Sujet CCP 2015-2017 : correction

Exercice 1(CCP 2015)
Question II.1. Une expression régulière représentant $L(\mathcal{E})$ est $ba(ba)^*|bba^*$.
Question II.2. Pour tout $n \in X^*$, $o \in Q$, $q \in Q$, $d \in Q$, $q = \delta^*(o, \Lambda) \wedge d = \delta^*(q, n) \Leftrightarrow q = o \wedge d = \delta^*(q, n) \Leftrightarrow d = \delta^*(o, n) \Leftrightarrow d = \delta^*(o, \Lambda.n)$.
Supposons que, pour un $m \in X^*$ fixé, pour tout $n \in X^*$, $q \in Q$, $d \in Q$, $q = \delta^*(o, m) \wedge d = \delta^*(q, n) \Leftrightarrow d = \delta^*(o, m.n)$. Soit $e \in X$. Alors $q = \delta^*(o, e.m) \wedge d = \delta^*(q, n) \Leftrightarrow \exists q' \in Q,\ \delta(o, e) = q' \wedge \delta^*(q', m) = q \wedge d = \delta^*(q, n) \Leftrightarrow \exists q' \in Q,\ \delta(o, e) = q' \wedge d = \delta^*(q', m.n) \Leftrightarrow d = \delta^*(o, e.m.n)$.
Par récurrence induction (ou récurrence sur la taille des mots), pour tout $m \in X^*$, $n \in X^*$, $o \in Q$, $q \in Q$, $d \in Q$, $q = \delta^*(o, m) \wedge d = \delta^*(q, n) \Leftrightarrow \delta^*(o, m.n)$.
Question II.3. $\sqrt{L(\mathcal{E})}$ est dénoté par l'expression $(ba)(ba)^*|b$.
Question II.4. Soit $m \in L$. Alors $m.m \in L^2$ donc $m \in \sqrt{L^2}$. Ainsi, $L \subset \sqrt{L^2}$.
Il n'y a en revanche pas d'incusion générale entre $\sqrt{L}^2$ et L. Par exemple, en notant $L = L(\mathcal{E})$ l'exemple de l'énoncé, $ba \in L$ et $ba \notin \sqrt{L}^2$; $ba \in \sqrt{L}$, $b \in \sqrt{L}$, $bab \in \sqrt{L}^2$ mais $bab \notin L$.
Question II.5.

Question II.6. Il s'agit de $(ba)(ba)^*|b$.
Question II.7. $Q_{\sqrt{A}}$ est un ensemble fini ; $i_{\sqrt{A}} \in Q_{\sqrt{A}}$; $T_{\sqrt{A}} \subset Q_{\sqrt{A}}$, $\delta_{\sqrt{A}}$ est définie sur $Q_{\sqrt{A}} \times X$ à valeurs dans $Q_{\sqrt{A}}$ car δ_A est défini sur $Q_A \times X$ à valeurs dans Q_A. L'automate $Q_{\sqrt{A}}$ est donc un automate fini déterministe complet.
Question II.8. Par définition, pour tout $o \in Q_{\sqrt{A}}$, $\delta^*_{\sqrt{A}}(o, \Lambda) = o = (o_i)_{0 \leq i \leq n} = (\delta^*_A(o_i, \Lambda))_{0 \leq i \leq n}$. Supposons que, pour $m \in X^*$ fixé, pour tout $o \in Q_{\sqrt{A}}$, $\delta^*_{\sqrt{A}}(o, m) = (\delta^*_A(o_i, m))_{0 \leq i \leq n}$. Soit $e \in X$. Soit $o \in Q_{\sqrt{A}}$. $\delta^*_{\sqrt{A}}(o, e.m) = \delta^*_{\sqrt{A}}(\delta_{\sqrt{A}}(o, e), m) = \delta^*_{\sqrt{A}}\big((\delta_A(o_i, e))_{0 \leq i \leq n}\big)$; de par l'hypothèse d'induction, $\delta^*_{\sqrt{A}}(o, e.m) = \big(\delta^*_A(\delta_A(o_i, e), m)\big)_{0 \leq i \leq n} = \big(\delta^*_A(o_i, e.m)\big)_{0 \leq i \leq n}$.
Par induction, pour tout $m \in X^*$, pour tout $o \in Q_{\sqrt{A}}$, $\delta^*_{\sqrt{A}}(o, m) = \big(\delta^*_A(o_i, m)\big)_{0 \leq i \leq n}$.
Question II.9. Soit $m \in X^*$. $m \in L(\sqrt{A})$ si et seulement si $\delta^*_{\sqrt{A}}(i_{\sqrt{A}}, m) \in T_{\sqrt{A}}$ si et seulement si $\big(\delta^*_A(q_i, m)\big)_{0 \leq i \leq n} \in T_{\sqrt{A}}$ si et seulement si il existe $j \in [\![0, n]\!]$ tel que $\delta^*_A(q_0, m) = q_j$ et $\delta^*_A(q_j, m) \in T_A$ si et seulement si $\delta^*_A(q_0, m.m) \in T_A$ si et seulement si $m.m \in L(A)$.
Question II.10. D'après la question précédente, $L(\sqrt{\mathcal{A}}) = \sqrt{L(\mathcal{A})}$.

