A}}, i_{\mathcal{A}'}), m) = (\delta^*_{\mathcal{A}}(i_{\mathcal{A}}, m), \delta^*_{\mathcal{A}'}(i_{\mathcal{A}'}, m))$.
Or comme $L(\mathcal{A}) = L(\mathcal{A}')$, $m \in L(\mathcal{A}) \iff m \in L(\mathcal{A}')$. $\boxed{\text{D'où } q \in F_{\mathcal{A}} \iff q' \in F_{\mathcal{A}'}}$.

21. Soit $\mathcal{A}, \mathcal{A}'$ deux automates accessibles qui acceptent le même langage. On considère $\mathcal{B}$ la partie accessible de $\mathcal{A} \times \mathcal{A}'$.
Montrons qu'il existe un morphisme d'automates $\varphi : \mathcal{B} -> \mathcal{A}$:
Considérons $\varphi : \begin{array}{ccc} \mathcal{B} & \to & \mathcal{A} \\ (q,q') & \mapsto & q \end{array}$.
D'après la question 16, comme $\mathcal{B}$ et $\mathcal{A}$ sont deux automates accessibles, il suffit de montrer que φ vérifie (2), (3) et (4) :
   — Comme $i_{\mathcal{B}} = (i_{\mathcal{A}}, i_{\mathcal{A}'})$, on a directement $\varphi(i_{\mathcal{B}}) = i_{\mathcal{A}}$, ce qui montre que φ vérifie (2)
   — Si $(q,q') \in Q_{\mathcal{B}}$, alors pour tout $\sigma \in \{a,b\}$, on a à la fois $\varphi(q,q') = q$ par définition de φ et $\varphi(\delta_{\mathcal{B}}((q,q'),\sigma)) = \varphi(\delta_{\mathcal{A}}(q,\sigma), \delta_{\mathcal{A}'}(q',\sigma)) = \delta_{\mathcal{A}}(q,\sigma)$. Donc φ vérifie (3).
   — Enfin, soit $(q,q') \in Q_{\mathcal{B}}$:
     ($\Rightarrow$) Si $(q,q') \in F_{\mathcal{B}}$, par définition de $\mathcal{B}$, $\varphi(q,q') = q \in F_{\mathcal{A}}$;
     ($\Leftarrow$) Si $\varphi(q,q') = q \in F_{\mathcal{A}}$, alors d'après la question 20, $q' \in F_{\mathcal{A}'}$, et donc $(q,q') \in F_{\mathcal{B}}$;
     Donc φ vérifie également (4)

Ainsi, on démontre bien $\boxed{\text{l'existence d'un morphisme d'automates entre } \mathcal{B} \text{ et } \mathcal{A}}$.
$\boxed{\text{La symétrie du problème entre } \mathcal{A} \text{ et } \mathcal{A}'}$ permet de conclure.

3.4 Diagramme d'automates

22. Montrons que $\equiv$ définit bien une relation d'équivalence :
   — Si $p \in Q_{\mathcal{B}}$ alors il existe une suite de longueur $0+1$ constituée du terme $p = q_0 = p$ donc $p \equiv p$. Ainsi $\boxed{\equiv \text{ est réfléxive}}$.
   — Soit $(p,q) \in Q^2_{\mathcal{B}}$ tel que $p \equiv q$, par définition, il existe une suite de longueur $k+1 \in \mathbb{N}^*$ constituée des termes $p = q_0, q_1, ..., q_k = q$, alors en considérant la suite constituée des termes $q = p_0, p_1, ..., p_{k-1}, p_k = p$, où $\forall\, 0 \leq j \leq k, p_j = q_{k-j}$. Ainsi on a :

$$\forall\, 0 \leq j < k,\ \varphi(p_{j+1}) = \varphi(q_{\underbrace{k-j-1}_{\in [\![1;k]\!]}}) = \varphi(q_{k-j}) = \varphi(p_j)$$

$$\text{ou } \psi(p_{j+1}) = \psi(q_{\underbrace{k-j-1}_{\in [\![1;k]\!]}}) = \psi(q_{k-j}) = \psi(p_j)$$