Exercice 2 (CCP 2017)
Q12.

```
type alphabet=list char;;
type mot=list char;;
type langage=list mot;;
```

Q13.

```
let prefixer p l = match l with
    | [] -> []
    | m::l0 -> (p@m)::(prefixer p l0);;
```

Q14.
Pour chaque mot du langage on effectue une concaténation qui prend $O(|p|)$ opérations. Ainsi, la complexité est en $O(|p|\times|l|)$.
Q15. On peut utiliser les chaînes de caractères de taille quelconque pour représenter un mot et les langages par des listes de chaînes de caractères.
Q16.

```
def prefixer(p,l):
    for i in range(len(l)):
        l[i]=p+l[i]
    return l
```

Q17.
Le nombre d'itérations est $O(|l|)$. La concaténation des chaînes de caractères est en $O(|p|)$. On a donc une complexité en $O(|p|\times|l|)$.
Q18.
On peut prendre $\mathcal{E}_1=(a|bb)(b|ab)^*$ et $\mathcal{E}_2=(c|ab)(a|cb)^*$.
Q19.

```
type etat=int;;
type transition=(etat*char*etat) list ;;
type automate={Q : etat list ; I : etat list ; T : etat list ; gamma : transition};;
let exp_III_1={ Q=[0;1;2;3] ;  I=[0] ;  T=[1] ;
gamma=[(0,'a',1);(0,'b',2);(0,'c',3);(1,'a',2);(1,'b',1);(1,'c',3);(2,'a',3);
(2,'b',1);(2,'c',3);(3,'a',3);(3,'b',3);(3,'c',3)]   };;
```

Q20.

```
let valider a =
    let rec sous_liste l1 l2 = match l1 with
        | [] -> true
        | x::q -> (List.mem x l2) && sous_liste q l2
    and graphe g liste = match g with
        | [] -> true
        | x::q -> let d,e,f=x in
                  (List.mem d liste)&&(List.mem f liste) in
    (sous_liste a.I a.Q)&&(sous_liste a.T a.Q)&&(graphe a.gamma a.Q);;
```

Q21.

```
let valider m a=
    let rec transition_fonction q lettre liste = match liste with
        | [] -> []
        | x::suite -> let d,e,f=x in
                      if d=q && lettre=e then f::(transition_fonction q lettre suite)
                      else transition_fonction q lettre suite in
     let rec transition_fonction_ens liste lettre = match liste with
         | [] -> []
         | x::q -> (transition_fonction x lettre a.gamma)@(transition_fonction_ens q lettre) in
     let rec transition_etoile liste mot= match mot with
         | [] -> liste
         | lettre::suite -> transition_etoile (transition_fonction_ens liste lettre) suite in
     let rec final liste = match liste with
         | [] -> false
         | x::q -> (List.mem x a.F) || final q in
     let rec aux liste = match liste with
         | [] -> false
         | x::q -> final (transition_etoile [x] m) || aux q in
     aux a.I ;;
```

Q22.
Pour chaque lettre du mot, on parcourt l'ensemble des transitions. La complexité est donc en $O(|mot|\times|transition|)$.
Q23. Les états sont représentés par des entiers, l'ensemble des états par un tableau d'entiers, les états finaux et initiaux par un tableau d'entiers, les transitions par un tableau de listes de triplets constitués d'un état, d'une lettre et d'un état. Un automate est un quadruplet dont le premier est une liste d'états, le deuxième une liste d'entiers états initiaux, le troisième une liste d'états finaux, le quatrième un tableau de listes de triplets.
Par exemple,

```
expl_III_1=( [0,1,2,3] , [0] , [1] ,
[ [(0,'a',1),(0,'b',2),(0,'c',3)], [(1,'a',2),(1,'b',1),(1,'c',3)], [(2,'a',3),(2,'b',1),(2,'c',3)], [(3,'a',3),(3,'b'
```

Q24.

```
def valider(a):
    boo=true
    for i in a[1]:
        boo=boo and (i in a[0])
    for i in a[2]:
        boo=boo and (i in a[0])
    for i in a[3]:
        boo=boo and (i[0] in a[0]) and (i[2] in a[0])
    return boo
```

Q25.

```
def accepter(m,a):
    Q=a[1]
    for i in range(len(m)):
        Q0=[]
        for x in Q:
            for t in a[3]:
                if x==t[0] and m[i]=t[1]:
                    Q0.append(t[2])
        Q=Q0
    for x in Q:
        if x in a[2]:
            return true
    return false
```

Q26.
On a $(a.b.c)||_S(a.b.d) = a.\big((b.c)||_S(a.b.d)\big) \cup a.\big((a.b.c)||_S(b.d)\big)$. De plus, $(b.c)||_S(a.b.d) = a.\big((b.c)||_S(b.d)\big) = a.b.(c||_S d)$ et $(c||_S d) = c.(\epsilon||_S d) \cup d.(c||_S \epsilon) = c.d.(\epsilon||_S \epsilon) \cup d.c.(\epsilon||_S \epsilon) = \{cd\} \cup \{dc\} = \{cd, dc\}$.
On a donc $(b.c)||_S(a.b.d) = a.b.\{cd, dc\} = \{abcd, abdc\}$. De même, $(a.b.c)||_S(b.d) = (b.d)||_S(a.b.c) = \{abdc, abcd\}$.
On a donc $(a.b.c)||_S(a.b.d) = \{aabcd, aabdc\}$.
Q27.
Montrons ce résultat par récurrence sur $\min(|m_1|, |m_2|)$.
Supposons que $\min(|m_1|, |m_2|) = 0$. Comme $m_1||_S m_2 = m_2||_S m_1$, on peut supposer que $|m_2| = 0$ donc $m_2 = \epsilon$. Alors, d'après les propriétés $s.m||_S\epsilon = \emptyset$, $x.m||_S\epsilon = x.(m||_S\epsilon)$; dans le premier cas, le langage n'a aucun mot et dans le second, il n'a pas de mot de longueur 0 ; par conséquent, si $m_1 \neq \epsilon$, $m_1||_S m_2$ ne contient pas le mot vide ; si $m_1 = \epsilon$, d'après la deuxième propriété, $m_1||_S m_2 = \{\epsilon\}$. Le résutlat est donc vrai pour $\min(|m_1|, |m_2|) = 0$.
Supposons le résultat vrai pour $\min(|m_1|, |m_2|) = n$. Supposons que $\min(|m_1|, |m_2|) = n+1$. Par symétrie des rôles (première propriété), on peut supposer que $m_1 = l.m$ avec $|m| = n$. Dans ce cas, $m_2 = l'.m'$. D'après la propriété 5, si $l = l'$ sont dans S, $m_1||_S m_2$ ne contient pas le mot vide ; si $l \neq l'$ sont dans S, d'après la propriété 6, $m_1||_S m_2$ ne contient pas le mot vide ; si $l \in S$ et $l' \notin S$ ou $l \notin S$ et $l' \in S$, d'après la propriété 7, $m_1||_S m_2$ ne contient pas le mot vide. Si l et l' sont dans $X\backslash S$, d'après la dernière propriété, $m_1||_S m_2$ ne contient pas le mot vide.
Ainsi, par récurrence, ceci démontre le résultat à établir.
Q28. L'énoncé n'est pas correct, notamment quant à la place des quantificateurs
Supposons que $s.m \in m_1||_S m_2$. D'après les règles de définition, le seul cas où un élément de $m_1||_S m_2$ commence par s est lorsqu'il existe m_1' et m_2' tel que $m_1 = sm_1'$ et $m_2 = sm_2'$. Dans ce cas, $m_1||_S m_2 = s(m_1'||_S m_2')$ donc, comme $sm \in m_1||_S m_2$, $m \in m_1'||_S m_2'$.
Réciproquement, supposons qu'il existe $(m_1', m_2') \in X^* \times X^*$ tels que $m_1 = sm_1'$ et $m_2 = sm_2'$ et $m \in m_1'||_S m_2'$. Alors, d'après la règle de définition, $m_1||_S m_2 = s.(m_1'||_S m_2')$ donc $sm \in m_1||_S m_2$.
Ainsi, $s.m \in m_1||_S m_2$ si et seulement si $\exists(m_1', m_2') \in X^* \times X^*$ tel que $m_1 = sm_1'$, $m_2 = sm_2'$ et $m \in m_1'||_S m_2'$.
Q29. Même remarque préliminaire qu'à la question précédente
D'après les règles de définition, $x.m \in m_1||_S m_2$ si et seulement si $\exists m_1' \in X^*$ tel que $m_1 = xm_1'$ et $m \in m_1'||_S m_2$ ou $\exists m_2' \in X^*$ tel que $m_2 = xm_2'$ et $m \in m_1||_S m_2'$.
Q30.

```
let concat x l = match l with
    | [] -> []
    | y::q -> (x::y)::(concat x q);;

let rec synchro_mot m1 m2 s = match m1,m2 with
```

```
    | [],[] -> [ [] ]
    | x::q,[] -> if List.mem x s then [] else concat x (synchro_mot q [] s)
    | [],x::q -> synchro_mot m2 m1 s
    | x::q1,y::q2 when List.mem x s ->
                        if x=y then concat x (synchro_mot q1 q2 s)
                        else if List.mem y s then []
                                else concat y (synchro_mot m1 q1)
    | x::q1,y::q2 when List.mem y s -> synchro_mot m2 m1 s
    | x::q1,y::q2 -> (concat x (synchro_mot q1 m2 s))@(concat y (synchro_mot m1 q2 s));;
```