   Soit, $\forall\, 0 \leq j < k,\ \varphi(p_j) = \varphi(p_{j+1})$ ou $\psi(p_j) = \psi(p_{j+1})$
   Ce qui montre $p \equiv q \implies q \equiv p$, et donc que $\boxed{\equiv \text{ est symétrique}}$.
   — Soit $(p,q,r) \in Q^3_{\mathcal{B}}$ tel que $p \equiv q$ et $q \equiv r$, alors il existe deux suites, de taille respective $l+1$ et $m+1$ ($(l,m) \in \mathbb{N}^2$), constituées des termes $p = q_0, q_1, ..., q_l = q$ et $q = r_0, r_1, ..., r_m = r$.
   Alors en considérant la suite de taille $k+1$ (où $k = l+m \in \mathbb{N}$) constituée des termes $p = q_0, q_1, ..., q_l = r_0, r_1, ..., r_m = r$, on a clairement :

$$\forall\, 0 \leq j < k,\ \varphi(p_j) = \varphi(p_{j+1}) \text{ ou } \psi(p_j) = \psi(p_{j+1}) \text{(même lorsque } j = l \text{ puisque } \varphi(q_l) = \varphi(r_0) = \varphi(r_1))$$

   Donc $p \equiv q \wedge q \equiv r \implies p \equiv r$, ce qui conclut sur la $\boxed{\text{transitivité de } \equiv}$.

On a ainsi montré que $\boxed{\equiv \text{ est une relation d'équivalence}}$.

23. Soit $(p,q) \in Q_{\mathcal{B}}^2$ tel que $p \equiv q$. Soit $p = q_0, q_1, ..., q_k = q$ la suite associée à $\equiv$.
Soit $\sigma \in \{a,b\}$, $j \in [\![0;k[\![$. Si $\varphi(q_j) = \varphi(q_{j+1})$, il découle :

$$\begin{aligned}\varphi(\delta_{\mathcal{B}}(q_j,\sigma)) &= \delta_{\mathcal{A}}(\varphi(q_j),\sigma) && \text{par propriété de morphisme de } \varphi\\ &= \delta_{\mathcal{A}}(\varphi(q_{j+1}),\sigma) \\ &= \varphi(\delta_{\mathcal{B}}(q_{j+1},\sigma)) && \text{à nouveau par propriété de } \varphi\end{aligned}$$

Le résultat est clairement similaire avec ψ, si on suppose $\psi(q_j) = \psi(q_{j+1})$.
Ainsi en définissant la suite $\Delta_0, \Delta_1, ..., \Delta_k$, avec $\forall j \in [\![0;k]\!],\ \Delta_j = \delta_{\mathcal{B}}(q_j,\sigma)$, alors

$$\forall\, 0 \leq j < k,\ \varphi(\Delta_j) = \varphi(\Delta_{j+1}) \text{ ou } \psi(\Delta_j) = \psi(\Delta_{j+1}) \text{ (d'après ce que l'on vient de montrer).}$$

D'où le résultat : $\boxed{\delta_{\mathcal{B}}(p,\sigma) \equiv \delta_{\mathcal{B}}(q,\sigma)}$.

24. Soit $(p,q) \in Q_{\mathcal{B}}^2$ tel que $p \equiv q$. Par symétrie de la relation $\equiv$, montrer que $p \in F_{\mathcal{B}} \implies q \in F_{\mathcal{B}}$ suffira à montrer l'équivalence.
Soit $p = q_0, q_1, ..., q_k = q$ la suite associée à $\equiv$.
Soit $j \in [\![0;k[\![$. Si $\varphi(q_j) = \varphi(q_{j+1})$, montrons que $q_j \in F_{\mathcal{B}} \implies q_{j+1} \in F_{\mathcal{B}}$:

$$q_j \in F_{\mathcal{B}} \implies \varphi(q_j) = \varphi(q_{j+1}) \in F_{\mathcal{A}} \implies q_{j+1} \in F_{\mathcal{B}} \text{ (par propriété de morphisme de } \varphi)$$

Le résultat est clairement similaire avec ψ, si on suppose $\psi(q_j) = \psi(q_{j+1})$.
Et donc, par une récurrence immédiate, $p \in F_{\mathcal{B}} \implies q \in F_{\mathcal{B}}$ et ainsi $\boxed{p \in F_{\mathcal{B}} \iff q \in F_{\mathcal{B}}}$

25.