Q31.

```
def concat(x,l):
    if len(l)==0:
        return l
    return [x+l[0];concat(x,l[1::])]

def synchro_mot(m1,m2,s):
    if len(m1)==0 and len(m2)==0:
        return [ [] ]
    if len(m1)==0:
        if m2[0] in s:
            return []
        else:
            return concat(m2[0],synchro_mot(m1,m2[1::],s))
    if len(m2)==0:
        return sycnhro(m2,m1,s)
    if m1[0] in s:
        if m2[0] in s:
            if m1[0]!=m2[0]:
                return []
            else:
                return concat(m1[0],synchro_mot(m1[1::],m2[1::],s))
        else:
            return concat(m2[0],synchro_mot(m1,m2[1::],s))
    else:
        if m2[0] in s:
            return synchro(m2,m1,s)
        else:
            return concat(m1[0],synchro(m1[1::],m2,s))+concat(m2[0],synchro(m1,m2[1::],s))
```

Q32.

```
let rec synchro_lang l1 l2 s =
    let rec aux m l = match l with
        | [] -> []
        | m2::q -> (synchro_mot m m2 s)@(aux m q) in
    match l1 with
        | [] ->[]
        | m1::q -> (aux m1 l2)@(synchro_lang q l2 s);;
```

Q33.

Q34.

```
let synchro a1 a2 s =
    let rec prod1 x l = match l with
        | [] -> []
        | y::q -> (x,y)::prod1 x q in
    let rec prod2 l1 l2 = match l1 with
        | [] -> []
        | x::q -> (prod1 x l2)@(prod2 q l2) in
    {Q=prod2 a1.Q a2.Q; I=prod2 a1.I a2.I; T=prod2 a1.T a2.T; gamma=prod2 a1.gamma a2.gamma};;
```

Q35.
L'ensemble $Q_1 \times Q_2$ est fini ; on a $I_1 \times I_2 \subset Q_1 \times Q_2$, $T_1 \times T_2 \subset Q_1 \times Q_2$ et $\gamma_{\mathcal{A}} \subset (Q_1 \times Q_2) \times X \times (Q_1 \times Q_2)$. L'automate $\mathcal{A}_1||_S\mathcal{A}_2$ est donc bien un automate fini.

Q36.
Supposons que $m = \epsilon$. Alors $((o_1, o_2), m, (d_1, d_2)) \in \gamma_{\mathcal{A}}^*$ si et seulement si $(o_1, o_2) = (d_1, d_2)$ si et seulement si $d_1 = o_1$ et $d_2 = o_2$ si et seulement si $(o_1, \epsilon, d_1) \in \gamma_1^*$ et $(o_2, \epsilon, d_2) \in \gamma_2^*$. D'après la question 27, $\exists(m_1, m_2) \in X^* \times X^*$ tel que $m = \epsilon \in m_1||_S m_2$ si et seulement si $m_1 = m_2 = \epsilon$. Ainsi, $((o_1), m, (d_1, d_2)) \in \gamma_{\mathcal{A}^*}$ si et seulement si $\exists(m_1, m_2) \in X^* \times X^*$ tel que $m \in m_1||_S m_2$ et $(o_1, m_1, d_1) \in \gamma_1^*$ et $(o_2, m_2, d_2) \in \gamma_2^*$.
Supposons que le résultat à établir soit vrai pour un mot m' : pour tout $(o_1, o_2, d_1, d_2) \in Q_1 \times Q_2 \times Q_1 \times Q_2$, $((o_1, o_2), m', (d_1, d_2)) \in \gamma_{\mathcal{A}}^*$ si et seulement si $\exists(m'_1, m'_2) \in X^* \times X^*$ tels que $m' \in m'_1||_S m'_2$ et $(o_1, m'_1, d_1) \in \gamma_1^*$ et $(o_2, m'_2, d_2) \in \gamma_2^*$.
Supposons que $m = sm'$. Alors $((o_1, o_2), m, (d_1, d_2)) \in \gamma_{\mathcal{A}}^*$ si et seulement s'il existe $q_1 \in Q_1$ et $q_2 \in Q_2$ tels que $((o_1, o_2), s, (q_1, q_2)) \in \gamma_{\mathcal{A}}$ et $((q_1, q_2), m', (d_1, d_2)) \in \gamma_{\mathcal{A}}^*$ si et seulement s'il existe $q_1 \in Q_1$ et $q_2 \in Q_2$ tels que $(o_1, s, q_1) \in \gamma_1$, $(o_2, s, q_2) \in \gamma_2$ et $((q_1, q_2), m', (d_1, d_2)) \in \gamma_{\mathcal{A}}^*$. Ceci équivaut, d'après l'hypothèse de récurrence, à : il existe $(q_1, q_2) \in Q_1 \times Q_2$ et $(m'_1, m'_2) \in X^* \times X^*$ tels que $(o_1, s, q_1) \in \gamma_1$, $(o_2, s, q_2) \in \gamma_2$, $m' \in m'_1||_S m'_2$, $(q_1, m'_1, d_1) \in \gamma_1^*$, $(q_2, m'_2, d_1) \in \gamma_2^*$. Ceci équivaut à : $\exists(m'_1, m'_2) \in X^* \times X^*$ tel que $m' \in m'_1||_S m'_2$ et $(o_1, sm'_1, d_1) \in \gamma_1^*$ et $(o_2, sm'_2, d_2) \in \gamma_2^*$. D'après la question 28 (rectifiée), ceci équivaut à $\exists(m_1, m_2) \in X^* \times X^*$ tel que $sm \in m_1||_S m_2$ et$(o_1, m_1, d_1) \in \gamma_1^*$ et $(o_2, m_2, d_2) \in \gamma_2^*$.
Supposons que $m = xm'$ avec $x \notin S$. Alors $((o_1, o_2), xm', (d_1, d_2)) \in \gamma_{\mathcal{A}}^*$ si et seulement s'il existe $(q_1, q_2) \in Q_1 \times Q_2$ tel que $((o_1, o_2), x, (q_1, q_2)) \in \gamma_{\mathcal{A}}$ et $((q_1, q_2), m', (d_1, d_2)) \in \gamma_{\mathcal{A}}^*$ si et seulement s'il existe $(q_1, q_2) \in Q_1 \times Q_2$ tel que $((o_1, x, q_1) \in \gamma_1$ et $o_2 = q_2$ et $(q_1, q_2), m', (d_1, d_2) \in \gamma_{\mathcal{A}}^*)$ ou $(o_1 = q_1$ et $(o_2, x, q_2) \in \gamma_2$ et $((q_1, q_2), m', (d_1, d_2)) \in \gamma_{\mathcal{A}}^*)$.
En utilisant l'hypothèse de récurrence, $((o_1, o_2), m, (d_1, d_2)) \in \gamma_{\mathcal{A}}^*$ si et seulement s'il existe $(q_1, q_2) \in Q_1 \times Q_2$ tels que $\big(\exists(m'_1, m'_2) \in X^* \times X^*$ tel que $(o_1, x, q_1) \in \gamma_1$ et $(q_1, m'_1, d_1) \in \gamma_1^*$ et $(o_2, m'_2, d_2) \in \gamma_2^*$ et $m' \in m'_1||_S m'_2\big)$ ou $\big(\exists(m'_1, m'_2) \in X^* \times X^*$ tel que $(o_2, x, q_2) \in \gamma_2$ et $(o_1, m'_1, d_1) \in \gamma_1^*$ et $(q_2, m'_2, d_2) \in \gamma_2^*)$ et $m' \in m'_1||_S m'_2\big)$ si et seulement si $\big(\exists(m'_1, m'_2) \in X^* \times X^*$ tel que $(o_1, xm'_1, d_1) \in \gamma_1^*$ et $(o_2, m'_2, d_2) \in \gamma_2^*\big)$ ou $\big(\exists(m'_1, m'_2 \in X^* \times X^*$ tel que $(o_2, x.m'_2, d_2) \in \gamma_2^*$, $(o_1, m'_1, d_1) \in \gamma_1^*\big)$.
D'après la question 29 (rectifiée), ceci équivaut à : il existe $(m_1, m_2) \in X^* \times X^*$ tels que $x.m \in m_1||_S m_2$ et $(o_1, m_1, d_1) \in \gamma_1^*$ et $(o_2, m_2, d_2) \in \gamma_2^*$.
Par récurrence ou induction, cela prouve le résultat.

Q37.
Soit $m \in X^*$. Alors $m \in L(\mathcal{A}_1||_S\mathcal{A}_2)$ si et seulement s'il existe $(o_1, o_2) \in I_1 \times I_2$, $(d_1, d_2) \in T_1 \times T_2$ tels que $((o_1, o_2), m, (d_1, d_2)) \in \gamma_{\mathcal{A}}^*$ si et seulement si (d'après la question précédente) $\exists(o_1, o_2, t_1, t_2) \in I_1 \times I_2 \times T_1 \times T_2$ et $\exists(m_1, m_2) \in X^* \times X^*$ tel que $m \in m_1||_S m_2$ et $(o_1, m_1, d_1) \in \gamma_1^*$ et $(o_2, m_2, d_2) \in \gamma_2^*$ si et seulement si $\exists(m_1, m_2) \in X^* \times X^*$ tel que $m \in m_1||_S m_2$ et $(\exists(o_1, d_1) \in I_1 \times T_1$ et $(o_1, m_1, d_1) \in \gamma_1^*)$ et $(\exists(o_2, d_2) \in I_2 \times T_2$ et $(o_2, m_2, d_2) \in \gamma_2^*$ si et seulement si $\exists(m_1, m_2) \in X^* \times X^*$ tel que $m \in m_1||_S m_2$ et $m_1 \in L(\mathcal{A}_1)$ et $m_2 \in L(\mathcal{A}_2)$ si et seulement si $m \in L(\mathcal{A}_1)||_S L(\mathcal{A}_2)$.
Ceci prouve que $L(\mathcal{A}_1||_S\mathcal{A}_2) = L(\mathcal{A}_1)||_S L(\mathcal{A}_2)$.